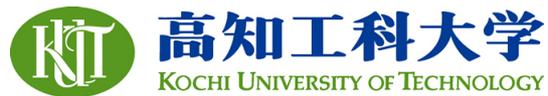


Title	Picking System for Automatic Harvesting of Sweet Pepper -Sensing and Mechanism-
Author(s)	EIZENTALS, Peteris
Citation	高知工科大学, 博士論文.
Date of issue	2016-09
URL	http://hdl.handle.net/10173/1417
Rights	
Text version	ETD



Kochi, JAPAN

<http://kutarr.lib.kochi-tech.ac.jp/dspace/>

Picking System for Automatic Harvesting of Sweet Pepper

Sensing and Mechanism

by

Peteris Eizentals

Graduate School of Engineering

Kochi University of Technology

A thesis submitted for the degree of

Doctor of Engineering

Kochi 2016

Abstract

It is widely recognized in the field of automatic harvesting that an additional sensing is required to deal with complicated cases when the visual cues alone do not provide the harvesting robot with the necessary information. The aim of this research was to develop novel methods for use in the automatic harvesting of sweet pepper that would allow improving the performance of the harvesting robot. The main question to answer was “How to detect the stem position if the visual information is unreliable?” Two general assumptions were made to base the research on, 1) only the cutting point is required for the cutter and 2) fruit position can be detected using the previously developed and verified sweet pepper recognition method. In order to find the stem position a concept of pose estimation was used, which was proposed by K. Kapach et al. in the article “Computer vision for fruit harvesting robots – state of the art and challenges ahead”. The concept was implemented by using model matching of a predefined model and the surface points of an actual sweet pepper fruit. The surface points were obtained by a LIDAR type laser range finder and the model matching was performed by Coherent Point Drift algorithm developed by Andriy Myronenko and Xubo Song. The hardware used for the developed method consists of a LIDAR laser range finder, an RGB USB webcam, a vertical slider, an LED array and the control electronics. The developed method calculates the orientation of a fruit in space and the position of the stem. This calculated stem position is supposed to be used by the harvesting robot to cut the found target pepper. Two experiments were performed to evaluate the developed method. First, the accuracy of the method was evaluated under the laboratory conditions by using a sweet pepper test object at a known inclination angle and stem position. A real sweet pepper fruit was positioned on a test platform and set to known inclination angles using a 2DOF manipulator. Then the pose of the target in space and the stem position was calculated by using the developed method. The result of the calculation was compared to the set position to assess the error of the calculation.

Secondly, a field testing was performed in a greenhouse on sweet pepper fruits. A test rig was positioned in the greenhouse and the developed method was executed to acquire the result of pose estimation and stem position calculation algorithm. The result was then projected on the visual image to evaluate the performance of the system.

It is recognized that the position of the stem calculated by the above described algorithm is merely a calculation and not actual stem position detection. Therefore means to verify that the stem is in the cutting position must be implemented. For this reason a touch sensor was developed, based on piezo effect. The sensor consists of two piezo stack actuators and a particular shape contact tip, which are all mounted together on the base of the cutter. The sensor works as follows: one of the actuators is being driven by a sine wave in one of the resonant frequencies of the sensor while the other actuator generates a charge due to this motion. The contact tip acts as a mechanical coupling between the actuators and also as an additional mass. The resonant frequency of any mechanical system depends on the stiffness and mass properties of the system. In this case, whenever some object, such as a stem, touches the contact tip of the sensor, the mass, and the stiffness properties are changed and consequently also the frequency response of the sensor. The shift of the resonant frequency causes the amount of charge generated by the second actuator to change. The change of the generated charge is then detected by the signal processing unit. As the change of the properties is directly dependent on the mass properties of the touching object and the force of the contact, the touch of a leaf is not detected due to the relatively small mass and stiffness change of the system. The position of the sensor allows a detection of a stem between the cutter blades but also makes it impossible for bigger objects such as fruit itself to touch the sensor. Several tests were performed to evaluate the performance of the developed sensor. First, the frequency response was measured to find the driving frequency. Secondly, each of the found major resonant frequencies was tested for stability in an extended time measurement to evaluate the quality of the used frequency. The most promising frequency was used as the driving frequency for all remaining tests. The following tests were performed: sensitivity test, movement test, and field testing. The sensitivity test was designed to evaluate the sensitivity of the sensor by pushing a pinpoint force to various parts of the sensing tip with a known force, which was measured by a dial tension gauge. The result was then compared to the force required to push a sweet pepper stem in a greenhouse. The movement test was designed to analyze the effect of the movement of a manipulator on the sensor output. The

experiment was performed by attaching the sensor to a manipulator and moving the manipulator randomly while recording the sensor output. Then the result was compared to the measurement of a static sensor to assess the impact of the movement. Lastly, the field testing was performed by using the sensor in a greenhouse. The sensor was attached to a manipulator, which was moved manually, and a physical contact was produced between the sensing tip and a sweet pepper stem while recording the output of the sensor. This experiment was performed for two reasons. First, it was necessary to verify that a contact with a sweet pepper stem changes the physical properties of the sensor significantly enough to affect the resonant frequency. Secondly, the amplitude of the sensor output change was examined to decide on the detection threshold value.

Incompatibility between the hardware used for the pose estimation algorithm and the hardware of the previous sweet pepper automatic harvesting robot prototype revealed a necessity for a new harvesting robot design. As a result, a new monorail type harvesting robot was designed. This design uses a single rail to move around a greenhouse thus solving the harvester positioning and movement problems. It also has vertical slider implemented separately from the working manipulator and works fully on batteries. The batteries are charged by a non-contact inductive power supply, which allows for fully automatic operation. The designed harvesting robot is currently at the development and assembling stage and is not yet fully tested although some parts have been tested separately.

The dissertation is constructed as follows. First, a literature review on automatic harvesting is given with the focus on methods used for target detection and grasping/detachment technologies. The state-of-art situation both in methodology and in hardware development is presented to demonstrate current achievements and obstacles to overcome. Afterward, the problems addressed by the current dissertation are explained together with the used methodology to solve them. A full description in great detail is given for all three of the developed systems described above and the tests performed, each in its own section. The results of performed tests are analyzed thoroughly and the implications of certain results are explained. Future work and improvement options are discussed in the end to outline the path for future researchers.

Acknowledgements

Foremost, I would like to express my deepest sense of gratitude to my supervisor *Prof. Koichi Oka* for having faith in my abilities and giving me this chance to study at Kochi University of Technology. His endless support and never-fading smile guided me through these years and made this research an enjoyable experience.

I would like to express my very sincere gratitude to *Lawrie Hunter* and *Paul Daniels* for their great contribution in my development as a researcher and, even more important, as a person. I deeply enjoyed our conversations about my research and other mysteries of life and I highly appreciate their help with improvement of my English language skills.

I am thankful to all the *members of the OKA lab* for their assistance during my research. I'm especially thankful to *Prof. Harada* and my friend and colleague *Annasi* for their great support and valuable discussions that led to fresh ideas.

I am also very thankful to all the *staff members of International Relations Division* for their hard work and support to every international student, who would need it.

I had a chance to meet many very interesting and intelligent people, some of whom became my closest friends here. I would like to express my gratitude to *Sergey, Monica* and *Roberto*, for helping me to deal with the initial culture shock and exploring Kochi together, to *Adam, Michelle, Rika, Erina, Ono, Vlad, Rebecca* and *Cristina* for being great friends both in moments of joy and trouble, to *Yuuki and his family* for introducing me to parts of Japanese culture I otherwise wouldn't discover, to all the *Thai community in KUT* for being open and friendly, and to *countless others* for making the time spent here an unforgettable experience.

Finally, I take this opportunity to express my profound gratitude to my family for their support and understanding through all these years. Their neverending support let me see the light even during the darkest of times.

To you all I dedicate this work.

Disclaimer

Some passages in this work have been quoted *verbatim* from the previous publications and work of the author (listed according to the state in time of writing the thesis):

Conference Publications

1. P. Eizentals, K. Oka, 2014. „The Use of Tactile Sensing in Green Pepper Harvesting Gripper”, in proceedings of SICE Annual Conference 2014, September 9 – 12, Sapporo, Japan.
2. P. Eizentals, K. Oka, 2015. „Automated Harvesting of Sweet Pepper in Greenhouse: Fruit Orientation Calculation Based on LIDAR Measurement”, in proceedings of the Robotics and Mechatronics Conference 2015, May 17 – 19, Kyoto, Japan.
3. P. Eizentals, K. Oka, 2015. „Green Pepper Stem Position Detection by Using a Piezo Sensor”, in proceedings of the 10th Asian Control Conference (ASCC) 2015, May 31 – June 3, Kota Kinabalu, Malaysia.
4. P. Eizentals, T. Tokunaga, K. Oka, 2016. „Design of a Monorail Type Green Pepper Automatic Harvesting Robot”, ”, in proceedings of the Robotics and Mechatronics Conference 2016, June 8 – 11, Yokohama, Japan.

Accepted Conference Papers

1. P. Eizentals, K. Oka, 2016. „Fruit Pose Estimation and Stem Touch Detection for Green Pepper Automatic Harvesting”, accepted for International Symposium on Experimental Robotics 2016, October 3 – 6, Tokyo, Japan.

Journal Paper Submissions

1. P. Eizentals, K. Oka. „3D Pose Estimation of Green Pepper Fruit for Automated Harvesting”, submitted to Computers and Electronics in Agriculture on January 4, 2016 (accepted for publishing, DOI: 10.1016/j.compag.2016.08.024).
2. P. Eizentals, A. Harada, K. Oka. „Fruit Stem Detection by a Piezo Sensor for Green Pepper Automatic Harvesting”, submitted to Computers and Electronics in Agriculture on April 22, 2016 (under review).

All of the quoted sources were originally written by the author of this thesis.

Table of Contents

Abstract	iii
Acknowledgements	vi
Disclaimer	vii
Table of Contents	ix
List of Figures	xii
List of Tables	xv
Abbreviations	xvi
1. Introduction	1
1.1 Background.....	1
1.2 Previous Research	2
1.3 Research Problem	3
1.4 Research Objectives and Methods.....	6
1.5 Organization of Dissertation	8
2. Robotics in Agriculture	10
2.1 Overview	10
2.2 Target Detection.....	14
2.2.1 <i>Visual Cues</i>	15
Color	16
Spectral Reflectance.....	17
Thermal Response.....	18
Texture	19
Shape.....	19
Combining Visual Cues	20
2.2.2 <i>Image Analysis Methods</i>	20
Thresholding.....	20
Clustering	21
Template Matching	22
Shape Inference	22
Machine Learning.....	23
2.2.3 <i>Present Research</i>	23

2.3 Harvesting End-Effectors.....	27
2.3.1 Gripping.....	27
2.3.2 Detachment.....	30
2.3.3 Current Research.....	32
2.4 Conclusions.....	33
3. Pose Estimation	34
3.1 Overview	34
3.2 Prototype I.....	35
3.2.1 Setup.....	35
3.2.2 Algorithm.....	36
3.2.3 Performance Evaluation.....	40
3.2.4 Conclusions.....	43
3.3 Prototype II.....	45
3.3.1 Setup.....	45
3.3.2 Algorithm.....	46
3.3.3 Performance Evaluation.....	51
Angle Test.....	51
Greenhouse Test.....	58
3.3.4 Speed of the System.....	62
3.3.5 Conclusions.....	63
3.4 Summary	65
4. Touch Detection	66
4.1 Background.....	66
4.2 Requirements.....	69
4.3 FSR Touch Sensor	70
4.3.1 Overview.....	70
4.3.2 Setup.....	72
4.3.3 Evaluation.....	73
4.3.4 Conclusions.....	75
4.4 Piezo Touch Sensor	75
4.4.1 Background	75
4.4.2 Overview.....	76
4.4.3 Piezoelectricity.....	77
4.4.4 Prototype I.....	80
Setup	80
Evaluation.....	81
Conclusions	86
4.4.5 Prototype II.....	87
Setup	87
Evaluation.....	89
Conclusions	95
4.5 Overall Conclusions.....	96

5. Monorail Continuum Automatic Harvesting Robot.....	97
5.1 Overview	97
5.2 Hardware.....	100
5.3 Working Algorithm	103
5.4 Position Control.....	104
5.5 Power Supply.....	106
5.5.1 Power Consumption	106
5.5.2 Charging	107
5.6 Manipulator Considerations.....	111
5.6.1 Kinematics	112
5.6.2 Development	114
5.7 Configuration Analysis.....	115
5.7.1 Structural Analysis.....	115
5.7.2 Harvesting Area.....	116
5.7.3 Turning	117
5.8 Cost Estimation	118
5.9 Conclusions.....	120
6. Summary.....	122
References	125
Appendices	133
Appendix A – Software Codes	133
<i>Vertical slider control code</i>	133
<i>Manipulator code</i>	135
<i>Piezo Control Code</i>	140
<i>Pose Estimation</i>	145
<i>Piezo Sensor</i>	155

List of Figures

Nr.	Caption	Page
Figure 1-1	The first prototype with 7DOF manipulator and end-effector	2
Figure 1-2	Japanese sweet pepper	3
Figure 1-3	Greenhouse setup information used in this study	5
Figure 1-4	Cutter – pincer end-effector, which was chosen for this study	7
Figure 2-1	Harvesting robot projects from the last 30 years for different fruits and vegetables	11
Figure 2-2	Timeline of research on automatic harvesting robots	12
Figure 2-3	Difference in histogram values depending on the illumination conditions for the fruit	17
Figure 2-4	Example of metamerism	18
Figure 2-5	Thermal response of citrus fruits versus the rest of canopy	18
Figure 2-6	Example of texture extraction method using edge detection	19
Figure 2-7	Result of k-means clustering on feature points	21
Figure 2-8	Shape of oranges reconstructed from the shape of edges	22
Figure 2-9	Comparing the difference between pixel values of fruit and leaf in HSV color space	24
Figure 2-10	Recognition algorithm	24
Figure 2-11	Example of recognition step by step	26
Figure 2-12	Example of a scissors type cutter with four-finger jaw gripper	29
Figure 2-13	Example of a thermal cutter with a two-finger jaw gripper and a suction cup	30
Figure 2-14	Example of a moving lip cutter with a suction cup	31
Figure 3-1	Pose estimation system Prototype I	36
Figure 3-2	Pose estimation setup for the greenhouse test	36
Figure 3-3	Pose estimation algorithm for the Prototype I	37
Figure 3-4	Target setup for the laboratory experiment	40
Figure 3-5	Setup for the laboratory experiment	41

Figure 3-6	LIDAR point clouds after filtering and pattern matching, front view	43
Figure 3-7	LIDAR point clouds after filtering and pattern matching, isometric view	43
Figure 3-8	Slider setup for the second prototype	46
Figure 3-9	Pose estimation algorithm of the Prototype II	48
Figure 3-10	Targets used for the laboratory test	52
Figure 3-11	Target setup for the laboratory test	53
Figure 3-12	Laboratory test results	55
Figure 3-13	The average error compared to the inclination angle for the affine transformation	58
Figure 3-14	The average error compared to the inclination angle for the rigid transformation	58
Figure 3-15	Greenhouse test setup	59
Figure 3-16	The template for greenhouse test result evaluation	60
Figure 3-17	Greenhouse test result examples	60
Figure 3-18	Comparing the frequency of receiving a particular score for both CPD transformations	61
Figure 4-1	Multi-finger gripper model with FSR sensors at fingertips	70
Figure 4-2	Working algorithm of fruit surface scanning by touch sensors	71
Figure 4-3	Sensor FSR400	72
Figure 4-4	Schematics of the measurement system	73
Figure 4-5	FSR test setup	74
Figure 4-6	Examples of FSR test result for a leaf, a fruit, and a branch	74
Figure 4-7	The end-effector used by the harvesting robot	76
Figure 4-8	Multilayer piezo stack actuator layer polarization and electrode configuration	78
Figure 4-9	Schematic representation of the measurement system	80
Figure 4-10	Developed sensor	80
Figure 4-11	Frequency response of the developed sensor	81
Figure 4-12	Stability test results	82
Figure 4-13	Test setup	82
Figure 4-14	Movement test result	83
Figure 4-15	Sensitivity test result	83
Figure 4-16	Repeatability test result	84
Figure 4-17	Settling test result	84
Figure 4-18	Touch stem test result	85
Figure 4-19	Cutting test result	85

Figure 4-20	The second prototype of the developed sensor	87
Figure 4-21	Functional diagram of the measurement system	87
Figure 4-22	Comparison between amplitude measurement and integral of squared amplitude for frequency response measurement	88
Figure 4-23	Frequency response of the developed sensor	89
Figure 4-24	Stability test results	90
Figure 4-25	FFT analysis result	91
Figure 4-26	Setup for the movement and greenhouse tests	91
Figure 4-27	Movement test result	92
Figure 4-28	Setup for the sensitivity and stability tests	92
Figure 4-29	Mode shape analysis results for 70 kHz frequency	93
Figure 4-30	Sensitivity test results	94
Figure 4-31	Example of a greenhouse test result (normalized)	95
Figure 5-1	Automatic harvesting robot developed by the research team of “Clever Robots for Crops”	98
Figure 5-2	The developed monorail continuum automatic harvesting robot	99
Figure 5-3	Movement system	100
Figure 5-4	Target recognition system	101
Figure 5-5	Functional schematics of the new harvesting robot design	102
Figure 5-6	Working algorithm of the new monorail harvester	103
Figure 5-7	Harvesting sequence	104
Figure 5-8	Functional schematic of the charging system; Top part is the charging station side while the bottom part is the harvester side	107
Figure 5-9	Robot with a continuum manipulator on the left, and during nuclear site examination test on the right	111
Figure 5-10	Mapping between the actuator space, arc parameters, and the task space	112
Figure 5-11	Arc parameters for one segment	113
Figure 5-12	Single-segment prototype of the continuum manipulator developed by Takaaki Tokunaga	114
Figure 5-13	FEM Structural analysis test results for an aluminum bracket with 3mm thickness, deformation on the left and stress on the right	115
Figure 5-14	Greenhouse setup with the position for the new harvesting robot	117
Figure 5-15	Increase in the harvesting area caused by the fixed segment	117
Figure 5-16	Rotation radius of the harvester the current configuration	118

List of Tables

Nr.	Caption	Page
Table 2-1	Harvesting robot projects reported in the period 1982 – 2012	13
Table 2-2	Most common manipulation strategies for agricultural robots	28
Table 3-1	Results of the laboratory test	42
Table 3-2	Angle test results, mean values of all 10 measurements	56
Table 3-3	Frequency of result error value under a set limit.	57
Table 4-1	Touch sensing techniques	68
Table 5-1	Power consumption of the main electronic parts of the harvesting robot prototype	106
Table 5-2	Parameters of the transfer (Tx) and receiving (Rx) coils of the inductive power supply	110
Table 5-3	General cost of the new harvesting robot prototype	119

Abbreviations

3D	–	Three Dimensional
AC	–	Alternating Current
ANN	–	Artificial Neural Networks
BW	–	Black and White Image
DC	–	Direct Current
DOF	–	Degree of Freedom
FEA	–	Finite Element Analysis
FEM	–	Finite Element Method
FFT	–	Fast Fourier Transform
FSR	–	Force Sensing Resistor
HSV	–	“Hue-Saturation-Value” color space image
LC	–	Resonant Circuit
LED	–	Light Emitting Diode
RGB	–	“Red-Green-Blue” color space image
SSD	–	Sum of Squared Differences
SVM	–	Support Vector Machines
TOF	–	Time of Flight
USB	–	Universal Serial Bus

1. Introduction

1.1 Background

Agriculture, despite being one of the most important fields for humanity, as it provides food resources, hasn't been affected by automation as much as other fields. The common trend in almost every aspect of our life for last several decades has been to automate every single process wherever it is possible. Rapid advances in computer technology and control systems make it possible for automation of almost anything, starting from such mundane tasks as cleaning (e.g. iRobot Roomba® series cleaning robots) to seemingly difficult technological tasks such as building bridges (e.g. MX3D Bridge Project). Automation, in general, is relatively easy if the environment is well-defined or controlled. Agriculture, however, is neither. Agriculture could be defined as a partially controlled environment as it is possible to define almost all parameters except for some important ones. For example, plants can be planted in known locations, but the place where plants will grow fruits can't be predicted or controlled with high accuracy. These and other reasons complicate automation of the harvesting process and are one of the main reasons why automatic harvesting robots still haven't reached the commercial production phase.

For the last few decades, researchers all over the world have been trying to develop automatic harvesting robots for various fruits and crops. The main focus, of course, has been on the major vegetables, such as tomatoes [1 – 4], bell peppers [5, 6] and cucumbers [7 – 9], and fruits, such as strawberries [10 – 12], kiwis [13] and grapes [14]. For several years researchers in KUT have been trying to develop an automatic harvesting robot for Japanese sweet pepper, locally known as *pīman* (ピーマン). Japanese sweet pepper (further in the text referred to as simply “sweet pepper”) is smaller than bell peppers with an average weight of approximately 40g (measurement obtained from commercially available specimens) and is one of the major vegetables in Japan. According to Ministry of Agriculture, Forestry and Fisheries of Japan collected statistical data for the fiscal year of 2013 (*Heisei 25*), from 41 officially

recognized major vegetables Sweet pepper is in 28th place in terms of the planted area (3360ha) and in 21st place in terms of the total production (145 300t). Kochi prefecture is the third largest Sweet pepper producer in Japan with 141ha of planted area and 13 000t of total production, which is 8.95% of the total production in Japan. In Kochi prefecture sweet pepper compared to other vegetables is in 7th place in terms of planted area and in 5th place in terms of total production, making it an important income source for the local farmers.

1.2 Previous Research

The first sweet pepper automatic harvesting robot prototype was developed by S. Kitamura and K. Oka [15] and it was later improved by S. Bachche and K. Oka [16]. The latest version of the robot prototype consisted of a 7DOF robotic arm which was mounted on a mobile platform (*fig. 1-1*). All electronics were mounted on the mobile platform and consisted of a central computer, power controller, and motor controllers. The target recognition was performed by using two CCD IR cameras in stereo configuration and an additional camera was mounted on the end effector of the robotic arm for the arm guidance in the harvesting process. The end effector of the latest prototype consisted of a scissor system that could perform the cutting and gripping tasks with a single movement (*fig. 1-1c*).

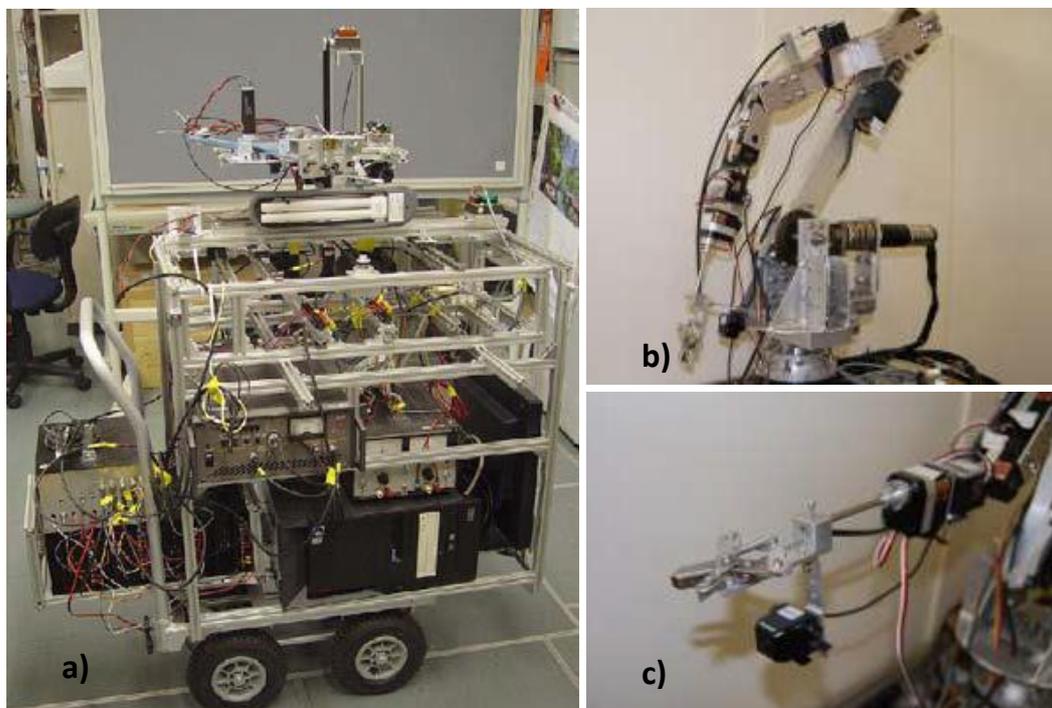


Figure 1-1 a) the first prototype with b) the 7DOF manipulator and c) end-effector

This latest prototype had many flaws, both in terms of mechanical development and actual functionality. For example, it didn't have means of moving itself in the greenhouse as the mobile platform had no motor for motion. Also, it had to be connected to an external power source via power cable as no battery was mounted on the platform during the development period. Furthermore, the electronics of the robot were not optimized for autonomous operation in terms of energy efficiency as autonomous systems that are designed to work on batteries should be as energy saving as possible. Despite the fact that the algorithm for target recognition was already mostly developed by previous researchers, the performance information of the harvesting robot prototype was mostly unavailable as the system wasn't tested in a greenhouse in its final form.



Figure 1-2 Japanese sweet pepper, this image demonstrates the difficulties that are to be tackled for automatic harvesting of sweet pepper, such as slantwise growth, dense foliage, and occlusions

1.3 Research Problem

There are several reasons why automatic harvesting of the sweet pepper is complicated, and they have been the main target of the continuous research:

- The detection of a target is not a trivial task due to the great similarity between the color of fruits and that of the surrounding environment. The previous research performed by S. Bachche and K. Oka concluded that the HSV color space can be used

to detect sweet pepper fruits in the night time by using artificial lighting or IR96 infra-red filter during day time. The acquired performance for both of these methods varied from 70 – 90% depending on the level of occlusions and illumination conditions [16]. This was partially verified also in this research, although some differences were found, which are described in more details in the later section.

- The currently applied cutting system requires information about the stem position in space, but none of the currently used recognition methods allows for accurate stem position detection. The usual method in such case would be to analyze the shape of the target segment and to assume that the stem is along the longest major axis of the smallest possible ellipse that can be fitted on the target segment in the image. This method, however, is unreliable in case of sweet pepper due to the frequently slantwise growth of sweet pepper fruits. Sweet pepper has a rather thick stem that allows the fruits to hang at steep angles, thus, the major axis of the fitted ellipse can be in a direction unrelated to the fruit direction as the fruit from the camera point of view can have an irregular shape. Furthermore, the shape of the found fruit available for analysis can be further distorted by occlusions by leaves or other fruits, or low segment quality. Due to these reasons, the chosen end-effector is well suited for the given task. A multi-finger gripper or a suction cup system would have to manipulate the fruit to position the stem for cutting. Such manipulations would require both accurate information on the current fruit pose in space and complex spherical joint type mechanism to perform the manipulation. The scissor-pincer system is capable of cutting and secure grasping of a fruit with a single motion therefore only information about the position of a stem is required to perform harvesting.
- Fruits often grow close together, which makes the recognition algorithm to detect a single fruit in case, when there are two or more fruits touching. In many cases, it is possible to recognize such clusters by the size of the segment in the image, but the result of such analysis is of Boolean type (true/false) and doesn't give information about the actual number of fruits in the cluster. Moreover, often fruits in the cluster are not entirely visible until one or more of the front fruits are harvested.
- Unless defoliated, sweet peppers have very dense foliage. Leaves are often covering fruits, greatly lowering the recognition rate. Furthermore, the dense foliage complicates a collision-free path planning for the manipulator of the harvesting robot.

Conventional robotic hands are ill-suited for obstacle avoidance in narrow space manipulation. Most of the developed harvesting robots deal only with fruits in front of them, and no publication up to my best knowledge addresses the problem of obstacle avoidance in the path planning in great details.

- Sweet pepper has a relatively wide growing space in the vertical direction. According to the greenhouse specifications used for this research (*fig. 1-3*), the height of the bed is 300mm and the height of planned harvesting area is approximately 1100mm. Standard lens camera at a close distance is unable to cover the whole area of interest, but fisheye lens cameras are susceptible to occlusions from objects in different heights. Some robots [17] use a camera mounted on the end-effector to solve this problem at the expense of the end-effector size thus sacrificing possible implementation of obstacle avoidance. The problem could be addressed by increasing the distance from the harvesting robot to the target plants, but that would require for increased space between the plan rows, which, in turn, would lower the overall productivity of the greenhouse.
- The movement in the greenhouse and the positioning of the robot is rarely addressed by the researchers in the literature. Only several researchers describe how their robot will navigate in the greenhouse and even then in a very general form [9, 11]. Sweet pepper harvesting is no different as the position in the row together with the general information on the already harvested area must be considered.

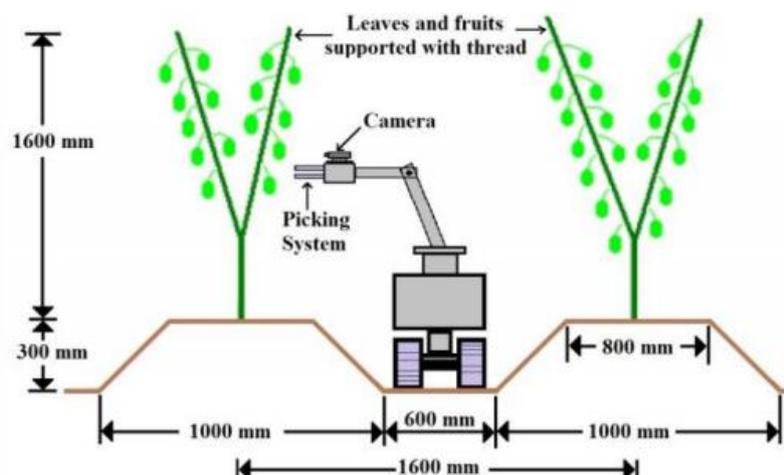


Figure 1-3 Greenhouse setup information used in this study (reproduced from [16]).

1.4 Research Objectives and Methods

The main research objective was changed during the course of the study. The initial objective was the development of an advanced gripper for the sweet pepper automatic harvesting robot. The requirements set for the advanced gripper were as follows:

- ✓ The ability to grasp a sweet pepper fruit of any size and shape – fruits of sweet pepper are often irregularly shaped and can vary in size, a perfect end-effector could be adjusted for all possible sizes;
- ✓ The ability to distinguish between different grasped objects – false positive recognition is possible in case of sweet pepper. If a leaf is recognized as a fruit, the end-effector should be able to detect that the grasped object is not a fruit;
- ✓ The ability to control the grasping force – damage to a fruit during harvesting is a serious problem for commercial use of automatic harvesting robots. The gripper must hold a fruit firmly but the force shouldn't exceed the one necessary for holding the fruit to limit the possible damage, which would, in turn, decrease the shelf life of the harvested fruits or make it completely useless for selling;
- ✓ The ability to grasp occluded fruits and single fruits from clusters – as mentioned before, occlusions and dense growing complicated the automatic harvesting of sweet pepper. The designed gripper should be able to securely grasp a fruit even if it is partially covered by a leaf or if it is close to other objects, such as other fruits or a branch;
- ✓ The ability to operate in confined environment – most of the harvesting end-effectors are bulky, which makes them unsuited for working in narrow spaces. Sweet pepper fruits often grow between or behind branches from the harvester perspective, thus, the gripper would often have to operate in narrow spaces to be able to harvest such fruits;
- ✓ Low maintenance – a fully autonomous system should require a low maintenance as possible.

During the course of the research, however, it was concluded that any attempts to solve one of these requirements by a sophisticated multifunctional end-effector would undermine one or more of the other requirement. For example, by designing a multi-finger robotic gripper with the ability to grasp fruits of different sizes we are introducing additional

actuators, which increase the overall size and weight of the gripper. This, in turn, makes the gripper bulky and less suited for working in the confined environment. By introducing sophisticated sensors and finger mechanisms the requirement for maintenance and regular checks is increased. An autonomous system must be as robust as possible, which makes use of sophisticated gripper systems infeasible. As a result, it was decided to use the cutter shown in *fig. 1-4*, as it is very compact and robust, and to improve the harvesting algorithm using this end-effector.



Figure 1-4 Cutter – pincer end-effector, which was chosen for this study

New requirements for the automatic harvesting method were set by the use of the selected end-effector. The information about the position of a stem becomes crucial for a successful harvesting. New methods were necessary to develop as the visual cues were unreliable in detection of the stem. **This research concentrated on the development of new and unconventional methods for use in the calculation of the position of green pepper stem.** The harvesting problems described above were addressed as follows:

- Target detection – the algorithm developed during the previous research was revised and improved. A new target recognition algorithm was developed by using the existing knowledge gained from the results of the previous research, but not using the existing software as for this research the main data processing software was Matlab (Matlab R2013, MathWorks Inc.). The developed algorithm is mentioned in more details in the Literature Review section but no accent is put on it as it was based on already existing research.
- Stem position detection – the stem position information is critical for using the selected end-effector. A novel stem position calculation algorithm was developed,

which is based on fruit pose estimation in space and stem position detection by tactile sensing. This method involves two main steps – 1) estimation of the fruit pose in space and 2) stem detection by touch. The method developed for fruit pose estimation and approximate stem position calculation is described in great details in Pose Estimation section while the method developed for stem detection is discussed in the Touch Detection section.

- The Occlusion avoidance, Field of view problem and Positioning in the greenhouse issue were all addressed by developing a new harvesting robot prototype, which is described in the later section of this dissertation.

1.5 Organization of Dissertation

The structure of this dissertation is organized as follows:

Introduction – this section introduces the research topic, gives a brief description of the automatic harvesting in general and justifies the choice of the target of the current research – Japanese sweet pepper. Short description of the previous research and necessary improvements are given. The requirements for the current research are set and explained.

Robotics in Agriculture – this section introduces with the current advances of the automatic harvesting and gives a deeper insight into the topics touched in the introduction. First, description of the history and current developments of robotics in agriculture are given. Further, different technologies of main parts of automatic harvesting robots, grippers, in particular, are discussed. Machine vision and target detection are analyzed in greater details by discussing the currently used recognition methods and comparing them to the one used in this research.

Pose Estimation – this section introduces the developed method for pose estimation of a sweet pepper fruit in space and position calculation of the stem. The section is organized in historical order, first introducing with the first prototype of the pose estimation setup and then the second. Both hardware and software algorithms are discussed in great details. The method evaluation tests are described and the test results are analyzed.

Touch Detection – this section introduces a novel topic in automatic harvesting, fruit and stem detection by touch. The section should be viewed as consisting of two parts. The first part

about FSR sensor is connected to the advanced gripper development stage of the study and should be recognized as historically the oldest part of this research. It introduces the development of touch sensor for use in multi-finger harvesting gripper. The second part about the developed piezo sensor for stem detection is closely related to the pose estimation as the sensor was developed as means of verification for the pose estimation calculation result. Both sections, however, give a complete description of the used hardware and the working algorithm. Description of performance tests is given together with an analysis of test results.

Monorail Harvesting Robot – this section introduces a novel harvesting robot prototype for sweet pepper automatic harvesting. During the course of the research, it was recognized that the hardware used for the pose estimation algorithm is incompatible with the old harvesting robot prototype. Therefore, it was decided to design a new harvesting robot prototype that would implement both of the developed methods, pose estimation and touch sensing, and would also solve the problems with the old prototype mentioned in the previous research section. A full description of the new design is given in this section together with analyses of various parts and characteristics, such as power consumption and movement in a greenhouse.

Summary – this section gives a short summary of the results of this research. All pros and cons of the developed systems in the context of the set requirements are described and analyzed. Only main conclusions are drawn here as each of the main sections has their own conclusion chapter.

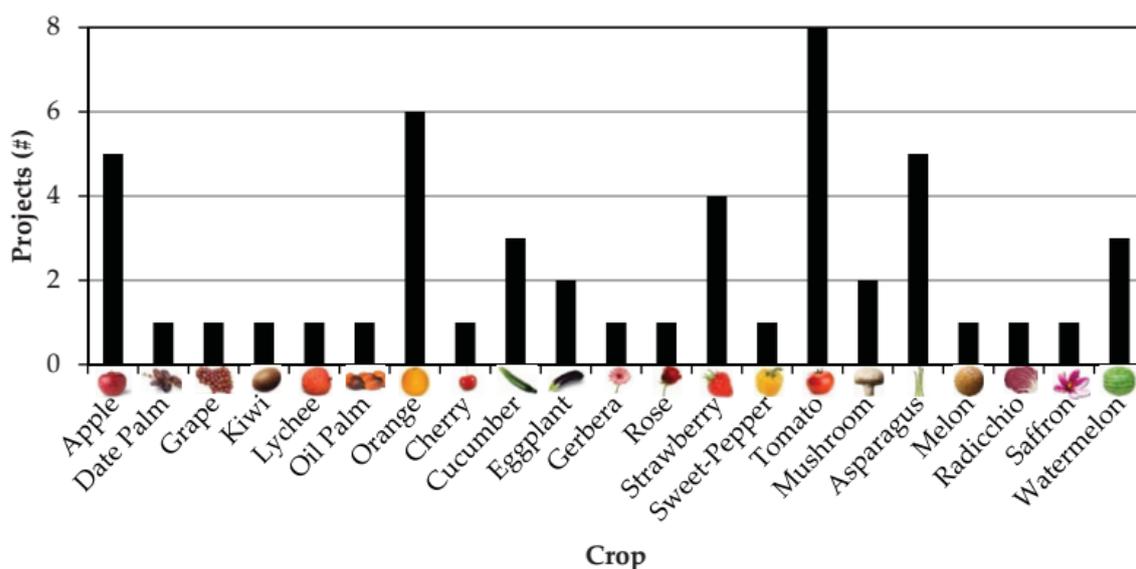
2. Robotics in Agriculture

2.1 Overview

Agriculture during the last few decades has been facing the problem of manual labor shortage. Nowadays farming is not a highly respected job by the young generation and therefore is rarely a career choice. As a result, the current farmer population is aging. The average age of farmers in Japan already is over 60 years old [18]. During the last two decades (1995 – 2015) the total area of the cultivated land has decreased by 12.1% (from 5.04 million ha to 4.496 million ha), the number of farming households has decreased by 59.6% (from 3.44 million to 2.155 million) and the amount of core agriculture workers has decreased by 46.2% (from 2.56 million to 1.754 million) [19]. With no one to replace the retiring farmers the agriculture in Japan in next few decades might have to face a serious crisis. Partial or full mechanization and automation of the agriculture processes is one of the most promising approaches to increasing the productivity of farming with the decreasing farmer population.

Mechanical harvesting systems are generally divided into two categories – mechanical harvesters and automatic harvesters. Mechanical harvesters achieve mass fruit removal by shaking the trees while automatic harvesters are designed for a single fruit harvesting method. The method used by mechanical harvesters is simple and efficient, especially when used in combination with chemical agents that help to loosen the mature fruits, but it also has several drawbacks, such as damage to the tree and fruits, harvesting both mature and immature fruits, not applicable to soft fruits, limited to only the tree type fruit harvesting and not usable in greenhouse. To solve these problems automatic harvesting was proposed by Schertz and Brown [20] in the early 1960s. The authors proposed individual-fruit harvesting as an alternative to mass harvesting to deal with the injury to trees and fallen fruits. Since then researchers worldwide have been working on automatic harvesting robots for different fruits and vegetables. A list of most advanced reported automatic harvesting robot projects for the

last 30 years (from 1982 – 2012) is given in [table 2-1](#) [21]. As it can be seen from [figure 2-1](#), the most attention has been given to tomato harvesting (8 of 50 projects), with orange picking robots being in the second place (6 of 50 projects) and apple and asparagus sharing the third place (5 of 50 projects). 21 of the listed 50 projects were for greenhouse automatic harvesting robot projects while the rest were for an orchard (16), indoor (2) and open field (11) harvesting. When divided by countries, Japan is the most active automatic harvesting robot developer with 15 out of 50 projects (30%), the USA follows with 7 projects and Italy with 5 projects.



[Figure 2-1](#) Harvesting robot projects from the last 30 years for different fruits and vegetables divided by the harvesting target (reproduced from [21]).

Automatic harvesting consists of three sub-tasks, target detection, secure target grasping and detachment of the target from the plant¹. A substantial error in any of these steps results in either unsuccessful harvesting attempt or damage to the fruit and/or the plant. Low harvesting success rate together with low speed compared to that of a human worker are the main reasons for reluctance to adopt automatic harvesting robots in commercial harvesting. The current research topics in automatic harvesting are aimed to the improvement of the performance of all of these three sub-tasks.

¹ Many researchers divide harvesting in different subtasks such as robot movement, recognition and manipulation [22] or recognition, manipulator movement and gripping tasks. I consider the movement of the robot itself and the manipulator to be issues of general robotics, therefore, the automatic harvesting specific tasks, in my opinion, are recognition, gripping, and detachment.

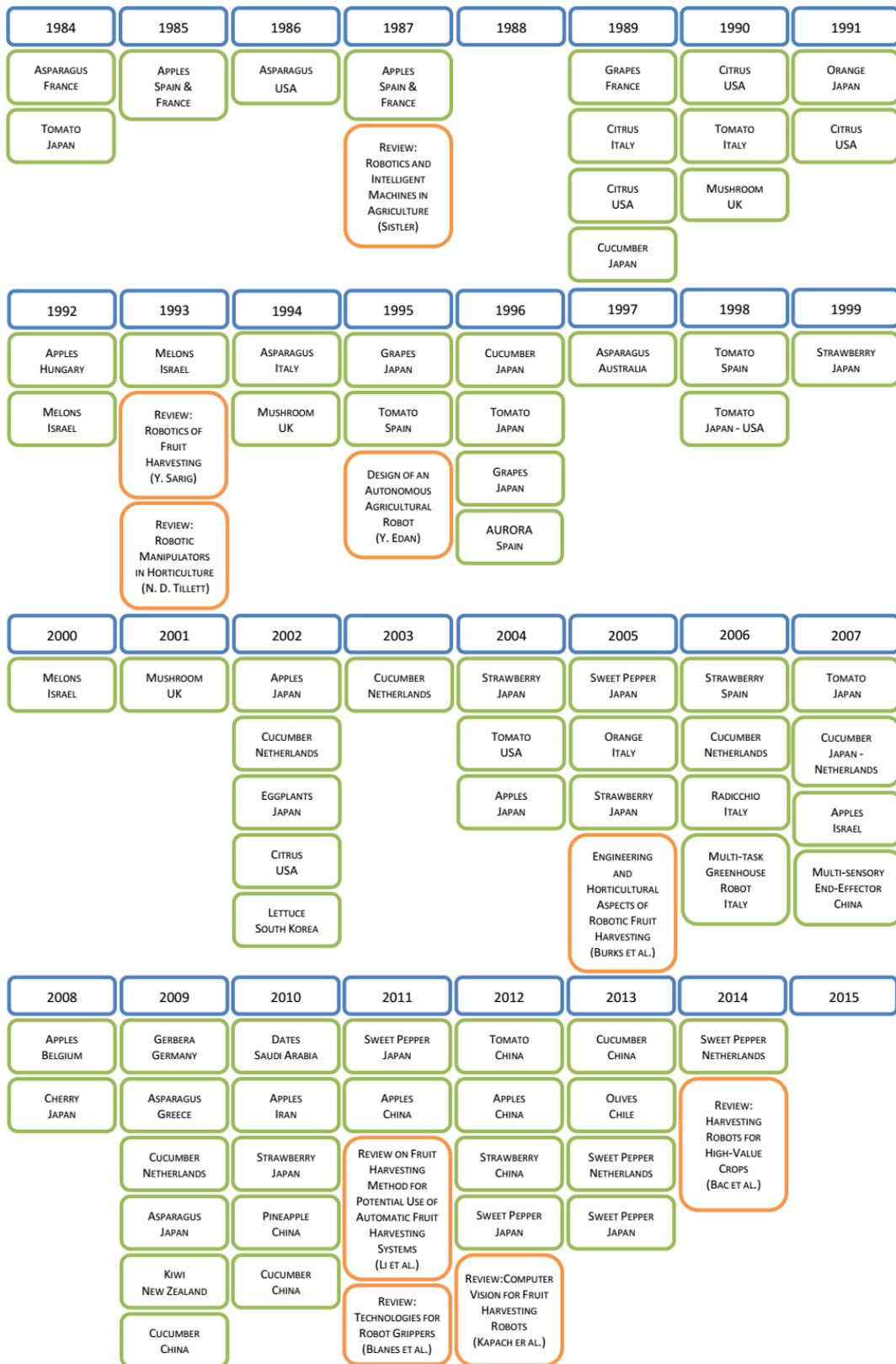


Figure 2-2 Timeline of research on automatic harvesting robots (green) and reviews of the field (orange) [22].

Table 2-1 Harvesting robot projects reported in the period 1982 – 2012 [21].

Environment	Crop	Location	Authors/Reference	Year	Autonomous
Orchard	Apple	China	Z. De-An, L. Jidong, J. Wei, Z. Ying, C. Yu	2011	Yes
		Belgium	J. Baeten, K. Donne, S. Boedrij, W. Beckers, E. Claesen	2008	Yes
		Hungary	L. Kassay, D. C. Slaughter	1993	No
		USA	P. W. Sites, M. J. Delwiche	1988	No
		France	A. Grand d'Esnon, R. Pellenc, G. Rabatel, A. Journeau, M. J. Aldon	1987	No
	Date palm	Saudi Arabia	A. A. Aljanobi, S. A. Al-Hamed, S. A. Al-Suhaibani	2010	No
	Grape	Japan	M. Monta, N. Kondo, Y. Shibano	1995	Yes
	Kiwi	New Zealand	C. Flemmer, R. Flemmer, A. Scarfe	2009	Yes
	Lychee	China	T. H. Liu, X. R. Zeng, Z. H. Ke	2011	Yes
	Oil palm	Thailand	A. Sittichareonchai, T. Khaorapapong, S. Limsiloratana	1995	No
	Orange	USA	B. S. H. Lee, U. A. Rosa	2006	No
		Italy	G. Muscato, M. Prestifilippo, N. Abbate, I. Rizzuto	2005	No
		Italy	A. Plebe, G. Grasso	2001	Yes
		Spain	R. Ceres, J. L. Pons, A. R. Jimenez, J. M. Martin, L. Calderon	1998	No
			A. R. Jimenez, R. Ceres, J. L. Pons	2000	
		France, Spain	F. Pla, F. Juste, F. Ferri	1993	Yes
USA		R. C. Harrell, P. D. Adsit, R. D. Munilla, D. C. Slaughter	1990	Yes	
		T. A. Pool, R. C. Harrell	1991		
Greenhouse	Cherry	Japan	K. Tanigaki, T. Fujiura, A. Akase, J. Imagawa	2008	Yes
	Cucumber	China	X. Tang, T. Zhang, L. Liu, D. Xiao, Y. Chen	2009	Yes
		Netherlands	E. J. Van Henten, J. Hemming, B. A. J. Van Tuijl, J. G. Kornet, J. Meuleman, J. Bontsema, E. A. Van Os	2002	Yes
		Japan	S. Arima, N. Kondo	1999	Yes
	Eggplant	Malaysia	W. I. Wan Ishak, W. H. Kit, M. A. Awal	2010	No
		Japan	S. Hayashi, K. Ganno, Y. Ishii, I. Tanaka	2002	Yes
	Gerbera	Germany	T. Rath, M. Kawollek	2009	Yes
	Rose	Netherlands	J. C. Noordam, J. Hemming, C. Van Heerde, F. Golbach, R. Van Soest, E. Wekking	2005	Yes
	Strawberry	Korea	K.-S. Han, S.-C. Kim, Y.-B. Lee, S.-C. Kim, D.-H. Im, H.-K. Choi, H. Hwang	2012	Yes
		China	Q. Feng, W. Zheng, Q. Qiu, K. Jiang, R. Guo, R.	2012	Yes
		Japan	S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, M. Kurita	2010	Yes
		Japan	F. Guo, Q. Cao, N. Masateru	2008	Yes
	Sweet pepper	Japan	S. Kitamura, K. Oka	2005	Yes
	Tomato	Japan	N. Kondo, K. Yamamoto, K. Yata, M. Kurita	2008	Yes
		USA	P. P. Ling, R. Ehasani, K. C. Ting, Y. Chi, N. Ramalingam, M. H. Klingman, C. Draper	2004	Yes

		Italy	F. Buemi, M. Massa, G. Sandini, G. Costi	1996	Yes
		Japan	N. Kondo, Y. Nishitsuji, P. P. Ling, K. C. Ting	1996	No
		Japan	S. Hayashi, O. Sakaue	1996	No
		France	S. Balerin, A. Bourly, F. Sevila	1991	Yes
		Japan	K. Namikawa, Y. Ogawa	1989	Yes
		Japan	N. Kawamura, K. Namikawa, T. Fujiura	1984	Yes
Indoor	Mushroom	UK	J. N. Reed, S. J. Miles, J. Butler, M. Baldwin, R. Noble	2001	Yes
		UK	J. N. Reed, R. D. Tillet	1994	Yes
Open field	Asparagus	Greece	A. P. Chatzimichali, I. P. Georgilas, V. D. Tourassis	2009	Yes
		Japan	N. Irie, N. Taguchi, T. Horie, T. Ishimatsu	2009	Yes
		USA	C. D. Clary, T. Ball, E. Ward, S. Fuchs, J. E. Durfey, R. P. Cavalieri, R. J. Folwell	2007	No
		Australia	G. Arndt, R. Rudziejewski, V. A. Stewart	1997	Yes
		USA	D. S. Humburg, J. F. Reid	1991	Yes
		USA, Israel	Y. Edan	1995	Yes
	Melon	USA, Israel	Y. Edan, D. Rogozin, T. Flash, G. E. Miles	2000	Yes
	Radicchio	Italy	M. M. Foglia, G. Reina	2006	No
	Saffron	Italy	M. G. Antonelli, L. Auriti, P. Beomonte Zobel, T. Raparelli	2011	Yes
	Watermelon	Japan	S. Sakai, M. Iida, K. Osuka, M. Umeda	2008	Yes
		Korea	H. Hwang, S. Kim	2003	Yes
		Japan	M. K. N. Tokuda, M. Suguri, M. Umeda, M. Iida	1995	Yes

2.2 Target Detection

Target detection is a machine vision task and the applied method depends on the harvested fruit and harvesting conditions. It can be called the bottleneck of automatic harvesting as even with a perfect gripping and cutting, the robot can't harvest more than it has detected. Extensive research has been done in the field to develop methods for fruit detection but practical success is still limited. Numerous challenges must be faced when developing a vision and detection algorithm for automatic harvesting. The main challenges are:

- *Color variations* – Color is one of the main features used for target recognition. Some fruits, such as apples, can have wide color spectrum even on a single ripe fruit. Other fruits, such as sweet peppers, some apple sorts, and cucumbers, have a color very similar to that of the surrounding foliage, which heavily complicates the detection.
- *Shape variations* – Shape of fruits such as sweet peppers and strawberries varies significantly from fruit to fruit, which makes use of shape feature unreliable. Even when applied to detection of round fruits, like apples, the shape feature detection

suffers from shape variations caused by occlusions and shadows, which distort the shape of the fruit in the image.

- *Reflectance* – Light reflected from the surface of fruits can affect the target detection as it appears as bright spots in the image. These bright spots can't be used for recognition as they don't carry any color or shape information related to the fruit.
- *Occlusions* – One of the most serious obstacles for both the recognition and gripping of target fruits as it is not easy to counter. The usual source of occlusions is leaves as they are the most common in foliage. Occlusions by other fruits and branches are also common. Removing leaves is a very labor demanding task, which, if done manually, would undermine the whole idea of automation.
- *Illumination and Shadows* – Amount of sunlight, the angle of the sun in the sky and the density of the foliage at the particular section influences illumination conditions shadows. The main feature used for recognition is color, which would be segmented by choosing an appropriate threshold value. In the case of changing illumination conditions, a global threshold value is not reliable for segmenting and should be adjusted depending on the ambient light. Shadows, on the other hand, can't be countered by changing the threshold of the detection as the change of color is local.

2.2.1 Visual Cues

Researchers have developed many visual detection methods and algorithms for fruit detection. From technological point of view the currently employed methods can be divided into following categories:

- *Single camera* – only one gray scale or color camera is used to obtain an image of the scene for analysis. The camera can be placed either on the frame of the robot [9, 23], on the end-effector [11] or in both positions [5]. A configuration of two cameras, both on the frame and the end-effector, allows the robot using the frame mounted camera for a general view while the end-effector mounted camera provides a closer view of the harvested target.
- *Stereo cameras* – the stereo configuration is one of the most common [5, 9, 11, 15, and many more] as it allows acquiring depth information from stereo images by a disparity map. Moreover, a fruit that is occluded for one camera might be better

visible for the second camera. Multiple cameras can also be used for handling illumination variations between the cameras [25].

- *Vision and range sensing* – it has been recognized in general machine vision that disparity maps often fail with providing an accurate depth information when it comes to handling images with a lot of similar patterns. Edges of leaves and branches in dense foliage can repeat frequently and typical disparity map algorithms, such as the Least squares, don't handle them well. Moreover, most of the disparity map algorithms fail when it comes to evenly colored surfaces of broad leaves and fruits, due to the correspondence problem [26]. Special disparity algorithms have been developed to handle weakly-textured images [27], but these approaches may miss details that can be crucial for harvesting such as occluding branches. Researchers have addressed this problem by using range sensors to measure the depth more directly [5, 28].
- *Spectral imaging* – researchers have recognized that certain fruits have different reflectance at certain wavelengths compared to that of the rest of foliage even if the color is very similar [16, 29, 30]. In such cases, spectral imaging provides with valuable information for target detection.
- *Hyperspectral imaging* – combining both standard color and spectral information can give an improved result compared to using each of these methods separately. The drawback, however, is an increase in acquisition and processing time of the image [38].

Where humans with little or no problems can pick out fruits that are not occluded or are partially occluded, machine vision faces serious difficulties. After an image of the scene has been acquired, special features or cues have to be extracted from the image to detect targets for harvesting. The most used features are color, spectral reflectance, texture, and shape.

Color

Many fruits change color when they turn ripe, like tomatoes and strawberries. In other cases, the color of fruits differs from that of the leaves regardless of the season. As a result, color is one of the most used visual cues when it comes to target detection both in the agricultural robotics and in the general machine vision. In agricultural robotics, the typically used colorspace is RGB [31 – 33] although other color spaces such as HSV [6, 11, 23] and L*a*b [34] are also employed. It has been recognized in the literature [35, 36] that color as a visual cue is very sensitive to illumination variations and shadows. The detection by using color is

performed by applying a threshold window to a single or multiple color channels, and pixels with a value in the range of the threshold are recognized as a fruit. Both shadows and change of illumination can change the value of pixels significantly enough for them to be outside the threshold window (see *fig. 2-3* for example). This is especially true for orchard and other outdoor applications, where movement of the sun provides a changing illumination, shadows from nearby objects and glare. Another drawback of color as the visual cue is unsuitability for fruits that have similar color with that of the surrounding environment such as green peppers, green apples, green citrus, cucumbers and other.

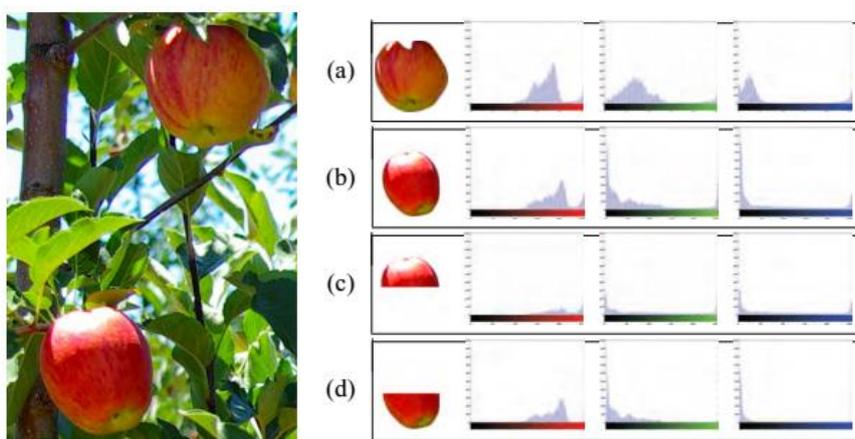


Figure 2-3 Difference in histogram values depending on the illumination conditions for the fruit (reproduced from [37]).

Spectral Reflectance

Spectral images are used when the color alone doesn't provide with a reliable way to separate a fruit from its surroundings. According to metamerism, objects with the same color don't necessarily have the same spectral signature (*fig. 2-4*). By using a specific light source or filters for the camera a quite different image can be acquired compared to the one visible in normal conditions. Ideally, one would use the entire spectral signature for analysis but such approach is both time and processing power consuming thus not suitable for real-time applications [38]. Narrow band filters are used instead to acquire an image at a certain, desired wavelength [9, 30]. Spectral imaging, however, has other drawbacks besides the increased acquisition and processing time. First, similar to the color cue, spectral reflectance is sensitive to the varying illumination conditions [9, 30]. Second, some parts of the foliage such as young leaves can have a similar spectral response to fruits [29]. Furthermore, both color cue and spectral reflectance suffer from issues related to occlusions.

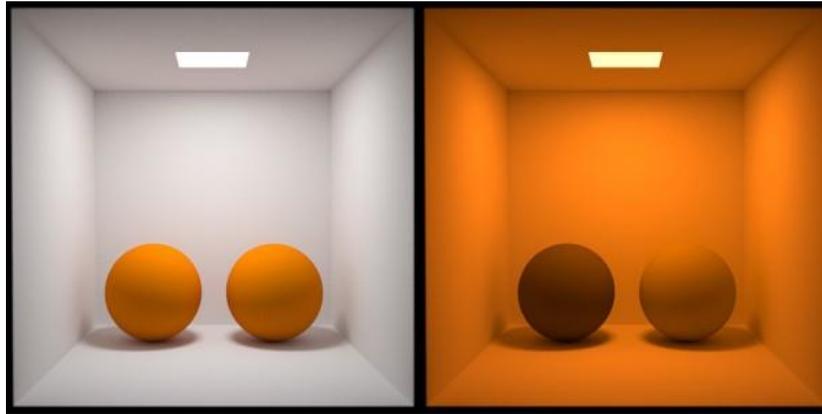


Figure 2-4 Example of metamerism, objects with the same color under normal illumination but with different spectral reflectance [<http://www.eclat-digital.com/metamerism/>].

Thermal Response

Thermal response is part of the spectral reflectance as it deals with the reflectance in the infrared range. The method is based on the fact that leaves accumulate less heat and emit it for a shorter time compared to fruits, and this difference can be measured and used for detection [36]. The thermal response method, however, has several significant flaws. First, the accumulated heat is directly proportional to the available sunlight during the particular time of the day. Furthermore, the position of a fruit and shadows from surrounding objects play an important role in how much heat each fruit will receive. During the day, there is a moment when both fruits and leaves have a similar thermal response and consequently, this method can't be used (fig. 2-5). Finally, both sides of the tree can't receive the same amount of sunlight simultaneously. As a result, *a priori* knowledge about the sunlight conditions and the position of the sun are necessary for a proper use of this method. Model of thermal response for all possible sunlight and weather conditions might be required to make this method feasible. As a result, this method hasn't received much attention in the literature.

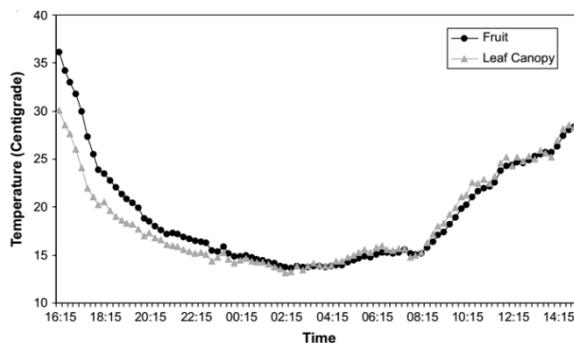


Figure 2-5 Thermal response of citrus fruits versus the rest of canopy (reproduced from [36]).

Texture

In a scenery where leaves and branches provide many edges, the relatively smooth surface of fruits can be easily distinguished from foliage by using edge detection techniques (*fig. 2-6*). This method works especially well in combination with other detection techniques. When combining color or spectral reflectance cue with edge detection, low edge density in the place of detection by another method will increase the confidence of correct recognition of a fruit [29, 39, 40]. Furthermore, it can be used to detect the real edges of the fruit in cases when the color feature provides an inaccurate segment edges due to self-shadowing of the fruit. Texture method is generally more stable in varying illumination conditions but doesn't work well with a sharp change of color such as in the case of multi-color apples or glare.

Shape

Most of the fruits are significantly different than the leaves in terms of shape. The shape is also the feature that is very robust to changes in illumination. As the shape extraction requires relatively high processing power, only quite recently the development of computers has allowed using shape analysis for real-time applications. This method is typically used for spherically shaped fruits [29, 41, 42] with some exceptions when the method was used for eggplants [43] and cucumbers [30]. Despite the benefit of robustness against illumination variations, use of shape is still limited as shape recognition is very sensitive to occlusions. In dense foliage conditions often fruits are visible only partially, which greatly changes the shape of the visible part of a fruit. This, in turn, deteriorates the result of shape recognition algorithms.



Figure 2-6 Example of texture extraction method using edge detection (Prewitt method).

Comparison between the original RGB image (left) and the extracted edges (right).

Combining Visual Cues

Only in rare cases a single cue gives sufficient information for accurate target detection. Ripe tomatoes, for example, have a significant difference in color compared to the foliage and therefore can be detected by using the color cue only. Automatic harvesting, however, has to deal also with fruits that have a color very similar to that of the surrounding environment and also might have to deal with unripe fruits. To increase the robustness of detection, multiple cues are used for recognition [44]. For example, a combination of color information and edge detection would allow dealing with fruit clusters that would otherwise be identified as a single object. Combining color with spectral reflectance and a shape related method might enhance the robustness when dealing with varying illumination conditions.

2.2.2 Image Analysis Methods

Methods of image analysis describe how exactly the visual cues are obtained from images to acquire information about possible fruit locations. Some of the most used methods are described here together with examples for a better understanding.

Thresholding

Applying a threshold is the fastest and simplest of image analysis methods so it is widely used in machine vision field in general and in automatic harvesting in particular. Threshold segmentation is performed by selecting a visual cue and deciding upon a value range for detection. If a pixel in the image has value within the selected range, the pixel is considered to belong to a fruit, otherwise, the pixel represents either foliage or background. The typical used visual cue for threshold segmentation is color, while also spectral reflectance [29] and texture [39] also can be used.

For example, to detect red apples in an image, we use *a priori* information that the color of target fruits is red for threshold value range. A binary image is created with pixel turned on (have a value of 1 or 255, depending on the format) in the same places, where pixels in the red channel of the original image are within the chosen threshold range, while the rest are turned off (have a value of 0). This mask shows where the detected fruits are and it can be used for further image processing to acquire the target positions.

Choosing a correct threshold range is a challenging task as the values of image pixels are heavily influenced by illumination variations as mentioned before. It is well known in general computer vision that use of a predefined global threshold is unreliable in situations

when a stable illumination can't be achieved [45]. Adaptive threshold methods have been proposed to deal with this issue, where the ambient illumination is used to adjust the threshold value [46, 47].

One of the main disadvantages of threshold segmenting, besides the challenge of choosing the right threshold value range, is ignorance to the shape information. A branch going over the middle of a fruit can "cut" the fruit in two separate segments in the image, which will be recognized as two individual fruits. In other cases, a cluster of fruits can be recognized as a single object rather than individual fruits. Combining threshold segmenting of the color cue with texture information could solve this problem.

Clustering

Clustering is a type of unsupervised computer learning method used to group pixels or regions of an image in segments so that pixels within any segment are similar while pixels across separate segments are different. By using one of the existing clustering algorithms pixels are segmented into three segments, fruits, foliage, and background. The most used clustering method in the field is *k-means* clustering as the number of existing clusters is known in advance [34, 48, 49]. Since clustering, in this case, uses the same information available to threshold segmenting, it also suffers from the same issues, such as illumination variations and ignorance to the shape information. Therefore, unless combined with other methods, clustering doesn't offer much of an improvement over threshold segmenting. Quite contrary, it has increased calculation time compared to the relatively very quick threshold segmenting. On top of that, it has been shown in the literature that clear separation between clusters of fruits, leaves, and background in some cases cannot be obtained (*fig. 2-7*).

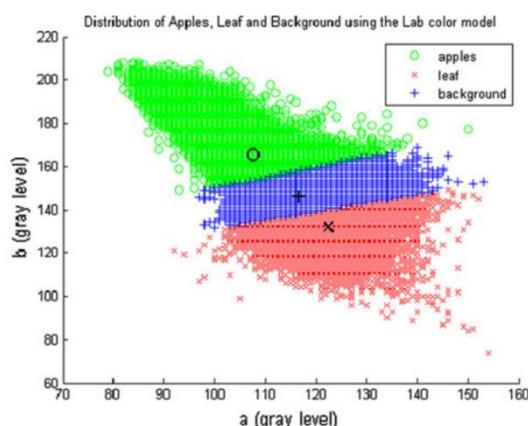


Figure 2-7 Result of *k-means* clustering on feature points, no clear separation between predefined classes of apples, leaves or background can be observed (reproduced from [34]).

Template Matching

Template matching is a method of image recognition where a specific pattern is compared to every part of the target image to find possible matches. The matching is done by moving the pattern over the image while the difference between the particular place in the image and the template is calculated. Typical approaches to calculating the difference are the sum of squared differences (SSD) and cross-correlation although specific features, like edges or corners, also can be used. A target is found if the difference between the pattern and a certain part in the image is under a predetermined threshold [41]. Template matching is not widely used in automatic harvesting as it is relatively time-consuming, which makes it unsuited for real-time applications. Moreover, variance in shape and size of fruits makes it difficult to find a good general model to use for matching.

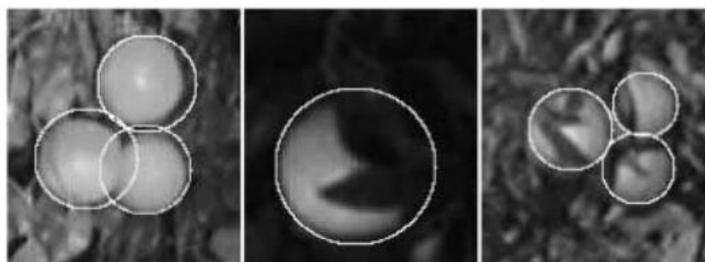


Figure 2-8 Shape of oranges reconstructed from the shape of edges, middle image shows a single occluded fruit while the right image shows occluded fruit cluster (reproduced from [25]).

Shape Inference

The shape is an important visual cue to be used by harvesting robots as most fruits have a distinct shape. Shape inference is a fundamental problem in machine vision, which is concerned with finding the most appropriate shape that fits some geometry found in an image. Points, lines, and other features are used to look for evidence if a particular shape is present. In a way, it is similar to the template matching method as it looks for the presence of a certain pattern in the image. Similarly, also the main difficulty lies in creating a robust enough shape model for fitting. Consequently, the currently existing shape inference methods are limited to spherical objects [25, 50] as modeling and fitting of a circle is mathematically relatively easy. One of the approaches described in the literature [25] employs adaptive edge fitting technique, where the spherical shape is detected by grouping edges in closed curves and labeling each curve as a separate object (see *fig. 2-8*). For each edge, the curvature and radius are calculated to estimate the position of a possible circle going through the edge. Once enough evidence for

the existence of a circle in a particular position is collected, the presence of a fruit is declared. The benefit of this method is the ability to calculate the shape of the fruit even if it is partially occluded.

Machine Learning

Machine learning is a subfield of computer science that deals with pattern recognition and computational learning. Machine learning algorithms build models from example inputs and later these models are used for making predictions on other data sets. Only three types of machine learning algorithms have been reported in the literature in the context of automatic harvesting, artificial neural networks (ANN) [34, 52, 53], support vector machines (SVM) [40, 51] and the already mentioned clustering algorithms. ANN method has been used for classification of pixels of different elements in an image such as fruits and background [52, 53]. The ANN is taught by “feeding” in input values, usually color features, to the input of the ANN while receiving binary classification in the output. The input values are examples of fruits and background, and the teaching phase is performed in a supervised mode to build a lookup table that matches each pixel to either fruit or background class [25]. A more sophisticated method was proposed in the literature [34], where three color spaces were used, RGB, HSV, and L*a*b, to classify pixels in predefined classes of apples, leaves, branches, ground and sky. The decision was made by majority vote.

2.2.3 Present Research

A sweet pepper detection algorithm was developed by S. Bachche and K. Oka [16]. This algorithm employed threshold segmentation method and both color and spectral segmentation were investigated. It was concluded that sweet pepper can be reliably detected either by color segmenting with an artificial illumination (success rate 84%) or during daylight by using infrared filter IR96 (success rate 84.35%). The segmenting itself was performed in the HSV color space V channel. During this research, however, it was discovered that the S channel is more reliable for target detection as V channel contained a lot of glare information from the artificial illumination source. Moreover, the pixel values of leaves and fruits in V channel overlapped thus complicating clear separation of fruits from leaves (*fig. 2-9*). Therefore, a new fruit detection algorithm was developed based on the information obtained from previously conducted research [15, 16].

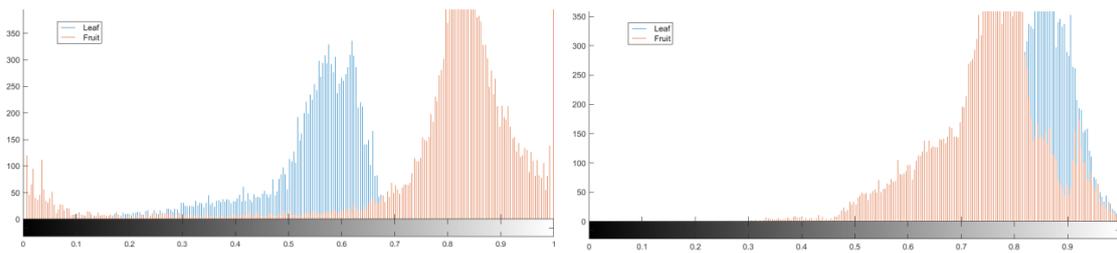


Figure 2-9 Comparing the difference between pixel values of fruit and leaf in HSV color space saturation channel (left) and value channel (right).

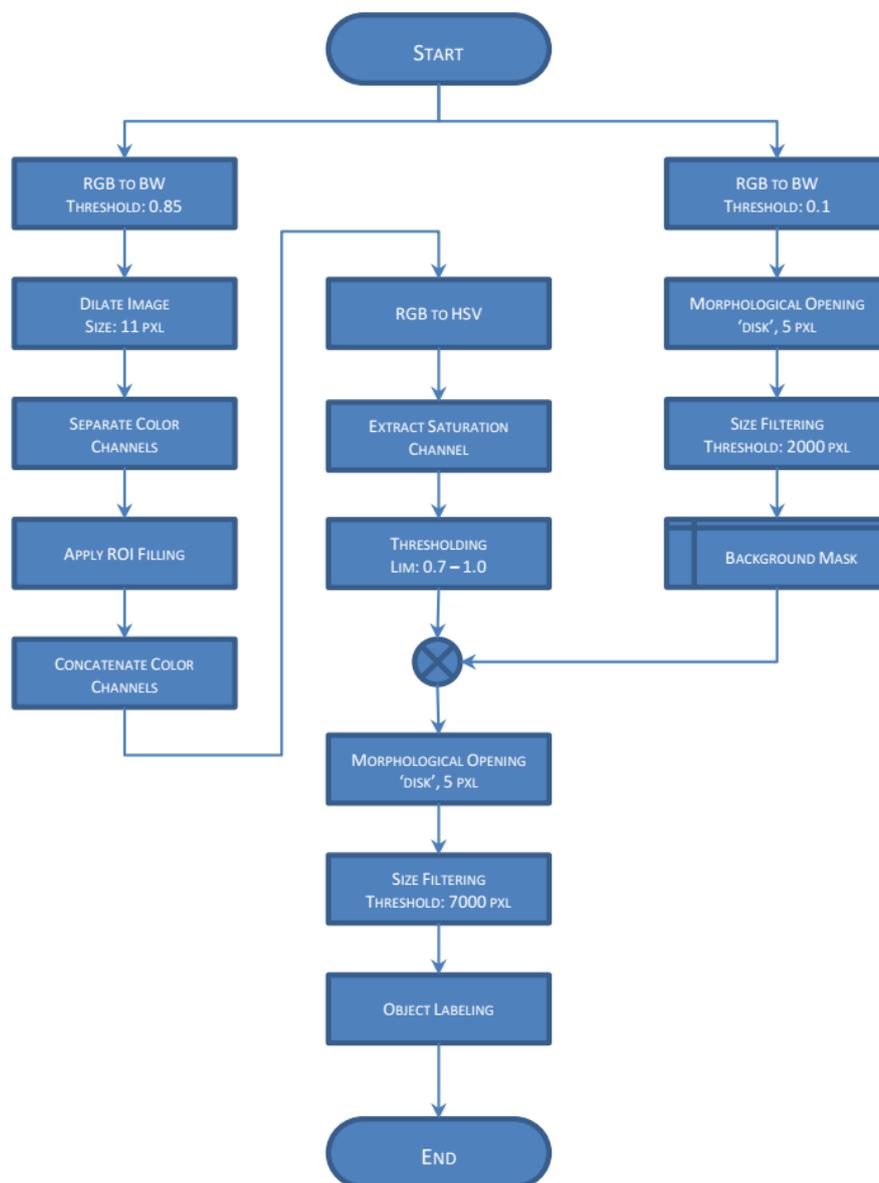


Figure 2-10 Recognition algorithm.

A single RGB camera was used in this research for image acquisition. The camera was calibrated according to the calibration method described in the literature [54] to obtain intrinsic parameters of the camera, such as focal length, principal point, skew coefficient, and distortion. The full algorithm of the target detection method used in this study can be seen in *fig. 2-10*. First, the background was removed from the acquired RGB image. The background information was extracted by first converting the color image to a grayscale format and then to a binary image with a threshold of 0.1. Afterward, the background was removed from the original image by applying a mask. This step is necessary as it was discovered that the background in HSV color space saturation (S) channel has similar value to that of fruits. This discovery doesn't pose much of a problem as the background is relatively easy to remove from the original image.

The next step deals with removing the glare from the image. Due to the highly reflective surface of the sweet pepper fruit, any light source creates a "hotspot", place on the pepper that reflects light to the camera. These hotspots must be removed as they interfere with the segmenting process. Such light reflection appears as a black region in the S channel and white region in the V channel. The hotspots are removed by performing smooth interpolation in the marked regions from outside inwards using pixel values from the borders of the hotspot.

Finally, the RGB image with removed background and hotspots is converted to the HSV color space. Only the saturation channel is used for segmenting in this algorithm as it has the greatest difference in pixel values between pixels of fruits and the rest of foliage. A wide threshold window was used in this research as other methods were used for removing false positives, such as size and position constraints. The lower threshold value was 0.5 and the higher threshold value was 1 (including). After Thresholding every segment with the size of <7000 pixels was removed to deal with the noise and small, immature fruits. Finally, the morphological opening was performed with a 20 pixels wide disk shaped structuring element to clean up borders of the detected fruit segments.

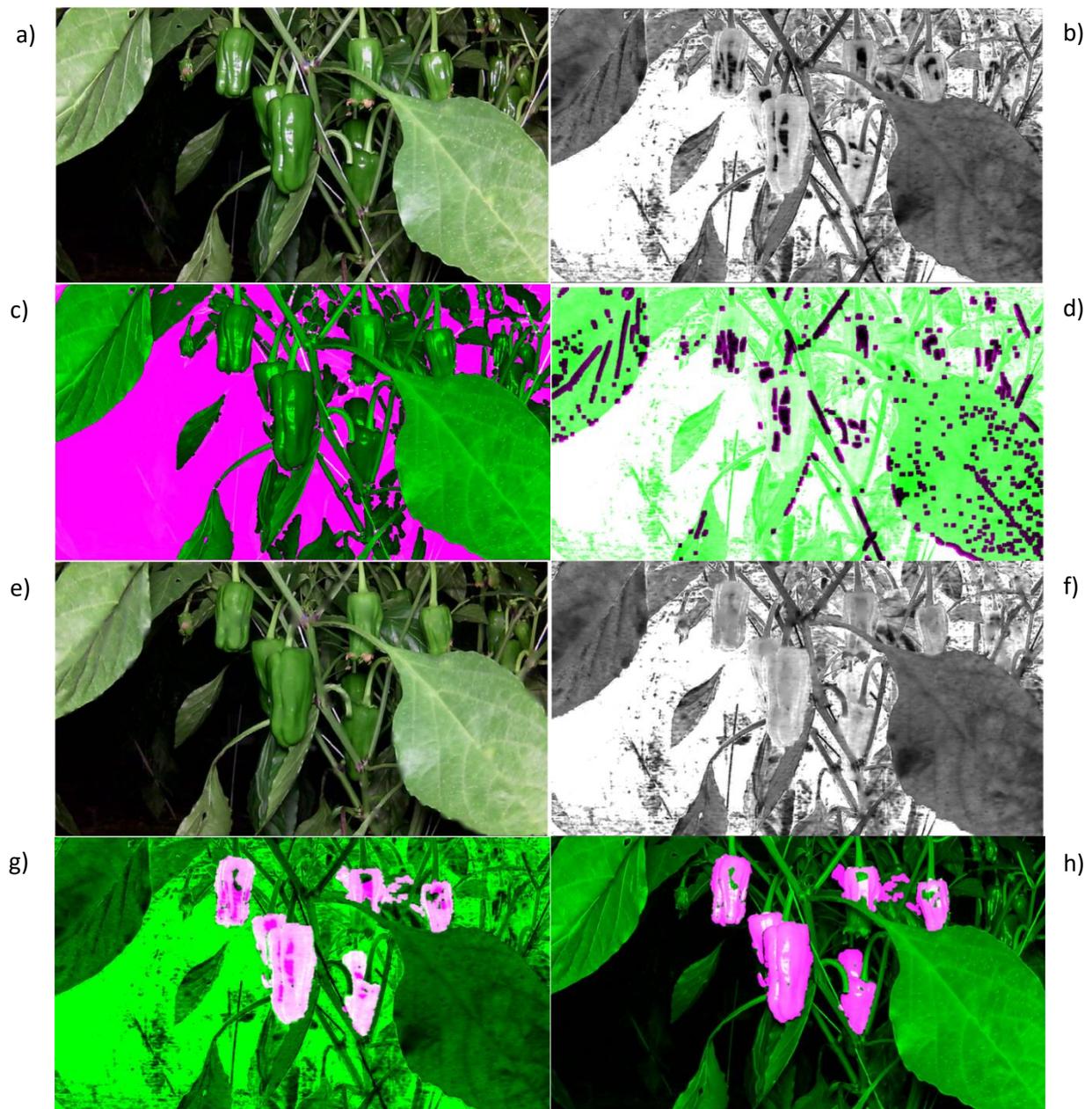


Figure 2-11 Example of recognition step by step: a) Original RGB image; b) HSV saturation channel; c) background (blue) that will be removed from the image; d) bright spots (red) that have to be removed; e) RGB image with removed bright spots; f) HSV saturation image with removed bright spots; g) saturation channel and h) RGB images after thresholding, all the closest sweet pepper fruits are detected.

2.3 Harvesting End-Effectors

End-effector is an important part of automatic harvesting robot as it is directly involved in removing a fruit from the plant. By using the wrong method it is possible to damage both the plant and the fruit during the harvesting process. An end-effector has to perform two tasks, secure gripping and detachment of the fruit from the plant. Basic requirements for harvesting end-effectors are:

- high-speed actuation;
- adaptation to different shapes;
- good gripping hold;
- minimal necessary pressure;
- gripping without damage to fruits;
- low maintenance;
- high durability;
- low weight;
- suitability for working with food products;
- low energy consumption.

Many different grippers have been developed during the years of research of automatic harvesting robots. This section will provide an insight into the most typical systems employed in harvesting robot end-effectors for gripping and detachment of fruits.

2.3.1 Gripping

Fresh fruits and vegetables are challenging for robotic manipulations due to their uneven shape and size. A significant size and shape variations can be observed even within the same category of fruits. Moreover, the fragile structure of fruit body makes them susceptible to bruising caused by the aggressiveness of harvest and postharvest processes. Studies have been done to evaluate the sensitivity of some fruits to bruising depending on their physical properties [55, 56]. Two typical manipulation strategies applied to fruit handling are contact gripping and air (table 2-2). Quite often both of these strategies are combined for a better performance.

Table 2-2 Most common manipulation strategies for agricultural robots [57].

Strategy	Method	Handling Ability				Damage Type				
		Gripping	Positioning	Orienting	Placing	Bruise	Tear	Break	Deformation	
Air	Vacuum	Suction cups	Yes	No	No	Yes	Low	Low	Low	Low
		Pipes	Low	No	Yes	No	Yes	Yes	Low	Low
	Pressure	Bernoulli	Yes	Low	No	Yes	No	Yes	Low	Low
		Blow	No	Yes	Low	Yes	No	No	Low	No
Contact	Gripper	Electric	Yes	No	Yes	Yes	Low	Low	Low	Low
		Pneumatic	Yes	No	Yes	Yes	Low	Low	Low	Low
		Hydraulic	Yes	No	Yes	Yes	Yes	Low	Yes	Yes
		Rubber	Yes	No	No	Yes	No	Low	No	Low
		Robot hands	Yes	No	Yes	Yes	Low	Low	No	No

Employing air suction cups is one of the most used methods for fruit holding and manipulation in automatic harvesting and postharvest processes. In this method, a fruit is held firm to a plastic or rubber suction cup by a pressure difference created by a pump inside the cup. For this method to work well the target fruit must have a thicker outer layer, which allows creating a vacuum, and a relatively uniform shape for the suction cup to fully cover. Therefore, it is usually employed for round fruits with relatively thick and smooth outer layer such as tomatoes [58] and apples, but it has also been used for fruits with relatively high shape variation, e.g. strawberries [11], cucumbers [9] and sweet peppers [59]. The main advantage of suction cup approach is the ease of implementation and combination with other gripping techniques, which for automatic harvesting is usually a contact gripping. The disadvantage, however, is poor performance on irregularly shaped, rough or dirty targets. Use of suction cups with multiple lips [60] and configuration of several suction cups [61] has been proposed for dealing with these fruit categories. Another drawback of vacuum suction mentioned in the literature is a possible damage to fruits with softer skin and high energy consumption of the vacuum pump.

Contact grasping strategy is the most used gripping method for automatic harvesting robots. A typical contact gripping end-effector consists of two or more jaw-like fingers, which are actuated either by an electric, pneumatic or hydraulic actuator(s). The electric actuators are fast, precise in terms of positioning and allow controlling the actuation force, but are more expensive compared to pneumatic systems and take up a lot of space on the end effector. Pneumatic actuators are easy to control with a valve, have high force and speed rating, low cost and high robustness, but the actuation typically is limited to full-open or full-close states.

Hydraulic actuators, on the other hand, allow easy speed and pressure control, but are too slow and heavy, and require a complex external power supply system. Therefore, hydraulic actuation method is rarely used for harvesting robots.

Hardware configuration-wise grippers for automatic harvesting employ two [17, 28, 58] or sometimes four-finger [23, 59] configuration. More fingers mean better grasping power but also bigger size. The size of the fingers can be important under dense foliage conditions as there might be no space near the fruit for fingers. Surprisingly, up to my best knowledge no publication in the field of automatic harvesting discusses methods on how to avoid gripping leaves or branches that are adjacent to the fruit to be harvested. None of the reported fruit detection algorithms describes a separate routine for verification if there is enough space next to the fruit for the fingers.

One of the flaws common to the currently existing harvesting robot end-effectors is the relatively large size, which together with the bulky industrial robotic manipulators, typically used for harvesting robots, makes them ill-suited for implementation of obstacle avoidance during harvesting. In case, when reaching inside the foliage is required, these end-effectors have a risk of getting tangled in the branches and leaves. Researchers tend to put a lot of additional electronics on the end-effector to aid with the detection process, this, in turn, makes the end-effector bulky and unsuited for working under dense foliage conditions.

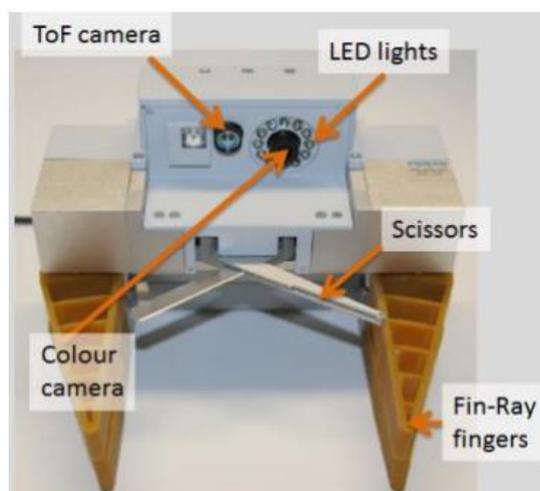


Figure 2-12 Example of a scissors type cutter with four-finger jaw gripper (reproduced from [59]).

2.3.2 Detachment

A detachment of a fruit from its plant is the third of main steps specific to automatic harvesting. Difficulties such as missing information of the stem position and occlusions complicate this task. Two general methods are employed, cutting of the stem or removing the fruit by twisting it until the stem breaks. The second approach is possible only for a limited number of fruits that are easy to detach when a certain level of ripeness is reached by the fruit. Typical examples for fruits that are harvested by twisting are tomatoes [58], apples [17] and citrus fruits [62]. Sometimes special abscission chemicals are used to weaken the connection of a fruit stem to the plant. The benefit of detachment by twisting is the ability to operate in the absence of information about the stem location as locating the stem in many cases is a complicated task. On the other hand, the danger of damaging the fruit and plant is present if the connection between the plant and fruit turns out to be strong as it is in the case of immature fruits. Also, the gripper must apply an additional gripping force to maintain a rigid grasp on the fruit while performing twisting, this, in turn, can damage the fruit if excessive force is used.

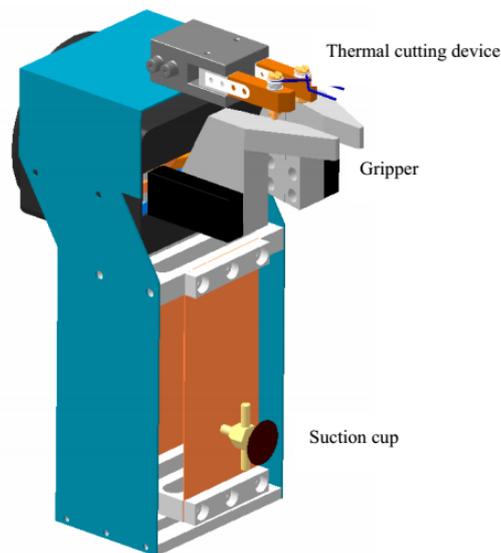


Figure 2-13 Example of a thermal cutter with a two-finger jaw gripper and a suction cup (reproduced from [9]).

Many different cutting systems have been developed based on the properties of target fruits. Most basic systems are scissor-like cutters [6, 59] that are attached to the end-effector (fig. 2-12). These cutters are easy to control, fast and reliable. The disadvantage of such design, however, is the fixed position of the cutter, which can turn into a problem if the stem after

securing the fruit is still outside the cutter range. Such situation is possible for fruits that have a tendency of slantwise growth such as bell peppers and sweet peppers. Another flaw that has been pointed out in literature is disease transfer between different plants [16, 22].

To solve this last problem, several researchers have proposed heated wire system for cutting the stem [9, 11, 16]. In these systems, a wire is heated to high temperature and then pressed against the stem. Heated wire systems require accurate information on the stem position as a contact between a fixed wire and the stem is necessary for cutting. On top of that, heating of the wire increases the energy consumption of the harvesting robot.

Moving lip cutter can be considered a subsection of scissor cutter systems [59, 63]. In this method, a movable semicircle ring(s) with attached blade can be rotated around the fruit until a contact with the stem is achieved. The benefit of this approach is the large working range as the blade will come in contact with the stem no matter how the stem is positioned. The main flaw is the inability to control the size of the blade. If the fruit is larger than the radius of the ring, the blade will damage the fruit. The blade can also damage any leaves, branches or other fruits that are in the path of the blade rotation.

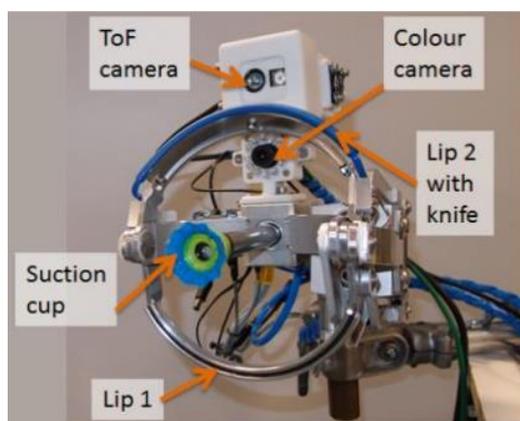


Figure 2-14 Example of a moving lip cutter with a suction cup (reproduced from [59]).

A laser cutting system for automatic harvesting robots has been introduced in the literature [28]. This system consists of two parts, a fiber-coupled laser diode, and a focusing lens. Although this configuration solves the disease transfer issue, it requires accurate information on stem position. Unfortunately, no information was provided in the literature about the maximum stem diameter that can be cut and the time required for cutting.

2.3.3 Current Research

From the analysis of a greenhouse of sweet pepper, it was concluded that the main problems for harvesting are dense foliage, occlusions by leaves and other fruits and slantwise growth of the fruits. To deal with such environment, the end-effector has to be as compact as possible to offer the possibility of reaching inside the foliage for harvesting. The size of the end effector is directly proportional to the complexity of the design and the amount of additional electronics mounted on the end-effector. To keep the end-effector compact it was decided to use a scissors-pincer system, similar to the one described in the literature [64].

The end-effector consists of scissors that work also as a two-jaw gripper and can perform cutting and gripping with a single motion. This particular design doesn't grip the fruit itself but the stem part that is left attached to the fruit after cutting. The main drawback of such design is the possibility of disease transfer between plants. A possible treatment for this issue is coating the active parts of the cutter with antimicrobial surfaces such as copper, silver or organosilanes. To evaluate the chosen end-effector design, it was compared to the requirements of harvesting robot end-effectors that were set in the beginning of this section:

<i>High-speed actuation</i>	Both cutting and gripping tasks are performed with a single motion. A high torque servo motor is used for actuation (Kondo KRS-6003HV, speed 0.22s/60 degrees at 11V power supply).
<i>Adaptation to shape</i>	Fruit shape variations are irrelevant as the fruit itself is not being gripped.
<i>Gripping hold</i>	Abrasive properties and hardness of the stem provides a good base for secure gripping by the stem.
<i>Minimum pressure</i>	No pressure on fruit is done.
<i>No damage to fruit</i>	No contact with the fruit is necessary under normal working conditions.
<i>Low maintenance</i>	No maintenance is required besides occasional checking of the cleanliness of the blade.
<i>High durability</i>	The end-effector is made of durable stainless alloys.

<i>Low weight</i>	No additional electronics or actuators are mounted on the end effector.
<i>Food handling</i>	Antimicrobial coating is required to provide protection against viruses, bacteria, and fungi.
<i>Low energy cons.</i>	Energy is consumed only at closing the end effector and to maintain the grip.

2.4 Conclusions

Automatic harvesting supposedly is the best option for solving the issue of decreasing farmer population in Japan and other countries. Various stages of automatic harvesting have been the subject of much research for the last few decades, but, despite the best effort of researchers worldwide, automatic harvesting robots still can't compete with skilled workers. In this section, a brief review was given of robotics in agriculture in general and development of automatic harvesting robots in particular. Special attention was paid to fruit detection strategies and basics of end-effector design as these two parts can be considered the bottleneck problem of the automatic harvesting. At the end of each particular subsection, a short description of the method used in this research is provided.

The methods for target detection and harvesting used in this research are relatively simple, when compared to existing methods reported in the literature, such as machine learning for detection, multi-finger robotic arm for grasping or laser cutting for the detachment. On other hand, guidelines for an automatic harvesting robot design [65] state that the methods employed should be as simple as possible to ensure robustness and simplicity.

3. Pose Estimation

3.1 Overview

The information about the pose of a fruit in space is important for several reasons. First, it allows calculating the optimal trajectory for the harvesting manipulator to approach the fruit. Secondly, an approximate position of the stem can be derived, assuming that basic information about the geometry of the fruit is available. The geometry information can be obtained from the visual information or from the width of the gripper if the fruit has been already grasped by the manipulator. Also, knowledge about the incline angle and direction of the fruit can help in case the manipulator is required to perform positioning of the fruit in place for cutting. Although the need of for pose estimation in the automatic harvesting has been pointed out in literature, up to the author's best knowledge no working algorithms had been published before the current research [37]. **The task of pose estimation is to give the information about the rotation and translation of an object in 3D space** with respect to the measuring system, in this case, the harvesting robot.

In the general machine vision, the pose estimation is usually acquired by registering a known 3D model over a 2D image by optimizing the rotation and translation parameters until a good estimate is found. This method requires a good 3D model and knowledge about the corresponding points in the 2D image. In automatic harvesting of sweet pepper, however, none of the required information is usually available. The irregularity of the shape of sweet pepper fruits makes it very complicated to generate a good universal 3D model. It is also hard to find all of the points in a 2D image that represents the found target fruit without adding extra points from adjacent objects or losing points that are outside the threshold value, such as dark shadows from surrounding objects or the fruit itself and bright areas caused by the

light source. Additionally, the great similarity between the color properties of a sweet pepper fruit and that of the surrounding foliage complicates an accurate target segmenting. These are the main reasons why it was decided not to use the conventional pose estimation methods. The method presented in this research, instead of performing model matching in the 2D image, performs a 3D point set registration by fitting a predefined general model on the point cloud that represents the surface of the fruit, measured by a laser range finder. The general algorithm works as follows. First, a target is found in the acquired image by using color segmentation in the HSV color space as described previously. As the next step, surface points are acquired for all objects in the area of interest. The obtained point cloud is segmented and filtered to retrieve only points that belong to the surface of the target fruit. Lastly, a model matching is performed by using a 3D point set registration method. The point set registration method finds the optimal fit for the predefined model to the measured data and provides us with a translation and rotation that are necessary for this fit.

3.2 Prototype I

3.2.1 Setup

The first prototype of the apparatus designed for the pose estimation of a sweet pepper fruit consisted of a 300mm long vertical slider (SA-S6AL-300-BE, Standard Units Supply Corp.), a CMOS USB webcam and a LIDAR type laser range finder (URG-04LX-UG01, Hokuyo Automatic Co. Ltd.) (*fig. 3-1*). The slider was controlled manually by the teaching box controller. The camera and the laser range finder were mounted on the slider. The vertical distance between the vertical center line of the camera image sensor and the laser measurement line was approx. 65mm and the horizontal distance between the sensor plane of the camera and the center of the laser was approx. 5mm. The vertical distance was verified by using a camera with the IR filter removed, and recording the position of the laser measurement line, and then the slider was moved until the marked line corresponded to the vertical center pixel of the camera (obtained from camera calibration).

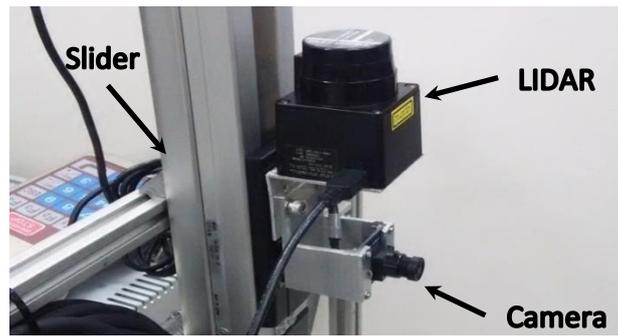


Figure 3-1 Pose estimation system Prototype I

The slider with the mounted measurement system was installed on a simple frame that could be moved in the greenhouse during testing (*fig. 3-2*). All data processing and control, with an exception for the slider positioning, was performed with a laptop by commercial software (Matlab R2013b, MathWorks, Inc.).



Figure 3-2 Pose estimation setup for the greenhouse test

3.2.2 Algorithm

The algorithm used for the first prototype consisted of 7 main steps (*fig. 3-3*). **(1)** First, the slider was moved until the target detection algorithm recognized a fruit in the image acquired by the webcam. After successful target recognition, the camera was moved to a position, from which the mass center of the detected target was approximately on the horizontal centerline of the camera in the image. This image was saved for **(2)** image processing step, where a more accurate segment of the detected target was acquired and the rest of the targets, if any, were removed from the image, leaving only the target of interest. After target registration, the slider was moved downwards to position the laser for **(3)** laser data acquisition. The laser was moved 190mm downwards to be in a position 125mm below the target center, and then the laser data was recorded for 250mm with 2.5mm step. **(4)** Next,

the acquired laser data was segmented with the DBSCAN (Density-Based Spatial Clustering for Applications with Noise) segmentation algorithm [66]. The DBSCAN algorithm sorts a 3D point cloud into separate clusters or noise by using two coefficients, epsilon (ϵ) and minimal points count (minPts), where the ϵ is the radius around the analyzed point and minPts is the number of points that is necessary to be within the distance ϵ for the point to belong to the cluster.

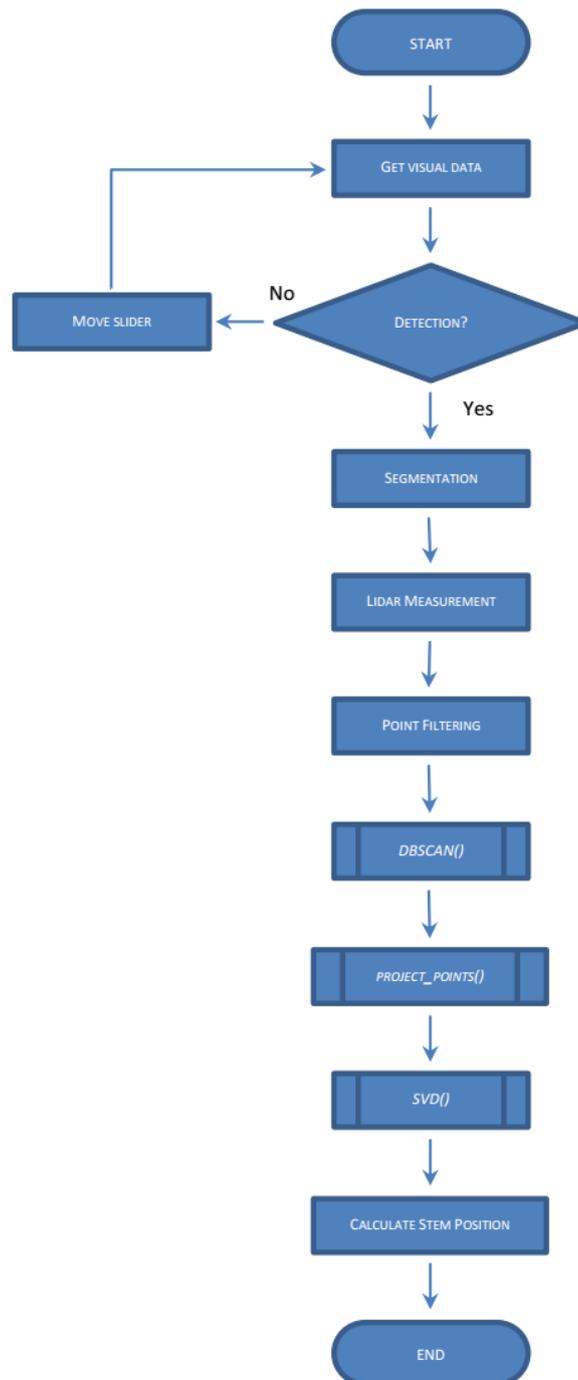


Figure 3-3 Pose estimation algorithm for the Prototype I

The coefficients used for this part of the study were 15 and 10 for ϵ and minPts respectively. These values were chosen by trial and error method by using several test data samples and varying the values to obtain the optimal result. **(5)** After clustering and noise removal each cluster was forward-projected on the target image by using intrinsic parameters of the camera, obtained from the camera calibration. Camera Calibration Toolbox for Matlab and a checkered pattern were used to obtain the intrinsic parameters, which are focal length (fc), principal point (cc), the skew coefficient (αc) and distortion coefficients (kc). The forward-projection is a calculation that takes a point in 3D coordinates and calculates the pixel position of the point on the image plane. The pixel coordinates were calculated as follows:

$$pixel_x = \begin{cases} \frac{nc}{2} - fc \left| \frac{P_Y}{P_X} \right|, & P_Y > 0 \\ \frac{nc}{2} + fc \left| \frac{P_Y}{P_X} \right|, & P_Y < 0 \end{cases} \quad (3-1)$$

$$pixel_y = \begin{cases} \frac{nr}{2} - fc \left| \frac{P_Z}{P_X} \right|, & P_Z > 0 \\ \frac{nr}{2} + fc \left| \frac{P_Z}{P_X} \right|, & P_Z < 0 \end{cases} \quad (3-2)$$

where

- $pixel_x$ = horizontal pixel location on the image
- $pixel_y$ = vertical pixel location on the image
- nc = number of columns in the image
- nr = number of rows in the image
- fc = focal length of the camera in pixels
- P_i = point P coordinate i in Euclidean space

The cluster with the most point within the area of the target segment in the image was chosen for the model matching. **(6)** The model matching was performed by using Euclidean or Rigid transform [67], which can be expressed as:

$$B = R * A + t \quad (3-3)$$

where

- A = the first dataset, model
- B = the second dataset, target
- R = rotation matrix
- t = translation vector

To find the R and t values, first both datasets have to be brought together by normalizing the centroids.

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (3-4)$$

$$centroid_A = \frac{1}{N} \sum_{i=1}^N P_A^i \quad (3-5)$$

$$centroid_B = \frac{1}{N} \sum_{i=1}^N P_B^i \quad (3-6)$$

where

- P_A = points in the dataset A
- P_B = points in the dataset B
- N = size of the dataset

The Singular Value Decomposition (SVD) method is used to calculate the rotation matrix R . First, a covariance matrix H is calculated using equation 3-7, and then the rotation matrix is calculated by equation 3-9.

$$H = \sum_{i=1}^N (P_A^i - centroid_A)(P_B^i - centroid_B)^T \quad (3-7)$$

$$[U, S, V] = SVD(H) \quad (3-8)$$

$$R = V * U^T \quad (3-9)$$

The calculated R value must be checked for reflection case, which gives a numerically correct but in terms of actual rotation false values. The verification is performed by calculating the determinant of the matrix R and correcting the matrix by changing the sign of the last column in case the determinant is found to be negative:

$$R = \begin{cases} R, & \text{if } |R| \geq 0 \\ R * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \text{if } |R| < 0 \end{cases} \quad (3-10)$$

Finally, the translation t was found as follows:

$$t = -R * centroid_A + centroid_B \quad (3-11)$$

The rotation is calculated as 3x3 rotation matrix $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ and the translation is a

3x1 vector $t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$.

Once the rotation R and translation t are determined, the position of the stem can be approximately calculated:

$$stem = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = R * \begin{bmatrix} 0 \\ 0 \\ l + \delta \end{bmatrix} + t \quad (3-12)$$

where

- s_i = stem cutting point position in Euclidean space
- l = length of the fruit, [mm]
- δ = distance from the top of the fruit to the cutting point, [mm]

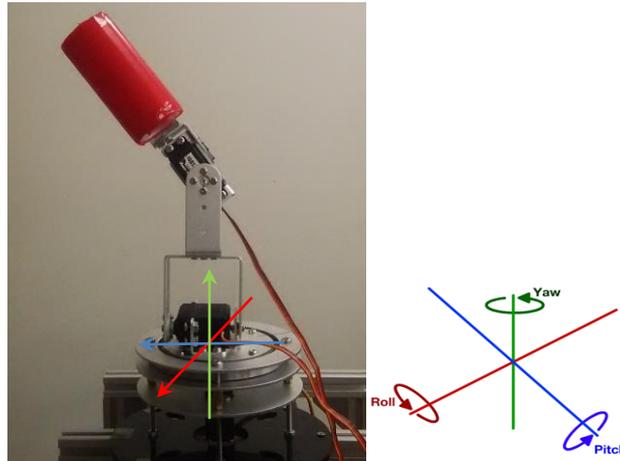


Figure 3-4 Target setup for the laboratory experiment

3.2.3 Performance Evaluation

The performance of the developed pose estimation method was evaluated by two separate tests. The first test was performed under laboratory conditions by using the developed method on a simple cylindrical object. The cylinder was attached to a 2DOF manipulator, which allowed positioning the cylinder at any incline angle (*fig. 3-4*). The predefined 3D model used for this experiment was the surface of a cylinder. The incline of the test target was set to 0°, 15° and 30° for both roll and pitch angles. As a result, measurement for 9 different angle configurations was acquired and analyzed. The aim of this experiment was

to determine the rotation angle calculation error of the method. Rotation angles were calculated from equations 3-13, 3-14 and 3-15 and were compared to the set values for evaluation.

$$\theta_{roll} = atan2(r_{32}, r_{33}) \quad (3-13)$$

$$\theta_{pitch} = atan2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}) \quad (3-14)$$

$$\theta_{yaw} = atan2(r_{21}, r_{11}) \quad (3-15)$$

where $atan2(y, x)$ is a two argument arctangent function that deals with the appropriate quadrant determination problem and the division by zero.

The results of the experiment can be seen in the table 3-1. The highest measured error was 6.25° for 75° roll and 60° pitch angle setup when the calculated roll angle was 81.25°. No evident relationship between the incline angle and the error value was observed within the tested incline angle range. An error of 6.25° for an 80mm long fruit ($\delta = 10mm$) would give stem position error of approx. 9.8mm, which is within the acceptable range for a cutter with 30mm opening width. Yaw angle was ignored as it doesn't influence the result for a symmetrical shape such as a cylinder.

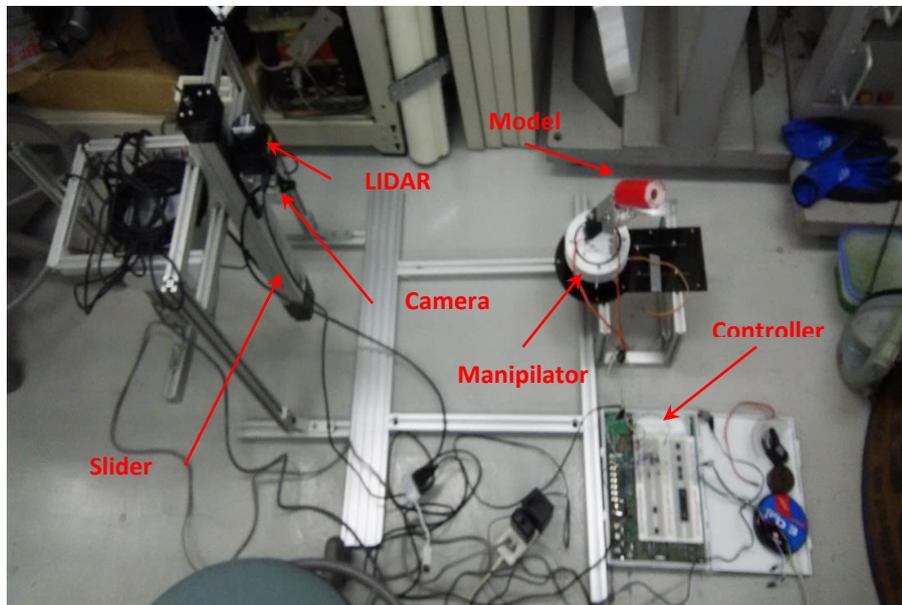


Figure 3-5 Setup for the laboratory experiment

Table 3-1 Results of the laboratory test

Set Angle, deg.		Calculated Angle, deg.		Error, deg.	
Roll	Pitch	Roll	Pitch	Roll	Pitch
90	90	89.57	95.48	0.43	-5.48
90	75	85.31	80.67	4.69	-5.67
90	60	89.88	80.67	0.12	-4.06
75	90	79.36	93.28	-4.36	-3.28
75	75	78.12	77.96	-3.12	-2.96
75	60	81.25	63.09	-6.25	-3.09
60	90	57.25	94.65	2.75	-4.65
60	75	63.31	78.49	-3.31	-3.49
60	60	61.88	64.72	-1.88	-4.72

The second test was performed in a greenhouse by acquiring a measurement of real sweet pepper fruits under the expected working conditions of the system. The measurement of several fruits was performed and the data was analyzed according to the above-described algorithm. The model used for this experiment was a half of truncated cone surface with 80mm height and 50 and 30mm top and bottom diameter respectively. While the calculated translation was in accordance with the laser measurement information, accurate evaluation of the calculated angles in for greenhouse measurements is complicated. A qualitative assessment was done for the angle accuracy by plotting a fruit model over the laser measurement and evaluating the match visually (*fig. 3-6*). It was concluded that the obtained results were reasonably accurate. The calculated angles for the analyzed fruits shown in *fig. 3-6* were 88.68° and 84.44° for roll and pitch angles of the fruit 1 respectively, and 71.04° and 82.44° for roll and pitch angles of the fruit 2 respectively. The calculated translation vectors and rotation matrices for fruits 1 and two were:

$$T_1 = \begin{bmatrix} -103.54 \\ 72.12 \\ 452.07 \end{bmatrix} mm$$

$$T_2 = \begin{bmatrix} -132.79 \\ 104.63 \\ 488.56 \end{bmatrix} mm$$

$$R_1 = \begin{bmatrix} 0.9953 & 0.0001 & 0.0968 \\ 0.0022 & 0.9997 & -0.0229 \\ -0.0968 & 0.0230 & 0.9950 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 0.9912 & 0.0315 & 0.1283 \\ 0.0117 & 0.9462 & -0.3234 \\ -0.1316 & 0.3220 & 0.9375 \end{bmatrix}$$

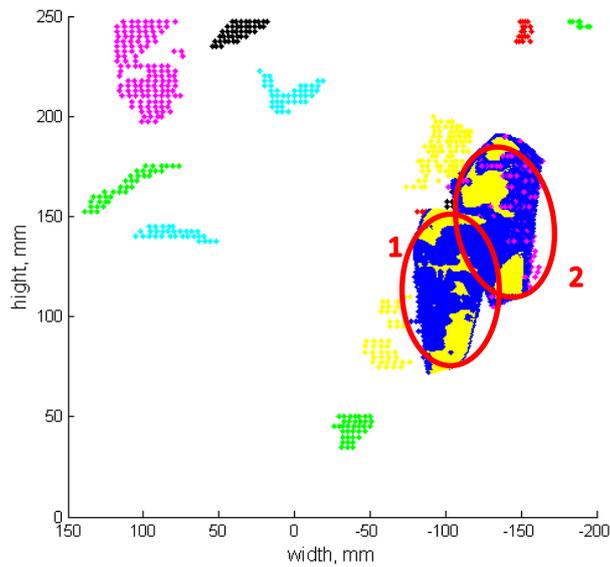


Figure 3-6 LIDAR point clouds after filtering and pattern matching, front view

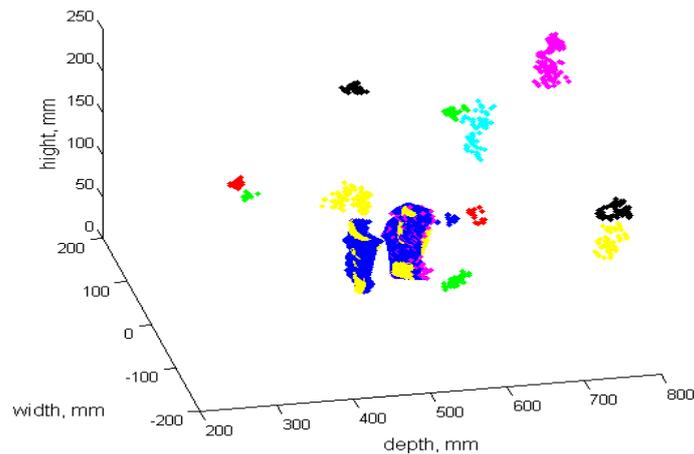


Figure 3-7 LIDAR point clouds after filtering and pattern matching, isometric view

3.2.4 Conclusions

A novel pose estimation method was developed and investigated. The proposed method calculates the pose of a fruit in space by using a model matching algorithm. The SVD algorithm was used to find the optimal translation and rotation, at which a predefined model fit the laser measurement of a sweet pepper fruit. Two tests were performed to evaluate the performance of the developed method, under the laboratory conditions and in a greenhouse. The laboratory test showed a good accuracy for the angle measurement. The max calculated angle error was 6.25° for the roll incline angle set to 75° . The method also proved acceptable performance during the greenhouse test. The DBSCAN algorithm was able to separate both clusters representing the detected sweet peppers. The method used for right cluster selection

proved to be reliable as both correct clusters were chosen. Qualitative analysis of the greenhouse test result showed that the calculated angle is relatively close to the actual. It was concluded that the developed method can be used for sweet pepper fruit pose estimation in automatic harvesting.

Despite the reasonably good performance of the system, several issues were detected. First, the accuracy of the laser range finder was proven to be unsatisfactory. Considering that the diameter of a sweet pepper is approx. 40mm, an accuracy of ± 30 mm can lead to a significant distance error. On top of that such accuracy leads to great slice-to-slice variations, which in turn deteriorate the angle calculation result. A more accurate laser range finder is necessary to solve this problem. An additional cause of inaccuracy in the laboratory test might be misalignments of the test platform. The measurement system and the target positioning system were two separate parts that were aligned without precision instruments, by using only a ruler, a protractor, and a bubble level, so the presence of alignment errors is likely.

The DBSCAN algorithm was able to cluster different objects in separate clusters as long as there was a reasonable distance between them. The algorithm was unable to separate objects that were touching each other, such as two overlapping leaves, a leaf covering a fruit and two closely touching fruits. Choosing different coefficients might be able to solve this problem, considering that the overall quality of the laser measurement is improved beforehand. If the coefficients will be chosen too small without improving the laser measurement accuracy, the clustering algorithm will make many small segments for a single object.

The speed of the main data processing parts, such as DBSCAN clustering algorithm and SVD model matching algorithm, depends on the number of points involved. For example, the worst case run complexity for DBSCAN is $O(n^2)$, where n is the size of the dataset, as the DBSCAN can visit each point in dataset multiple times, if the point is part of different clusters. Each laser measurement returns 682 points so the size of dataset for the greenhouse test was 68 200 points. For comparison, using DBSCAN to cluster a full dataset of 68 200 points required 654.83s to calculate, while the same calculation after removing unnecessary point and obtaining dataset of 6422 points required only 4.61s. This suggests that all unnecessary points must be removed before further data processing.

3.3 Prototype II

3.3.1 Setup

A second pose estimation system was built based on the conclusions drawn from the experience with the first prototype (*fig. 3-8*). Several modifications were implemented to improve the performance and to make the system closer to the final version to be used in the harvesting robot. First, the length of the vertical slider was changed from 300mm to 1200mm by developing a new slider. The length of 1200mm is the approximate intended working range based on the used greenhouse setup. The new slider consisted of a stepper motor, sliding structure (RSR40-1200, Misumi Group Inc.), aluminum base and a timing belt with pulleys. The control of the slider was changed from manual to fully automatic by using an Arduino Uno R3 microcontroller board with Adafruit Motor Shield V2 for driving the stepper motor. The movement accuracy of the slider was approx. 0.4mm, and the micro stepping feature was used for displacements less than 10mm to ensure smooth movement. The stress test was performed by attaching an extra weight equal to twice the weight of the intended load to evaluate the performance of the slider. No step skipping was observed so it was decided that use of an encoder for the position verification is unnecessary. A limit switch was used instead to verify when the slider has reached the home position during initialization.

The second major improvement was a change of the used laser range finder. The used laser with an accuracy of $\pm 30\text{mm}$ (URG-04LX-UG01) was changed to a similar model with the accuracy of $\pm 10\text{mm}$ (URG-04LX). In addition, the reading of the laser was changed from the Matlab built-in serial object interface to an MEX code (CPort mini toolbox, developed by Eyal Doron [68]) that used low-level C code for communication with the serial port. This modification improved the communication speed from 1.2s to 0.3s per reading, thus greatly improving the measurement speed.

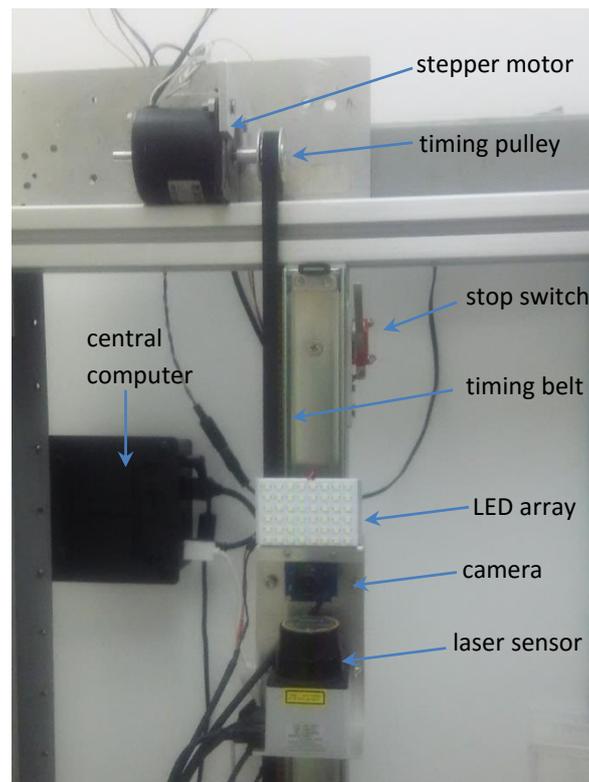


Figure 3-8 Slider setup for the second prototype

Besides the mentioned major improvements, several smaller changes were made. An LED array was mounted on top of the camera to allow the system to take measurements during night time, which is the planned working time of the system. A low power computer (Gigabyte Brix GB-BXBT-2807, Giga-byte Technology Co., Ltd.) was used as the main control and data processing unit instead of a laptop. The chosen computer model works on 9W power and is well suited for the particular application as the robot is meant to work on battery instead of an external power supply.

3.3.2 Algorithm

The proposed algorithm can be divided into two major parts, target detection on the left side of the flowchart and model matching on the right side (*fig. 3-9*). First, searching for targets is performed by moving the camera in a vertical direction. Image acquisition and processing are performed after each 10mm of the slider movement. HSV color space saturation band is used for target recognition as it has the most significant difference between gray values of sweet pepper fruits and those of leaves. The measurement is performed at nighttime with artificial illumination from an LED array to eliminate changing illumination from the dynamic sunlight conditions. The background has to be removed from the RGB image as it

has similar saturation value to that of a fruit when the image is converted to the HSV color space. A wide threshold window is used in the HSV image for fruit detection.

Size and shape constraints are used to remove false positives resulting from the wide threshold window. The pixel of the mass center is calculated for each recognized target in an image and only targets with a mass center within the distance of 30 pixels in the vertical direction are registered. Having the mass center of a target at the same vertical position in the image as the central pixel of the camera indicates that the target and the camera are at the same height, which will be used for the laser measurement. The center pixel of the camera together with other intrinsic parameters was obtained using a camera calibration toolbox in Matlab designed by Jean-Yves Bouguet [54]. Fruit mass center pixel, pixel list of the fruit segment, position of the slider at the moment of image acquisition and the acquired image, in which the target was detected, are saved for each registered target in a target feature list. A check is performed for each detected target to verify that it is original and was not previously registered by comparing the horizontal position of the new target to those of the targets registered in the last three steps. The choice for three step difference was based on the assumption that two mature sweet peppers are unlikely to have mass centers in roughly the same horizontal position and with vertical position difference less than 30mm. In the case of an already registered target, the mass center pixel distance to the camera center pixel for the new target is compared to that of the old one, to determine if the new registration is closer to the center. The feature list of the previous detection is overwritten if the new position is decided to be closer to the image center. The laser measurement is performed for each registered target in a $\pm 50\text{mm}$ range around the found center position with 2mm step in the vertical direction. In the case of a measurement range overlap for different fruits, only one measurement is done and the result is saved in the feature list for each fruit at the particular height. Once the laser measurement for all registered targets has been done, the acquired distance points are first processed by removing points outside the harvesting area. The laser range finder used in this study has a measurement range up to 5.6m in 240 degrees wide angle (step angle approx. 0.3615°), while the harvesting area is 400mm wide ($\pm 200\text{mm}$ from the center) and 800mm long zone in front of the harvester. The points outside the harvesting area are discarded to reduce the time consumption for data processing in the next steps. Point segmentation and identification is necessary before the model matching can be performed. The target surface points are identified by projecting each of the remaining laser rangefinder

measurement points on the image plane of the camera and keeping only points corresponding to the target segment.

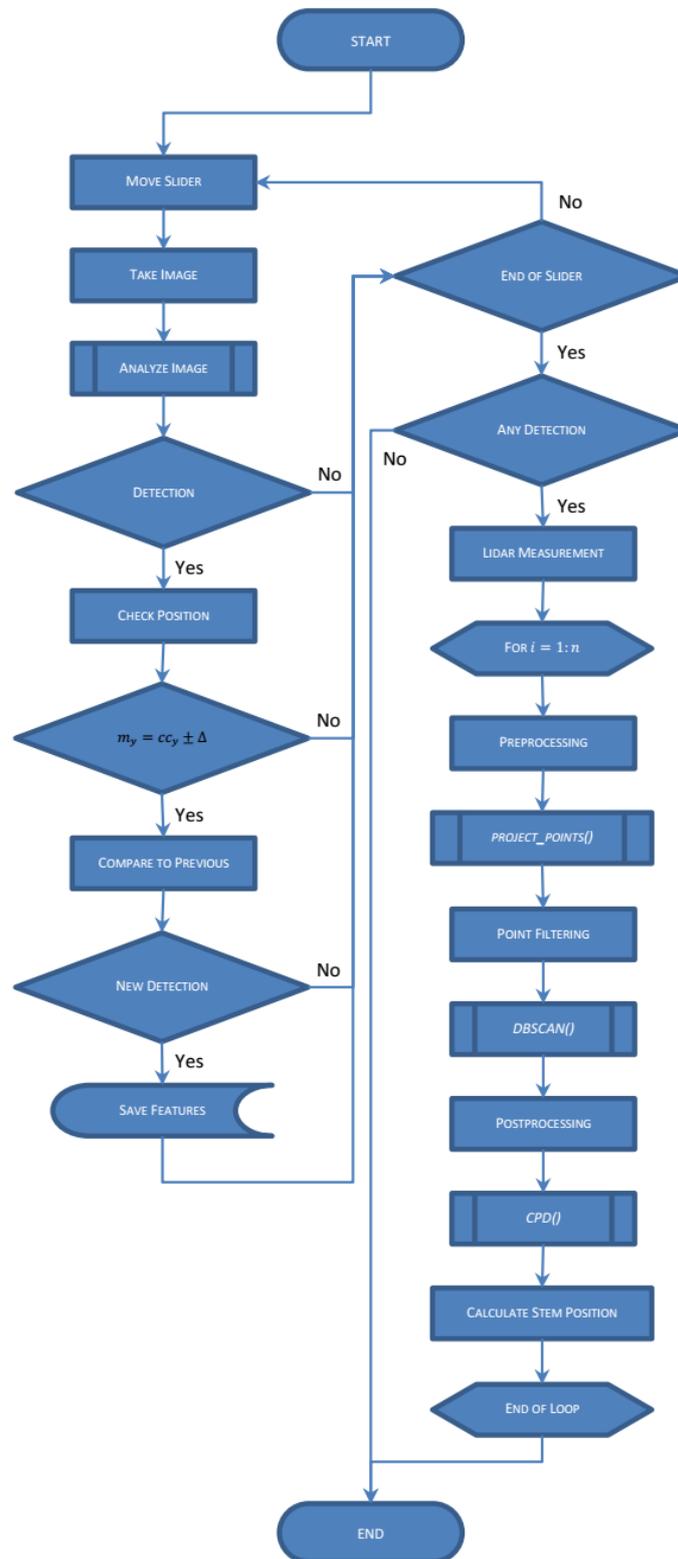


Figure 3-9 Pose estimation algorithm of the Prototype II

Forward projection is performed using a function from camera calibration toolbox `project_points2()`, which works as follows [54]. Let's assume that we have a point in the world coordinate frame $P = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$. The same point transferred to the camera coordinate frame can be described with the following equation:

$$C = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = R_c * P + T_c \quad (3-16)$$

where

- R_c = rotation matrix showing the rotation of the camera coordinate frame with respect to the world coordinate frame
- P = the point coordinates in the world coordinate frame
- T_c = translation vector showing the difference between the origins of the world coordinate frame and that of the camera

The pixel coordinates PX of pinhole projection of the point P on the image plane are calculated as follows:

$$px = \begin{bmatrix} px_x \\ px_y \end{bmatrix} = \begin{bmatrix} \frac{c_x}{c_z} \\ \frac{c_y}{c_z} \end{bmatrix} \quad (3-17)$$

Due to lens misalignment and production imperfections, radial and tangential distortion exists and must be included. The normalized point coordinate after including the lens distortion is calculated by

$$px_d = \begin{bmatrix} px_{dx} \\ px_{dy} \end{bmatrix} = (1 + kc_1 r^2 + kc_2 r^4 + kc_5 r^6) px + dx \quad (3-18)$$

where

- r = a vector from the image origin to the normalized image projection
- kc = distortion matrix (from intrinsic camera parameters)
- dx = tangential distortion vector

Values r and dx can be calculated from equations 3-19 and 3-20:

$$r = \sqrt{px_x^2 + px_y^2} \quad (3-19)$$

$$dx = \frac{2kc_3px_xpx_y + kc_4(r^2 + 2px_x^2)}{kc_3(r^2 + 2px_y^2) + 2kc_4px_xpx_y} \quad (3-20)$$

The final pixel coordinates are calculated by multiplying the normalized (distorted) coordinate vector, acquired from equation 3-21, with a camera matrix KK :

$$\begin{bmatrix} pixel_x \\ pixel_y \\ 1 \end{bmatrix} = KK \begin{bmatrix} px_{dx} \\ px_{dy} \\ 1 \end{bmatrix} = \begin{bmatrix} fc_1 & ac * fc_1 & cc_1 \\ 0 & fc_2 & cc_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} px_{dx} \\ px_{dy} \\ 1 \end{bmatrix} \quad (3-21)$$

where

- $pixel_x$ = column coordinate of the pixel in the image
- $pixel_y$ = row coordinate of the pixel in the image
- fc_i = horizontal (1) and vertical (2) focal distance expressed in pixels
- ac = skew coefficient, the angle between x and y sensor axes
- cc_i = principal point coordinates, the center pixel of the image plane

Points corresponding to the target segment are used for the DBSCAN point cloud clustering. The result from point cloud clustering is used for model matching phase, where the measured surface points are used as a reference for point set registration to a predefined model. The model used is half of a truncated cone surface, which roughly represents the shape of a sweet pepper fruit. The initial parameters used for the model are 15mm and 20mm and 80mm for the small radius, the big radius, and the length, respectively. The point set registration is done by using a coherent point drift (CPD) algorithm developed by A. Myronenko and X. Song [69]. The CPD algorithm aligns two datasets by iteratively solving a probability density estimation problem, thus finding the rotation and translation of a fruit with respect to the coordinate frame of the laser range finder with center at the homing position of the vertical slider. The obtained rotation and translation values are then used to calculate the approximate position of the stem for cutting operation. The equation used to calculate the cutting point S can be written as follows:

$$S = R * I + T \quad (3-22)$$

$$\begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3-23)$$

where

- S = stem cutting point coordinates

- R = rotation matrix obtained from the CPD algorithm
- I = cutting point initialization vector
- T = translation vector obtained from the CPD algorithm

The cutting point initialization vector contains the coordinates of the cutting point with respect to the fruit:

$$I = \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} = \begin{bmatrix} 0 \\ sc * l + \delta \\ sc * r_{top} \end{bmatrix} \quad (3-24)$$

where

- sc = scaling factor calculated by the CPD algorithm
- l = length of the model
- δ = distance from the top of a fruit to the cutting point
- r_{top} = top radius of the model

Three different transformation methods for model handling are offered by the CPD algorithm, rigid, affine and non-rigid. Only the first two methods were used in this study as the non-rigid proved to be too time-consuming for a practical application in an automatic harvesting robot. The rigid method allows the algorithm to scale the model for searching of the best fit, while the affine method instead of scaling allows the model to be deformed in all three dimensions independently.

3.3.3 Performance Evaluation

Angle Test

An experiment was performed in a laboratory environment to evaluate the accuracy of the proposed method. A set of 10 sweet pepper fruits were used as targets for this test (fig. 3-10). The targets were attached to a test platform and positioned in a known inclination. The test platform consisted of two 1DOF joints allowing to set the rotation of the test objects along X and Z axis by using servo motors (fig. 3-11). The set position of the cutting point in space was calculated by a simple trigonometry:

$$setCP = \begin{bmatrix} scp_x \\ scp_y \\ scp_z \end{bmatrix} = \begin{bmatrix} -l_3 \cos \alpha \\ l_1 + l_2 \sin \beta + l_3 (\sin \alpha \cdot \sin \beta) \\ l_2 \cos \beta + l_3 (\sin \alpha \cdot \cos \beta) \end{bmatrix} \quad (3-25)$$

where

- l_1 = distance from the base of the platform to the lower joint β
- l_2 = distance from the lower joint β to the top joint α
- l_3 = distance from the top joint α to the cutting point
- α = rotation about axis Z
- β = rotation about axis X



Figure 3-10 Targets used for the laboratory test

Both cutting point positions the set and the calculated had to be converted to a common coordinate system for comparing. The coordinate system of the test platform was chosen as the base coordinate system for this test. The conversion of the measured points to the new base coordinate system was done as follows:

$$\begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{old} \\ y_{old} \\ z_{old} \end{bmatrix} + \begin{bmatrix} 0 \\ H_{\Sigma} - (H + \gamma) \\ \mu \end{bmatrix} \quad (3-26)$$

where

- H_{Σ} = distance between coordinate systems in vertical direction
- H = the found height of the target
- γ = correction factor for the vertical direction
- μ = distance between coordinate systems in horizontal direction

The term γ was introduced for alignment of the laser and the camera coordinate system, as the measurement was performed starting from 50mm below the coordinate H registered by the camera. The angles of the joints α and β were set from 0° to $\pm 45^\circ$ with 15° step giving in total 49 measurements per fruit. It must be noted that the testing platform and

the slider platform were two separate units and, although a special care was taken to align the testing platform with the linear slider, a misalignment might had been present and has to be considered when interpreting the data.

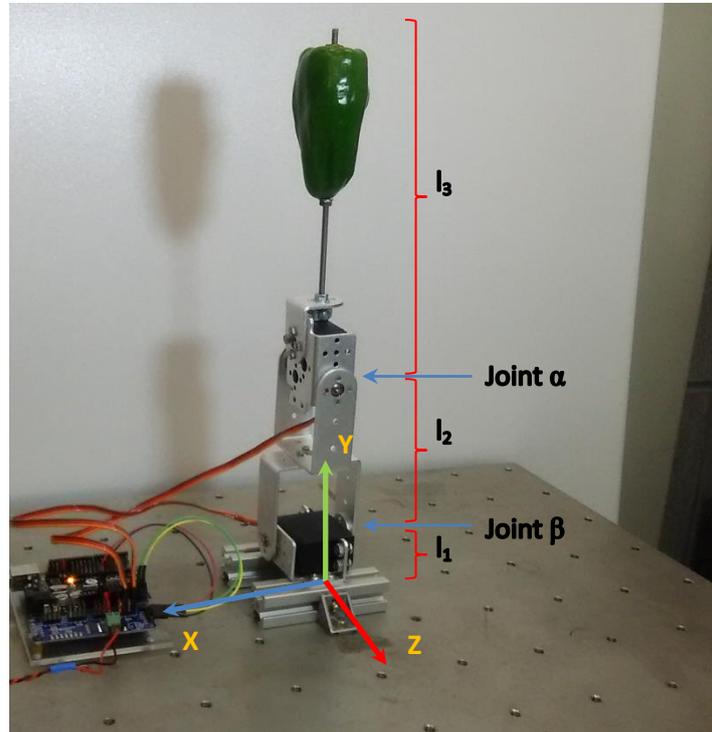


Figure 3-11 Target setup for the laboratory test

Both the rigid and the affine CPD transformation methods were used for data analysis to decide on which is better for the intended purpose. Error for all three axes X, Y, Z and the total error were calculated for every angle combination of all 10 test targets. The total error was calculated according to the equation $e_{total} = \sqrt{e_x^2 + e_y^2 + e_z^2}$, while the individual axes errors, e_x , e_y and e_z were calculated by subtracting the position calculated by the algorithm from the set position. All mean errors are summarized in [table 3-2](#) and also shown in [fig. 3-12](#) for better comprehension. As it can be seen by comparing [figures 3-12a](#) and [3-12b](#), the affine transformation had better overall mean total error compared to the rigid transformation. It can also be noted that the error increased significantly when the rotation $\alpha = 0^\circ$ and the rotation $\beta > 0^\circ$. This increase in the total error came from an increase in Z error, which was caused by extra points from the top of a fruit. In the described rotation the fruit is positioned with the top slanted in the direction of the laser. In such situation the laser takes measurement of the top of the fruit and the algorithm uses these points for model matching

(*fig. 3-12g and 3-12h*). As a result, the calculated position was shifted slightly along the Y axis and significantly along the Z axis. Similar result was obtained for rotation $\beta = -45^\circ$, when points from the bottom of a fruit were measured instead of the ones from top. Surprisingly, this last position didn't deteriorate the result for the rigid transformation, but on contrary, the rigid transformation had the best results when the rotation $\beta < 0^\circ$. The *fig. 3-12b* also suggests that the result for the rigid transformation was biased to the rotation $\alpha < 0^\circ$ direction. This bias could be explained by misalignment of the test platform and the measurement system, although no clearly visible bias could be observed for the affine transformation. Other explanation could be an increased difference between the shape of the target and the model accidentally occurring more often on one side. This would explain the absence of the bias in the affine result as the affine transformation was found to be less influenced by the shape of the target. No apparent relationship could be observed between the set angle and the error along X axis for the rigid transformation, while a strong relationship was observed between rotation α and error along X axis for the affine transformation (*fig. 3-12c*). This result can be explained with the increased freedom that the affine transformation has over the change of the shape of the model. When the inclination of the target was increased in the direction of X axis (rotation α), the shape of the model was changed instead of the rotation angle. The rigid transformation doesn't have the option to change the shape, as it is limited to only scaling, so the best fit is obtained by changing the rotation, thus better representation of the actual angle is obtained.

The rigid transformation handled the Y axis error better compared to the affine transformation when the angles of inclination for rotations α and β were $\pm 30^\circ$ and $\pm 45^\circ$, but the result for Z axis error was significantly better for the affine transformation. One of the possible explanations for the Z axis error might be a wrong model choice. The measured sweet pepper surface points didn't represent a half of truncated cone surface, but more like a segment. The rigid transform has to maintain the shape of the model so it compromises by changing the distance, but the affine transform is allowed to flatten the model to better suit the data.

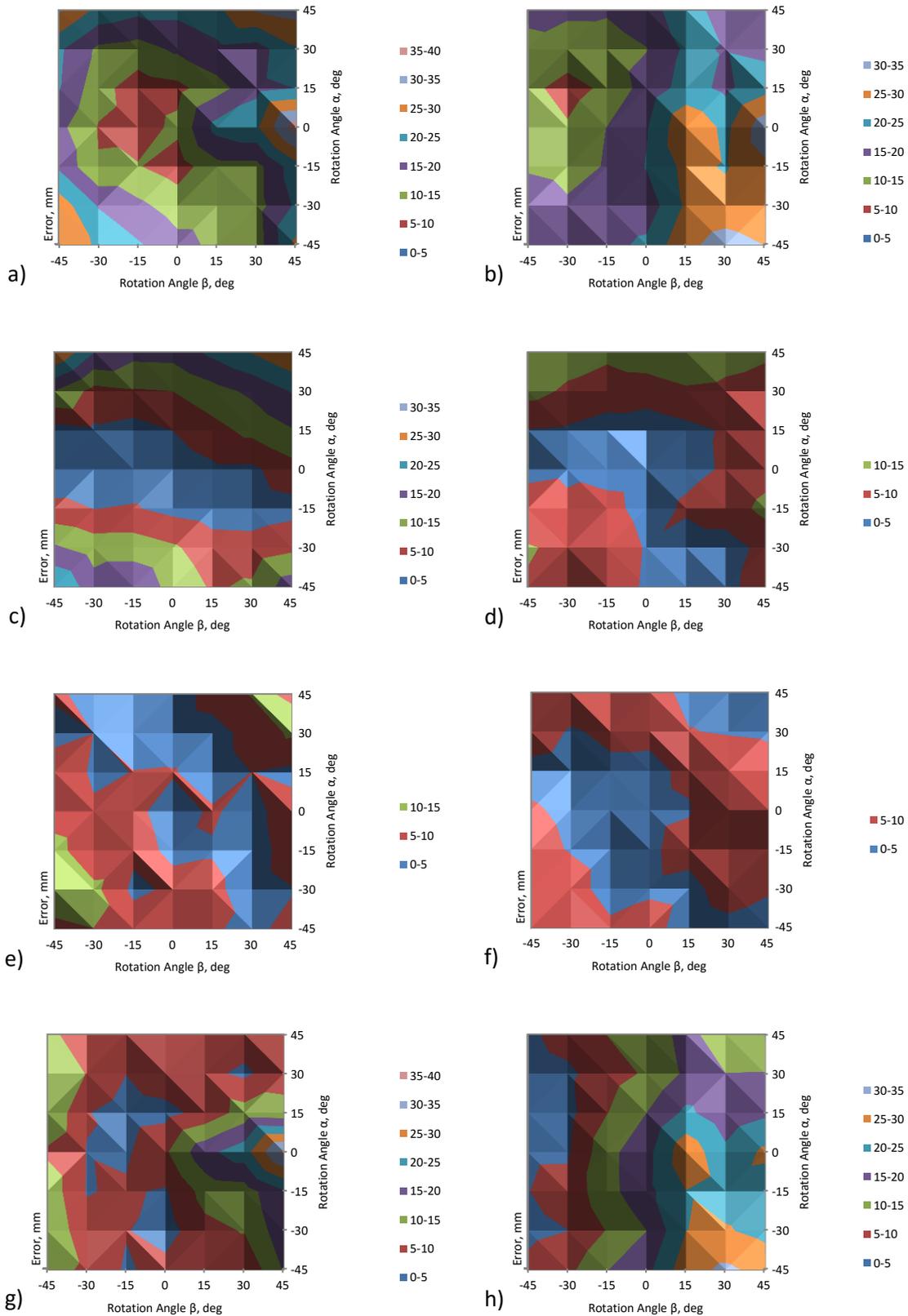


Figure 3-12 Laboratory test results, relationship between inclination angle and a) total error for affine; b) total error for rigid; c) X error for affine; d) X error for rigid; e) Y error for affine; f) Y error for rigid; g) Z error for affine and h) Z error for rigid

Table 3-2 Angle test results, mean values of all 10 measurements

Rotation [deg]		Affine								Rigid							
		Error X [mm]		Error Y [mm]		Error Z [mm]		Total Error [mm]		Error X [mm]		Error Y [mm]		Error Z [mm]		Total Error [mm]	
Rot β	Rot α	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
-45	-45	25,0	8,8	8,7	6,2	12,4	5,5	28,6	4,6	9,8	5,8	9,7	7,2	6,2	4,2	18,5	5,8
-45	-30	15,7	7,2	13,1	6,1	14,9	7,6	27,9	6,3	10,2	4,6	8,5	7,1	3,8	4,1	19,8	5,7
-45	-15	6,4	4,2	12,1	7,0	13,3	8,5	19,3	7,3	8,0	3,1	6,0	6,5	4,9	5,5	14,0	5,9
-45	0	1,0	3,1	7,2	5,4	9,1	7,7	16,4	5,2	3,4	3,6	5,3	6,3	4,3	5,4	12,1	6,1
-45	15	3,0	3,3	8,5	6,3	14,9	9,8	20,4	7,3	4,8	5,3	4,0	4,9	3,9	4,7	11,6	5,6
-45	30	15,0	5,1	2,9	1,7	14,4	6,6	20,9	5,0	11,7	5,0	6,7	4,1	2,6	5,6	14,5	5,2
-45	45	30,4	6,9	5,8	4,7	11,6	5,7	31,0	6,2	14,6	6,4	5,0	3,0	4,1	4,6	15,4	6,2
-30	-45	17,1	8,0	10,0	3,4	8,9	5,1	24,2	5,4	7,7	4,6	6,6	6,6	8,6	3,5	15,8	5,5
-30	-30	12,8	4,0	11,4	4,1	6,2	7,2	19,7	3,3	9,4	4,3	6,3	5,5	5,0	3,6	16,0	5,3
-30	-15	4,9	3,3	5,7	3,6	4,8	4,1	10,2	2,8	6,0	3,7	4,8	5,0	6,6	3,5	12,5	3,7
-30	0	1,5	3,3	3,6	5,3	4,1	2,5	9,5	4,2	4,1	4,0	4,4	4,1	4,3	3,7	10,6	4,3
-30	15	4,2	3,2	4,3	3,5	5,3	6,2	11,8	4,7	4,4	3,8	3,5	3,8	3,4	4,2	7,8	4,5
-30	30	8,3	4,6	5,0	3,5	9,8	9,1	14,0	8,5	9,3	5,7	4,7	3,6	4,0	4,8	13,4	6,0
-30	45	20,4	4,5	3,4	4,9	6,6	8,5	23,7	5,9	12,2	5,4	6,9	2,8	7,1	4,1	15,6	4,9
-15	-45	19,1	4,6	7,8	4,6	4,4	4,5	20,6	4,1	8,5	3,9	5,6	4,8	14,8	4,2	18,4	4,3
-15	-30	12,8	3,1	3,9	4,8	5,3	4,2	16,2	2,7	9,3	4,0	3,7	5,3	10,2	5,6	16,1	5,4
-15	-15	5,4	3,2	6,1	3,0	5,6	4,2	10,3	4,5	6,9	3,0	3,7	3,8	13,4	4,4	15,5	3,7
-15	0	2,1	2,4	5,7	3,3	4,3	3,6	7,9	3,7	2,8	3,0	4,4	3,7	13,5	4,4	15,1	4,6
-15	15	2,9	3,1	5,1	2,0	2,6	2,6	7,5	3,1	2,8	3,2	4,5	3,8	9,8	4,9	11,2	5,6
-15	30	9,5	3,2	3,5	2,2	5,1	4,3	12,8	2,8	7,7	3,9	5,6	3,4	7,7	4,9	12,4	5,4
-15	45	16,7	3,7	2,6	3,3	8,5	5,0	21,4	4,7	11,1	4,7	5,2	3,2	10,1	4,4	15,7	5,6
0	-45	13,1	3,9	5,3	3,5	6,7	4,4	16,5	3,7	4,1	3,7	6,1	4,3	18,5	4,3	21,1	4,5
0	-30	10,7	4,7	5,9	3,9	2,8	2,9	12,8	3,7	4,5	2,2	4,6	4,5	15,0	4,4	17,1	3,9
0	-15	3,5	2,6	4,4	3,0	4,5	4,6	7,6	4,9	2,9	2,4	4,7	3,8	18,4	4,0	20,3	4,0
0	0	1,7	2,0	3,8	3,8	9,3	5,1	11,1	5,6	2,4	2,7	4,4	3,4	18,3	2,9	19,6	3,2
0	15	5,1	4,0	5,3	3,5	3,5	3,5	8,6	5,0	1,5	3,5	4,5	3,5	15,0	5,7	17,1	6,5
0	30	10,0	4,1	2,4	4,5	8,0	4,7	16,0	4,3	7,9	5,1	5,4	3,2	10,7	4,6	14,8	6,2
0	45	17,0	5,0	2,5	4,1	6,7	5,3	21,4	5,8	12,5	5,5	6,0	3,0	12,0	5,5	18,5	6,5
15	-45	9,4	5,0	5,3	3,1	6,6	8,1	14,0	7,0	2,9	2,5	3,5	3,1	28,1	7,5	28,8	6,9
15	-30	7,4	5,1	5,9	2,6	8,5	6,4	12,7	6,7	2,8	1,9	4,2	3,8	25,0	6,6	25,8	5,2
15	-15	2,3	1,8	4,7	3,2	11,9	6,6	13,5	6,3	5,7	2,2	5,8	3,8	24,9	5,4	26,5	3,9
15	0	4,0	2,9	5,4	4,4	18,5	6,0	21,8	6,0	3,8	3,1	4,8	3,7	28,4	4,5	29,3	4,0
15	15	8,3	2,6	3,2	3,3	6,6	5,0	13,2	4,8	3,5	3,5	5,5	3,0	21,1	4,4	21,9	4,4
15	30	13,1	3,7	3,3	3,8	6,2	5,0	18,7	4,1	9,4	5,0	5,1	3,8	15,8	4,2	20,9	5,1
15	45	21,3	5,0	6,9	2,6	5,7	4,9	24,0	5,0	14,0	4,8	4,4	4,4	16,7	5,2	22,3	6,6
30	-45	10,8	4,2	4,3	2,1	5,7	7,2	12,3	7,1	4,1	2,3	4,3	27,8	31,1	6,8	32,2	20,9
30	-30	7,1	3,9	3,2	1,5	10,3	6,9	13,1	6,2	2,8	2,3	6,2	5,1	25,5	5,6	27,0	4,3
30	-15	2,1	5,7	3,0	4,2	12,6	6,4	13,3	7,5	7,4	4,0	6,1	4,3	22,9	6,9	24,5	4,8
30	0	3,9	2,5	3,7	5,9	23,0	7,0	24,2	7,7	5,4	3,5	7,8	4,9	21,1	6,7	23,8	5,4
30	15	11,5	2,8	5,0	4,6	12,1	6,2	18,5	5,5	5,6	3,1	6,8	4,3	16,8	6,7	18,9	5,7
30	30	16,1	4,0	6,6	3,8	4,0	6,2	17,8	6,0	8,8	3,6	4,8	3,7	15,1	5,6	19,7	5,1
30	45	25,4	4,8	10,4	4,3	7,6	4,9	30,6	4,9	12,1	3,2	4,3	3,3	11,9	4,5	17,7	4,7
45	-45	17,8	4,9	5,9	1,9	15,3	10,4	26,2	9,8	6,6	3,8	3,6	3,2	28,3	7,2	29,9	6,5
45	-30	8,9	4,3	4,8	8,7	18,2	10,1	20,8	10,1	6,3	4,0	5,3	4,6	23,5	6,7	25,5	5,2
45	-15	3,2	5,1	6,9	4,3	19,6	6,1	23,1	6,2	11,4	4,8	7,1	5,6	22,3	8,9	29,1	6,3
45	0	8,1	1,5	7,1	3,0	36,3	10,2	37,8	9,6	7,9	4,6	8,8	6,4	26,1	9,9	33,2	7,8
45	15	14,7	4,1	3,7	3,5	11,7	12,9	20,1	12,2	6,8	3,2	6,0	4,3	19,1	11,5	23,4	10,9
45	30	20,3	4,8	10,9	6,0	8,1	11,9	24,4	11,8	6,9	2,4	4,6	4,5	15,1	7,2	18,4	5,8
45	45	29,9	6,4	9,7	3,9	6,5	9,8	32,0	9,4	11,3	2,6	4,9	4,3	11,3	6,1	17,6	4,8

Table 3-3 Frequency of result error value under a set limit.

Limit, mm	Affine				Rigid			
	e_x , %	e_y , %	e_z , %	e_{total}	e_x , %	e_y , %	e_z , %	e_{total}
<25	92.7	99.4	91.4	77.6	99.8	99.2	84.1	75.5
<20	84.1	98.6	87.1	58.4	98.8	98.4	72.9	54.7
<15	70.2	95.3	75.3	38.0	91.4	95.5	53.5	29.2

The whole dataset (490 measurements) was analyzed to determine how frequently the result was within certain limits. When the limit was set to 25mm, the result with the error less than the set limit for the affine transformation occurred in as many as 92.7%, 99.4%, 91.4% and 77.6% of the total amount of measurements for e_x , e_y , e_z and the total error e , respectively. The same calculation for the rigid transform gave result 99.8%, 99.2%, 84.1% and 75.5% for e_x , e_y , e_z and the total error e , respectively. Results for limits of 25mm, 20mm and 15mm can be found in [table 3-3](#). These results suggest that the rigid transformation performed better in terms of error e_x (rotation around Z axis), while the affine transformation performed better in terms of e_z . Both methods performed similarly in terms of e_y , which is a product of both e_x and e_z as the height of the found cutting point depends on both rotations around X and Z axis. It should be noted that this overall statistical result shown in [table 3-3](#) is greatly deteriorated by results for the inclination angle of 45°. This is especially true for the affine result. For example, if the statistical result is calculated only for inclination angles up to 30°, the total error was less than 25mm in 91.6%, less than 20mm in 81.6% and less than 15mm in 61.2% of cases. The average error value by inclination angles for both transformations is shown in [fig. 3-13](#) and [fig. 3-14](#). The data suggest that the rigid transformation was less responsive to the changes of the angle compared to the affine transformation. The deterioration of e_x value for affine transformation was found to be proportional to the increase of the inclination angle. From these results, it can be concluded that the affine transform is more suited to be used for angles up to 30°, but the rigid transform works better than the affine transform for the inclination angles over 30°.

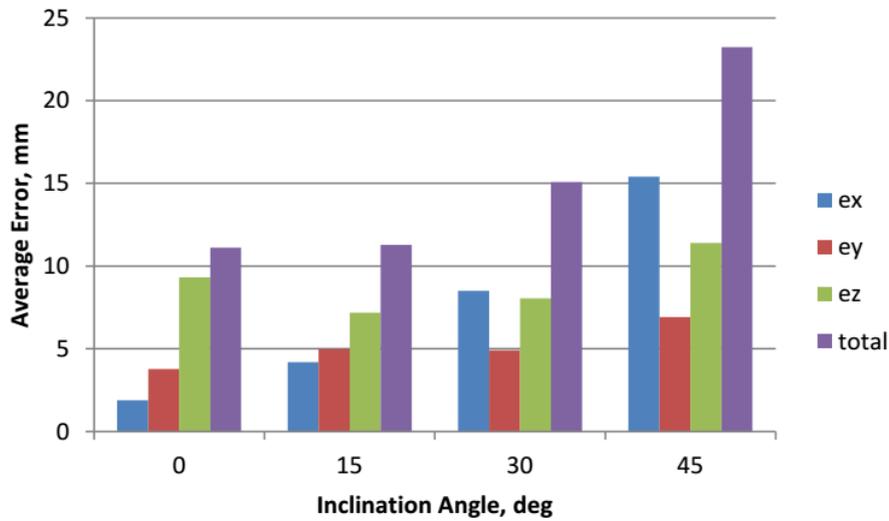


Figure 3-13 The average error compared to the inclination angle for the affine transformation

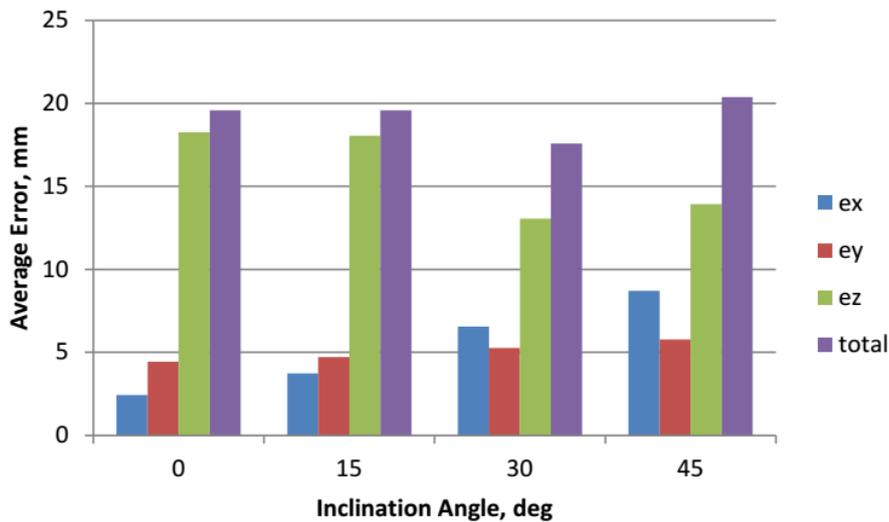


Figure 3-14 The average error compared to the inclination angle for the rigid transformation

Greenhouse Test

A field test was performed to evaluate the performance of the method in a greenhouse environment. A different linear slider was used for the field test for the sake of mobility. The used slider was SA-S6AL-300-BE (Standard Units Supply Corp.) with a travel distance of 300mm. The slider was mounted on a test platform that could be moved freely in the greenhouse (*fig. 3-15*), and the position was controlled manually using a controller. First, the slider was positioned at the height, where the mass center of a fruit in the image coincided with the vertical centerline of the camera. After that, the slider was lowered by 10mm to position the laser 50mm below the center position of the target. The measurement was performed for 100mm distance with 2mm step in the vertical direction, and the obtained data

was processed as described previously. In total 107 different, randomly picked sweet pepper fruits were chosen as test targets for this experiment, which were divided into 4 categories depending on the level of occlusions:

Type I – no occlusions ;

Type II – slightly occluded, around 10% of the fruit are covered;

Type III – partially occluded, around 30% of the fruit are covered;

Type IV – very occluded, over 50% of the fruit are covered.



Figure 3-15 Greenhouse test setup

Accurate actual stem position measurement for reference purpose under greenhouse conditions is a complicated task. To overcome this problem indirect comparison method was used for the result evaluation. To verify the accuracy of the calculation, the calculated cutting point location together with the calculated center axis of the fruit were projected on the image taken for the fruit recognition. This information allowed a visual comparison of the estimated cutting point and the actual stem position in the image. The projected position was then evaluated by giving points based on how close it was to the stem (*fig. 3-16*). From the total 107 fruits chosen for this test, 68 were type I (no occlusions), 28 were type II (slight occlusions), 6 were type III (around 30% of the fruit was occluded) and 5 were type IV (over 50% of the fruit was occluded). Points were given to both transformation methods of the CPD algorithm and the mean was calculated to decide on the best method.

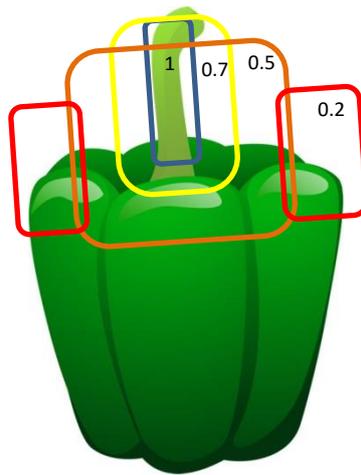


Figure 3-16 The template for greenhouse test result evaluation



Figure 3-17 Greenhouse test result examples – a) type I; b) type II; c) type III; d) type IV

According to the results of this test under the greenhouse conditions the affine transformation performed significantly better than the rigid with mean score 0.78 versus 0.69 for the rigid transformation. For 47 targets both methods showed similarly good results, for 38 targets performance of the affine transformation was better than that of the rigid, for 18 targets the rigid transformation showed a better result, but for 4 targets both methods failed. For 13 out of 107 instances, the result was deteriorated due to a failed segmenting by DBSCAN

algorithm. In 10 of 13 occasions the DBSCAN segmenting separated the points of the fruit into several smaller segments, but in 3 of 13 occasions it wasn't able to separate two fruits that were close together. There are two possible reasons for DBSCAN fail, the first reason is wrong parameter values for eps and minPts, and second is a low laser measurement quality. If the instances of DBSCAN error were ignored in order to check the CPD performance without the influence of failed DBSCAN segmenting, the mean scores were 0.82 and 0.74 for the affine and the rigid transformation, respectively. The affine method worked particularly well compared to the rigid when the targets were occluded and when the shape of a fruit was significantly different than that of the model. The affine transformation had average score 0.71 and 0.80 for type III and type IV targets, respectively, while for the same targets the average scores of the rigid transformation were only 0.54 and 0.65. *Figure 3-18* shows a comparison between the affine and the rigid transformations in terms of frequency of receiving a particular score. The results of the greenhouse test were in accordance with the angle test results and suggest that the affine transformation is more suited for the given task compared to the rigid transform.

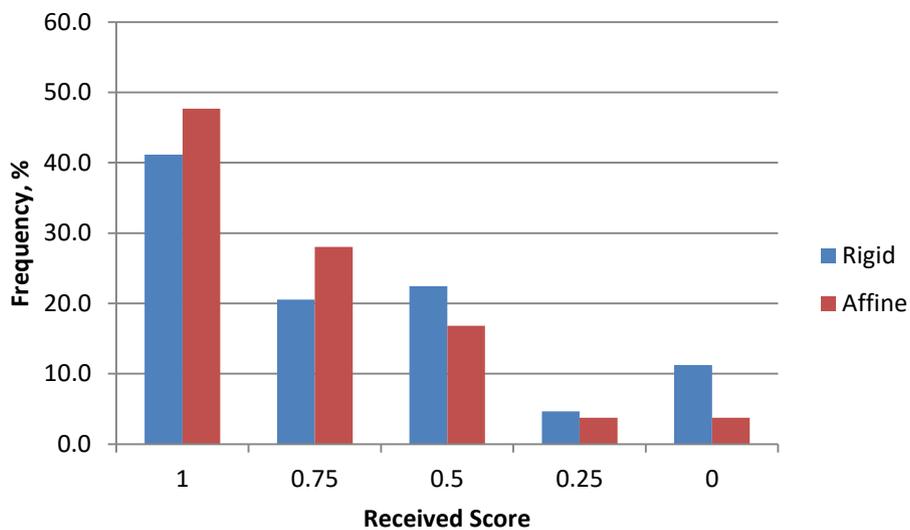


Figure 3-18 Comparing the frequency of receiving a particular score for both CPD transformations

3.3.4 Speed of the System

The speed of operation is a very important characteristic of any autonomous system. It is especially true for an automatic harvesting robot, as it is compared to the working speed of a human labor. Even if a harvesting robot would have a harvesting success rate of 100%, it wouldn't be practically useful if the harvesting time for one fruit would be long. Researchers in the field often have to compromise between accuracy and speed by lowering the image resolution or by choosing simpler but faster algorithms over more sophisticated but slower ones. One of the main drawbacks of the proposed system is the increased data acquisition and processing time. Instead of using relatively simple stereovision for target detection and position calculation a more complex data acquisition and processing algorithm was chosen to solve the problems with sweet pepper harvesting discussed in the introduction section. There are three main time-consuming steps in the current proposal, image acquisition, laser measurement and data processing. The image acquisition and target identification take approx. 0.7s per image. Normally it would be considered a good speed, but the high frequency of image acquisition must be taken into account as this process is repeated after each 10mm of slider movement for 1m in total movement distance. The total image acquisition time including the time for mechanical movement of the slider with the current setup was 75.4s. The time consumption in this step could be improved by increasing the distance between image acquisition points, but that might lead to inaccuracy in target height detection. More study has to be done for camera movement step size and other image acquisition parameter optimization.

The laser measurement with the current setup takes approx. 0.3s per slice. In total 51 slices were taken per fruit with 2mm step in the vertical direction to cover all 100mm length. The control software is optimized to save as much time as possible when working with multiple targets by taking only one measurement per any given height and saving the result for each of the fruits at that particular position. The time spent for measurement of a single separate target (51 laser measurement and the mechanical movement of the slider) was approx. 31.4s. To decrease the time spent for this step, the number of measurements can be changed by either choosing a bigger step size or by decreasing the length of the measurement area, which is currently 100mm. This length was chosen with the consideration that most mature sweet pepper fruits have approx. 80 – 90mm length, but in some cases, they can be up to 120mm long. The adaptive method can be developed that varies the length of the

measurement area according to the fruit size to save time for shorter fruits and not lose accuracy for longer ones.

The last time-consuming step is data processing, which consists of several parts as described in the algorithm section. The most time-consuming part is the CPD algorithm with the average total consumed time in this study of 14.50s for the rigid transformation and 15.79s for the affine transformation (with a maximum number of iterations `opt.max_it = 30`). The speed of the CPD algorithm depends on the number of data points used, maximum number of iterations and the chosen transformation method. The number of points used for the CPD algorithm could be decreased by reducing the amount of points used for building surfaces of the model and targets. Currently, the surfaces used for point matching have an initial size of 100x100 points, which results in point matching of two matrices with the maximum possible size of 3x100x100. When the point count was changed to 50x50 points, the CPD calculation time changed from 14.50s and 15.79s to 4.90s and 5.01s for rigid and affine transformations, respectively. It must be noted that the results of this study were obtained by using 100x100 point surfaces, and the quality loss for lowering the point count, if any, was not investigated.

The speed of the control software could be further increased by translating the processing software from Matlab to C/C++ environment. It is generally known that a code written in C/C++ is faster than or as fast as its analog written in Matlab. This premise was already proven in this research by using MEX code for communication with the laser instead of Matlab built-in serial object, therefore decreasing the time spent for laser measurement from 1.2s per slice to 0.3s per slice. Naturally the final version of the control code for a harvesting robot should be written in C/C++ or similar programming language, but the Matlab environment is a very useful tool for the development process, especially for systems dealing with image processing. In general, the speed of the system can still be optimized without losing much of the performance, but more study has to be done for fine tuning of the used parameters.

3.3.5 Conclusions

A second prototype of the hardware for pose estimation method was built and evaluated. This prototype was built to be directly implemented in the harvesting robot without further changes. Several small changes in the algorithm were also made compared to the algorithm of the first prototype. First, the order was slightly changed to optimize the point dataset before using the time-consuming functions. In this prototype point, forward projection

is performed before the DBSCAN clustering to remove all points outside the target segment in the image. Point projection algorithm is relatively fast compared to the clustering algorithm. The main task of the clustering algorithm now was to remove the noise around the fruit and to solve fruit cluster problem in case several fruits were segmented into a single segment in the image. Secondly, point forward projection was improved by implementing the lens distortion in the calculation. Lastly, the SVD algorithm was changed to the CPD algorithm for the model matching. Although the CPD algorithm is slower compared to the SVD, it is a better choice for the intended task as it can perform scaling and affine transformation. This, in turn, can deal with the irregularly shaped fruits and fruits with a size different than that of the model. The SVD algorithm wasn't able to change the size of the model, which made it unreliable in cases when the shape of a fruit was significantly different than that of the model.

Two tests were performed to evaluate the accuracy of the proposed method. The first test was performed under laboratory conditions on targets with a known inclination angle to verify the accuracy of the method. In this experiment, the method was capable of calculating the position of a sweet pepper stem within 25mm of the actual position in 77.6% of cases using the affine transformation as CPD method and in 75.5% of cases using the rigid transformation. The second test was performed to verify the performance of the method under greenhouse conditions. The qualitative analysis of the field test result showed acceptable stem position calculation in 81 of 107 cases for the affine transformation and in 66 of 107 cases for rigid transformation. The obtained results from both tests suggest that the proposed method can be used with good results for stem position calculations for fruits with inclination angle up to 30 degrees.

The two main reasons observed for unsatisfactory result were occlusions and failure of the DBSCAN clustering. Occlusions that cover the lower part of a fruit do not affect the performance of the method as much as occlusions of the top part. In case of covered lower part the system treats the case as if it would be a shorter fruit while in case of covered top the found position of the stem would be lower than the actual. Sweet pepper plants tend to have thick foliage and, therefore, many occlusions. To solve this problem, a special method could be implemented, based on fuzzy logic or other machine learning techniques. This method would look for evidence to suggest that the top of the fruit is occluded. The evidence could be, for example, rather short but thick fruit with a leaf right on top of it.

The DBSCAN clustering was the main reason within the developed system for partial or complete failing to correctly calculate the stem position. In several occasions due to noisy laser measurement the clustering algorithm separated the surface of a target fruit into several smaller segments from which only the largest was selected for further calculations. Therefore, in future work, much attention should be given to improving the point cloud segmentation. More specifically, fine tuning of the constants used by the DBSCAN should be performed. As the worst case scenario, the use of DBSCAN could be completely discarded.

Another important topic for improvement is the model used for fitting. Irregularly shaped fruits are frequent, therefore, either finding an optimized version of the model to fit or developing an approach to handling with the irregular shape fruits must be considered. Although the results of this study were promising for use of the proposed method in automatic harvesting, the accuracy and success rate of the method is still not high enough for this method to be used as the final decision for the position of a cutting point, but it can be used as a good first guess where to look for the stem with other sensors mounted on the end effector.

3.4 Summary

The necessity for fruit pose estimation in automatic harvesting robots for providing with essential information for harvesting has been stated in the literature. Despite the existing research in general machine vision regarding the pose estimation in space, to authors' best knowledge no working algorithms have been presented in the agriculture field. To improve the performance of sweet pepper harvesting robot a fruit pose estimation algorithm was developed. In this algorithm a laser range finder was used to obtain surface points of the harvesting target and model fitting method was used to calculate the rotation and translation of the fruit in space. The calculated rotation was then used to determine the stem position and the cutting point. The system was tested both under laboratory conditions and in a greenhouse, where actual working conditions of the harvesting robot were simulated. Results of these tests suggested that the developed method is well suited for the intended task. It is worth noting that the method is not limited to pose estimation of sweet pepper and can be used in automatic harvesting in general, as long as a good generic model of the target fruit is available. However, this method can't be used for completely round fruits such as tomatoes and apples.

4. Touch Detection

4.1 Background

Touch is a very important sense for humans and animals that give vital information about the surrounding environment. It is the reason why such task as object handling without the visual aid is relatively easy for humans but very complex for robots. The importance of touch or tactile sensing in robotics was recognized in the 1980s and it has been the subject of much research ever since [70]. In robotics any device that gives information about shape, texture, softness, temperature, vibration or shear and normal forces, by physical contact or touch, can be called a touch or tactile sensor. Most of the information interpretation for touch sensors of humans is done unconsciously or intuitively from years of experience. A child has to learn how to handle objects with the proper strength to not drop them and also to not damage fragile objects by applying too much strength. In robotics, such year's long experience and *a priori* information from previous failed attempts is unavailable therefore accurate sensor feedback and interpretation is even more critical. It is interesting to note that one of the early researchers in this field Harmon concluded that touch sensing is unlikely to be adopted for applications that deal with living systems such as medicine and agriculture due to too much of technical difficulties to produce a return on investment [71]. Indeed, no serious attempts to use tactile sensing in agriculture have been reported in the literature. Only several end-effectors use some kind of sensory feedback, such as force sensors to determine if a proper gripping force is being used [28] or proximity sensors for additional guidance information [5, 28]. None of the most developed automatic harvesting robots use tactile sensing at any form [5, 9, 11], which can be due to several reasons.

1. Despite almost four decades of research, sensing technologies are still not developed enough to be used with confidence. Sensor information interpretation and handling must be done with extreme care, or else, instead of improving, the performance of the robot can be deteriorated. When it comes to autonomous robots, researchers tend to choose the simple but reliable methods over the advanced ones.
2. Tactile sensing requires additional computation power, which until several years ago wasn't readily available. The speed of the system is one of the key characteristics of a harvesting robot as it has to compete in speed with a skilled human worker. It also must be noted that automatic harvesting robots ideally are supposed to work on the battery to gain complete autonomy. As a result, high-performance computers cannot be used. During the last decade, however, the computer development has allowed for low power consumption computers with relatively high computation speed.
3. In our daily life we depend heavily on our visual sense, and only those, who are visually impaired, truly understand the importance of touch sensing. Similarly, it is possible that many researchers underestimate the possibilities of the tactile sensing or are intimidated by the possible difficulties and complications, and as a result choosing already verified methods.

Automatic harvesting relies heavily on the machine vision as the means of target detection [37]. Despite decades of effort, no harvesting robot can report a 100% success rate when combining visual detection with harvesting [21]. Even if the machine vision would be able to detect 100% of targets in an image, the success of the harvesting would still be limited by the successful detachment of the targets. Pulling off by force can't be used for many fruits and vegetable as it can lead to damage to the fruit. In such cases, the stem of a fruit must be cut, which requires information about the stem position. Localization of stem in the visual data is often complicated due to the similarity between the stem and other branches nearby or even far behind the harvested fruit [23]. The importance of tactile sensing is clearly apparent in applications, where the use of other sensing modalities doesn't guarantee the required success rate [72]. In this research, this problem is addressed by proposing a stem localization by use of a tactile sensor. All general types of tactile sensor transduction techniques were analyzed in order to choose the most appropriate for the given task and intended working conditions (table 4-1).

Table 4-1 Touch sensing techniques [72]

Transduction technique	Modulated parameter	Advantages	Disadvantages
Capacitive	Change in capacitance	Excellent sensitivity Good spatial resolution Large dynamic range	Stray capacitance Noise susceptible Complexity of measurement electronics
Piezoresistive	Change in resistance	High spatial resolution High scanning rate in mesh Structured sensors	Lower repeatability Hysteresis Higher power consumption
Piezoelectric	Strain (stress) polarization	High-frequency response High sensitivity High dynamic range	Poor spatial resolution Dynamic sensing only
Inductive LVDT	Change in magnetic coupling	Linear output Uni-directional measurement High dynamic range	Moving parts Low spatial resolution Bulky in size Poor reliability More suitable for force or torque measurement
Optoelectric	Light intensity change Light spectrum change	Good sensing range Good reliability High repeatability High spatial resolution Immunity from EMI	Bulky in size Non-conformable
Strain gauges	Change in resistance	Sensing range Sensitivity Low cost Established product	Calibration Susceptible to temperature Susceptible to humidity Design complexity EMI induced errors Non-linearity Hysteresis
Multi-component	Coupling of multiple intrinsic parameters	Ability to overcome certain limitations via combination of intrinsic parameters	Discrete assembly Higher assembly costs

The most straightforward touch sensor is a limit switch, which consists of an open contact that is closed by the force of a touch. Although technically it could be possible to construct a limit switch, which is closed only by the stem and not leaves, by choosing proper stiffness parameters, the use of limit switches was ruled out from the point of view of the working conditions. The main reason against open contact sensors, such as limit switches, is the humidity present in a greenhouse. The general guidelines for using limit switches are against use in high temperature and humidity environments. First, the condensate from the humidity in the air can short circuit an open switch thus leading to a false positive sensor reading. The second problem is vulnerability to corrosion. High temperature and humidity can lead to arc discharge during opening and closing of the switch, which in turn will lead to the corrosion of the contact surface of the switch. Corroded surfaces have lower conductivity characteristics and can cause decreased sensitivity or complete malfunction of the sensor.

Capacitive sensor technology was found to be more suited for the given working conditions, as it is possible to completely screen the active components from the humidity. The issue, however, is the inability to distinguish between a fruit and a leaf as a capacitive sensor was found to be equally sensitive to both. Moreover, the capacitive sensing technology doesn't need a direct contact and can sense an object from a distance if properly tuned. This can lead to unexpected behavior and false positive detection error in case a leaf is nearby the stem or simply from moving the sensor too close to the fruit. Due to these drawbacks, implementation of capacitive sensors for touch detection in automatic harvesting is complicated.

Use of force sensing resistors (FSR^{*}) was selected as the most promising sensing technique for the early research stage application described previously and an in-depth examination was performed. Despite the susceptibility to the temperature and humidity conditions, it has near linear force to conductivity relationship, which allows analyzing the touched object by the force the object is able to apply. A full description of the performed tests and their results are given in sections below. Later, however, a special sensor based on piezo effect was designed to deal with the drawbacks that were recognized for the FSR sensor.

^{*} Abbreviation *FSR* is a trademark of Interlink Electronics Inc. and can be used in this context only because force sensing resistors from the particular manufacturer were used.

4.2 Requirements

During the course of this research, two different sets of requirements were established depending on the problem being solved at the time. In the early stage of the research, the addressed problem was the development of a sophisticated gripper system that would be able to grasp a fruit without an accurate visual guidance. The target end-effector was a multi-finger robotic arm, and the main requirement was to be able to determine on fruit size and orientation by using tactile sensing instead of visual data. The sub-requirements were as follows:

- To be able to detect a contact with a sweet pepper fruit;
- To be able to distinguish between a sweet pepper fruit and a stem or a leaf;
- To be able to work in a greenhouse environment.

In the later stage of the research, however, the target platform for the sensor and the main requirement was changed. The approach to the harvesting was changed from using a

multi-finger robotic gripper to using a scissors-pincer type cutter. The new end-effector required only stem position for harvesting thus the new main requirement for the touch sensor was stem detection.

4.3 FSR Touch Sensor

4.3.1 Overview

Force sensing resistors (FSR) were investigated for potential use in sweet pepper automatic harvesting. The main requirement for the sensor was to be able to detect a contact between the sensor and a sweet pepper fruit. The initial intended position of the sensors was the fingertips of a multi-finger robotic gripper (*fig. 4-1*). The designed fruit harvesting algorithm was as follows (*fig. 4-2*). First, the visual recognition algorithm would be used to search for fruits. Once a fruit would be found, the position of the fruit would be calculated using stereo imaging. The robotic manipulator would then move the end-effector with a closed finger to the fruit position until a contact would be detected.

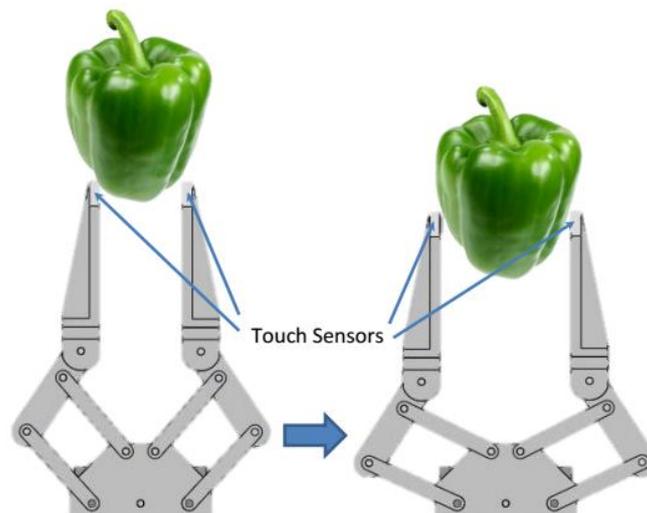


Figure 4-1 Multi-finger gripper model with FSR sensors at fingertips

After the initial contact, the fingers would be opened gradually while slowly moving the end-effector forward until each finger would reach the side of the pepper. This approach would allow scanning the surface of the sweet pepper and finding the optimal finger opening width for proper fruit gripping without requiring accurate information from the visual recognition. This part is particularly important because accurate fruit dimension information in

case of sweet pepper is difficult to obtain due to reasons discussed in previous sections. Once a proper gripping is established, fruit manipulation for cutting purposes is possible.

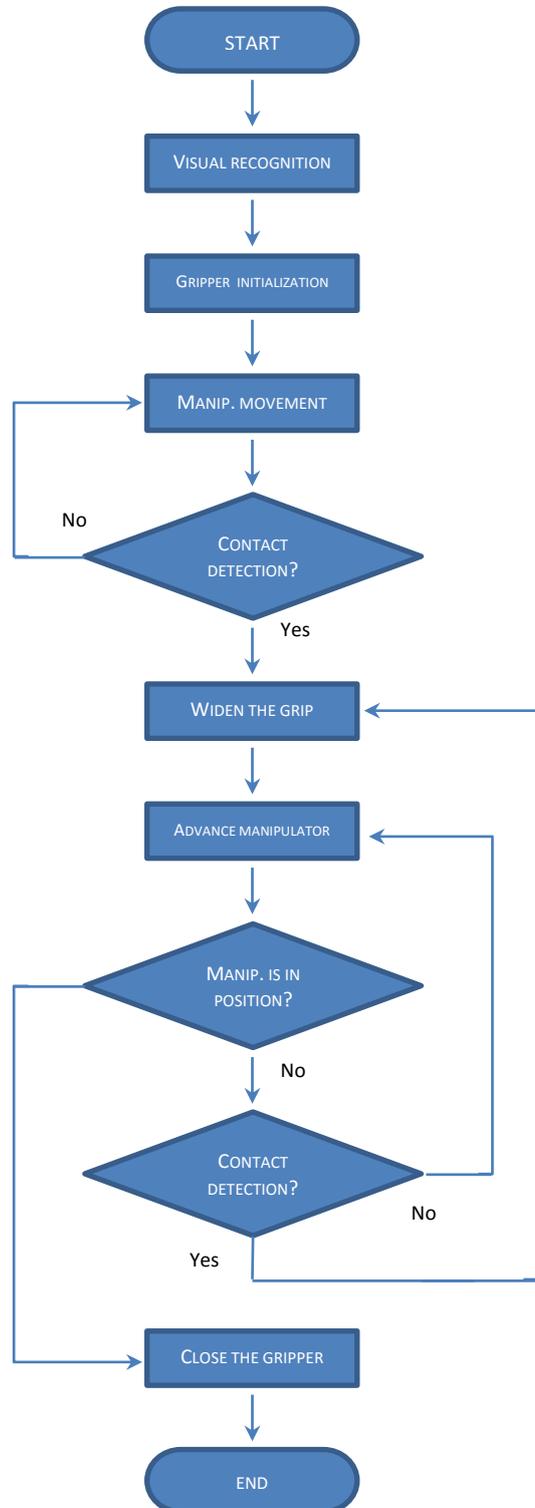


Figure 4-2 Working algorithm of fruit surface scanning by touch sensors

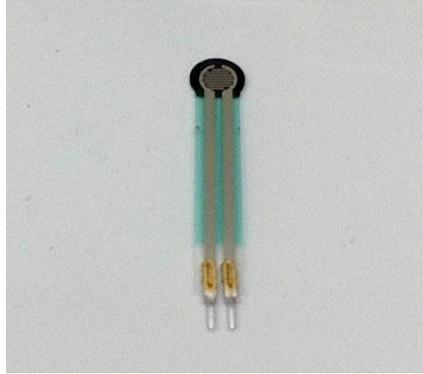


Figure 4-3 Sensor FSR400

4.3.2 Setup

In this study FSR400 (Interlink Electronics Inc.) force sensing resistor was used (*fig. 4-3*). This sensor has a round sensitive area with 5.08mm diameter and 1mm wide insensitive ring around. The theoretical actuation force according to the datasheet is approx. 0.2N. A simple measurement schematic was used, where the sensor was connected in a voltage divider configuration, and the measurement was performed by a 10-bit ADC. A 100k resistor was set as the fixed resistor in the voltage divider as per manufacturer's recommendations. The output voltage and resistance of the FSR sensor relationship can be described by the following equation:

$$V_{out} = V_{in} \frac{R_f}{R_s + R_f} \quad (4-1)$$

where

- V_{in} = input voltage, [5V]
- R_f = fixed resistance of the voltage divider, [100k Ω]
- R_s = resistance of the sensor, [Ω]

The same value in terms of an ADC measurement can be calculated from equation

$$V_{out} = \frac{ADC_{val} \cdot V_{in}}{ADC_{bit}} \quad (4-2)$$

where

- ADC_{val} = the measured ADC value
- ADC_{bit} = the bit depth of the ADC used

The output of the voltage divider was connected to the ADC input of an 8-bit microcontroller (PIC16F887, Microchip Technology Inc.) according to the *figure 4-4*. Movement of the sensor

was achieved by a simple linear slider, which was controlled by the second microcontroller through Quadstepper Motor Control Board (SparkFun Electronics). The ADC reading was performed with the rate of approx. 70SPS and it was sent to a computer through TTL to USB converter.

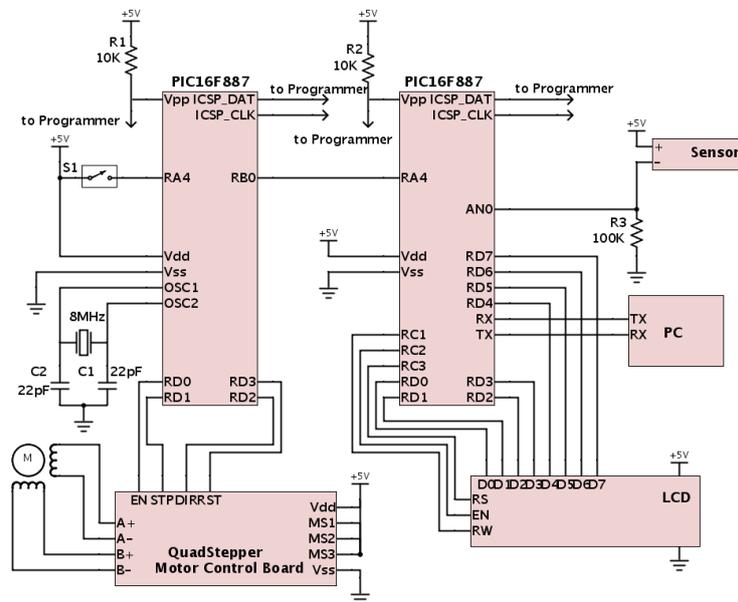


Figure 4-4 Schematics of the measurement system.

4.3.3 Evaluation

According to the requirements, the sensor had to be able to detect a contact with a sweet pepper fruit. To verify the approximate force necessary to move a sweet pepper fruit, a measurement in a greenhouse was performed with a dial strain gauge. The measured force was starting from approx. 0.18N, which is close to the lower limit of the sensitivity of FSR. A simple test rig was built to test the performance of the sensor in a greenhouse (fig. 4-5). The sensor was attached to the linear slider by an aluminum rod with a diameter of 3mm, which is less than the diameter of the active surface area of the sensor. Such setup was chosen to eliminate the insensitive area of the sensor. Any touch in the insensitive outer ring of the sensor would bend the sensor slightly, which in turn will be detected. This is an important modification as, considering the shape of the sensor and geometry of the fruit, any contact would most likely occur at the sides of the sensor.

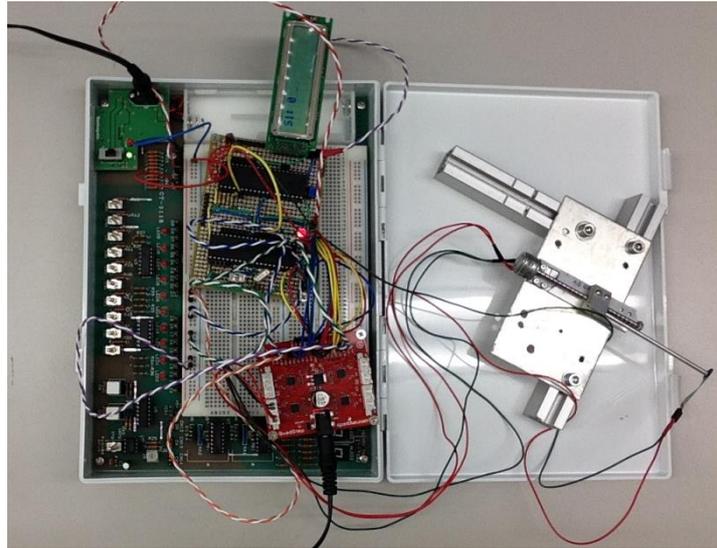


Figure 4-5 FSR test setup.

A contact between the sensor and a test object was induced by moving the slider towards the target while reading the sensor output. The results were in accordance with the expectations. No change in the FSR resistance was observed after a contact with a leaf. When the sensor was tested by introducing a contact with a fruit, in 8 of 17 cases (47%) the sensor output showed clear detection, in 2 of 17 cases the output reading had several spikes above the noise but not enough for detection, but none of the rest measurements showed any signs of change. The mean output of the sensor after a detected contact was 1.31V (0.71V standard deviation) while the max noise level was 0.39V. See *figure 4-6* for examples of measurement for contact with leaf, branch and fruit.

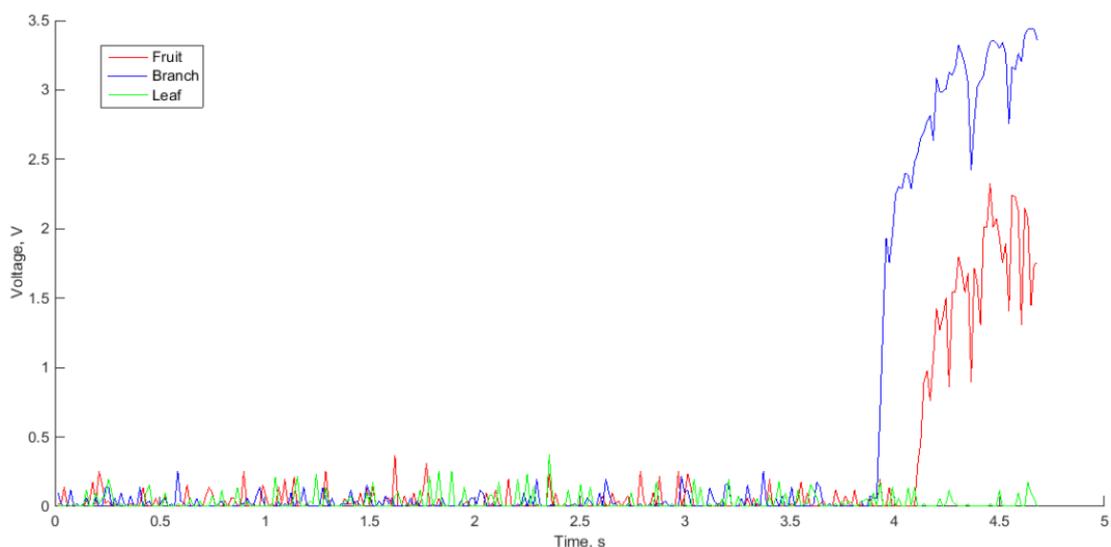


Figure 4-6 Examples of FSR test result for a leaf, a fruit, and a branch.

4.3.4 Conclusions

A simple test in the greenhouse was performed to evaluate the suitability of force sensing resistors for use in automatic harvesting. The results of the performed tests suggest that the sensitivity of the FSR sensor chosen for this experiment is insufficient for the intended task. The sensor was able to detect the contact only 8 of 17 of cases. During the successful detections of the contact with a sweet pepper fruit, the mean output of the sensor was calculated to be 1.31V, while during a contact with a major branch it was 2.84V. No change in output was detected for touch with a leaf. Observations during the test suggest that the main reason for unsuccessful measurements was relatively low resistance force from the fruits. In many cases, the slider pushed the target out of the way without registering any contact. All of the contacts were on the side of the sensor. The attempt to counter the insensitivity of the outer section of the sensor surface was only partially effective. The adhesive used to attach the sensor to the base rod was found to stiffen the base of the sensor thus making it harder to bend. It was concluded that the FSR technology itself can be used in automatic harvesting for tasks where contact with the sensitive surface can be assured. For the task described in this section, however, the performance of the tested sensor was found to be insufficient. The problem of the insensitive area could be solved by manufacturing a custom shape FSR sensor that would match the shape of the fingertips, but such task was found to be too complicated to be reasonable.

4.4 Piezo Touch Sensor

4.4.1 Background

During the course of this study, it was decided to use already existing scissor type cutter-pincer end-effector (*fig. 4-7*). This end-effector performs cutting and gripping of the stem with a single motion, therefore, only the stem position is necessary for harvesting. The localization of stem is performed as described in the section *Pose Estimation*. The pose estimation algorithm calculates the position of the stem from the estimated pose of a fruit in space in cases when the stem is not visible due to occlusions or fruit orientation. The presented sensor was developed as means of verification to confirm that a stem is in the calculated position. The V-shape end-effector is convenient to position a stem in the center of the scissors, where the sensor is mounted. The end-effector design, together with the sensor

placement, theoretically allow for successful detection, if the calculated stem position has a lateral error less than half of the opening width of the cutter.



Figure 4-7 The end-effector used by the harvesting robot.

4.4.2 Overview

The experience gained from the experiments with the FSR sensor indicated that none of the transduction techniques in their basic form fit the requirements and can be used directly. Therefore, a custom sensor must be designed for the specific purpose that would have all the specific drawbacks eliminated. Piezoelectric transduction technique was chosen as the most promising and versatile. The problem of detecting touch with an object was viewed from a different perspective. In theory, when two objects are connected a third object is created with physical and mechanical properties of the connected bodies. In this case properties of one of the objects are known, but not the other. When analyzed from this perspective, a touch between two objects can be considered as adding an extra mass to one of the objects and changing the stiffness properties. There are many mass measurement systems available based on piezoelectricity. The most common mass measurement technique by piezoelectricity is a measurement of the natural or resonant frequency of the system. The resonant frequency of any simple mechanical system can be expressed by equation

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (4-3)$$

where

- k = stiffness of the system
- m = mass of the system

If an additional mass M is added to the system, the resonant frequency changes as follows:

$$f'_0 = \frac{1}{2\pi} \sqrt{\frac{k+\Delta k}{m}} \sqrt{\frac{m}{m+M}} \quad (4-4)$$

This shift in the resonant frequency depends on the amount of the mass added M and the force of the contact, which changes the stiffness properties k .

The use of piezoelectric materials for mass detection has been researched for already more than half century [73]. It was found that the resonant frequency shift is linearly proportional to the added mass when a small mass was uniformly distributed over a piezoelectric crystal. This gave a base for development of many different piezo-based mass measurement systems, such as femtogram mass measuring system developed by Pang W. et al. [74], or macro scale mass measurement system development by Rabih A.A.S. et al [75] etc. The design used in this research was inspired by a muscle stiffness measuring device designed by Han H. and Kim J. [76]. The developed sensor consists of two adjacent multilayer piezo stack actuators attached to a base and covered by a contact tip that acts like a mechanical coupling. One of the stack actuators (from now on referred to as “the driving” actuator) is actuated with a sine wave signal with a frequency equal to one of the resonant frequencies of the sensor. The mechanical coupling through the contact tip transfers this motion to the second (“the reading”) actuator, which in turn generates charge due to this motion. Any contact with the contact tip affects the sensor system by increasing mass of the system and changing the stiffness properties. Due to this change, the frequency of the driving signal is not equal to the resonant frequency so the amplitude of generated charge is significantly lower. The generated charge is measured by an ADC and sent to a PC for processing and evaluation.

4.4.3 Piezoelectricity

A piezoelectric material is a material that generates a charge when a stress is applied (direct piezo effect) or changes its dimensions when a voltage is applied (inverse piezo effect). The direct piezo effect converts mechanical energy into electrical energy and therefore is often used in sensor technology, whereas the inverse piezo effect is more suited for actuator development as it converts electrical energy into mechanical. The relationship between electric and elastic properties of piezoelectric elements in simplified form can be represented as follows:

$$D = d \times T + \varepsilon^T \times E \quad (4-5)$$

$$S = s^E \times T + d \times E \quad (4-6)$$

where

- D = electric flux density
- T = mechanical stress
- E = electrical field
- S = mechanical strain
- d = piezoelectric charge coefficient
- ϵ^T = permittivity (for $T = \text{constant}$)
- s^E = compliance or elasticity coefficient (for $E = \text{constant}$)

A single piezo element provides relatively short displacement. To increase the performance without increasing the voltage necessary for actuation, many individual layers are mechanically connected in series and electrically connected in parallel (fig. 4-8). When a voltage is applied to one of the piezo stacks, the length of that stack increases according to equation

$$\Delta L_0 \approx n \cdot V \cdot d_{33}, \quad (4-7)$$

where

- n = number of stacks in the actuator
- V = operating voltage
- d_{33} = strain coefficient in the longitudinal polarization direction

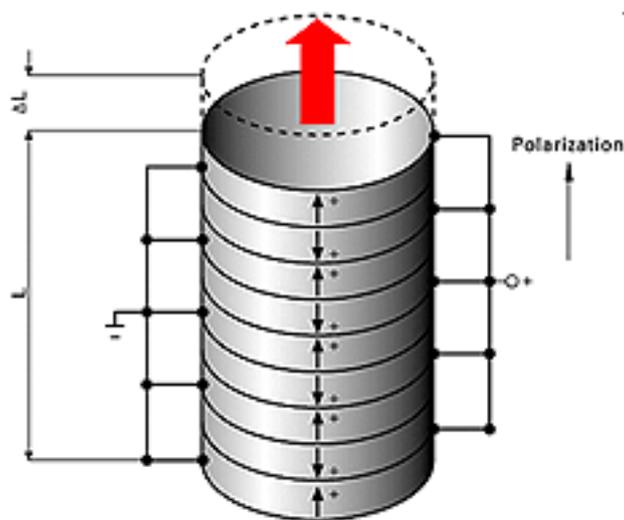


Figure 4-8 Multilayer piezo stack actuator layer polarization and electrode configuration.

In this particular design, however, the free end of the driving actuator is connected to the second actuator through the contact tip, which acts as mechanical coupling. The force acting on the second actuator can be described as follows:

$$F_{max} \approx k_{T1} \Delta L_0 \left(1 - \frac{k_{T1}}{k_{T1} + k_S}\right), \quad (4-8)$$

where

- k_{T1} = stiffness of the driving piezo actuator
- ΔL_0 = max nominal displacement without external force or restraint
- k_S = stiffness of the contact tip in combination with the second actuator that acts as an external spring

When an external force is applied to the contact tip, the dimensional change of the second actuator and consequently the generated charge is modified:

$$\Delta L \approx \frac{F - F'}{k_{T2}}, \quad (4-9)$$

where

- F = force generated by the first actuator
- F' = counteracting force
- k_{T2} = stiffness of the second piezo stack actuator

Note (1): It is generally not advised to apply pulling forces to a piezo stack actuator, but in our case the pulling was applied by an actuator of the same type so the generated pull will not be greater than what the actuator can withstand. Furthermore, the driving piezo stack was actuated with a low peak voltage (10V for the first prototype and 2V for the second prototype), which was significantly less than 150V that is generally recommended for this type of actuators for full stroke application.

Note (2): One of the arguments against the use of piezoelectric elements for this particular application is high temperatures that can be reached in a greenhouse. The standard working temperature of a piezo stack actuators is in the range from -20° to +85°, however, stack actuators that can work in a range of -40° to +150° can be produced by selecting an appropriate adhesive for bonding the layers. Moreover, by using a special soldering technique the working range can be further increased to from -271° to +200°. As the temperature in the greenhouse does not reach outside this range, it was concluded that piezo stack actuators can be safely used in the greenhouse environment.

4.4.4 Prototype I

Setup

The first prototype of the developed sensor consisted of two piezo stack actuators with dimensions 3x3x5mm (Newcastle Innovation Ltd.). Schematic representation of the measurement system is shown in *fig. 4-9*. The contact tip was made of Polylactic Acid (PLA) with a 3D printer in a trapezoidal prism shape with a slightly curved top (*fig. 4-10*). The driving signal was generated by an arbitrary waveform generator AWG2005 (Sony Tektronix Inc.) and it was a sine wave with 28 kHz frequency and 10V amplitude.

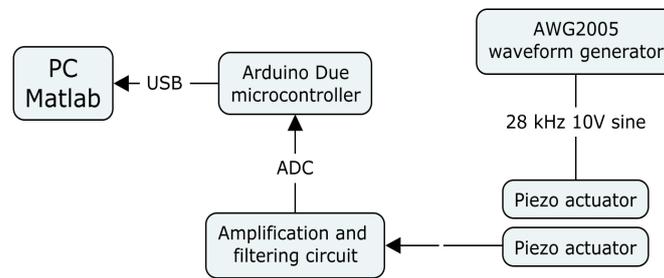


Figure 4-9 Schematic representation of the measurement system.

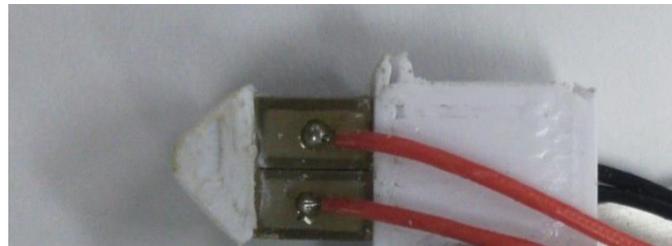


Figure 4-10 Developed sensor, consisting of two piezo stack actuators and a contact tip.

The resonant frequency of the sensor system was determined by measuring the sensor output while changing the frequency of the driving signal from 1 kHz to 40 kHz with 1 kHz increments. The measured frequency response can be seen in *fig. 4-11*. The charge generated by the reading piezo stack was first sent to the amplification circuit, which had a 10x gain and which did full rectification of the signal. The signal from the amplifier circuit was then read by a 12-bit ADC of an Arduino DUE microcontroller board. The default Arduino Due ADC sampling time is 40us, which gives 2500SPS sampling rate. According to the Nyquist Sampling theorem, the sampling rate must be “*greater than twice the highest frequency component of interest in the measured signal*” [77]. The necessary sampling rate for the working frequency of 28 kHz must be greater than 56 kSPS which is more than 20x more than the default Arduino DUE ADC

reading speed. High-speed ADC measurement with the sampling rate of approx. 1MSPS (994 kSPS) was achieved by proper register configuration and usage of the DMA controller [78]. In the microcontroller 256 individual ADC measurements were combined into a data packet, which was sent to the computer for processing through the native USB port. This data packet was used to calculate the amplitude of the generated charge. All data processing was performed with commercial software Matlab (Matlab R2013b, MathWorks, Inc.).

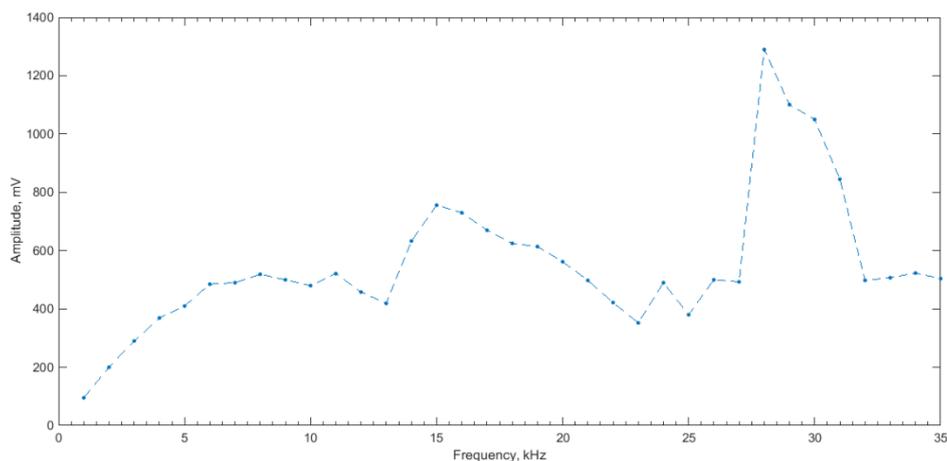


Figure 4-11 Frequency response of the developed sensor.

Evaluation

To properly interpret information from a sensor, behavior of the sensor must be well understood. Several different tests were performed to study the viability of the sensor for the given task.

Stability test – to verify the sensor’s stability when operated for long periods of time the sensor was continuously operated for 20 min and the data packet was received once per second. The test results revealed a slight value drift over time as shown in *fig. 4-12*. The mean value for the stability test was 2074mV with 29.5mV standard deviation. During the test, mean amplitude fluctuated between the minimum value of 2019mV and the maximum value of 2134mV, which is a 5.39% difference. This peculiarity indicated a need for sensor calibration, which was tested later, during the fruit stem detection test.

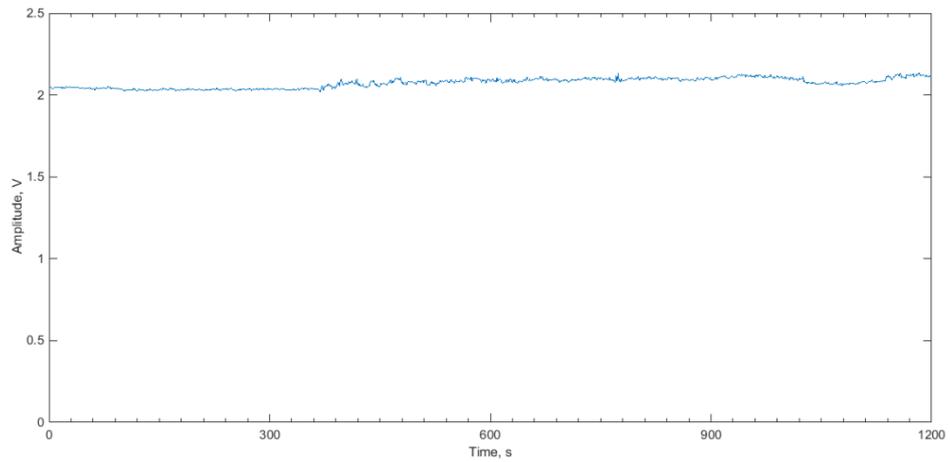


Figure 4-12 Stability test results.

Movement test – to analyze the impact of arm movement on measurement the sensor was attached to a robotic arm (*fig. 4-13*) and the readings were acquired while the arm was moving. The test results showed a slight influence of arm movement on the reading, which can be seen in *fig. 4-14*. The mean amplitude for a static arm was 1307mV (Std. dev. 1.4mV), while for a moving arm it was 1293mV (Std. dev. 5.1mV). Although the change is relatively small (1.05%), it still must be compensated for in the control software for accurate measurement.

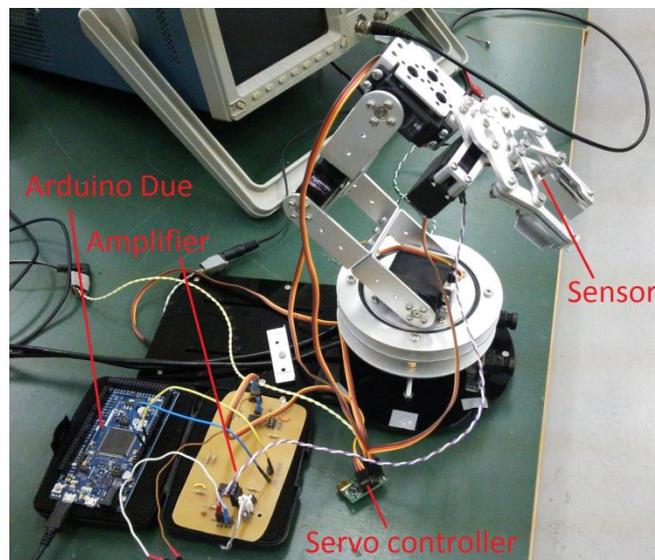


Figure 4-13 Test setup

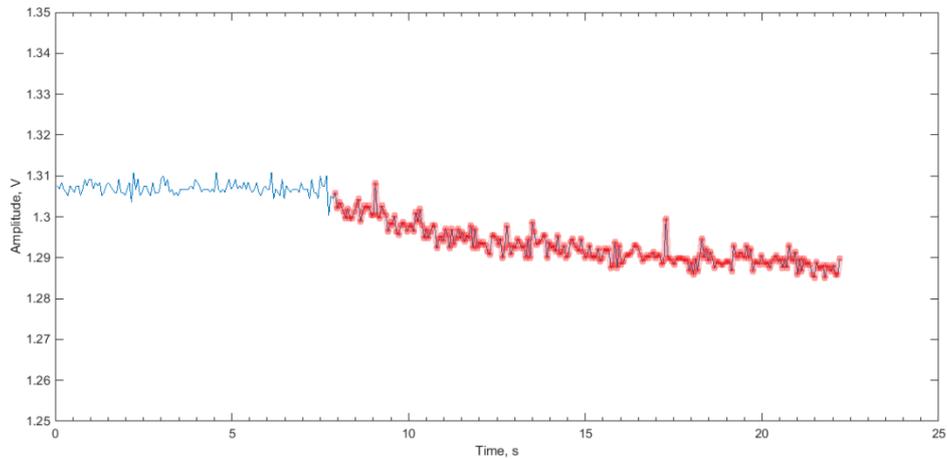


Figure 4-14 Movement test result, measurement after movement marked red

Sensitivity test – to analyze the sensitivity of the sensor a known force was applied to the sensor using a tension gauge. This test confirmed that the sensor was able to measure force as low as 2g. As it can be seen in *fig. 4-15*, when 2g force was applied to the sensor, the mean amplitude decreased by 3.21%, which is significantly more than the change caused by arm movement.

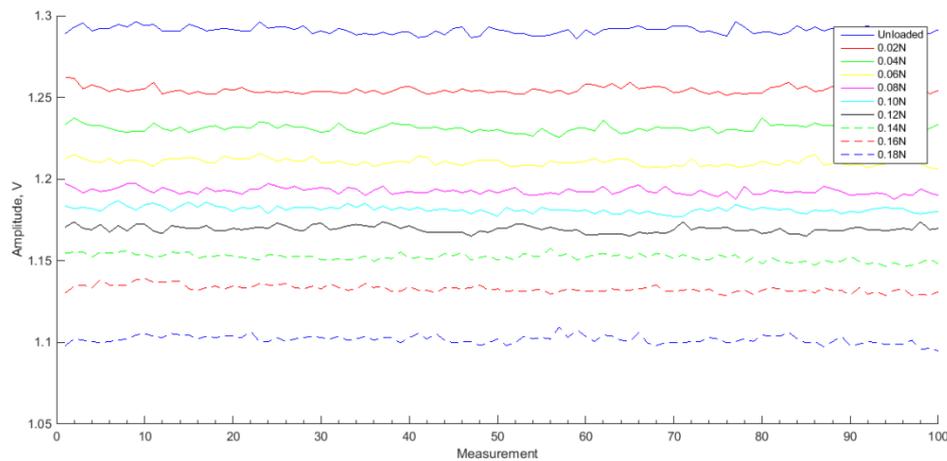


Figure 4-15 Sensitivity test result

Repeatability test – to ascertain that same force to the sensor within a short period of time will give the same output a constant pressure was applied to the sensor several times with short breaks between measurements. The test results were in good agreement with the assumption that measurement by the system is stable and accurate in relatively short time periods (under 5 min) since no significant amplitude drift was observed. The results of the repeatability test are shown in *fig. 4-16*. The measurement was done 4 times with 3 breaks.

The measurement was stopped during the breaks, and the measurement time and break time were both 15s. The mean amplitude value and the standard deviation were 1079mV and 1mV respectively.

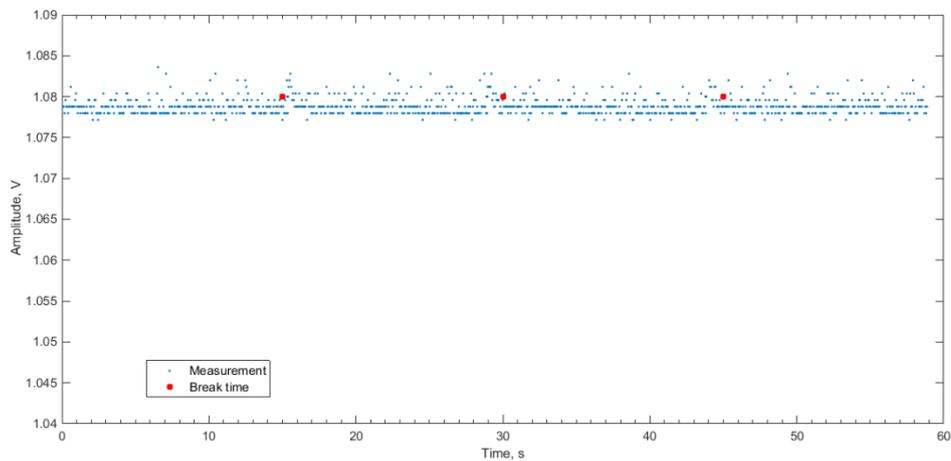


Figure 4-16 Repeatability test result, break time is not included in the graph but only indicated with markers

Settling test – to measure sensor settling time first measurement was done on an unloaded sensor, and then the pressure was applied to the sensor for 10s, measurement was continued few seconds after the force was released. The results of the settling test showed a settling time of 60ms (time for one measurement, marked red in *fig. 4-17*), which is insignificant for an application in real time. When the contact between sensor and object was canceled, the reading returned to the unloaded value almost instantly. Therefore, there is no need to let the sensor rest after contact with an object.

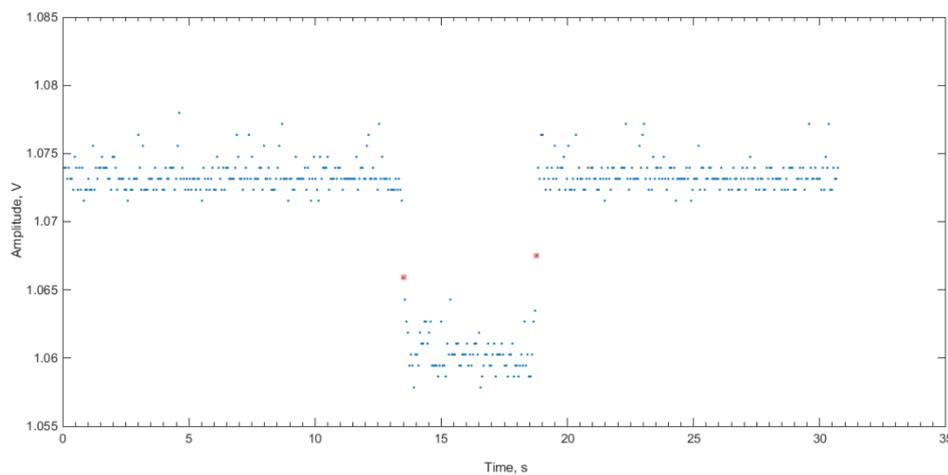


Figure 4-17 Settling test result; transition of the sensor output takes a single measurement to settle

Touching a stem – to verify the ability of the sensor to detect contact with a sweet pepper stems the sensor was pressed against a sweet pepper stem several times with rest periods between contacts. The test results were promising for practical application of the sensor. Contact with the stem noticeably changed the sensor reading as can be seen in *fig. 4-18*. The mean amplitude for an unloaded sensor in this test was 1345mV (Std. dev. 3mV) while for a loaded sensor it was 1281mV (Std. dev. 9mV). The difference between the mean amplitude of loaded and unloaded sensor was 4.76%, which is adequate for detection.

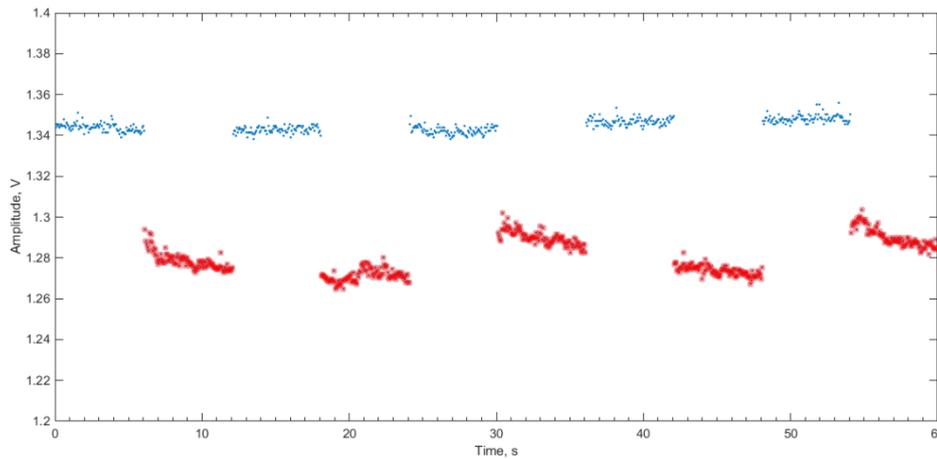


Figure 4-18 The result of touch stem test, measurements with contact marked red

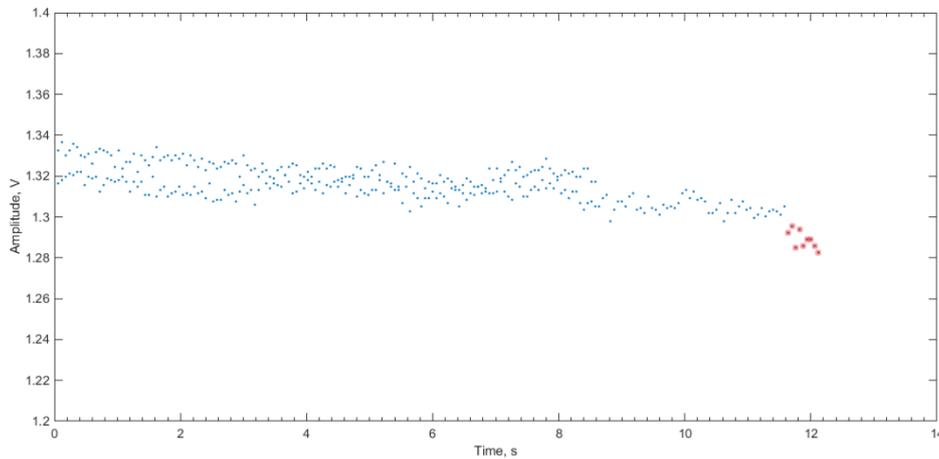


Figure 4-19 Cutting test result, measurements after contact marked red

Cutting test – to verify that the measurement speed is sufficient for application in real time and that the sensor will detect contact with a stem, the sensor was moved to a predetermined position using the robotic arm while holding a sweet pepper stem in its path. A successful sweet pepper harvesting attempt was performed in a laboratory environment (*fig. 4-19*). During this test, the sensor was recalibrated each time after all motors had moved one

step. Recalibration was found to have no significant effect on harvesting time as the calibration procedure took only 60ms.

Conclusions

A novel touch sensor for sweet pepper stem detection based on piezoelectric effect was developed. Several tests were executed to assess the performance of the designed sensor system. The results of the tests support sensor viability in sweet pepper harvesting robots. The sensor's sensitivity was proven to be sufficient to detect contact with sweet pepper stem, and the stability parameters indicated reliable performance. One of the limitations of this sensor was its relatively small active surface, which could be solved by changing the shape of the contact tip. The second issue was dependence on the signal generator. The signal generator used for this prototype was too bulky and energy inefficient to be used on a harvesting robot.

4.4.5 Prototype II

Setup

Several important modifications were done to the design of the sensor system to improve the performance. The contact tip was changed from the plastic trapezoidal prism to a complex shape, which was handmade out of aluminum (*fig. 4-20*). Several different shapes were designed and tested to find the most appropriated.



Figure 4-20 The second prototype of the developed sensor

The second improvement in hardware was the change of the signal generator. The bulky SWG2005 was changed to an Arduino Due controlled AD9850 DDS signal generator with the ability to generate a sine wave with frequency up to 40 MHz. The driving signal amplitude was only 0.2V, which was found to be acceptable both in terms of actuating the sensor and for saving space and energy. As it can be seen from the functional diagram of the measurement system (*fig. 4-21*), the amplifier circuit was removed and the amplification was done in the software instead by changing the data processing algorithm.

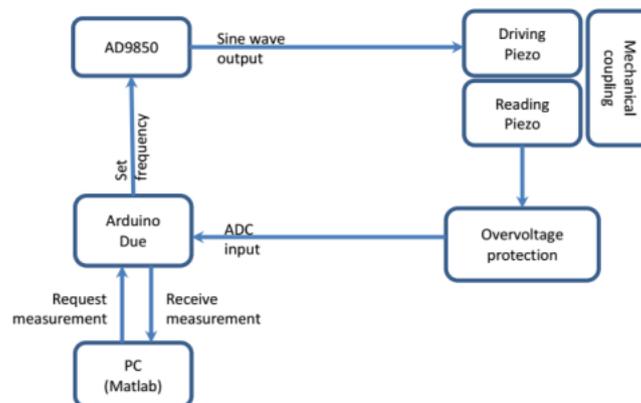


Figure 4-21 Functional diagram of the measurement system

The size of a single data packet was increased to 512 measurements. Instead of calculating the amplitude of 10 measured cycles, the generated charge was calculated by calculating the integral of the produced signal according to the following procedure. First, the DC offset was removed from the signal by subtracting the mean value from each time-domain signal point. Then the signal was squared and the integral of the signal was calculated by using trapezoidal numerical integration. The value used in analysis can be approximately derived from the following equation:

$$Q' = \int (\sigma V)^2 dV, \quad (4-10)$$

where

- Q' = value used for analysis
- σ = constant for scaling the actual voltage to the measured ADC bin value, [1241]
- V = measured ADC signal consisting of 512 measurements

With such method, even small changes in the signal amplitude can be detected (see *fig. 4-22* for comparison between amplitude and the integral of squared amplitude). For simplicity, one such data packet consisting of 512 ADC readings further in the text will be referred to as “a measurement”.

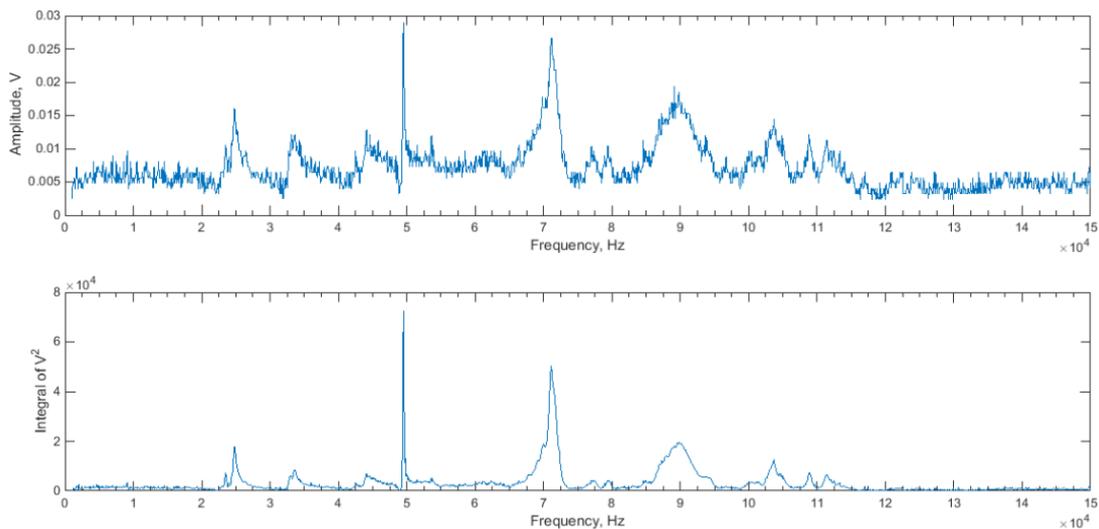


Figure 4-22 Comparison between amplitude measurement and integral of squared amplitude for frequency response measurement. The benefits of using the later method are evident, the amplitude of the signal was increased by a factor of 10^5 and no noise can be observed.

Evaluation

To verify the suitability of the designed sensor for the intended task, a thorough evaluation was performed. The main requirement for the sensor was to be able to detect contact with a sweet pepper stem while being insensitive to a contact with leaves. Following tests were performed for the assessment: frequency response test, stability test, sensitivity test, movement test, and field testing in a greenhouse.

The frequency response test was performed by changing the frequency of the driving sine signal from 1 kHz to 150 kHz with a 100 Hz increment while measuring the generated charge. The purpose of this test was to detect all resonant frequencies of the sensor in the tested range to decide on the most appropriate “actuating frequency”. The generated charge was acquired by the ADC and processed as described in the previous section. 10 consecutive frequency response tests were performed to assess the stability of the frequency response and the resulting calculated integral of the measured signal versus frequency plot is shown in *fig. 4-23*. As it can be seen, there are two most prominent resonant frequencies within the tested frequency range at 49.4 kHz and at 71 kHz. Other peaks have considerably lower amplitude. It can also be noted that the frequency response has a low deviation suggesting good sensor stability. These two resonant frequencies were chosen as the driving frequencies for the sequent tests.

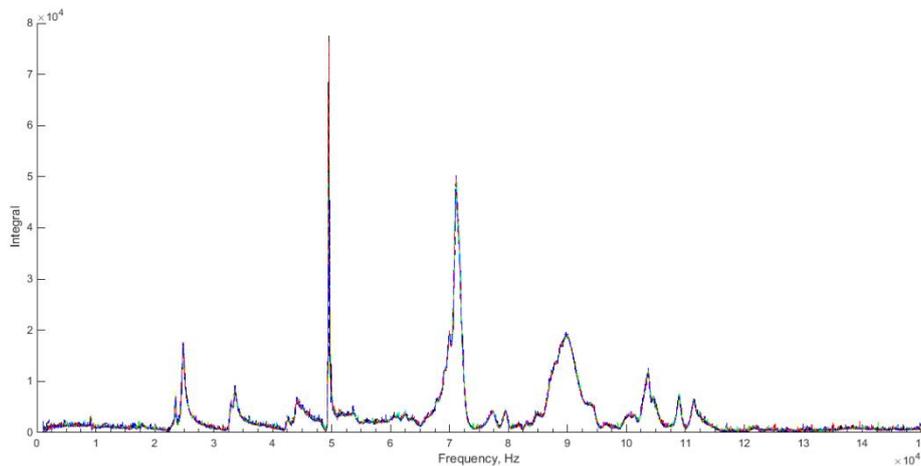


Figure 4-23 Frequency response of the developed sensor

The stability test was performed to evaluate the behavior of the sensor in a longer duration measurement. 10 consecutive stability tests were performed, and each test consisted of 6000 measurements with a 100ms between each measurement. Time used for a single

reading and data processing was measured to be approx. 13 ± 1 ms. The resulting time spent for each test was approximately 11 minutes and the resting time between two consecutive tests was 10 minutes, during which the sensor was turned off and the measurement circuit was reset. Both 49.4 kHz and 71 kHz resonant frequencies were used for this test to determine the stability of the response for each used frequency. According to the test results (fig. 4-24), the generated voltage for the driving frequency of 49.4 kHz had a large mean standard deviation of 2.83×10^3 (approx. 12% of the mean value), compared to 401.2 (approx. 1.3% of the mean value) for driving frequency of 71 kHz.

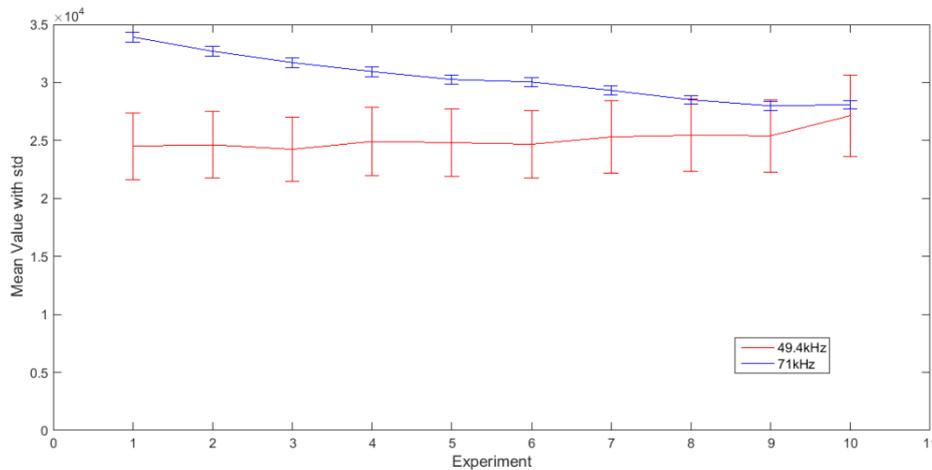


Figure 4-24 Stability test results

The reason for the instability of the measurement with the driving frequency of 49.4 kHz is not well understood. One possible explanation is a presence of frequency components that periodically increased the amplitude of the generated charge, consequently magnifying the integral of the signal. FFT analysis showed a presence of a low-frequency component in the measured signal, but the same component was present also in the 71 kHz measurement (fig. 4-25). The instability could also be explained by the sharpness of the resonant peak at 49.4 kHz. The frequency deviation of the used signal generator might be sufficient to occasionally slightly deviate from the set frequency. The maximum change in the mean value of the measured signal during a single 10min experiment was approx. 1.73% of the mean value for the resonant frequency of 71 kHz and 2.1% for the resonant frequency of 49.4 kHz. The reason for this value shift is assumed to be due to the characteristic behavior of piezo elements, such as drift, hysteresis, and creep. The change of the mean amplitude in prolonged measurement was concluded to be insignificant, but a considerable deviation between individual experiments was observed, due to which a global predefined detection threshold value for touch detection

can't be used, and the sensor must be calibrated each time it is turned on by measuring the mean value of the measurements for the first few seconds. It was also decided to re-calibrate the sensor after every 10 minutes to counteract the output drift.

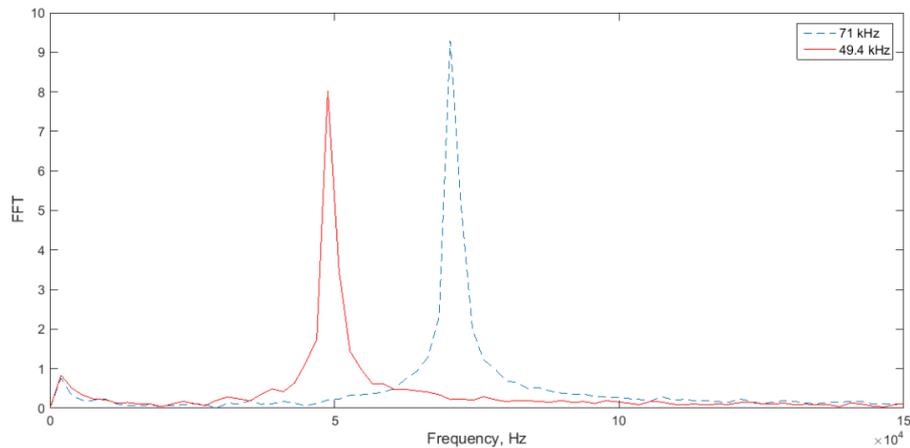


Figure 4-25 FFT analysis result

During **the movement test**, the influence of the manipulator movement on the sensor output was assessed by performing measurements while moving a manipulator with an attached end-effector and sensor. A simple 3 DOF robotic arm was used as the manipulator for this experiment (see **fig. 4-26** for the test setup). The control system for the manipulator consisted of an Arduino UNO, Adafruit 16-Channel 12-bit PWM/Servo driver for servo control and a joystick button type controller. The manipulator was moved randomly while the sensor output was recorded. The experiment was repeated 10 times to analyze the stability of the result. The standard deviation was calculated and compared to that of the stability test to detect deterioration of the stability of the sensor output. Only the 71 kHz driving frequency was used for the movement test due to the unstable results of the 49.4 kHz driving frequency in the previous test.



Figure 4-26 Setup for the movement and greenhouse tests

The results of the movement test showed that the movement of the sensor had only a very slight influence on the measured sensor output. The standard deviation of the measurement with a moving sensor ranged from 602.47 (1.5%) to 868.35 (2.2%) with the average of 652.38 (fig. 4-27). For comparison, the standard deviation calculated for the stability test results ranged from 366.19 (1.24%) to 430.02 (1.37%) with the average of 401.2. The value in the brackets is the numerical value compared to the mean value of the particular measurement.

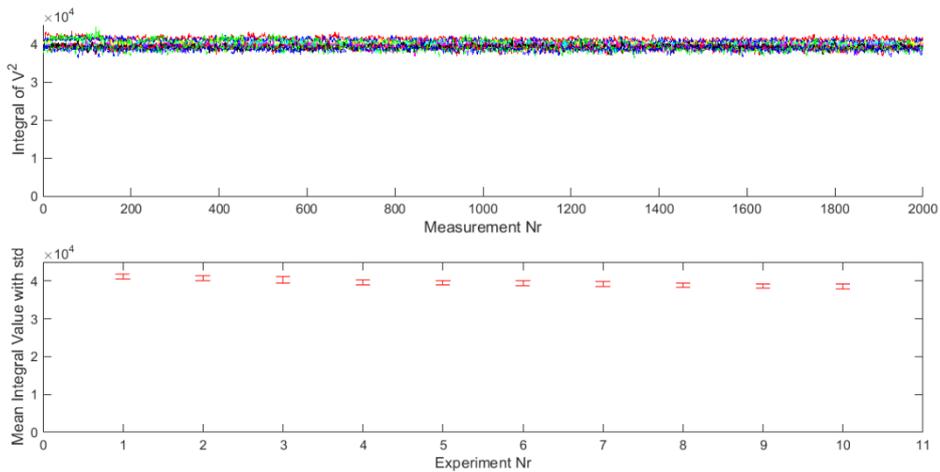


Figure 4-27 Movement test result

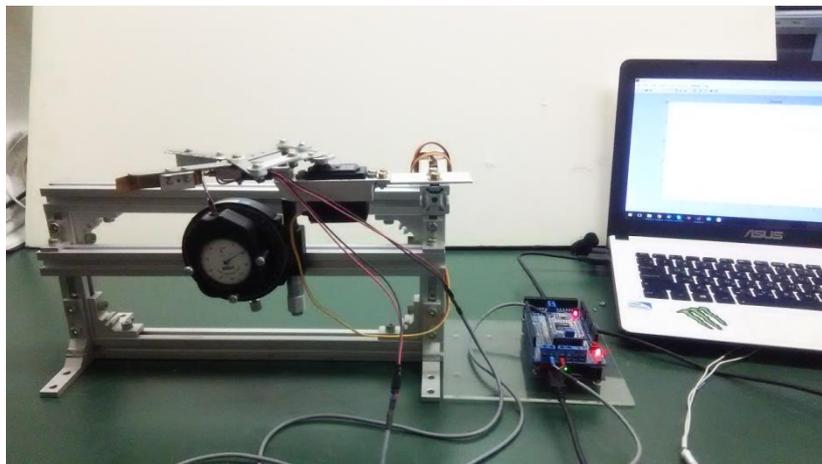


Figure 4-28 Setup for the sensitivity and stability tests

The Sensitivity test was used to evaluate the sensitivity of the sensor to a touch by pressing a pin to certain parts of the contact tip with a known force. The used force was from 0.03 to 0.3 N with 0.03 N increments and it was measured by dial tension gauge (fig 4-28). The results of the sensitivity test were in accordance with the results of the mode shape analysis

(fig. 4-29). It was experimentally proven that the most sensitive parts of the contact tip are parts with the greatest displacement for the mode shape of the particular resonant frequency.

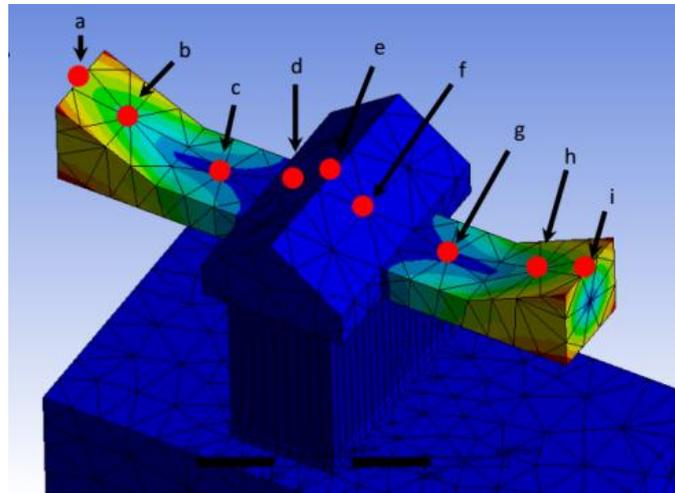


Figure 4-29 Mode shape analysis results for 70 kHz frequency, which was the closest resonant frequency to the driving frequency found by FEA

As it can be seen from the *fig. 4-30*, the most sensitive parts of the sensing tip were the wings (sections a – c and g – i). The reading of the sensor started to change when a pinpoint force of approx. 0.06N were applied to sections g and i. Sections d and f remained insensitive in all of the tested range while the section e started to detect a touch from 0.18 to 0.21N. A slight difference between the sensitivity of the side a – c and the side g – i could be observed, and the side with sections g – i appeared to be more sensitive. Such result wasn't surprising as the sensing tip was manufactured by hand with relatively large tolerances. A misalignment of the sensing tip on the actuators should also be considered. As the most sensitive parts of the sensing tip were the wings, which are intended to be the main contact surface with a stem, it was concluded from the test results that the sensitivity of the sensor was acceptable, as the force necessary to move a stem was measured to be starting from approx. 0.18N to over 0.3N depending on the stem position and characteristics of the branch the fruit was attached to (measurement obtained by the same dial tension gauge).

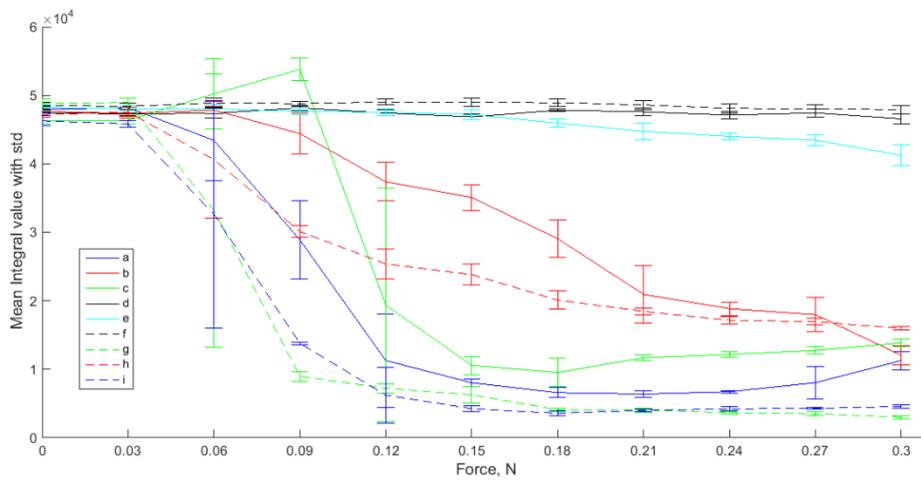


Figure 4-30 Sensitivity test results

During **the field testing**, 25 sweet pepper fruits were randomly selected. The manipulator was moved manually with a controller to induce contact between the sensor and the stem of the chosen sweet peppers while recording the sensor output (test setup the same as in [fig. 4-26](#)). The change of the generated charge was in the range of 11.18% to 73.15% with the average of 31.37% (see [fig. 4-31](#) for an example). The value of generated charge change was obtained by comparing the mean measurement value before and after contact. The great variance between the results of this test can be explained by differences in touch quality and force from case to case. This result shows that the decrease of the generated charge due to contact with a stem is significantly higher than the standard deviation. The results of this test can be used to select the value for a detection threshold. Considering the calculated lowest value of 11.18% in this experiment and the greatest calculated standard deviation of 2.2% from the movement test for a moving manipulator, an ideal detection threshold would be in the range from 5% to 8% of the drop in generated charge compared to a value obtained during calibration. No change in the sensor output was observed during a contact with leaves.

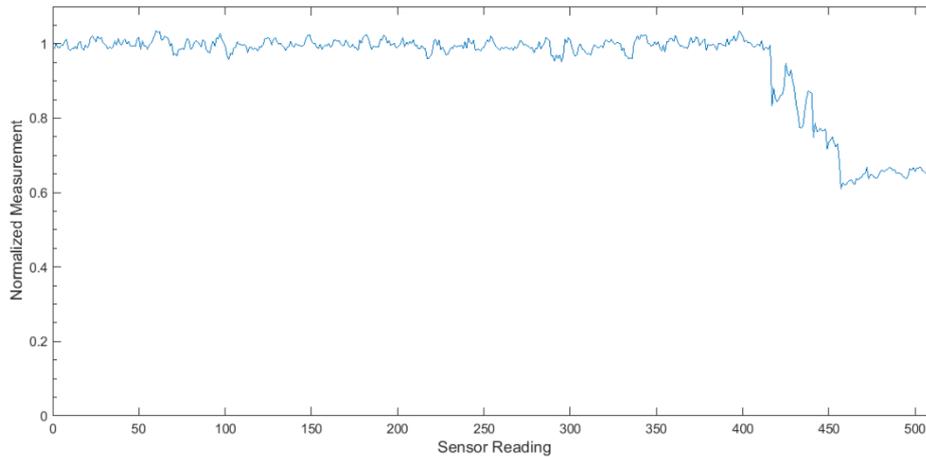


Figure 4-31 Example of a greenhouse test result (normalized)

Conclusions

A sweet pepper stem position calculation algorithm had been developed previously, which gave an approximate location of the cutting point based on the fruit pose estimation. In this section, a piezo-based touch sensor was described for stem position verification in the calculated position. The developed sensor consists of two piezo stack actuators, and a contact tip mounted at the base of the cutter of the sweet pepper harvesting robot. Contact between a stem and the sensor should occur when the stem would be in the cutting position.

Several tests were performed to evaluate the performance of the proposed sensor. The frequency response test revealed that there are two most prominent resonant frequencies within the tested range that could be used for driving the sensor at 49.4 kHz and at 71 kHz. The stability test showed that the result from the resonant frequency of 71 kHz was much more stable compared to that for the resonant frequency of 49.4 kHz. The movement test confirmed that the influence of the manipulator movement on the sensor result was negligible and increased the standard deviation of the sensor reading by only approx. 1% of the mean calculated generated charge value. The sensitivity test confirmed that the sensitive parts of the sensing tip can detect touch with forces starting from approx. 0.06N. The field testing in a greenhouse confirmed that the sensitivity of the sensor was sufficient to detect contact with a sweet pepper stem, with the average change of the calculated sensor output being approx. 31.37%. The results of the performed tests strongly suggest a good suitability for this sensor to the intended task.

4.5 Overall Conclusions

A new approach for detecting the sweet pepper fruit and stem for automated harvesting was introduced in this section. In this approach, the visual information obtained by a harvesting robot is supplemented with tactile information, acquired by a specially designed sensor. The initial requirement for the sensor was to be able to detect a contact with a sweet pepper fruit. Several different sensing technologies were examined to find the most appropriate technique for this particular purpose. Sensors based on force sensing resistors showed promising results, but the reliability was found to be unsatisfactory. One of the main drawbacks of FSR sensors for this application is the insensitive outer area, which is where different layers of the sensor are attached together. Production of a custom shaped FSR sensor would be necessary to eliminate this drawback, but high costs of such solution together with the uncertainty of the success were strong arguments against advancing this idea any further.

During the course of this research, the end-effector used by the final harvesting robot prototype was decided upon, which in turn changed the requirements to the sensor. The new harvesting method required stem detection instead of the fruit detection so a novel piezo-based touch sensor was developed for stem detection by touch. Two prototypes of the sensor were manufactured and tested. The first prototype showed a good performance, thus assuring that the chosen technology can be used for the intended purpose, but certain features had to be improved. The second prototype was designed using the first prototype as base knowledge and eliminating the issues discovered for the first prototype. Improvements were done both for the hardware and for the software. The contact tip was redesigned to achieve improved sensitivity and increased active surface. The measured characteristic value was changed from the amplitude of the generated charge signal to the integral of the signal over one data packet of 512 measurements. This greatly increased the sensitivity as it can be seen by comparing the change in value after the contact with a stem for the first prototype (*fig. 4-19*) with that of the second prototype (*fig. 4-31*). It can be concluded that a tactile sensor for stem detection was successfully designed. The results of executed tests suggest a good suitability for the given task. It is hoped that the results of this research will encourage other researchers in the automatic harvesting field to experiment with the use of unconventional methods for improving the performance of automatic harvesting robots.

5. Monorail Continuum Automatic Harvesting Robot

5.1 Overview

During the course of this research, an incompatibility between the hardware of the existing sweet pepper harvesting robot prototype and that used by the pose estimation algorithm was recognized. The main issue was absence of a simple and convenient way to mount the vertical slider on the frame of the harvesting robot. The slider can't be mounted in front of the manipulator as it would constrain the moving range of the manipulator. By mounting the slider behind the manipulator, the manipulator would obstruct the view of the slider. Mounting the slider on either of sides would reduce the overall stability of the harvester and it would be at risk of tipping over during the movement of the manipulator or the robot itself. The most advanced automatic harvesting robot that is using vertical sliders is a sweet pepper harvesting robot designed by the "Clever Robots for Crops" team in Netherlands [59]. The vertical slider of the sweet pepper harvesting robot is used for holding the target recognition system, which consists of two color cameras, TOF camera, and an illumination rig, and is mounted on a separate module (*fig. 5-1*). This solution doubled the length of the harvesting robot, thus a lot of space is necessary for turning and storage of the robot.

The existing sweet pepper harvesting robot prototype, which was developed by the research team in Kochi University of Technology, besides the hardware incompatibility also had other issues that were to be solved, such as:

Power supply The existing prototype relied on an external power supply through a power cord. This approach is not recommended for autonomous systems with long travel range.

Movement system The movement system wasn't developed and the prototype was mounted on a mobile platform that lacked an actuating system for movement and had to be pushed manually.

Positioning No system was developed for the guidance of the robot in the greenhouse.

Obstacle avoidance The used manipulator wasn't suited for obstacle avoidance and no obstacle-free trajectory planning was implemented.

Hardware optimization The parts and electronics used for building the robot were not optimized in terms of weight and efficiency. Energy efficiency is an important characteristic of any battery based autonomous system.

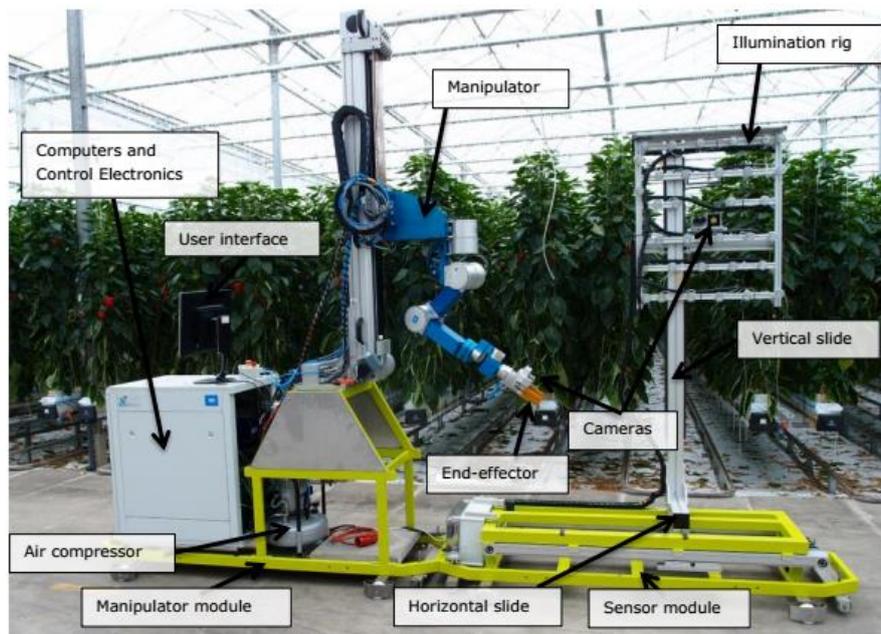


Figure 5-1 Automatic harvesting robot developed by the research team of “Clever Robots for Crops” [59]. Vertical sliders are used for both the manipulator and recognition system.

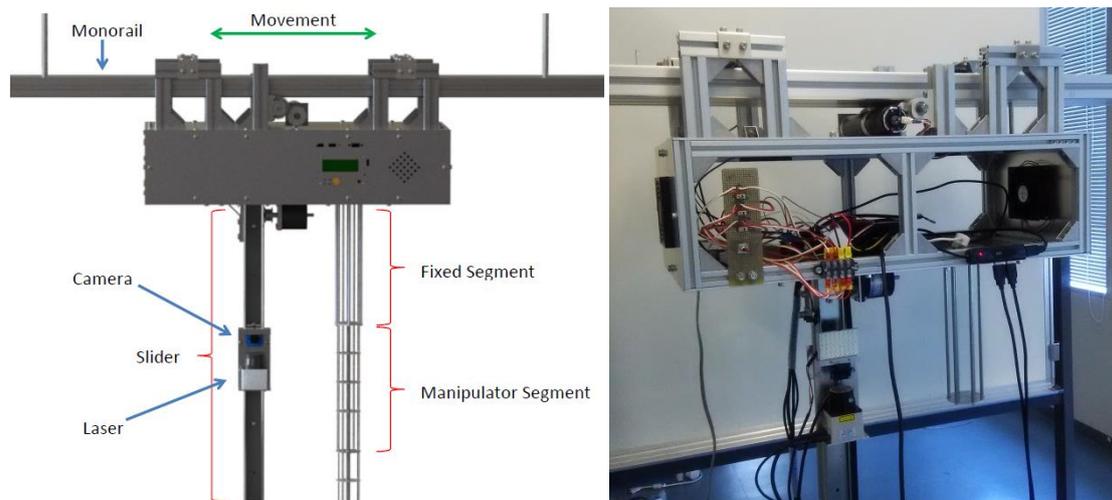


Figure 5-2 The developed monorail continuum automatic harvesting robot, CAD model (left) and half assembled prototype (right).

To overcome these listed issues and to be able to implement the newly developed pose estimation technology a novel monorail type harvesting robot was designed (*fig. 5-2*). The new harvesting robot prototype consists of all the same typical parts of harvesting robots, such as a movement system, a recognition system, and a picking system, but the configuration of the parts differs significantly from that of conventional harvesting robots. The main body of the robot contains all of the electronics and moves along a monorail above the pathway between the plant rows. Both the recognition and the picking systems are attached to the main harvester body. The recognition system consists of a camera, an LED array, and a laser range sensor, which are mounted on a vertical slider. The use of a vertical slider eliminates the possible blind zone as the camera can be moved in a vertical range of over 1m. The picking system consists of a four segment continuum manipulator. This manipulator was chosen to employ its ability to operate in narrow spaces and to introduce obstacle avoidance in sweet pepper automatic harvesting.

This section introduces the new sweet pepper harvesting design, gives description and analysis of the used hardware and harvesting algorithm. Special attention is dedicated to the choice of manipulator and control system. Power consumption and cost analysis are given in the end to show the feasibility of the developed harvester.

Note: During the time of writing this dissertation the new harvesting robot prototype was in the stage of assembly and programming. As a result, no performance information was available and only results of analysis of separate parts of the new prototype will be provided in

this section. The final version of the designed robot might differ significantly from the one described here, and performance test results will be published later in literature.

5.2 Hardware

The hardware of the designed robot can be divided into four main systems – the movement system, the recognition system, the picking system and the main control system. The main tasks of the movement system (*fig. 5-3*) are to keep the robot on the railway and to control the movement along the railway. The movement along the rail is accomplished by a roller, which is actuated by a DC motor (Maxon, 148867) with a gearhead that ensures a sufficient torque for movement and prevents any movement of the robot on inertia. The railway itself is an aluminum frame (A6063S-T5) of size 120x40mm, which is attached to the ceiling of a greenhouse. The frame has a groove along the center line that is used for guidance of the robot by two guiding rods, one on each end of the robot. The robot is held on the rail by four wheels, two on each side of the robot. Positions of the wheels were chosen with respect to the movement roller in the center to ensure the stability of the harvester during movement along the rail and movement of the manipulator when the position of the center of mass is changed.

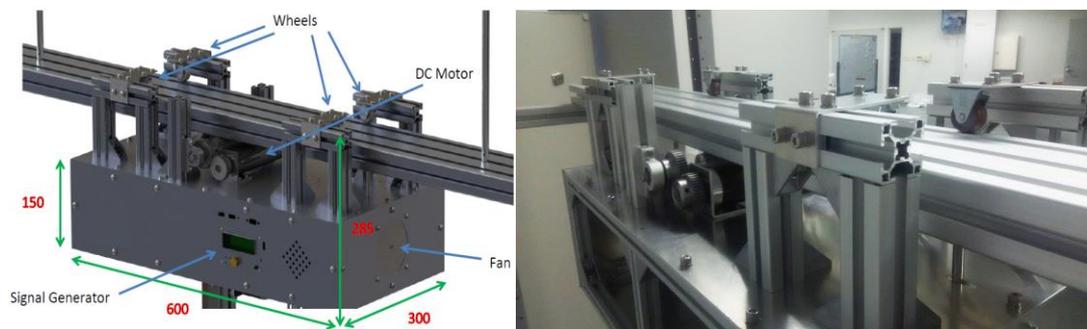


Figure 5-3 Movement system, CAD model with dimensions (left) and assembled prototype (right).

The recognition system is identical to the one described in the section *Pose Estimation – Prototype II*. It consists of a 1200mm long vertical slider with a laser range finder, a USB camera and an LED array mounted on the slider as shown in *fig. 5-4*. This system is used for target recognition and position calculation of the stem as already described in the previous section.

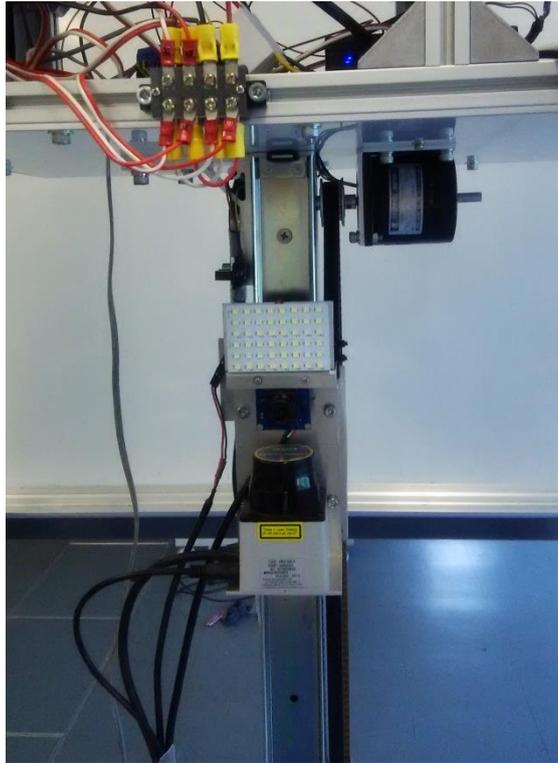


Figure 5-4 Target recognition system, consisting of a laser range finder, a camera, and an LED array

According to the planned design, the manipulator system will consist of a continuum type robotic arm and the end-effector. The designed continuum arm manipulator will consist of four 300mm long segments, each actuated by three tendons, and a backbone. A 250mm long fifth segment was placed at the base of the manipulator to increase the reaching length of the manipulator and consequently the harvesting area without increasing the number of actuators necessary complexity of coordination. Continuum type robotic arms have been the subject of much research during the last decades. Despite a relatively complicated control and lack of rigidity, continuum manipulators offer several advantages over the rigid link manipulators, such as high degree of freedom, obstacle avoidance and conforming to complex movement trajectories. The disadvantages of continuum manipulators are sagging and a high number of actuators. The main reason for choosing this type of manipulator was the offered obstacle-free trajectory planning.

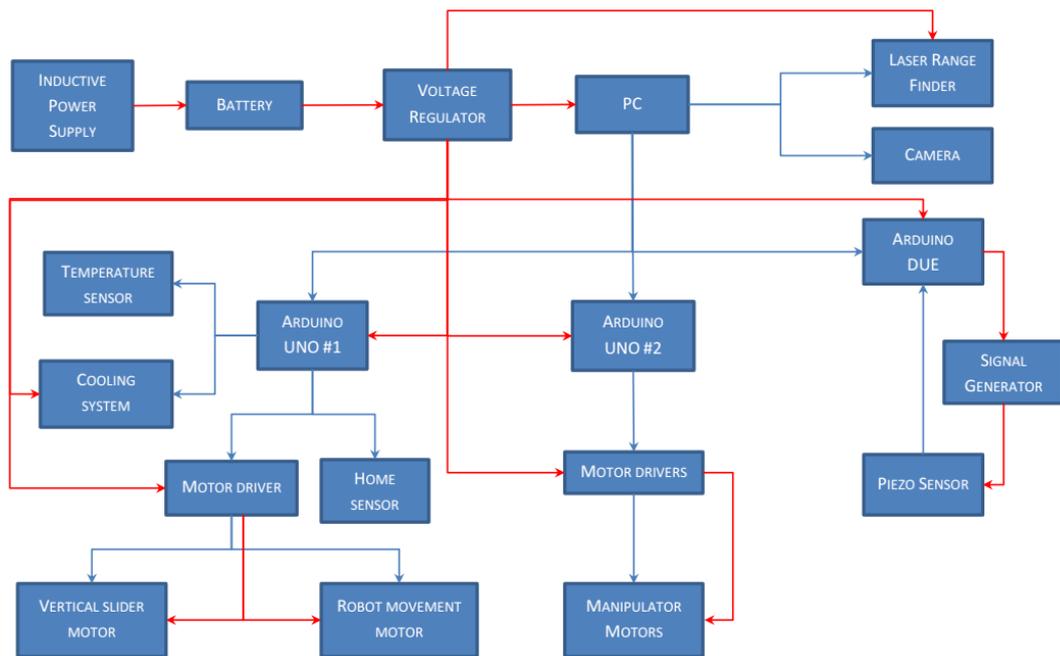


Figure 5-5 Functional schematics of the new harvesting robot design; the blue line represents information flow while the red line – energy flow.

The main control system is placed in the main body of the harvesting robot under the rail. The control system consists of the central processing unit of the robot, which is a low power consumption computer (Gigabyte Brix GB-BXBT-2807), two additional Arduino UNO boards and one Arduino DUE board. One of the Arduino UNO boards is equipped with an Adafruit Motor Shield v2 and Arduino Motor Shield v3, and its main tasks are the control of the linear slider movement and of the monorail DC motor. The second Arduino UNO board is dedicated to the control of the manipulator. The Arduino DUE board is used for fast data acquisition from the piezo sensor and is reading the ADC measurement of the piezo sensor at approx. 1MSPS speed and sending it to the central computer. The system is powered using four 12V 15000mAh Li-ion battery packs. Functional schematic of the control system for the new harvesting robot design is shown in *fig. 5-5*.

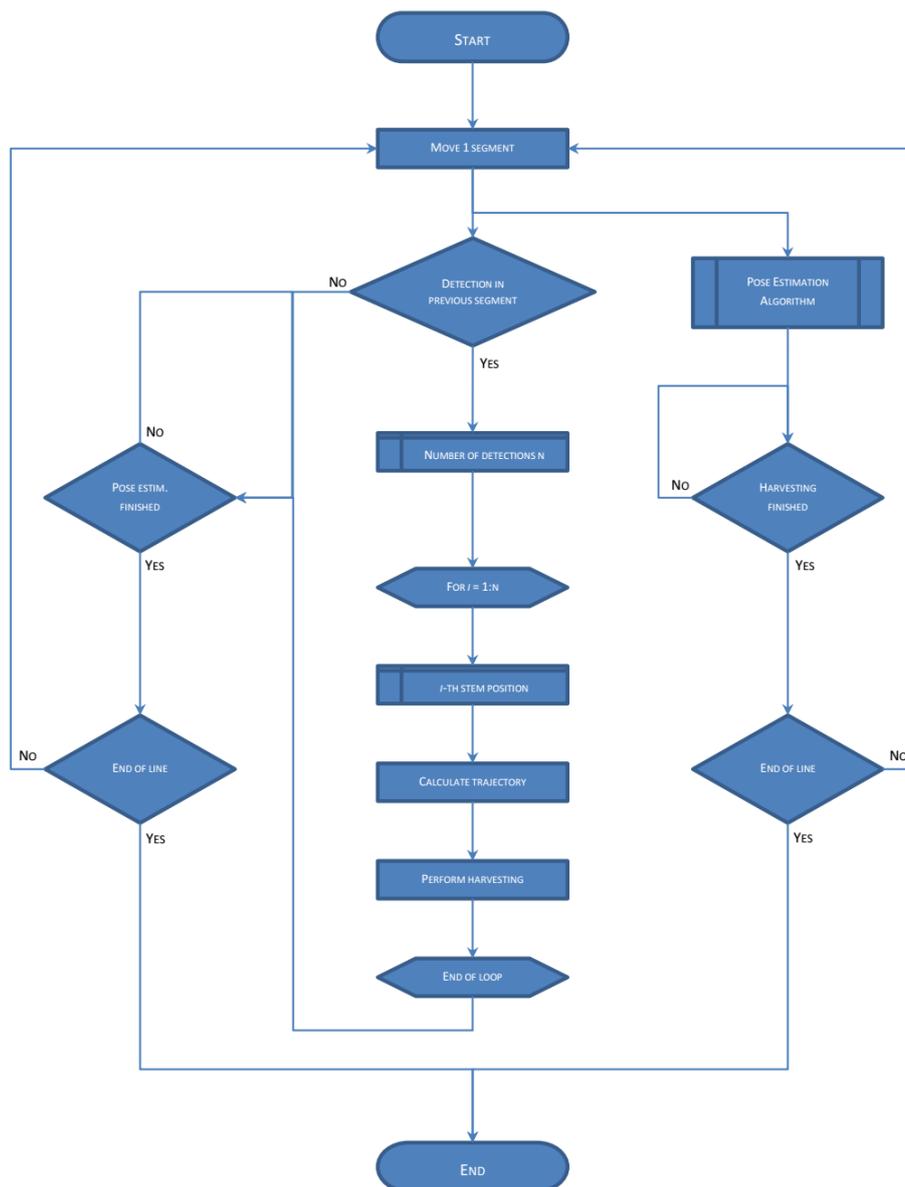


Figure 5-6 Working algorithm of the new monorail harvester; pose estimation algorithm block is identical to the one described in the section *Pose Estimation – Prototype II*.

5.3 Working Algorithm

The harvesting algorithm is divided into two actions that work simultaneously (*fig. 5-6*). To explain the complete algorithm, let's assume that the harvester has started a new harvesting cycle and has arrived at the beginning of sweet pepper bed (*fig. 5-7a*). The initial position of the harvester is with the detection side aligned to the end of the bed. The harvester then moves $\frac{1}{2}$ of its length ("one segment") to position the recognition system in front of the

sweet pepper plants (*fig. 5-7b*). The recognition and stem pose estimation is performed for this segment as described previously (see section *Pose Estimation – Prototype II*). Once the recognition algorithm is finished, the robot moves 1 segment to position the picking system in front of the part of plants, for which the recognition was performed, and to position the recognition system in front of a new segment (*fig. 5-7c*). From this point on the recognition system and the harvesting system are working simultaneously until the end of the plant bed is reached (*fig. 5-7d*). The robot moves to the next section when both systems have finished their particular tasks.

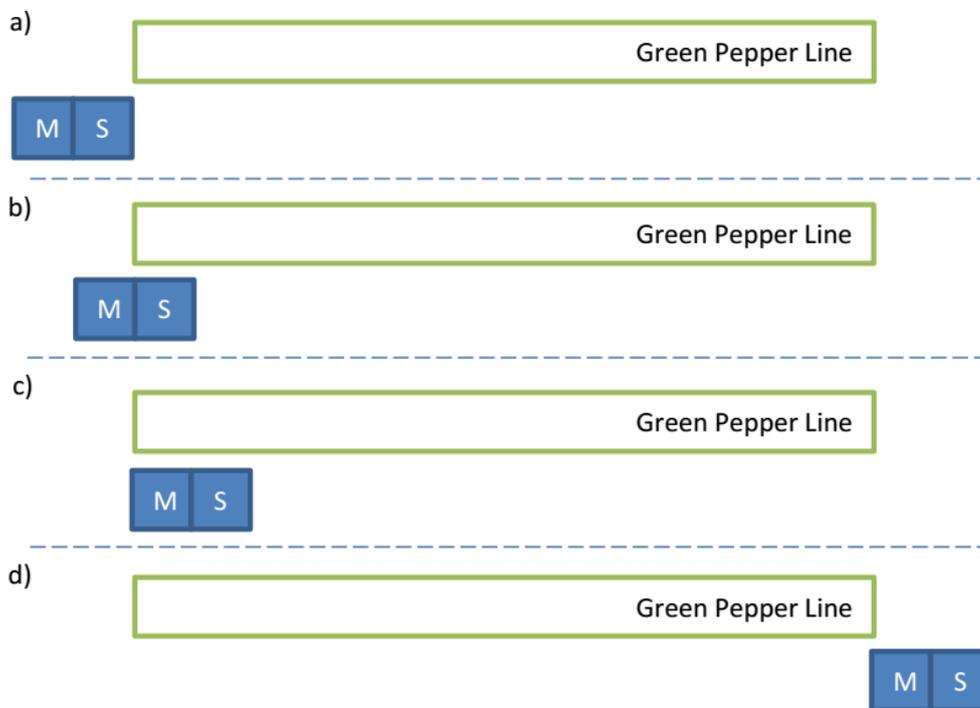


Figure 5-7 Harvesting sequence; Harvester is divided into two sections, sensory (S) and manipulator system (M).

5.4 Position Control

Almost no attention in the literature about automatic harvesting robots is dedicated to the movement of a harvester and the position control in a greenhouse. Typically it is assumed to be a task of general robotics, and researchers in the field of automatic harvesting concentrate to more specific tasks, such as target recognition and harvesting end-effector

development. However, the working environment in a greenhouse presents some characteristics that have to be taken into account when developing an automatic harvesting robot. First, the rows of plants leave limited space for navigation and operation of the robot. Plant beds are usually placed as close as possible to each other to increase the overall productivity of the greenhouse. This affects not only robot dimensions but also the navigation and operation strategy. Secondly, uneven terrain present in a greenhouse is prone to wheel slippage, which makes information of internal sensors, such as inertial sensing and rotary encoders, unreliable for use in navigation. As a result, navigation by using predefined mapping information alone is unsafe in an environment, where even an error of few centimeters could mean a collision. To overcome these issues constant monitoring of surroundings from supervision cameras and proximity sensors are used alone or in combination with a predefined map [11, 79].

In our monorail harvesting robot design, many of these issues are solved. First of all, the used rail provides a fixed pathway, from which no deviation in the lateral direction is possible. The method of using rails for the guidance of the robot is not new and has been already used for ground-based harvesting robots, such as cucumber harvesting robot designed by Van Henten et al. [9]. In our design guidance rods are used to maintain accurate lateral position. Maximum possible lateral error for the current setup can be $\pm 1\text{mm}$ as the groove width of the rail is 10mm but the diameter of the guidance rods is 8mm. Second, the main body of the harvester is positioned above the top plant level, thus, only the recognition and picking systems are penetrating the space between the rows of plants. Both of these systems require relatively minimal space compared to the size of a full robot body so the rows of plants can be put as close as possible to allow the access of sunlight but not worry about extra space for the movement of a harvesting robot. Finally, using a rail gives a convenient way of navigation in the greenhouse. By printing special marks on the rail and using a barcode scanner it is possible to implement a system of navigation with a fine accuracy. Separate marks can be used to coordinate the robot by giving the row number, position in the row and the total distance from the initial point, all of this in one mark. Furthermore, printing these marks on the rail with a fixed step, e.g. with one segment distance (half of the robot length) from each other, provides the robot with stepwise positioning that doesn't suffer from accumulative error.

5.5 Power Supply

5.5.1 Power Consumption

One of the key characteristic parameters of any electrical system is power consumption. It is even more crucial for autonomous systems that are required to be powered by a battery. The power consumption must be as low as possible to increase the active working time and to decrease the need for large size batteries that would make the system bulky. The working time of the designed harvesting robot is during the dark period of the day, which is typically from 20:00 to 4:00 in the summer time, giving approximately 8h long working cycle. All major electric parts and their respective power consumption are listed in [table 5-1](#).

Table 5-1 Power consumption of the main electronic parts of the harvesting robot prototype.

Part	Consumption per unit, W	Amount of units	Total consumption, W
PC	9	1	9
Arduino UNO	0.25	2	0.5
Arduino DUE	1.4	1	1.4
Slider Motor	6.1	1	6.1
Manip. motor	4	12	48
Railway motor	5	1	5
Cooler Fan	2.9	2	5.8
Laser Range Finder	2.5	1	2.5
Signal Generator	0.8	1	0.8
Motor Controller	1	3	3
Camera	1	1	1
Total:			83.1

As it can be seen, the total calculated power consumption of the system in case when all actuators are working is **83.1W**. To compensate for power loss in voltage regulators and the extra current draw for peak current working conditions, the power consumption is rounded up to **100W**. As the working voltage of the harvester is 12V, the calculated current draw from the battery was calculated as follows:

$$I = \frac{P}{V} = \frac{100W}{12V} \approx 8.33A \quad (5-1)$$

where

- I = current draw from the battery, [A]
- P = power consumption of the system, [W]
- V = working voltage of the system, [V]

The system is using four 15000mAh batteries, which gives 7.2h working cycle at 100W power consumption and 8.66h working time when the power consumption is 83.1W. The average expected system working time is approximately 8h.

5.5.2 Charging

The charging of the harvesting robot will be performed by an *inductive power supply*. The inductive power supply is a wireless power transfer system that uses the electromagnetic field for energy transfer. As a result, no power cables are necessary and fully autonomous operation can be achieved. Furthermore, the transmitter side of the inductive power supply is powered by a battery, which in turn is charged from a solar panel as the main power source. Optionally, the system can be plugged also to the AC power grid to be used when the solar power is insufficient to charge the main battery, for example, during a prolonged rainy period. According to the statistical information, Kochi prefecture has approximately 226 sunny days per year (average from 1981 – 2010) [80]. On top of that, the average solar power in Japan is 4.3 – 4.8kWh/m² per day [81], therefore solar power can be used as the main power source for a greenhouse power supply as our robot requires less than 1kWh per day (approx. 720Wh + losses in the charging system). Full functional schematics of the charging system can be seen in *fig. 5-8*. First, the main battery is charged from a solar panel or, in the case of low output from the solar panel, from the main AC power grid. A microcontroller circuit controls when to turn on the AC power supply and when to turn on the transmitter circuit. The transmitter circuit is turned on only when the robot is in the home position. Transmitter circuit consists of an LC oscillator and a flat coil, which is built in the homing position of the robot. The receiving coil is built on the side of the robot. Both sides of the robot can be used for increased efficiency.

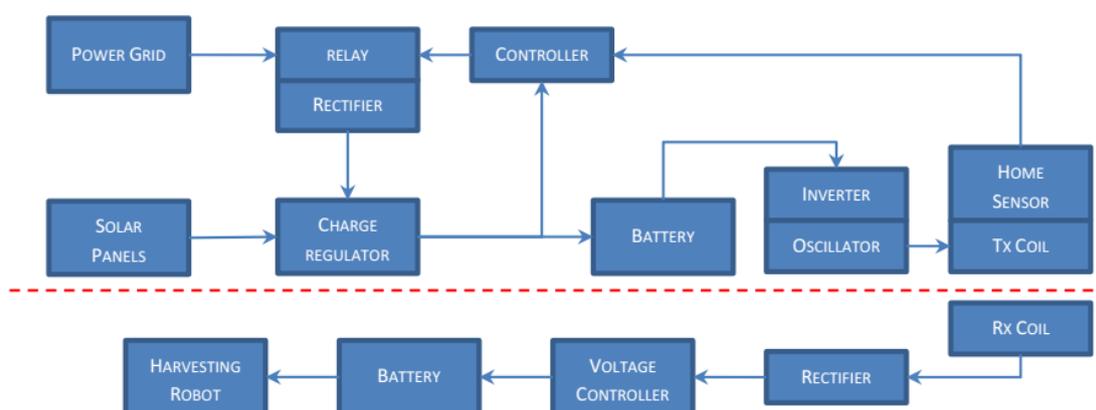


Figure 5-8 Functional schematic of the charging system; Top part is the charging station side while the bottom part is the harvester side.

For the charging system to work sufficiently, all four batteries must be charged during the idle period of the robot. Let's assume that the idle period is 15h. During this time, approximately 4 x 15 000mAh of energy must be delivered to the batteries. To charge these batteries with a constant voltage mode 12V voltage must be delivered. Losses must be taken into account when calculating the required power from the receiver. The used voltage stabilizer, which is connected to the battery, requires minimum starting voltage from 8V and has an efficiency of over 90% according to the manufacturer. For simplicity of calculation, the used efficiency of the stabilizer was rounded to 90%. The output from the inductive power supply is in AC form. A full wave rectifier is used to convert the voltage from the receiving coil to DC voltage. The necessary AC voltage for the rectifier input can be calculated from equation:

$$V_{DC} = \frac{2V_{AC}}{\pi} \quad (5-2)$$

$$V_{AC} = \frac{V_{DC} \cdot \pi}{2} = \frac{8.9 \cdot 3.14}{2} \approx 14V \quad (5-3)$$

To deliver a sufficient current to the batteries, in total 4A current must be transmitted. Therefore, the total required power of the inductive power supply is

$$P = I \cdot V = 4 \cdot 14 \approx 56W \quad (5-4)$$

The harvesting robot design has a free space on the sides with a total area of 4x 270x140mm. That means that each of four coils should receive 14V at 1A current. The required coil resistance according to the Ohms law is 14Ω, which is hard to achieve by using a copper wire. Therefore, a *Kanthal™* wire is used to decrease the coil size and increase the resistance. Kanthal is an iron-chromium-aluminium (FeCrAl) alloy, which is used for resistance and high-temperature applications. It has resistivity of $1.45 \frac{\Omega mm^2}{m}$ while the resistivity of copper, for example, is $0.017 \frac{\Omega mm^2}{m}$. The use of kanthal allows producing a coil of necessary resistance for our power requirement. The parameters of the chosen receiver coil are as shown in [table 5-2](#). By using AWG17 (1.15mm diameter) size kanthal wire the necessary wire length for a single coil is approximately 10.1m. The inductance of a flat coil can be calculated from equation:

$$L = \frac{N^2 \cdot A^2}{30A - 11D_i} \quad (5-5)$$

where

N = number of turns

D_i = inner diameter
 A = coefficient, which is calculated as follows:

$$A = \frac{D_i + N(W+S)}{2} \quad (5-6)$$

where

W = wire diameter
 S = turn spacing, gaps between the wires.

By using the equations 5-5 and 5-6 the calculated inductance of one receiving coil is 129.43 μ H. Inductive power supplies usually have power losses in the range of 20 – 40%. Therefore, the voltage in the transmitting coil should be up to 40% higher. Assuming that the system is tightly coupled, i.e. misalignments and the distance between the coils are minimal; the voltage gain from the primary coil to the secondary can be approximated from equation:

$$\frac{V_2}{V_1} \propto k \sqrt{\frac{L_2}{L_1}} \quad (5-7)$$

where

V_1 = voltage in the transmitter coil
 V_2 = voltage in the receiver coil
 k = magnetic coupling factor
 L_1 = inductance of the transmitter coil
 L_2 = inductance of the receiver coil

The magnetic coupling factor is a coefficient from 0 to 1 where 1 represents perfectly coupled coils so that all the flux generated by the transmitter coil penetrates the receiver coil. In reality, this factor is lowered by losses due to leakage and position misalignment. It can be calculated as

$$k = \frac{M}{\sqrt{L_1 L_2}} \quad (5-8)$$

where

M = mutual inductance of the coils

The coupling coefficient k for transformers is from 0.95 to 0.99, but for inductive power supplies, it is in the range from 0.2 to 0.7, where the lowest end is for loosely coupled systems. The calculation of the mutual inductance at this point is impossible, as parameters of the

second coil are unknown. Assuming that we can position our robot in the charging position with a high accuracy, for the sake of calculation, the magnetic coupling factor k was chosen 0.6. The voltage in the transmitting coil was chosen to be 40% higher than that of the receiving coil, which is 20V. Using the equation 5-7 the required conductance for the second coil is approximated:

$$L_2 \propto \frac{L_1 V_2^2}{k^2 V_1^2} \approx 733.73 \mu H \quad (5-9)$$

To achieve such inductance the transmitting coil parameters were calculated and are shown in table 5-2. According to the Ohm's law, the current in such coil for voltage of 20V will be 0.37A, and the power will be approx. 7.41W. Using the tuning of resonant frequencies, far more power is in the oscillating field than is being delivered to the transmitting coil.

Table 5-2 Parameters of the transfer (Tx) and receiving (Rx) coils of the inductive power supply.

Parameter	Tx Coil	Rx Coil
Coil Type	Single row, multi-layer	
Wire Material	Kanthal	
Wire Size	AWG16, 1.3mm	AWG17, 1.15mm
Wire Length	38.735m	10.1m
Nr. of Turns	90	39
Inner Diameter	20mm	40mm
Outer Diameter	254mm	130mm
Resistance	54Ω	14.1Ω
Inductance	815.45μH	129.43μH

The transmitted power of an inductive power transfer system is proportional to the operating frequency and the flux linkage

$$P_T \propto (f_0 \cdot L \cdot \hat{I})^2 \propto (f_0 \cdot \hat{B})^2 \quad (5-10)$$

This equation shows that the transmitted power can be increased by either increasing the operating frequency of the flux density in the air gap. The flux density can be increased by a proper shielding. The operating frequency of the system f_0 can be tuned by the capacitance C , which is connected to the LC circuit according to the equation:

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \quad (5-11)$$

The operating frequency and, consequently, the capacitor value for the LC circuit should be chosen by fine-tuning the system performance. The goal of this calculation, however, was to show that the inductive power transfer can deliver the necessary power for charging the batteries of the harvesting robot. A design and optimization of such power supply system is a separate study on its own.

5.6 Manipulator Considerations

A snake-arm robot, more commonly referred to as continuum robot or manipulator can be defined as an infinite-degree-of-freedom manipulator with an elastic structure [82]. The hyper-redundancy of degrees of freedom has been the key motivation for researchers worldwide for the past two decades to work on developing many different hardware and control solutions. One of the main advantages of continuum manipulators over the conventional rigid link robotic arms is the ability to form complex shapes, which is a very important characteristic for an obstacle-free trajectory planning. The most typical application for continuum manipulators is working in dangerous conditions or complex, congested environments, such as searching for survivors in collapsed building or inspection of power plant reactors (*fig. 5-9*). On the other hand, the additional degrees of freedom introduce great complexity to the control of the manipulator. Many different hardware and control methods have been introduced by researchers, starting from attempts to modify the already existing rigid link manipulator control methods [83] to many different geometrical [84] or analytical [85] approaches.



Figure 5-9 Robot with a continuum manipulator on the left, and during nuclear site examination test on the right [OC Robotics].

5.6.1 Kinematics

The pose of each point in space of conventional rigid link manipulators can be derived from the link lengths and joint angles. In the case of continuum manipulators, the elasticity of the used materials must be considered together with the forces applied on the manipulator both from its own actuators and from the external environment. The general mathematical models rarely take this into consideration in the closed form control. The most typical approach to the manipulator control is from a geometrical point of view, by assuming that the manipulator is a constant stiffness rod, which bends in a predictable manner. This assumption allows applying Euler-Bernoulli beam theory to calculate the curvature of the beam for an applied force. The bending can be described by a finite set of arc parameters, which are used as the control variables by converting them analytical frame transformations. By assuming a piecewise constant curvature, kinematics are decomposed in two mappings (*fig. 5-10*). The first mapping describes the conversion from the joint or actuator space q to the arc parameters of curvature (k , ϕ , and l). The second mapping defines the task space x by using these arc parameters. Task space describes the required pose in space of the end tip and the shape of the manipulator. The conversion from the actuator parameters to the arc variables is a robot specific task, which depends on characteristics of the used actuation system. The conversion between arc parameters and task space is robot independent task and can be applied to all systems that can be approximated as piecewise constant curvature arcs. The actuator space depends on the particular arm design and it can be described by tendon lengths or pneumatic pressure.

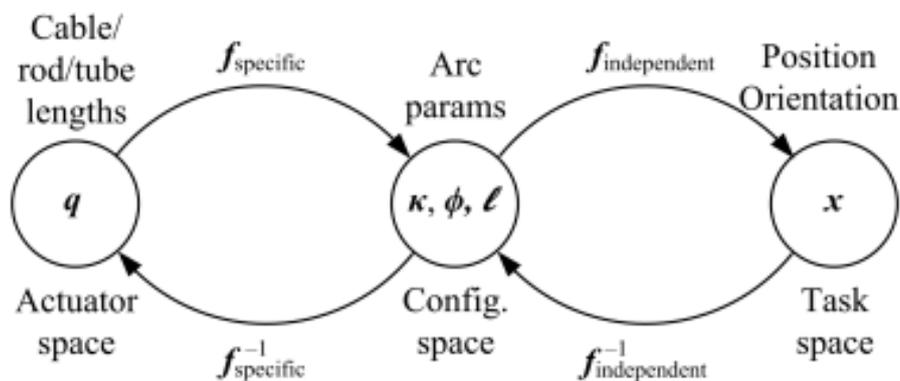


Figure 5-10 Mapping between the actuator space, arc parameters, and the task space [82]

The arc is described by three variables – k , ϕ , and l , where k is the curvature of the arc, ϕ is the rotation angle of the manipulator out of the default working plane x-z, and l is the length of

the arc (fig. 5-11). This nomenclature is accepted in the general literature on the topic and is used by most control methods.

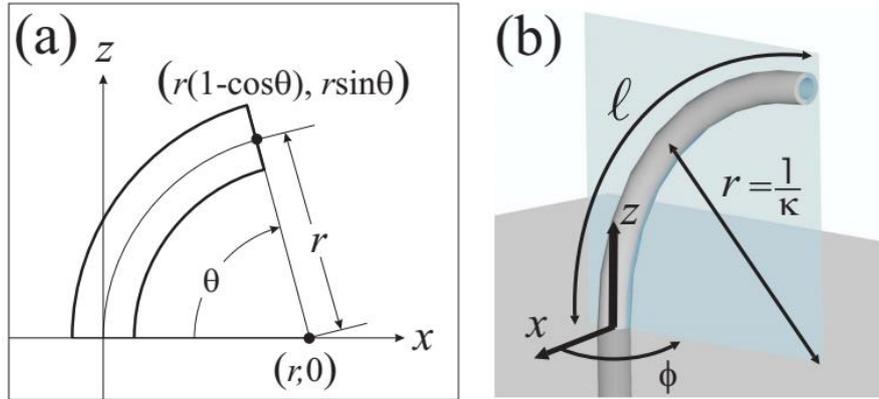


Figure 5-11 Arc parameters for one segment [82].

Forward kinematics is straightforward. The end point pose for a single segment continuum manipulator can be calculated from the following equation

$$f_{indep.} = \begin{bmatrix} \cos \phi \cos ks & -\sin \phi & \cos \phi \sin ks & \frac{\cos \phi(1-\cos ks)}{k} \\ \sin \phi \cos ks & \cos \phi & \sin \phi \sin ks & \frac{\sin \phi(1-\cos ks)}{k} \\ -\sin ks & 0 & \cos ks & \frac{\sin ks}{k} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-12)$$

where

s = point on the arc

$$s \in [0 \ l] \quad (5-13)$$

This equation can be applied to the multi-segment case by calculating transformation for each segment and then multiplying the resulting matrices. Despite having many different methods introduced by researchers, it has been proven in literature that all these methods arrive at the same result when constant piecewise curvature is assumed [82]. Inverse kinematics, on the other hand, is a challenging task in case of manipulators with a redundant degree of freedom. The general practice is to start with the conversion from the task space to the arc parameters required to achieve a certain end-effector orientation in space. Several methods have been developed to deal with the seemingly infinite number of possible orientations of the manipulator [83 – 86]. The conversion from the arc parameters to the actuator control

variables depend on the manipulator design. Most of these methods, however, ignore the real world effects on the manipulator, such as gravitation force and friction, which can significantly affect the accuracy of positioning. Mass of the manipulator and payload introduces sagging and the manipulator does not behave as a constant curvature arc anymore.

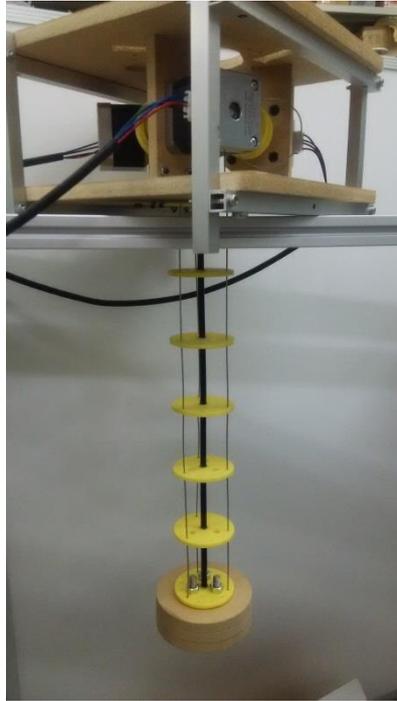


Figure 5-12 Single-segment prototype of the continuum manipulator developed by Takaaki Tokunaga.

5.6.2 Development

A hardware and control algorithm for the manipulator of the monorail harvesting robot is under development by Takaaki Tokunaga (徳永貴昭, Department of Intelligent Mechanical Systems Engineering, Kochi University of Technology) according to the proposed manipulator design. The manipulator consists of an elastic backbone, several spacer disks, and actuating tendons. The backbone is a plastic coated steel spiral, which is elastic enough to bend easily but regains the straight shape as soon as the bending force is released. An additional benefit for such backbone is that the center of the spiral can be used for running wires to the electronics mounted on the end-effector. The spacer disks are made of plastic (50mm diameter and 5mm thickness) and are rigidly attached to the backbone equidistant from each other, 6 disks per segment. The disks are used to guide tendons along the

manipulator and to ensure a correct bending shape. Each segment has three driving tendons that are positioned equidistant from the central backbone and each other. The tendons are actuated by using stepper motors, which are positioned at the base of the manipulator (fig. 5-12). Preliminary results of the study by Takaaki Tokunaga suggest that this configuration of manipulator is not well suited for working with a load and might suffer from positioning inaccuracies due to its own weight. More research must be done to improve the manipulator prototype and implement it in the harvesting robot.

5.7 Configuration Analysis

Various parts of the developed harvesting robot design were analyzed to verify the suitability for the intended application. A proper analysis of critical parts is crucial for early design fault detection and prevention.

5.7.1 Structural Analysis

The structural analysis must be performed for parts that are required to be durable to ensure the safety of the robot. In the current design, only four brackets are holding the whole weight of the robot. Monorail harvesting robot will be positioned approx. 2m above the ground so falling due to a broken holding part could mean a serious damage to the electronics and sensory system. FEM static load analysis was performed to calculate if the chosen part dimensions are sufficient to withstand the mass of the harvesting robot. 100N force, which is more than twice the actual planned load, was applied on a bracket in the same manner as it would be in the actual robot. The choice to apply such load was done considering the extra load that will come from the force with which the moving roller is pushing on the rail from the bottom.

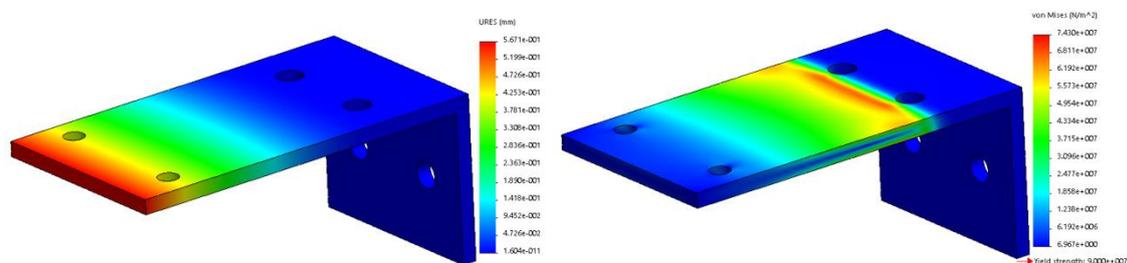


Figure 5-13 FEM Structural analysis test results for an aluminum bracket with 3mm thickness, deformation on the left and stress on the right.

The performed FEM analysis showed that with such load the maximum stress (155.2MPa) is greater than the yield strength (90MPa) if the thickness of the wheel bracket is 2mm, resulting in 1.83mm deformation. If the thickness of the bracket is increased to 3mm, the maximum stress is 75.75MPa, which is lower than the yield strength, resulting in plastic deformation of just 0.57mm. These results suggest that the thickness of the wheel brackets should be 3mm. Other possible option would be to change the material from aluminum to stainless steel, as even 2mm thick stainless steel bracket would have sufficient durability for the given task as the maximum calculated stress from FEM was 154.2MPa while the yield strength of AISI 316 stainless steel is 172.4MPa.

5.7.2 Harvesting Area

The harvesting area is a very important characteristic for an automatic harvesting robot. It is defined by the area that is within physical reach of the used manipulator and the size of this area should be sufficient to include all of the possible fruit growing positions. Sweet pepper in the greenhouse is grown in a hanging type setup by attaching the plants to strings in a “V” shaped pattern. The height of plants can reach up to two meters so a relatively large harvesting area has to be covered by the manipulator. The length of a continuum manipulator depends on the length of each individual segment and the total number of segments. Making a single segment too long will negatively affect the maneuverability and path-conforming of the manipulator as it will increase the length of each individual arc. The increase in the number of segments allows for making more complex shapes but also greatly increases the complexity of control and the total number of actuators needed. The chosen segment length and count in this study were 300mm and 4 segments respectively. This gives a manipulator with the total length of 1200mm and requires 12 motors to actuate. According to the greenhouse specification presented in *fig. 5-14*, the required harvesting manipulator distance in a worst case scenario is approximately 1300mm. To increase the harvesting area without increasing the number of segments or segment length, it was decided to use an extra fixed segment at the base of the manipulator. The chosen fixed segment length was 250mm, but in theory, this length could be increased further up to approx. 600m, increasing it further would result in possible contact between the end-effector and ground. This modification increased the harvesting area by approx. 11% (*fig. 5-15*).

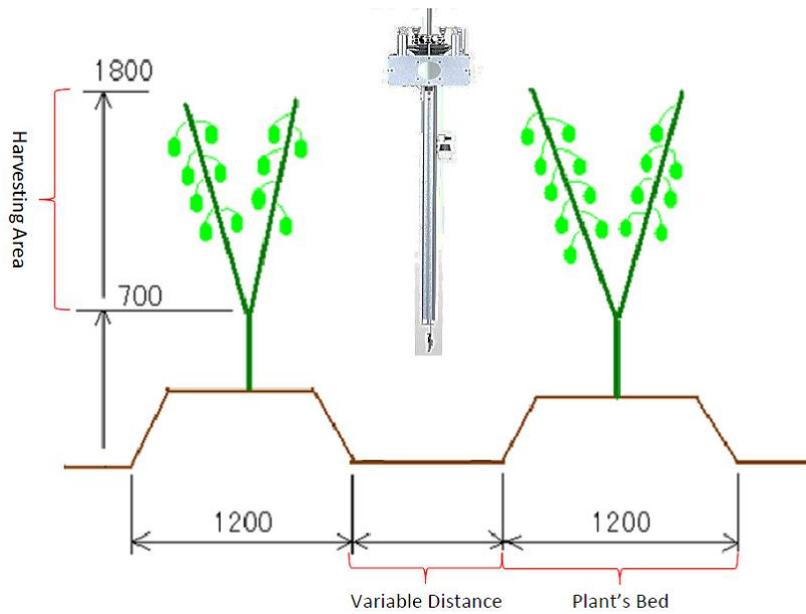


Figure 5-14 Greenhouse setup with the position for the new harvesting robot.

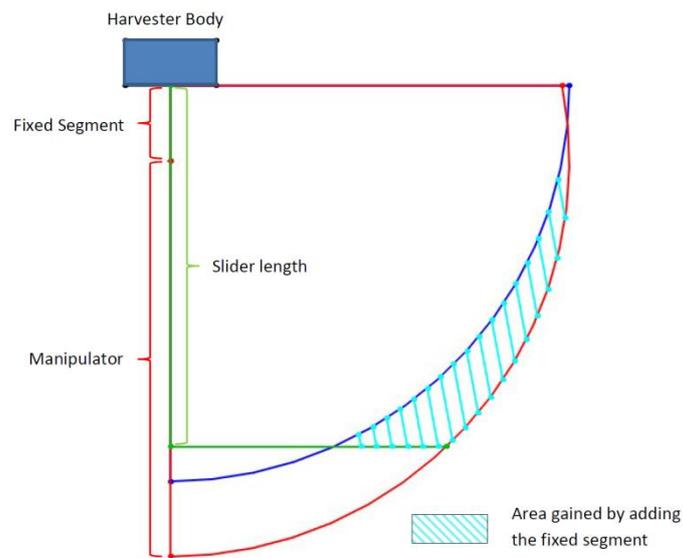


Figure 5-15 Increase in the harvesting area caused by the fixed segment.

5.7.3 Turning

According to the design, several parts of the movement system can be rearranged to change the radius of the arc that can be used for turning. The current setup allows for turning with a radius of 695mm (fig. 5-16), which was chosen to decrease the width of pathway necessary between plant rows from 400mm to 200mm. Further decrease of the pathway

width is possible but it would significantly decrease the amount of sunlight available to the lower parts of plants.

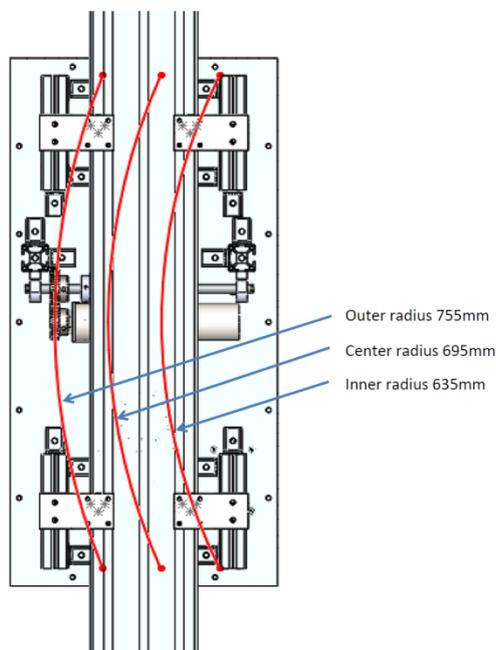


Figure 5-16 Rotation radius of the harvester the current configuration.

5.8 Cost Estimation

Cost is an important characteristic of an automatic harvesting robot as it directly affects the cost of harvested fruits. An automatic harvesting robot should cost as much as or less than what would cost to hire a human worker with the same performance for the whole life cycle of the robot. The life cycle of typical commercial industrial robots is around 10 years, but in this case, harsh working conditions (heat and humidity) and amount of electronics involved increase the breakdown probability and depreciation. The life cycle of this robot is assumed to be approximately 5 years. Consequently, the annual amortization costs are 20% of the initial robot net value.

A simplified list of the costs for parts of the new harvesting robot prototype is provided in the [table 5-3](#). Note that this is an approximate cost calculation, as few of the used parts were custom-made by hand. Also, exact price for some motors was not available as they were second-handed. As it can be seen, the total cost of the involved parts is 460 517 yen (around 4000€). When comparing the electronics versus the body parts, the cost is 255 768 yen and 204 749 yen for electronics and the body parts respectively. The main reason for high

body parts cost was several relatively expensive custom-made parts that were responsible for 74.31% of the total body parts cost, while the rest was for standard components. As for electronics, the most expensive part was the laser range finder, which made up for 58.65% of the total cost of electronics. Many small parts, such as screws and bolts, were not included in this calculation. With the assumed life cycle of 5 years, the annual amortization costs would be approximately 92 103 yen (20%). Unfortunately, it is impossible to compare the cost of this harvesting robot prototype with that of other automatic harvesting robot projects as researchers, in general, don't provide such information.

Table 5-3 General cost of the new harvesting robot prototype; value in parenthesis is the percentage of the total cost of the particular subsection.

Section	Cost, ¥	% of the Total
Robot	460 517	100%
Electronics (total)	255 768	55.54% (100%)
Main Control	34 340	7.46% (13.43%)
Manipulator	48 650	10.56% (19.02%)
Recognition	159 365	34.61% (62.31%)
Stem Detection	13 413	2.91% (5.24%)
Body Parts (total)	204 749	44.46% (100%)
Standard	52 599	11.42% (25.69%)
Custom-made	152 150	33.04% (74.31%)
Railway (1m)	5857	

5.9 Conclusions

A novel monorail based hanging type sweet pepper automatic harvesting robot prototype has been developed. The new design was analyzed for feasibility for the working conditions and the set requirements. The new prototype has following improvements over the old harvesting robot prototype:

- ✓ *Positioning accuracy* – the monorail configuration allows higher positioning accuracy due to almost complete elimination of the lateral error and due to providing a stable base for movement. The rail is also planned to be used for longitudinal positioning through special markings on the rail for the robot to use as orienteers.
- ✓ *Pose estimation* – the new design has all the hardware necessary for implementing the developed pose estimation algorithm.
- ✓ *Touch sensing* – a special piezo-based touch sensor was designed to be used with this system by attaching it to the end effector. The sensor was designed for detection of a contact with a sweet pepper stem and is used in combination with the pose estimation algorithm.
- ✓ *Field of view* – introduction of the linear slider allows eliminating the blind zone of cameras for target detection. This is particularly important for sweet pepper as the height of plants and close proximity between plants and the camera create a significant blind zone for the camera, where fruits can't be detected.
- ✓ *Optimization* – all electronics and hardware were chosen considering their particular size and power consumption.
- ✓ *Power supply* – batteries were chosen according to the calculated power consumption and the required working time. Moreover, the selected charging method with a non-contact power supply eliminates the need for a person to connect the robot to a power supply, making a completely autonomous operation possible.
- ✓ *Obstacle-free harvesting* – use of continuum manipulator allows implementing of obstacle-free manipulator trajectory planning. The 3D information from the laser range finder will be used for obstacle and occlusion detection around the target fruit.

The introduced design, however, currently has two drawbacks. First, modification of the greenhouse by implementing a railway through the entire greenhouse is required. This also includes building the homing position, where the robot will be charged, with all involved

electronics. Such modifications can be expensive, depending on the size of the greenhouse. On the other hand, any greenhouse, that would be a subject for implementation of automatic harvesting, would require certain modifications, depending on requirements of the particular robot. Secondly, in the current design a special box is missing for the harvested fruits. It is necessary to design a container with a level sensing and detachability options to be used by the robot.

6. Summary

The aim of this research was to investigate novel methods to be used for automatic harvesting of Japanese sweet pepper. Several problems were recognized when analyzing growing conditions in a greenhouse of the sweet pepper. First, the dense foliage greatly complicates automatic harvesting. Occlusions of fruits by other fruits or leaves are very common. Many fruits grow deep in the canopy and can't be easily harvested without the information of positions of all surrounding objects in space for obstacle-free manipulator path planning. Furthermore, the manipulator used for the harvesting robot must be able to conform to a complex trajectory to avoid touching the foliage and possibly cause damage to a plant in the case when the end-effector or any part of the manipulator gets tangled in the canopy.

Secondly, it is not uncommon for fruits of sweet pepper to grow slantwise due to the nearby objects and the thick stem. As a result, it can't be automatically assumed that the stem of a fruit will be positioned on strictly on the top of the fruit. Information about the pose of the fruit in space is required to correctly estimate the stem location as it often can't be detected by the visual recognition. Typically stereo vision would be used for calculation of a disparity map of the scene to obtain the depth information. In this case, however, the currently available real-time performance disparity map calculation algorithms fail to provide an accurate disparity map for such complex scenery as that present in a typical image acquired in the greenhouse. Moreover, most of the disparity map algorithms are unreliable when working with low color variation images such as foliage and fruits of sweet pepper. Consequently, it was concluded that novel methods are required to deal with the mentioned issues.

A novel sweet pepper pose estimation and stem location calculation algorithm was developed during this research. This algorithm deals with the slantwise growth and depth information issues. A laser range finder is used to acquire point cloud of surface points of a fruit and its surroundings. Afterward, a point set registration algorithm is used to match a model to the measured surface points of the fruit and determine its position in space. It is possible to calculate an approximate location of the stem from the fruit pose information and information about the size of the fruit. Additionally, the hardware used for the pose estimation algorithm, namely, the vertical slider, also deals with one additional problem that was recognized – limited field of vision. Due to the height of sweet peppers a single robot frame mounted camera is unable to observe the entire sweet pepper plant at the short distance that

is provided between the camera and a plant under greenhouse conditions. This issue was solved by using a mobile camera instead of a static one by mounting it on the vertical slider and acquiring a series of images instead of a single image at a fixed height.

The performance of the developed pose estimation algorithm was evaluated both under laboratory and greenhouse conditions. The laboratory tests showed that the method can calculate the stem position with an error less than 25mm in 91.6% of cases and less than 15mm 61.2% of cases for inclination angles up to 30°. The average error for fruits with 30° inclination angle was 15mm. The greenhouse test confirmed the feasibility of the developed method for actual application in a greenhouse as a very good and good result was achieved in 75.7% of all cases.

The main recognized issue preventing a better result from the pose estimation method was the quality of the laser range finder measurement. The used laser range finder has an accuracy of $\pm 10\text{mm}$, which in the worst case scenario can introduce up to 20mm big error. It also can have a heavy influence on the result of the segmentation which directly affects the quality of the point cloud used for pose estimation. The second source of error can be the model used for point set matching. It is very hard to create a good generic model for the model matching, which is the main reason for using the affine transformation as it can modify the shape of the model.

It was recognized that the stem position information obtained from the pose estimation algorithm is a calculation, which must be verified by an actual detection. Therefore, a piezo touch sensor was developed to verify if the stem of a fruit is in the cutting position. The developed sensor was designed to be able to detect a touch to a stem while remaining insensitive to leaves. The performance of the sensor was studied in details in the laboratory and verified on a field study in a greenhouse. The results of laboratory tests showed very good performance and reliability of the sensor while the field testing confirmed that the sensor is capable of detecting touch with a sweet pepper stem under normal working conditions.

A novel monorail automatic harvesting robot was designed to address the issues of the old design, namely, incompatibility between the hardware used by the pose estimation method and the existing automatic harvesting robot prototype, inability of the used manipulator to perform obstacle avoidance during harvesting process and other issues described in detail in the particular section. The new prototype is using a monorail to move in a greenhouse and has all the electronics and actuators hanging under the railway. Such

configuration allows for decreased space between plant rows required for the robot movement, thus increasing the productivity of the greenhouse by increasing the space available to grow crops. Other improvements were also included, such as continuum manipulator that is capable of obstacle avoidance and complex harvesting path.

It has been recognized that the developed methods of pose estimation and stem detection increase the overall harvesting time. To solve this problem it is suggested to use machine learning or fuzzy control to make a decision when to use these novel methods for the harvesting as they are necessary only for complicated cases. In a case, when the position of a stem is obvious from the visual cues, only one laser range finder measurement is required for acquiring the depth information. Selective harvesting method choice could negate the disadvantage of long data acquisition time that is present in the developed methods. As mentioned in the literature review section, machine learning is a promising technique to be used in automatic harvesting in general and provides many solutions for both recognition and path planning. Study on the introduction of machine learning and artificial intelligence in automatic harvesting robots is strongly advised.

References

- [1] N. Kondo, Y. Nishitsuji, P. P. Ling, K. C. Ting, 1996. „Visual Feedback Guided Robotic Cherry Tomato Harvesting”, in *Transactions of the ASAE*, Vol. 39(6), pp. 2331 – 2338.
- [2] N. Kondo, K. Yamamoto, K. Yata, M. Kurita, 2008. „A Machine Vision for Tomato Cluster Harvesting Robot”, in *Proceedings of the ASAE Annual International Meeting*, Providence, RI (Vol. 5, pp. 3111 – 3120).
- [3] P. P. Ling, R. Ehasani, K. C. Ting, Y. Chi, N. Ramalingam, M. H. Klingman, C. Draper, 2004. „Sensing and End-Effector for a Robotic Tomato Harvester”, in *Proceedings of the ASAE Annual International Meeting*, Ottawa, ON.
- [4] S. Hayashi, O. Sakaue, 1996. „Tomato Harvesting by Robotic System”, in *Proceedings of the ASAE Annual International Meeting*, Phoenix, AZ.
- [5] J. Hemming, W. Bac, B. Van Tuijl, R. Barth, J. Bontsema, E. Pekkeriet, E. Van Henten, 2014. „A Robot for Harvesting Sweet-Pepper in Greenhouses”, in *Proceedings of the International Conference of Agricultural Engineering*, Zürich, Switzerland 6 – 10 July 2014.
- [6] S. Kitamura, K. Oka, 2005. „Recognition and Cutting System of Sweet Pepper for Picking Robot in Greenhouse Horticulture”, in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Niagara Falls, ON (Vol. 4, pp. 1807 – 1812).
- [7] S. Arima, N. Kondo, 1999. „Cucumber Harvesting Robot and Plant Training System”, *Journal of Robotics and Mechatronics*, Vol. 11(3), pp. 208 – 212.
- [8] X. Tang, T. Zhang, L. Liu, D. Xiao, Y. Chen, 2009. „A New Robot System for Harvesting Cucumber”, in *Proceedings of the American Society of Agricultural and Biological Engineers Annual International Meeting*, Reno, NV (Vol. 6, pp. 3873 – 3885).

- [9] E. Van Henten, J. Hemming, B. Van Tuijl, J. Kornet, J. Meuleman, J. Bontsema, E. Van Os, 2002. „An Autonomous Robot for Harvesting Cucumbers in Greenhouses”, *Autonomous Robots*, Vol. 13(3), pp. 241 – 258.
- [10] K.-S. Han, S.-C. Kim, Y.-B. Lee, S.-C. Kim, D.-H. Im, H.-K. Choi, H. Hwang, 2012. „Strawberry Harvesting Robot for Bench-Type Cultivation”, *Journal of Biosystems Engineering*, Vol. 37(1), pp. 65 – 74.
- [11] Q. Feng, W. Zheng, Q. Qiu, K. Jiang, R. Guo, 2012. „Study on Strawberry Robotic Harvesting System”, in *Proceedings of the Computer Science and Automation Engineering (CSAE)*. (Vol. 1, pp. 320 – 324).
- [12] S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, M. Kurita, 2010. „Evaluation of a Strawberry-Harvesting Robot in a Field Test”, *Biosystems Engineering*, Vol. 105(2), pp. 160 – 171.
- [13] C. Flemmer, R. Flemmer, A. Scarfe, 2009. „Robotics in Horticulture”, in *Proceedings of the Australia and New Zealand Avocado Growers Conference*, Cairns, Australia.
- [14] M. Monta, N. Kondo, Y. Shibano, 1995. „Agricultural Robot in Grape Production System”, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Japan (Vol. 3, pp. 2054 – 2509).
- [15] S. Kitamura, K. Oka, 2005. „Recognition and Cutting System of Sweet Pepper for Picking Robot in Greenhouse Horticulture”, in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Niagara Falls, ON (Vol. 4, pp. 1807 – 1812).
- [16] S. Bachche, 2013. „Automatic Harvesting for Sweet Peppers in Greenhouse Horticulture”, *Ph.D. Dissertation*, Kochi University of Technology, Kochi, Japan.
- [17] D. M. Bulanon, T. Kataoka, 2010. „A Fruit Detection System and an End-Effector for Robotic Harvesting of Fuji Apples”, *Agricultural Engineering International: the CIGR Ejournal*, Vol. 12(1), pp. 203 – 210.
- [18] S. Pongchompu, K. Tsuneo, P. Pongchompu, 2012. „Aspects of the Aging Farming Population and Food Security in Agriculture for Thailand and Japan”, *International Journal of Environmental and Rural Development*, Vol. 3(1), pp. 102 – 107.
- [19] Ministry of Agriculture, Forestry and Fisheries, „Annual Reports 1995 – 2015”, retrieved from: <http://www.maff.go.jp/e/index.html>, last accessed: 03.06.2016.

- [20] C.E. Schert, G.K. Brown, 1968. „Basic Considerations in Mechanizing Citrus Harvest”, *Trans. ASAE*, Vol. 11, pp. 343 – 346.
- [21] C. W. Bac, E. J. van Henten, J. Hemming, Y. Edan, 2014. „Harvesting Robots for High-Value Crops: State-of-the-Art Review and Challenges Ahead”, *Journal of Field Robotics*, Vol. 31(6), pp. 888 – 911, DOI: 10.1002/rob.21525.
- [22] Bachche S., 2015. „Deliberation on Design Strategies of Automatic Harvesting Systems: A Survey”, *Robotics*, Vol. 4, pp. 194 – 222, DOI: 10.3390/robotics4020194.
- [23] N. Kondo, K. Yamamoto, H. Shimizu, K. Yata, M. Kurita, T. Shiigi, M. Monta, T. Nishizu, 2009. „A Machine Vision System for Tomato Cluster Harvesting Robot”, *EAEF*, Vol. 2(2), pp. 60 – 65.
- [24] E. J. Van Henten, D. A. Van't Slot, C. W. J. Hol, L. G. Van Willigenburg, 2009. „Optimal Manipulator Design for a Cucumber Harvesting Robot”, *Computers and Electronics in Agriculture*, Vol. 65(2), pp. 247 – 257.
- [25] A. Plebe, G. Grasso, 2001. „Localization of Spherical Fruits for Robotic Harvesting”, *Machine Vision and Applications*, Vol. 13(2), pp. 70 – 79.
- [26] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, 2006. „A Comparison and Evaluation of Multiview Stereo Reconstruction Algorithms”, in *IEEE Conf. Comp. Vision and Pattern Recognition*, pp. 519 – 528.
- [27] Q. Yang, C. Engels, A. Akbarzadeh, 2008. „Near Real-Time Stereo for Weakly-Textured Scenes”, in *Proceedings of British Machine Vision Conference*, 2008.
- [28] J. Liu, P. Li, Z. Li, 2007. „A Multi-Sensory End-Effector for Spherical Fruit Harvesting Robot”, in *Proceedings of the IEEE International Conference on Automation and Logistics*, August 18 – 21, 2007, Jinan, China
- [29] H. Okamoto, W. S. Lee, 2009. „Green Citrus Detection Using Hyperspectral Imaging”, *Computers and Electronics in Agriculture*, Vol. 66, pp. 201 – 208.
- [30] T. Yuan, L. Wei, F. Qingchun, Z. Junxiong, 2010. „Spectral Imaging for Greenhouse Cucumber Fruit Detection Based on Binocular Stereovision”, in *ASABE Annual International Meeting*, June 2010.
- [31] M. Hannan, T. Burks, D. M. Bulanon, 2007. „A Real-Time Machine Vision Algorithm for Robotic Citrus Harvesting”, in *ASABE Annual International Meeting*, June 2007.

- [32] L. Bin, W. Maohua, W. Ning, 2010. „Development of a Real-Time Fruit Recognition System for Pineapple Harvesting Robots”, in *ASABE Annual International Meeting*, June 2010.
- [33] H. Jiang, Y. Peng, Y. Ying, 2008. „Measurement of 3D Locations of Ripe Tomato by Binocular Stereo Vision for Tomato Harvesting”, in *ASABE Annual International Meeting*, July 2008.
- [34] J. Wachs, H. Stern, T. Burks, V. Alchanatis, 2010. „Low and High-Level Visual Feature Based Apple Detection from Multi-Modal Images”, *Precision Agriculture*, Vol. 11(6), pp. 717 – 735.
- [35] M. Hannan, T. Burks, 2004. „Current Developments in Automated Citrus Harvesting”, in *ASABE Annual International Meeting*, August 2004.
- [36] D. M. Bulanon, T. Burks, V. Alchanatis, 2009. „Image Fusion of Visible and Thermal Images for Fruit Detection”, *Biosystems Engineering*, Vol. 103(1), pp. 12 – 22.
- [37] K. Kapach, E. Barnea, R. Mairon, Y. Edan, O. Ben-Shahar, 2012. „Computer Vision for Fruit Harvesting Robots – State of Art and Challenges Ahead”, *Int. J. Comput. Vis. Robot*, Vol. 3, pp. 4 – 34.
- [38] O. Safren, V. Alchanatis, V. Ostrovsky, O. Levi, 2007. „Detection of Green Apples in Hyperspectral Images of Apple-Tree Foliage Using Machine Vision”, in *transactions of the ASABE*, Vol. 50(6), pp. 2303 – 2313.
- [39] J. Zhao, J. Tow, J. Katupitiya, 2005. „On-Tree Fruit Recognition Using Texture Properties and Color Data”, in *IEEE International Workshop on Intelligent Robots and Systems*, August 2005.
- [40] J. Rakun, D. Stajniko, D. Zazula, 2011. „Detecting Fruits in Natural Scenes by Using Spatial-Frequency Based Texture Analysis and Multiview Geometry”, *Computers and Electronics in Agriculture*, Vol. 76(1), pp. 80 – 88.
- [41] F. Zhang, N. Zhang, 2008. „Applying Joint Transform Correlator in Tomato Recognition”, in *ASABE Annual International Meeting*, July 2008.
- [42] Y. Liu, B. Chen, J. Qiao, 2011. „Development of a Machine Vision Algorithm for Recognition of Peach Fruit in a Natural Scene”, *transactions of the ASABE*, Vol. 54(2), pp. 695 – 702.
- [43] S. Hayashi, K. Ganno, Y. Ishii, I. Tanaka, 2002. „Robotic Harvesting System for Eggplants”, *Japan Agricultural Research Quarterly*, Vol. 36(3), pp. 163 – 168.

- [44] H. Patel, R. Jain, M. Joshi, 2011. „Fruit Detection Using Improved Multiple Features Based Algorithm”, *International Journal of Computer Applications*, Vol. 13(2), pp.1 – 5.
- [45] V. S. Nalwa, 1993. „A Guided Tour of Computer Vision”, Addison-Wesley Publishing, Reading, MA, USA.
- [46] K. Kane, W. Lee, 2006. „Spectral Sensing of Different Citrus Varieties for Precision Agriculture”, in *ASABE Annual International Meeting*, July 2006.
- [47] Y. Chi, P. Ling, 2004. „Fast Fruit Identification for Robotic Tomato Picker”, in *ASABE Annual International Meeting*, August 2004.
- [48] R. Chinchuluun, W. Lee, 2006. „Citrus Yield Mapping System in Natural Outdoor Scenes Using the Watershed Transform”, in *ASABE Annual International Meeting*, July 2006.
- [49] D. M. Bulanon, T. Kataoka, 2010. „Fruit Detection System and an End-Effector for Robotic Harvesting of Fuji Apples”, *International commission of Agricultural and Biosystems Engineering Journal*, Vol. 12(1), pp. 203 – 210.
- [50] A. Jimenez, R. Ceres, J. Pons, 2000. „A Vision System Based on a Laser Range-Finder Applied to Robotic Fruit Harvesting”, *Machine Vision and Applications*, Vol. 11, No. 6, pp.321 – 329.
- [51] D. Kong, D. Zhao, Y. Zhang, J. Wang, H. Zhang, 2010. „Research of Apple Harvesting Robot Based on Least Square Support Vector Machine”, in *Proceedings of the International Conference on Electrical and Control Engineering*, June, pp. 1590 – 1593.
- [52] D. M. Bulanon, T. Kataoka, H. Okamoto, S. Hata, 2004. „Determining the 3D Location of the Apple Fruit During Harvest”, in *Proceedings of the Automation Technology for Off-Road Equipment Conference*, October 2004.
- [53] M. Regunathan, W. Lee, 2005. „Citrus Fruit Identification and Size Determination Using Machine Vision and Ultrasonic Sensors”, *ASABE Annual International Meeting*, July 2005.
- [54] J. Bouguet, 1999. „Camera Calibration Toolbox For Matlab”, available at: <http://www.vision.caltech.edu/bouguetj/index.html>, last accessed: 27.11.2015.
- [55] R. Lewis, A. Yoxall, M. Marshall, L. Canty, 2008. „Characterising Pressure and Bruising in Apple Fruit”, *Wear*, Vol. 264, pp. 37 – 46.
- [56] M. Van Zeebroeck, 2007. „The Effect of Fruit Factors on the Bruise Susceptibility of Apples”, *Postharvest Biol Technol*, Vol. 46, pp. 10 – 19.

- [57] C. Blanes, M. Mellado, C. Ortiz and A. Valera, 2011. „Review. Technologies for Robot Grippers in Pick and Place Operations for Fresh Fruits and Vegetables”, *Spanish Journal of Agricultural Research*, Vol. 9(4), pp. 1130 – 1141.
- [58] M. Monta, N. Kondo, K. Ting, 1998. „End-Effectors for Tomato Harvesting Robot”, *Artificial Intelligence Review*, Vol. 12, pp. 11 – 25.
- [59] J. Hemming, J. Bontsema, W. Bac, Y. Edan, B. van Tuijl, R. Barth, E. Pekkeriet, „Final Report: Sweet-Pepper Harvesting Robot”, *Report to the European Commission in the 7th Framework Programme*.
- [60] M. Sdahl, B. Kuhlenkoetter, 2006. „CAGD-Computer Aided Gripper Design for a Flexible Gripping System”, *Intl J Adv Robot Syst*, Vol. 2(2), pp. 135 – 138.
- [61] G. Mantriota, 2007. „Optimal Grasp of Vacuum Grippers with Multiple Suction Cups”, *Mechanism and Machine Theory*, Vol. 42, pp. 18 – 33.
- [62] S. J. Flood, 2006. „Design of a Robotic Citrus Harvesting End-Effector and Force Control Model Using Physical Properties and Harvesting Motion Tests”, *Ph.D. Dissertation*, University of Florida, Florida, USA.
- [63] T.A. Pool, R.C. Harrell, 1991. „An End-Effector for Robotic Removal of Citrus from the Tree”, *Trans. ASABE*, Vol. 34, pp. 373 – 378.
- [64] B. Jia, A. Zhu, S. X. Yang, G. S. Mittal, 2009. „Integrated Gripper and Cutter in a Mobile Robotic System for Harvesting Greenhouse Products”, *Proceedings of the 2009 IEEE International Conference on Robotics and Biomimetics*, December 19 – 23, 2009, Guilin, China.
- [65] P. Gay, P. Piccarolo, D. R. Aimonino, R. Deboli, 2008. „Robotics for Work and Environment Safety in Greenhouse”, *International Conference*, September 15 – 17, 2008, Ragusa, Italy.
- [66] T. N. Tran, K. Drab, M. Daszykowski, 2013. „Revised DBSCAN Algorithm to Cluster Data with Dense Adjacent Clusters”, *Chemometrics and Intelligent Laboratory Systems*, Vol. 120, pp. 92 – 96.
- [67] N. Ho, 2013. [Blogpost] „Finding Optimal Rotation and Translation Between Corresponding 3D Points”, retrieved from: http://nghiaho.com/?page_id=671, last accessed: 03.06.2016.

- [68] E. Doron, 1999. „CPORT Minitoolbox for Matlab”, retrieved from: <http://www.mathworks.com/matlabcentral/fileexchange/161-cport>, last accessed: 03.06.2016.
- [69] A. Myronenko, X. Song, 2010. „Point Set Registration: Coherent Point Drift”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32(12), pp. 2262 – 2275.
- [70] L. Harmon, (1984). „Tactile Sensing for Robots”, *Robotics and Artificial Intelligence*, pp. 109–158.
- [71] M. Lee, 2000. „Tactile Sensing: New Directions, New Challenges”, *The International Journal of Robotics Research*, Vol. 19(7), pp. 636.
- [72] M. I. Tiwana, S. J. Redmond, N. H. Lovell, 2012. „A Review of Tactile Sensing Technologies with Applications in Biomedical Engineering”, *Sensors and Actuators A*, Vol. 179, pp. 17 – 31.
- [73] G. Sauerbrey, *Phys. Verhandl.*, 8 (1957).
- [74] W. Pang, L. Yan, H. Zhang, H. Yu, E. S. Kim, W. C. Tang, 2006. „Femtogram Mass Sensing Platform Based on Lateral Extensional Mode Piezoelectric Resonator”, *Applied Physics Letters* Vol. 88, 243503; DOI: 10.1063/1.2213975.
- [75] A. A. S. Rabih, J. O. Dennis, M. H. Md khir, M. A. Abdullah, 2013. „Mass Detection Using a Macro-Scale Piezoelectric Bimorph Cantilever”, *Proc. of the IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 26 – 27 November 2013, Kuala Lumpur, Malaysia.
- [76] H. Han, J. Kim, 2013. „Active Muscle Stiffness Sensor Based on Piezoelectric Resonance for Muscle Contraction Estimation”, *Sensors and Actuators A*, Vol. 194, pp. 212 – 219.
- [77] National Instruments, 2015, „White Paper: Acquiring an Analog Signal: Bandwidth, Nyquist Sampling Theorem, and Aliasing”, retrieved from: <http://www.ni.com/white-paper/2709/en/#>, last accessed: 03.06.2016.
- [78] F. Kapel, 2013. [Blogpost] „Fast analogRead with Arduino Due”, retrieved from: <http://frenki.net/2013/10/fast-analogread-with-arduino-due/>, last accessed: 03.06.2016.
- [79] A. Mandow, J. M. Gomez de Gabriel, J. L. M. Rodriguez, V. F. Munoz, A. Ollero, A. Garcia-Cerezo, 1996. „The Autonomous Mobile Robot AURORA for Greenhouse Operation”, *IEEE Robotics & Automation Magazine*, pp. 18 – 28.

- [80] Japan Meteorological Agency, „Average Sunshine a Year in Japan”, retrieved from: <https://www.currentresults.com/Weather/Japan/sunshine-annual-average.php>, last accessed: 03.06.2016.
- [81] F. A. Saad, 2016. „The Shock of Energy Transition”, *Partridge Publishing*, Singapore, ISBN: 978-1-4828-6495-3.
- [82] R. J. Webster III and B. A. Jones, 2010. „Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review”, *International Journal of Robotics*, Vol. 29(13), pp. 1661 – 1683, DOI: 10.1177/0278364910368147.
- [83] M. W. Hannan and I. D. Walker, 2002. „Kinematics and the Implementation of an Elephant’s Trunk Manipulator and Other Continuum Style Robots”, *Journal of Robotic Systems*, Vol. 20(2), pp. 45 – 63, DOI: 10.002/rob.10070.
- [84] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. D. Walker, 2008. „A Geometrical Approach to Inverse Kinematics for Continuum Manipulators”, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Acropolis Convention Center Nice, France, Sept, 22 – 26.
- [85] B. He, Z. Wang, Q. Li, H. Xie and R. Shen, 2012. „An Analytic Method for the Kinematics and Dynamics of a Multiple-Backbone Continuum Robot”, *International Journal of Advanced Robotic Systems*, Vol. 10, 84:2013, pp. 1 – 13, DOI: 10.5772/54051.
- [86] S. Neppalli, M. A. Csencsits, B. A. Jones, I. D. Walker, 2009. „Closed-Form Inverse Kinematics for Continuum Manipulators”, *Advanced Robotics*, Vol. 23, pp. 2077 – 2091, DOI: 10.1163/016918609X12529299964101.

Appendices

Appendix A – Software Codes

Vertical slider control code

```
#include <math.h>
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_PWMServoDriver.h"

#define LIM      2 // limit switch pin
#define UP       BACKWARD // map slider direction with stepper direction
#define DOWN     FORWARD

// creat Adafruit motor shield object and configure the motor
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_StepperMotor *myMotor = AFMS.getStepper(200, 2);

int currentPos, targetPos; // declare variables for position
unsigned int s2t,i; // steps to take
int diff; // difference
const float dps = 0.4; // distance per step, mm

void setup()
{
  AFMS.begin(); // set up motor
  myMotor->setSpeed(30); // speed of the motor, rpm
  pinMode(LIM, INPUT); // set up pins for limit switch
  Serial.begin(115200); // open serial communication
  while(digitalRead(LIM) == 0) // move slider to home position
  {
    myMotor->step(1, UP, DOUBLE);
  }
  currentPos = 0; // initialize the step count
  Serial.write(currentPos); // send message that slider is ready
}

void loop()
{
  if(Serial.available()) // if new info
```

```

{
  targetPos = Serial.read();
  while(!Serial.available()){ // wait for new byte
  targetPos += Serial.read() << 8;
  if(targetPos == 0)
  {
    while(digitalRead(LIM) == 0) // homing
    {
      myMotor->step(1,UP,DOUBLE);
    }
    Serial.write(currentPos);
    currentPos = 0;
  }
  else
  {
    diff = currentPos - targetPos;
    diff = abs(diff);
    s2t = round(diff/dps); // calculate steps to take
    if(s2t < 25) // use microstepping feature
    {
      if(currentPos < targetPos)
      {
        for(i=0;i<s2t;i++)
          myMotor->step(1,DOWN,MICROSTEP);
      }
      if(currentPos > targetPos)
      {
        for(i=0;i<s2t;i++)
          myMotor->step(1,UP,MICROSTEP);
      }
      currentPos = targetPos;
      Serial.write(currentPos); // send feedback about position
    }
    else // use full phase movement
    {
      if(currentPos < targetPos)
      {
        for(i=0;i<s2t;i++)
          myMotor->step(1,DOWN,DOUBLE);
      }
      if(currentPos > targetPos)
      {
        for(i=0;i<s2t;i++)
          myMotor->step(1,UP,DOUBLE);
      }
      currentPos = targetPos;
      Serial.write(currentPos);
    }
  }
}
}

```

Manipulator code

```
// Servo 1 - base rotation
// Servo 2 - 1st joint
// Servo 3 - 2nd joint
// Servo 4 - 3rd joint
// Servo 5 - cutter, controlled by software
// Used board: Adafruit 16 channel PWM control board
// Angle info is received from Matlab

// to calculate the values needed you can use equations:
// pulse = (1/freq)*(end/4095)*1000
// end = (pulse*freq*4095)/1000
// puls = 1.5 will put roughly in middle, 1 - min position, 2 - max
position

// communication uses 12 bytes, start byte, 10 servo angle bytes and
command byte
// start byte is 0xAA or 170 in hex, each servo angle is divided in
two bytes, LSB first
// command byte is 0xBB for arm movement and 0xCC for cutter movement

#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <Math.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

#define servoMin 0
#define servoMid 300
#define servoMax 600

#define servo1 0
#define servo2 2
#define servo3 6
#define servo4 12
#define servo5 10

byte button1, button2, rs1t;
const int ctrlStep = 2;
uint16_t servo1pos, servo2pos, servo3pos, servo4pos, servo5pos;
uint16_t servo1posNew, servo2posNew, servo3posNew, servo4posNew;
uint16_t servo5posNew;
unsigned int s2t, firstByte, commandByte;
unsigned long int i;
int tmp1, tmp2, tmp3, tmp4, tmp5;
uint8_t buff[10] = {0};

void setup()
{
  Serial.begin(115200);
```

```

pwm.begin();
pwm.setPWMFreq(50);
delay(10);

pinMode(0, INPUT_PULLUP);
pinMode(1, INPUT_PULLUP);
pinMode(2, INPUT_PULLUP);
pinMode(3, INPUT_PULLUP);
pinMode(4, INPUT_PULLUP);
pinMode(5, INPUT_PULLUP);
pinMode(6, INPUT_PULLUP);
pinMode(7, INPUT_PULLUP);

pwm.setPWM(servo1, 0, 300);
servo1pos = 300;
delay(10);
pwm.setPWM(servo2, 0, 250);
servo2pos = 310;
delay(10);
pwm.setPWM(servo3, 0, 300);
servo3pos = 300;
delay(10);
pwm.setPWM(servo4, 0, 300);
servo4pos = 300;
delay(10);
pwm.setPWM(servo5, 0, 300);
servo5pos = 300;
delay(10);
}

void loop()
{
  serialEvent();
  button1 = PIND;
  delay(20);
  button2 = PIND;
  rslt = button1 | button2;
  if((rslt & B00000001) == 0)
  {
    servo1pos += ctrlStep;
    if(servo1pos > servoMax) servo1pos = servoMax;
  }
  if((rslt & B00000010) == 0)
  {
    servo1pos -= ctrlStep;
    if(servo1pos < servoMin) servo1pos = servoMax;
  }
  if((rslt & B00000100) == 0)
  {
    servo2pos += ctrlStep;
    if(servo2pos > servoMax) servo2pos = servoMax;
  }
}

```

```

if((rslt&B00001000) == 0)
{
    servo2pos -= ctrlStep;
    if(servo2pos < servoMin) servo2pos = servoMax;
}
if((rslt&B00010000) == 0)
{
    servo3pos += ctrlStep;
    if(servo3pos > servoMax) servo3pos = servoMax;
}
if((rslt&B00100000) == 0)
{
    servo3pos -= ctrlStep;
    if(servo3pos < servoMin) servo3pos = servoMax;
}
if((rslt&B01000000) == 0)
{
    servo4pos += ctrlStep;
    if(servo4pos > servoMax) servo4pos = servoMax;
}
if((rslt&B10000000) == 0)
{
    servo4pos -= ctrlStep;
    if(servo4pos < servoMin) servo4pos = servoMax;
}
pwm.setPWM(servo1,0,servo1pos);
pwm.setPWM(servo2,0,servo2pos);
pwm.setPWM(servo3,0,servo3pos);
pwm.setPWM(servo4,0,servo4pos);
}

```

```

void serialEvent()
{
    if(Serial.available() > 11)
    {
        firstByte = Serial.read();
        if(firstByte == 170)
        {
            buff[0] = Serial.read();
            buff[1] = Serial.read();
            buff[2] = Serial.read();
            buff[3] = Serial.read();
            buff[4] = Serial.read();
            buff[5] = Serial.read();
            buff[6] = Serial.read();
            buff[7] = Serial.read();
            buff[8] = Serial.read();
            buff[9] = Serial.read();
            commandByte = Serial.read();

            switch(commandByte)
            {
                case 187:

```

```

servo1posNew = ((uint16_t)buff[0]) | (((uint16_t)buff[1])<<8);
servo2posNew = ((uint16_t)buff[2]) | (((uint16_t)buff[3])<<8);
servo3posNew = ((uint16_t)buff[4]) | (((uint16_t)buff[5])<<8);
servo4posNew = ((uint16_t)buff[6]) | (((uint16_t)buff[7])<<8);

tmp1 = servo1posNew - servo1pos;
tmp1 = abs(tmp1);
tmp2 = servo2posNew - servo2pos;
tmp2 = abs(tmp2);
tmp3 = servo3posNew - servo3pos;
tmp3 = abs(tmp3);
tmp4 = servo4posNew - servo4pos;
tmp4 = abs(tmp4);

s2t = max(tmp1,tmp2);
s2t = max(s2t,tmp3);
s2t = max(s2t,tmp4);

for(i=0;i<s2t+1;i++)
{
    if(servo1pos != servo1posNew)
    {
        if(servo1pos < servo1posNew)
        {
            servo1pos++;
        }
        else
        {
            servo1pos--;
        }
        pwm.setPWM(servo1,0,servo1pos);
        delay(5);
    }
    if(servo2pos != servo2posNew)
    {
        if(servo2pos < servo2posNew)
        {
            servo2pos++;
        }
        else
        {
            servo2pos--;
        }
        pwm.setPWM(servo2,0,servo2pos);
        delay(5);
    }
    if(servo3pos != servo3posNew)
    {
        if(servo3pos < servo3posNew)
        {
            servo3pos++;
        }
        else

```


Piezo Control Code

```
#undef  HID_ENABLED
#define W_CLK      8 // Pin 8 - connect to AD9850 module (CLK)
#define FQ_UD     9 // Pin 9 - connect to freq update pin (FQ)
#define DATA     10 // Pin 10 - connect to AD9850 module (DATA)
#define RESET     11 // Pin 11 - connect to reset pin (RST).
#define ADC_CTRL  32 // Turn on gate to let signal get to A0 pin
#define INP_BUFF  512
#define OUT_BUFF  1024 // must be 2x INP_BUFF
#define pulseHigh(pin) {digitalWriteDirect(pin,HIGH);
digitalWriteDirect(pin,LOW);}

inline void digitalWriteDirect(int pin, boolean val){
    if(val) g_APinDescription[pin].pPort -> PIO_SODR =
g_APinDescription[pin].ulPin;
    else g_APinDescription[pin].pPort -> PIO_CODR =
g_APinDescription[pin].ulPin;
}

volatile int16_t      flag = 0;
uint8_t              res_f[4] = {0}, rec_buf[7] = {0};
uint16_t             adcIn[INP_BUFF] = {0};
uint32_t             resonant_freq, freq;
boolean              single1 = false, single2 = false;
boolean              freerun1 = false, freerun2 = false;

void tfr_byte(byte data)
{
    for (int i=0; i<8; i++, data>>=1)
    {
        digitalWrite(DATA, data & 0x01);
        pulseHigh(W_CLK); //after each bit sent, CLK is pulsed high
    }
}

void sendFrequency(double frequency)
{
    int32_t freq = frequency * 4294967295/125000000;
    for (int b=0; b<4; b++, freq>>=8)
    {
        tfr_byte(freq & 0xFF);
    }
    tfr_byte(0x000); // Final control byte, all 0 for 9850 chip
    pulseHigh(FQ_UD);
}

void restart_sampl( void )
{
    flag = 1;
}
```

```

    adc_configure_trigger(ADC, ADC_TRIG_SW, 1);
    adc_start(ADC);
    while(flag){} // wait new pull
}

void adc_setup ()
{
    pmc_enable_periph_clk(ID_ADC);
    adc_init(ADC, SystemCoreClock, 21000000UL, ADC_STARTUP_FAST);
    NVIC_EnableIRQ (ADC_IRQn);
    adc_disable_all_channel(ADC);
    adc_enable_interrupt(ADC, ADC_IER_RXBUFF);
    ADC->ADC_RPR = (uint32_t) adcIn; // DMA buffer
    ADC->ADC_RCR = INP_BUFF;
    ADC->ADC_PTCR = 1;
    adc_set_bias_current(ADC, 0x01);
    adc_enable_channel(ADC, ADC_CHANNEL_7); // AN0
    ADC->ADC_MR = (ADC->ADC_MR & 0xFFFFF7F); //DISABLE
    adc_configure_trigger(ADC, ADC_TRIG_SW, 1);
    adc_start(ADC);
}

void ADC_Handler (void)
{
    if ((adc_get_status(ADC) & ADC_ISR_RXBUFF) == ADC_ISR_RXBUFF) {
        ADC->ADC_MR = (ADC->ADC_MR & 0xFFFFF7F); // FREERUN-OFF
        ADC->ADC_RPR = (uint32_t) adcIn; // DMA buffer
        ADC->ADC_RCR = INP_BUFF;
        flag = 0;
    }
}

uint32_t Piezo_calib(uint32_t start_freq, uint32_t end_freq)
{
    uint32_t best_freq;
    uint16_t best_amp = 0;
    for(uint32_t ii = start_freq; ii < end_freq; ii += 100)
    {
        sendFrequency(ii);
        delay(10);
        restart_sampl();
        if(max_array(adcIn, INP_BUFF) > best_amp)
        {
            best_amp = max_array(adcIn, INP_BUFF);
            best_freq = ii;
        }
    }
    return(best_freq);
}

int max_array(uint16_t a[], unsigned int num_elements)
{
    int i, max_num = -32000;

```

```

    for (i=0; i<num_elements; i++)
    {
        if (a[i]>max_num)
        {
            max_num = a[i];
        }
    }
    return(max_num);
}

void setup()
{
    SerialUSB.begin(0);
    while(!SerialUSB);
    pinMode(FQ_UD, OUTPUT);
    pinMode(W_CLK, OUTPUT);
    pinMode(DATA, OUTPUT);
    pinMode(RESET, OUTPUT);
    pinMode(ADC_CTRL, OUTPUT);
    digitalWrite(ADC_CTRL, HIGH);
    pulseHigh(RESET);
    pulseHigh(W_CLK);
    pulseHigh(FQ_UD); // this pulse enables serial mode
    adc_setup();
    delay(1);
    resonant_freq = Piezo_calib(30000, 150000);
    digitalWriteDirect(RESET, HIGH);
    res_f[0] = resonant_freq;
    res_f[1] = resonant_freq >> 8;
    res_f[2] = resonant_freq >> 16;
    res_f[3] = resonant_freq >> 24;
    SerialUSB.write((uint8_t *)res_f, 4);
    SerialUSB.flush();
}

void loop()
{
    serialEvent();
    if(single1) // single measurement of the residual signal
    {
        pulseHigh(RESET);
        pulseHigh(W_CLK);
        pulseHigh(FQ_UD);
        sendFrequency(freq);
        delay(5);
        digitalWriteDirect(RESET, HIGH);
        restart_sampl();
        SerialUSB.write((uint8_t *)adcIn, OUT_BUFF);
        SerialUSB.flush();
        single1 = false;
    }
    if(single2) // single measurement with the signal on
    {

```

```

pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD);
sendFrequency(freq);
delay(5);
restart_sampl();
SerialUSB.write((uint8_t *) adcIn, OUT_BUFF);
SerialUSB.flush();
digitalWriteDirect(RESET, HIGH);
single2 = false;
}
if(freerun1) // continuous measurement of the residual signal
{
pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD);
sendFrequency(freq);
delay(5);
digitalWriteDirect(RESET, HIGH);
restart_sampl();
SerialUSB.write((uint8_t *) adcIn, OUT_BUFF);
SerialUSB.flush();
}
if(freerun2) // continuous measurement
{
pulseHigh(RESET);
pulseHigh(W_CLK);
pulseHigh(FQ_UD);
sendFrequency(freq);
delay(5);
restart_sampl();
SerialUSB.write((uint8_t *) adcIn, OUT_BUFF);
SerialUSB.flush();
}
}
}

void serialEvent()
{
if(SerialUSB.available() > 6)
{
rec_buf[0] = SerialUSB.read(); // initiate communication
if(rec_buf[0] == 170) // if received the communication byte
{
rec_buf[1] = SerialUSB.read(); // read mode select byte
rec_buf[2] = SerialUSB.read(); // frequency select MSB
rec_buf[3] = SerialUSB.read();
rec_buf[4] = SerialUSB.read();
rec_buf[5] = SerialUSB.read(); // frequency select LSB
rec_buf[6] = SerialUSB.read(); // end of communication
if(rec_buf[6] == 204)
{
freq = 0;
freq = (uint32_t) rec_buf[2] << 24;
}
}
}
}

```


Pose Estimation

```
% add folder and all subfolders of the code directory
addpath(genpath('D:/Pose Estimation Code'));

try
    tmp = cportwrite(lidar, 'QT', 'LF');
    rc = cportclose(lidar);
    fprintf('\nClosed Serial Ports\n');
catch
    fprintf('\nNo Serial Ports were Open\n');
end

clear all
delete(instrfindall)
delete(imaqfind)
close all
imaqreset
clc
warning('off', 'all');
tStart = tic;

% CONSTANTS
motor_step = 10;           % slider step in mm
max_pos = 750;            % how far to go down
cam_center = 360;         % the center pixel of camera, from calibration
delta = 10;               % lidar correction coefficient
step = 2;                 % small step during laser measurement
lidar_cor = 40;           % distance between lidar and camera center
fruit_size = 100;
d_max = 500;              % max distance
d_min = 100;              % min distance
side_lim = 200;          % boundary to sides
eps = 10;                 % DBSCAN segmenting coefficient
min_pts = 15;             % DBSCAN segmenting coefficient
h_fruit = 80;             % hight of the model
r1 = 20;                  % model top radius
r2 = 15;                  % model bot radius
min_segm = 7000;          % threshold for bwareaopen()
segm_thr = 0.2;           % threshold for im2bw
trg_thr = 0.8;            % how many points must be in target area

% options for Coherent Point Drift
opt.method = 'affine';
opt.viz = 1;              % don't show visualization
opt.max_it = 15;          % maximum iteration count
opt.fgt = 2;              % use Fast Gauss Transform and fine tune
                           % using truncated kernel approximations
opt.rot = 1;              % estimate strictly rotation
opt.scale = 1;            % estimate scaling
se = strel('disk', 5);
```

```

se1 = strel('disk',10);
se2 = strel('disk',25);
% Create the model
hf = linspace(0,h_fruit); % hight of the fruit
phi = linspace(0,pi); % angle to work with
[hf,phi] = meshgrid(hf,phi); % make grid for the angle and fruit hight
r = r2 - hf*(r2-r1)/h_fruit; % make the slope gradient
xmdl = r.*sin(phi); % generate point x coordinates
ymdl = r.*cos(phi); % generate point y coordinates
zmdl = -hf; % generate point z coordinates
xmdl = xmdl(:);
ymdl = ymdl(:);
zmdl = zmdl(:);
mdl = [xmdl,ymdl,zmdl]; % generate model matrix
xq_mdl = linspace(min(ymdl),max(ymdl));
yq_mdl = linspace(min(zmdl),max(zmdl));
[xq_mdl,yq_mdl] = meshgrid(xq_mdl,yq_mdl);
FM =
scatteredInterpolant(mdl(:,2),mdl(:,3),mdl(:,1),'natural','none');
VM = FM(xq_mdl,yq_mdl);
mdl_new = [xq_mdl(:),yq_mdl(:),VM(:)];

% INITIALIZATION
fprintf('\nSetting Up the Communications... ');

% SET UP THE CAMERA
vid = videoinput('winvideo',1); % select input device
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.TriggerRepeat = inf;
vid.ReturnedColorspace = 'rgb';
%src.BacklightCompensation = 'off';
vidRes = vid.VideoResolution;
nc = vidRes(1);
nr = vidRes(2);

% LIDAR SETUP
lidar = cportopen('COM8'); % depends on the used port
pause(0.1);
conf = cportconfig(lidar,'BaudRate',115200,...
    'ByteSize',8,'ReadIntervalTimeout',1);
if isempty(conf)
    error('Failed to connect Lidar');
end
tmp = cportwrite(lidar,'SCIP2.0','LF');
pause(0.1);
[ch,err] = cportreset(lidar);
tmp = cportwrite(lidar,'VV','LF');
pause(0.1);
[ch,err] = cportreset(lidar);
tmp = cportwrite(lidar,'BM','LF');
pause(0.1);
[ch,err] = cportreset(lidar);

```

```

% ARDUINO SETUP
ard_uno = serial('COM7'); % depends on the used port
set(ard_uno, 'BaudRate', 115200, 'terminator', '', 'Timeout', 0.1);
fopen(ard_uno); % Arduino will reset and do homing
while(ard_uno.BytesAvailable == 0)
    pause(0.01);
end
fread(ard_uno);
fprintf('done');
fprintf('\nHoming the slider and initializing the system. ');
pause(20); % wait for slider to finish homing
tInit = toc(tStart);

% SEARCH FOR TARGETS
fprintf('\nSearching for targets...\n');
target_index = 0; % detected object index
X = 0; % x position in the image
Y = 0; % y position in the image
H = 0; % actual position with respect to the slider
I = zeros(nr,nc,3,1); % variable to save the images
BW = zeros(nr,nc,1); % variable to save the found segments
BB = zeros(4,1); % variable to save the bounding boxes

s_l = 0.5;
s_h = 1.1;

start(vid);
for position = motor_step:motor_step:max_pos
    img = getdata(vid,1,'uint8');
    fwrite(ard_uno,position,'uint16','sync');
    flushdata(vid);

% -----
% ----- This part deals with recognition -----
% ----- change according to the method used -----
% -----

% remove the background
bck = im2bw(img,0.1);
se = strel('disk',5);
bck = imopen(bck,se);
bck = bwareaopen(bw,2000);

% get rid of the bright spots
fprintf('\nRemoving the hotspots...')
hotSpots = im2bw(img,.85);
hotSpots = hotSpots&bckGrnd;
hotSpots = imdilate(hotSpots, true(11));
im_r = img(:,:,1);
im_g = img(:,:,2);
im_b = img(:,:,3);
r_new = roifill(im_r,hotSpots);

```

```

g_new = roifill(im_g,hotSpots);
b_new = roifill(im_b,hotSpots);
imNEW = cat(3,r_new,g_new,b_new);

hsv_im = rgb2hsv(imNEW);
s = hsv_im(:,:,2);
bw = s > s_l & s < s_h;
bw = bw&bck;
bw = imopen(bw,sel);
bw = bwareaopen(bw,min_segm);

%-----
% this part looks for objects and registers new object positions
STATS = regionprops(img_trg,'Centroid','BoundingBox');

if ~isempty(STATS)
    nD = length(STATS); % number of detections
    obj_pos = zeros(nD,2); % variable to save object positions
    for obj = 1:nD % check all the found objects
        cx = STATS(obj).Centroid(1);
        cy = STATS(obj).Centroid(2);
        % look for an object near the center of camera
        if cy > (cam_center-30) & cy < (cam_center+30)
            % check if the object has been already registered
            if target_index > 0
                if any(H > (position-21))
                    fprintf('\nthere was a detection before');
                    o2cX = X(H > (position-21));
                    o2cY = Y(H > (position-21));
                    if any(o2cX(o2cX > (cx-20) & o2cX < (cx+20)))
                        fprintf(' and the center is similar');
                        Yval = o2cY(o2cX > (cx-20) & o2cX < (cx+20));
                        idx = ismember(Y,Yval);
                        if abs(cam_center-cy) < abs(cam_center-Yval)
                            fprintf(', but new is better');
                            H(idx) = position; % replace the old
                            X(idx) = cx;
                            Y(idx) = cy;
                            I(:, :, :, idx) = img;
                            BW(:, :, idx) = img_trg;
                            BB(:, idx) = STATS(obj).BoundingBox;
                        else
                            fprintf(', this is the same target');
                        end
                    else % this is a different fruit
                        fprintf('\nthe centers are different');
                        target_index = target_index + 1;
                        H(target_index) = position;
                        X(target_index) = STATS(obj).Centroid(1);
                        Y(target_index) = STATS(obj).Centroid(2);
                        I(:, :, :, target_index) = img;
                        BW(:, :, target_index) = img_trg;
                    end
                end
            end
        end
    end

```

```

        BB(:,target_index)=STATS(obj).BoundingBox;
    end
else
    fprintf('\n\na new target');
    target_index = target_index + 1;
    H(target_index) = position;
    X(target_index) = STATS(obj).Centroid(1);
    Y(target_index) = STATS(obj).Centroid(2);
    I(:,:,,target_index) = img;
    BW(:,:,target_index) = img_trg;
    BB(:,target_index) = STATS(obj).BoundingBox;
end
else
    fprintf('\n\nthis is the first detection')
    target_index = target_index + 1;
    H(target_index) = position;
    X(target_index) = STATS(obj).Centroid(1);
    Y(target_index) = STATS(obj).Centroid(2);
    I(:,:,,target_index) = img;
    BW(:,:,target_index) = img_trg;
    BB(:,target_index) = STATS(obj).BoundingBox;
end
end
end
end
clear X Y % delete x and y as we will use them later again
cur_pos = position;
fprintf('\n\nNumber of detections: %d',target_index)
stop(vid);
tVid = toc(tStart) - tInit;

% LASER MEASUREMENT
start_pos = max(H(:)) + 0.5*fruit_size - lidar_cor;
stop_pos = min(H(:)) - 0.5*fruit_size - lidar_cor;
fwrite(ard_uno,start_pos,'uint16','sync');
h_total = zeros(fruit_size/step + 1,target_index);
for h = 1:target_index % make matrice with all the heights
    h_total(:,h) = (H(h) + 0.5*fruit_size - lidar_cor):-step:(H(h) -
0.5*fruit_size - lidar_cor);
end
pause(5); % let the slider get to the position if it is far
L = zeros(682,(fruit_size/step + 1),3,target_index);
C = ones(target_index,1);
pos_idx = zeros(target_index,1);
for h_go = start_pos:-step:stop_pos % go through all steps
    if any(ismember(h_go,h_total))
        for i = 1:target_index
            if ismember(h_go,h_total(:,i))
                pos_idx(i) = 1;
            else
                pos_idx(i) = 0;
            end
        end
    end
end

```

```

end
fwrite(ard_uno,h_go,'uint16','sync');
cur_pos = h_go;
pause(0.1);
[px,py] = readLidar(lidar);
py(1:257) = nan;
px(1:257) = nan;
py(513:end) = nan;
px(513:end) = nan;
for j = 1:target_index
    if pos_idx(j)
        L(:,C(j),1,j) = px;
        L(:,C(j),2,j) = py;
        L(:,C(j),3,j) = C(j)*step;
        C(j) = C(j) + 1;
    end
end
end
if abs(cur_pos - h_go) == 10
    fwrite(ard_uno,h_go,'uint16','sync');
    cur_pos = h_go;
    while(ard_uno.BytesAvailable == 0)
        pause(0.01);
    end
    fread(ard_uno);
end
end
tLid = toc(tStart) - (tInit + tVid);

% PROCESSING THE DATA
R = zeros(3,3,target_index); % initialize rotation matrices
t = zeros(3,1,target_index); % initialize translation vectors
s = zeros(1,target_index); % initialize scaling factors
for obj_nr = 1:target_index
    fprintf('\nSegmenting the Target in the Image... ');

    % Take LIDAR data and apply the first filtering
    px = L(:, :, 1, obj_nr);
    py = L(:, :, 2, obj_nr);
    pz = L(:, :, 3, obj_nr);
    closeX = px(:);
    closeY = -py(:); % reverse y direction
    closeZ = pz(:);
    closeX(isnan(px(:))) = [];
    closeY(isnan(py(:))) = [];
    closeZ(isnan(pz(:))) = [];
    pts = closeX <d_min|closeX>d_max|closeY<-side_lim|closeY>side_lim;
    X = closeX(pts == 0);
    Y = closeY(pts == 0);
    Z = closeZ(pts == 0);
    figure;plot3(X,Y,Z,'r. '); % remove from the final code
    % forward-project points
    fprintf('\nForward Projection of the Lidar data... ');

```

```

x4segm = 0;
y4segm = 0;
z4segm = 0;
fc = [1331;1330];
cc = [737;294];
kc = [-0.4105;0.2105;0.0024;-0.0009;0];
alpha_c = 0;
data = [-Y(:),-Z(:),X(:)]';
om = zeros(3,1);
T = [9,78,0]';%tz,tx regulates on x, ty on y
pixels = project_points2(data,om,T,fc,cc,kc,alpha_c);
ptMap = zeros(nr,nc);
vldPtIdx = 1;
for i = 1:length(pixels(1,:))
    xp = round(pixels(1,i));
    yp = round(pixels(2,i));
    isValid_x = xp>BB(1,obj_nr)&&xp<(BB(1,obj_nr)+BB(3,obj_nr));
    isValid_y = yp>BB(2,obj_nr)&&yp<(BB(2,obj_nr)+BB(4,obj_nr));
    if isValid_x && isValid_y
        ptMap(yp,xp) = 1; % save the point index
        x4segm(vldPtIdx) = data(1,i);
        y4segm(vldPtIdx) = data(2,i);
        z4segm(vldPtIdx) = data(3,i);
        vldPtIdx = vldPtIdx + 1;
    end
end

figure;imshowpair(uint8(I(:,:, :,obj_nr)),ptMap);
figure;imshowpair(uint8(I(:,:, :,obj_nr)),uint8(BW(:,:, :,obj_nr)));

% Segmenting points
fprintf('\nObtaining target points... ');
pts4segm = [x4segm;y4segm;z4segm]';
[labs,labscore] = dbscan(pts4segm,eps,min_pts);
segm_idx = 0;
j = 0;
for i = 1:max(labs) % check which segment is the biggest
    tmp = sum(labs == i);
    if tmp > j
        j = tmp;
        segm_idx = i;
    end
end
x4segm(labs ~= segm_idx) = [];
y4segm(labs ~= segm_idx) = [];
z4segm(labs ~= segm_idx) = [];

% Creating the model
fprintf('\nCreating the model... ');
% interpolation
xx = x4segm; % take measurement data
yy = y4segm;
zz = z4segm;

```

```

xq_trg = linspace(min(xx),max(xx));
yq_trg = linspace(min(yy),max(yy));
[xq_trg,yq_trg] = meshgrid(xq_trg,yq_trg);
FT = scatteredInterpolant(xx(:),yy(:),zz(:),'natural','none');
VT = FT(xq_trg,yq_trg);
trg_new = [xq_trg(:),yq_trg(:),VT(:)];
trgX = trg_new(:,1);
trgY = trg_new(:,2);
trgZ = trg_new(:,3);
trgX(isnan(VT)) = [];
trgY(isnan(VT)) = [];
trgZ(isnan(VT)) = [];
mdlX = mdl_new(:,1);
mdlY = mdl_new(:,2);
mdlZ = mdl_new(:,3);
mdlZ = -mdlZ + repmat(rl,size(mdlZ)); % turn around the model
mdlX(isnan(VM)) = [];
mdlY(isnan(VM)) = [];
mdlZ(isnan(VM)) = [];
% make equal datasets by randomly deleting points
if length(trgX) > length(mdlX)
    del_nr = length(trgX) - length(mdlX); % how many to remove
    idx = randperm(size(mdlX,1),del_nr);
    trgX(idx) = [];
    trgY(idx) = [];
    trgZ(idx) = [];
else
    del_nr = length(mdlX) - length(trgX);
    idx = randperm(size(trgX,1),del_nr);
    mdlX(idx) = [];
    mdlY(idx) = [];
    mdlZ(idx) = [];
end

% Find the pose in space
fprintf('\nCalculating the orientation in space... ');
A = [mdlX(:),mdlY(:),mdlZ(:)];
B = [trgX(:),trgY(:),trgZ(:)];
[Transform,~] = cpd_register(B,A,opt);
R(:,:,obj_nr) = Transform.R;
t(:,:,obj_nr) = Transform.t;
s(1,obj_nr) = Transform.s;
% calculate the angles
[roll,pitch,yaw] = R2angle(Transform.R);
fprintf('\nCalculated angles:');
fprintf('\n\tRoll: %.2f',roll);
fprintf('\n\tPitch: %.2f',pitch);
fprintf('\n\tYaw: %.2f',yaw);
tProc = toc(tStart) - (tLid + tVid + tInit);
% get stem position
figure;
imshowpair(uint8(I(:,:,obj_nr)),ptMap);
hold on

```

```

t_proj = project_points2(Transform.t,om,T,fc,cc,kc,alpha_c);
xp = round(t_proj(1));
yp = round(t_proj(2));
plot(xp,yp,'y.','markersize',10);

stem_base = [0,-(Transform.s*h_fruit-70),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit-60),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit-50),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit-40),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit-30),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit-20),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit-10),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));

```

```

plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'r.','markersize',10);
stem_base = [0,-(Transform.s*h_fruit+10),0];
stem_pos = Transform.R*stem_base' +
repmat(Transform.t,1,size(stem_base,1));
stem_proj = project_points2(stem_pos,om,T,fc,cc,kc,alpha_c);
xp = round(stem_proj(1));
yp = round(stem_proj(2));
plot(xp,yp,'b.','markersize',30);
hold off
end

fprintf('\n\nEND OF PROGRAM\n');
fprintf('\nTime spent for system initialization: %.2f seconds',tInit);
fprintf('\nTime spent for video capture: %.2f seconds',tVid);
fprintf('\nTime spent for laser measurement: %.2f seconds',tLid);
fprintf('\nTime spent for data processing: %.2f seconds',tProc);
fprintf('\nTotal time spent: %.2f seconds\n',toc(tStart));

tmp = cportwrite(lidar,'QT','LF');
rc = cportclose(lidar);
fclose(ard_uno);
delete(instrfindall)
delete(imaqfind)

```

Piezo Sensor

```
delete(instrfindall)
clear all
clc

packet_size = 1024; % one measurement in bytes
mode2set = 4; % 3 for residual, 4 for continuous

% open the communication
ardDue = serial('COM5');
set(ardDue, 'BaudRate', 115200, 'Terminator', 'LF', ...
    'InputBufferSize', packet_size, 'Timeout', 0.5);
fopen(ardDue);

figure;
result = zeros(packet_size/2, 1);
result(:, :) = nan;
subplot(2, 1, 1);
h1 = plot(result);
axis([0 packet_size/2 0 4100]);
title('Measurement');
subplot(2, 1, 2);
q_r2 = zeros(packet_size/2, 1);
q_r2(:, :) = nan;
h2 = plot(q_r2, 'b. ');
title('Integral');
drawnow;

res_freq = 0;
max_val = 0;

while (~ardDue.BytesAvailable)
    pause(0.01);
end
rf = fread(ardDue, 4);
bin_res_freq = [dec2bin(rf(4), 8) dec2bin(rf(3), 8) dec2bin(rf(2), 8)
dec2bin(rf(1), 8)];
resonant_freq = bin2dec(bin_res_freq);
fprintf('Measured resonant frequency by ArdDue is %u
\n', resonant_freq);

% Do calibration check by going through frequencies from 20k to 150k
for i = 20:1:150
    freq = i*1000;
    bfreq = dec2bin(freq, 32);
    bf1 = bin2dec(bfreq(1:8));
    bf2 = bin2dec(bfreq(9:16));
    bf3 = bin2dec(bfreq(17:24));
    bf4 = bin2dec(bfreq(25:32));
    data2send = [170 mode2set bf1 bf2 bf3 bf4 204];
```

```

flushinput(ardDue);
fwrite(ardDue,data2send);
for tm = 1:0.01:5
    if ardDue.BytesAvailable
        break;
    else
        pause(0.01)
    end
end
if ardDue.BytesAvailable
    buffer = fread(ardDue);
    b_data = dec2bin(buffer,8);
    j = 1;
    for k = 1:2:packet_size
        tmp_data{j} = [b_data(k+1,:) b_data(k,:)];
        j = j+1;
    end
    r = bin2dec(tmp_data);
    r2 = r - repmat(mean(r),size(r));
    integ = trapz((r2).^2);
    if integ > max_val
        max_val = integ;
        res_freq = i*1000;
    end
else
    fprintf('\nNo data received');
end
end

fprintf('Measured resonant frequency by Matlab is %u \n',res_freq);

freq = max(res_freq,resonant_freq); % choose the highest from two
detected
bfreq = dec2bin(freq,32);
bf1 = bin2dec(bfreq(1:8));
bf2 = bin2dec(bfreq(9:16));
bf3 = bin2dec(bfreq(17:24));
bf4 = bin2dec(bfreq(25:32));
data2send = [170 mode2set bf1 bf2 bf3 bf4 204];
flushinput(ardDue);
fwrite(ardDue,data2send);

% define constants
thr = 0.20; % detection threshold
Fs = 1/(1e-6); % sampling frequency
L = 2048; % length of the signal
T = 1e-6; % sampling period
xx = 160; % initial arm position X
yy = 0; % initial arm position Y
zz = 160; % initial arm position Z
moveStep = 2; % arm displacement step in mm
cc = 250; % pulse width value to close the cutter
co = 360; % pulse width value to open the cutter

```

```

% initialize the hardware
move_arm(ardUno,xx,yy,zz);

% set up the timer for the measurement
tmr = timer('ExecutionMode','FixedSpacing','Period',0.1);
tmr.TimerFcn = @sensor;
start(tmr);

% main arm control
while true
    fprintf('\nCurrent position (x,y,z): %f.0,%f.0,%f.0',xx,yy,zz)
    fprintf('\nChoose the control method:')
    fprintf('\n\t\t1. by coordinates')
    fprintf('\n\t\t2. by direction')
    fprintf('\n\t\t3. exit program')
    choice = input('\nChoice: ');
    switch choice
        case 1
            xx = input('\nCoordinate X: ');
            yy = input('\nCoordinate Y: ');
            zz = input('\nCoordinate Z: ');
            move_arm(ardUno,xx,yy,zz);
        case 2
            while true
                moveDir = input('\nDirection (u/d/l/r/f/b) or e for
exit: ','s');
                switch moveDir
                    case 'u'
                        zz = zz + moveStep;
                        move_arm(ardUno,xx,yy,zz);
                    case 'd'
                        zz = zz - moveStep;
                        move_arm(ardUno,xx,yy,zz);
                    case 'l'
                        yy = yy + moveStep;
                        move_arm(ardUno,xx,yy,zz);
                    case 'r'
                        yy = yy - moveStep;
                        move_arm(ardUno,xx,yy,zz);
                    case 'f'
                        xx = xx + moveStep;
                        move_arm(ardUno,xx,yy,zz);
                    case 'b'
                        xx = xx - moveStep;
                        move_arm(ardUno,xx,yy,zz);
                    case 'e'
                        break;
                end
            end
        case 3
            break;
    end
end

```

```

    end
end

stop(tmr);
fclose(ardDue);
fclose(ardUno);

% set up the timer function
function sensor(src,event)
    flushinput(ardDue); % clear the serial buffer
    while(ardDue.BytesAvailable < 1024) % wait till the buffer fills
        tic;
        while toc < 0.001
            end
        end
    end
    buffer = fread(ardDue); % read the serial buffer
    j = 1;
    g = 1;
    for k = 1:2:1024
        LSB = dec2bin(buffer(k),8);
        MSB = dec2bin(buffer(k+1),8);
        data = [MSB,LSB];
        result(j,:) = bin2dec(data);
        j = j + 1;
    end
    r2 = result - repmat(mean(result),size(result));
    % calculate the integral
    q_r2(g) = trapz((r2).^2);
    if q_r2(g) < thr
        fprintf('\nTouched')
    end
    if g > 512
        q_r2 = q_r2(2:end);
        g = 512;
    else
        g = g + 1;
    end
end

% calculate the ttf
tmp = abs(fft(r2(:))/L); % do fft analysis
fft_r2 = tmp(1:L/2+1); % get rid of the mirror effect
fft_r2(2:end-1) = 2*fft_r2(2:end-1);
f = Fs*(0:(L/2))/L; % define fft X axis

% display
set(h1,'YData',r2);
set(h2,'YData',q_r2);
set(h3,'XData',f,'YData',fft_r2);
drawnow;
end

```