Autonomous Anthropomorphic Robotic Arm to Monitor Plant Growth in a Laboratory

A thesis presented in partial fulfilment

of the requirements for the degree of

Masters

in

Engineering

at

Massey University

Palmerston North

New Zealand

Mark Seelye

2011

Abstract

An autonomous anthropomorphic robotic arm was designed, fabricated and programmed for monitoring of plant tissue grown in a modified *in vitro* clonal plant propagation system being developed by The New Zealand Institute for Plant & Food Research. The custom fabricated aluminium robotic arm uses a vertical linear ball shaft and high speed stepper motors to provide arm joints movements enabling the arm to swivel 180 degrees horizontally. Sensors located at the end of the arm are used to monitor plant growth and the immediate growing environment. This includes a compact colour zoom camera on a pan and tilt mount for image capturing, red, green and blue (RGB) colour sensors to monitor leaf colour as well as temperature, relative humidity and carbon dioxide sensors. The robotic arm is capable of reaching over multiple trays (600mm x 600mm) of plantlets. Captured plant tissue images are processed using innovative algorithms to determine tissue or whole plant growth rates over specified time periods. Leaf colour sensors provide information on tissue health status when compared to predetermined optimum values. Custom software fully automates the operation of the arm and the data capture, allowing the arm to return to specified sites (i.e. individual plantlets) at set time intervals to identify subtle changes in growth rates and leaf colour. This allows plant nutrient levels and the immediate environment to be regularly adjusted in response to continuous sensing resulting in optimised rapid growth with minimal human input.

Acknowledgements

In completing this Masters degree in Mechatronic Engineering, I would like to take this opportunity to sincerely thank:

My supervisor, Dr Gourab Sen Gupta from the School of Engineering and Advanced Technology (SEAT), Massey University, Palmerston North, New Zealand for his guidance and expertise throughout this project from start to finish.

Mr Clive Bardell, Mr Kerry Griffiths and Mr Greg McLeay from the School of Engineering and Advanced Technology's (SEAT) metal workshop and CNC lab for their guidance and expertise in teaching me the hands on skills required to build the robotic arm and program and running of the CNC machine.

Associate professor Donald Bailey from the School of Engineering and Advanced Technology (SEAT), for his input into colour transformations leading to the calibration of the colour sensors.

Dr Rory Flemmer from the School of Engineering and Advanced Technology (SEAT) for his input into the control algorithms required to control the robotic arm and input into the image analysis.

Mr John Seelye, Scientist, Breeding Technologies Group, The New Zealand Institute for Plant and Food Research Limited, Palmerston North, New Zealand for sponsoring the entire project.

Table of Contents

List of Figures

List of Tables

1. Introduction

Visual monitoring of plant growth as a means of identifying the health status of individual plants or plantation crops has been undertaken widely throughout the world. This however relies on human judgment and very subtle changes are not readily identifiable by humans, or identified too late. Continuous monitoring of plants around the clock by humans is impractical, costly and inaccurate. This issue has lead to Plant & Food Research investigating the use of a robotic system that can monitor propagated plant growth around the clock with minimal human input. Research monitoring the health status of plants has determined simple effective methods such as monitoring plant leaf colour are highly effective. All plants have an optimal “healthy” colour, with very subtle changes away from “healthy” leaf colour implying the plant may be experiencing stress. These stress levels can be attributed to a lack of nutrients, and by adjusting nutrient levels it is possible to optimise the growth of a healthy plant.

This Master’s project aimed to develop a complete automated solution for the New Zealand Institute of Plant & Food Research to monitor the growth of small propagated plants using vision and colour. Captured plant tissue images were processed using innovative algorithms to determine tissue or whole plant growth rates over specified time periods. Leaf colour sensors provided information on tissue health status when compared to predetermined optimum values.

Due to a confidentiality agreement with Plant & Food Research no in-depth discussion of the application and use of this system is provided and no information is presented on the actual propagation system, as it is not the focus of this project. This thesis discusses in-depth the development of the robotic system, its control systems and the range of sensors used.

Ultimately, a unique system has been developed for Plant & Food Research which makes use of a number of electrical and computer systems engineering theories. A prototype robotic arm has been designed, developed and constructed, which has been integrated with motors, controllable using specific

electronic components and custom computer software. A number of sensors have been successfully introduced, tested, calibrated and integrated into the robotic system including colour, proximity, CO_2, temperature and humidity systems. The system required the use of vision, with custom algorithms being developed to identify plant growth rates. The entire system was integrated into a fully automated package. This allowed the system to autonomously return to specified sites (i.e. individual plantlets) at set time intervals to identify subtle changes in growth rates and leaf colour. This provided the potential for plant nutrient levels and the immediate environment to be routinely adjusted in response to continuous sensing resulting in optimised rapid growth with minimal human input.

This thesis consists of a number of sections, describing the process followed for the successful completion of the automated solution. The scope of this project was to source or develop a robotic system which could be integrated into an existing environment consisting of propagated plant material held in trays located on multiple levels of shelves. Section 3 describes the electronic components that were required to allow for a successful interface and communication between the computer and robotic system. Once a robotic system had been sourced / developed and was controllable. A number of sensors (Section 4) were acquired, calibrated and integrated into the system that allowed colour readings of plant media to be obtained. Consistent sensor reading needed to be taken at fixed heights. Environmental information needed to be obtained including temperature, humidity and CO_2 levels. Vision was required to identify plant growth rates which were calculated using innovative algorithms. Finally, each component required custom software to not only interpret data readings but also to integrate and automate the entire system, eliminating human input to provide a complete automated solution, outlined in Section 5.

The use of colour sensors to monitor plant colour and growth rates can be adapted to numerous applications within the horticulture industry. The system presented in this thesis has been tailor-made for Plant & Food Research and their working environment, however future developments could include the improved development of the robotic system and customising it for other

working environments. Further calibration of the sensors could improve the accuracy of the system. One future development would be to allow the system to move horizontally, allowing more shelves to be accessed. Alternative sensors, such as NIR (Near Infrared Sensors) for decisions on the health status of plants could also be investigated.

The work detailed in this thesis builds on a project entitled 'Remote Plant Monitoring System' [1] in which a basic robotic arm was constructed using standard components and remote control (RC) servo motors. The system housed a compact colour zoom camera located at the end of the arm (Figure 1). The entire system was controlled through custom software with both the arm and camera programmed to be controlled wirelessly via a joystick. While the project empowered researchers to move the robotic arm and camera to monitor plants, without the need to leave the office, automation of this basic system was considered a more useful tool.

Figure 1: Robotic arm designed as part of Honours project

As a result of the work completed as part of this Masters project a number of papers have been published and/or accepted for publication, these include:

(Full articles can be seen in the appendix – page 109):

1. Seelye, M., Sen Gupta, G., & Seelye, J., Bailey, D. (2011). Low Cost Colour Sensors for Monitoring Plant Growth in a Laboratory. *Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2011)*, pp 972-977.

2. Seelye, J., Seelye, M., Gupta, G. S., & Mullan, A. (2011). Opportunities for non invasive, real-time sensing of plant tissue growth. *Proceedings of the 19th Biennial Meeting of the New Zealand Branch of the International Association for Plant Biotechnology, Hammer Springs, 8-11 Feb 2011. pp 11.*

3. Seelye, M., Sen Gupta, G., & Seelye, J. (2010). Autonomous Anthropomorphic Robotic Arm using Vision and Colour to Monitor Plant Growth in a Laboratory. *Proceedings of Electronics New Zealand Conference, ENZCon 2010*, pp 33-38.

4. Seelye, M., Sen Gupta, G., Seelye, J., & Mukhopadhyay, S. C. (2010). Camera-in-hand Robotic System for Remote Monitoring of Plant Growth in a Laboratory. *Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2010)*, pp 809-814.

2. Literature Review

In today's world, robotic and automated products have become increasingly common in all economic sectors. The International Organization for Standardization defines a robot in ISO 8373 as: "an automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications". [2]

2.1. Robotics in the Workplace

Robotics and automation systems are relatively new inventions, though early automated operations date back to the early 1400s. Since the 1960s robotic systems have become more common as technology moved forward. George Devol was credited with inventing the first Industrial robot, called "Unimate" (Figure 2) which worked on the General Motors assembly line in 1961.

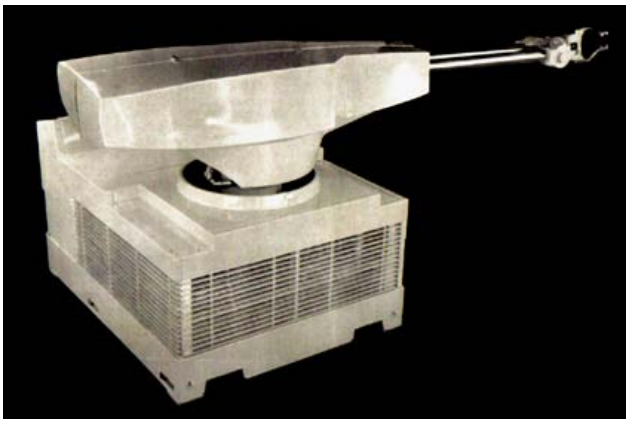


Figure 2: The original Unimate, considered to be the first industrial robot. (http://sharedrobotics.com/robot/the-unimate)

Since then, much time and research has been spent developing robots for specific tasks. Robotic systems are now used in all corners of the globe, ranging in operation, complexity, size and cost. Figure 3 (left image) demonstrates a flow diagram for a basic robotic system, with an advanced robotic system, the Kuka KR 100 robotic system used on a number of production lines worldwide (right image).

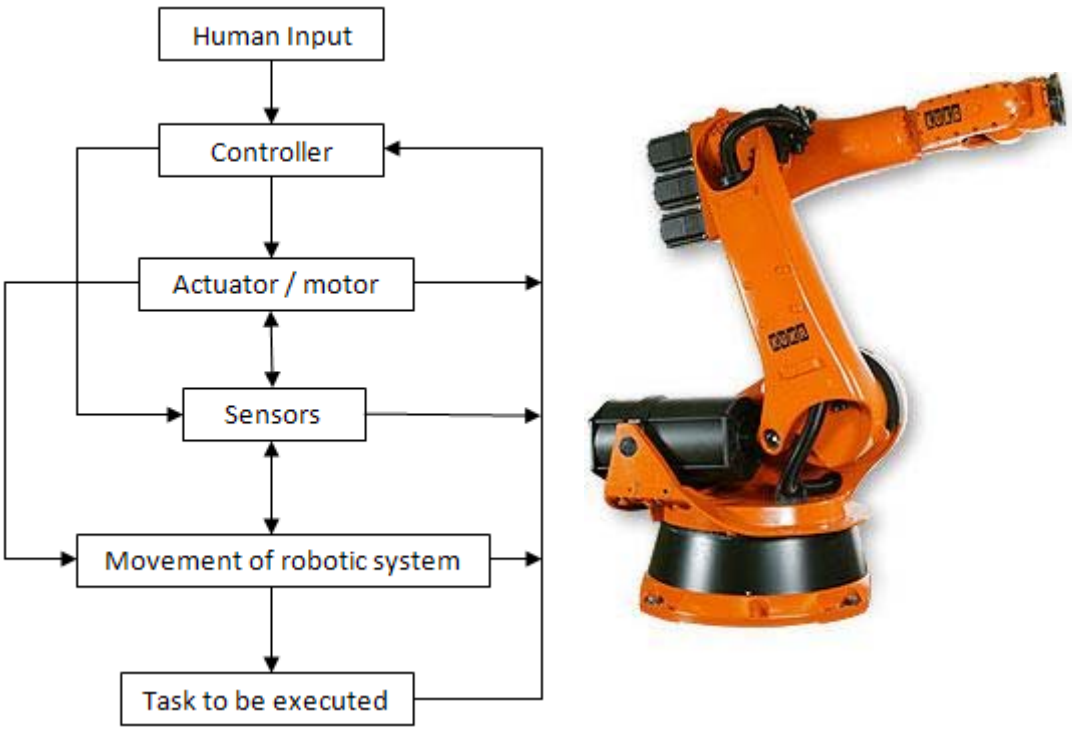


Figure 3: Basic concept of a robotic system (left) with an example of a modern day Kuka KR 100 robot (right). (http://www.robots.com/kuka.php?robot=kr+100)

Over the past decade there has been a continual push towards more automation in the horticulture industry. Now as robots become more sophisticated and reliable, we are beginning to see them used to undertake routine, often repetitive, tasks which are often expensive to do using a highly paid labour force. As this technology moves forward possible applications soon becoming realties.

Recent research articles include the current status of the core technologies required for the successful development of multiple autonomous robotic system for weed control, all of which use vision and artificial intelligence to identify weed growth [3], [4], [5], [6]. Advancement in electronics has empowered engineers to build robots that are capable of operating in unstructured environments using an image based system for identification of objects and their positions [7]. Camera-in-hand robotic systems are becoming increasingly popular. In these a camera is mounted on the robot, usually at the hand, to provide an image of the objects located in the robot's workspace with an emphasis on position – control in an environment with static objects [8].

Kataoka *et al*, identify that the status of crop growth during growth stages is important information used for crop cultivation tasks and management [9]. The paper introduces the means of determining the vegetation cover area of soybean and sugar beet crops over a set period of time using a vision system. A relationship was developed to determine the vegetation growth by comparing images acquired using vision to the actual plant dimensions. The benefits of being able to monitor plant growth over time include cost savings to the farmers and saving of the environment, with the information being used to predict future plantations and schedules. Ideally information collated in real time using remote sensing technologies, (which is non-destructive), is advantageous, as shown by the use of a Sony colour camera placed in front of a spray booth, to acquire images located 2.48m above the ground at an angle of 79.18 degrees to the ground. The images captured were processed using algorithms to determine the intensity of the RGB content of each pixel, with the green information being enhanced to distinguish between plant material and the background. Plants were selected randomly and manually measured to be able to make comparisons. The height of soybeans was measured while the sugar beet leaves were measured (as sugar beet grows along the ground).

Using two methods (Gompertz curve and exponential function) the results showed a strong correlation for the soybean height of 0.945 (Gompertz curve) and 0.858 (exponential function) and for sugar beet 0.895 (Gompertz curve) and 0.858 and 0.958 (exponential function).

The authors identified that the crop growth images could be used for estimating growth rates using the appropriate function (either Gompertz curve or exponential function). The paper states that while the project was successful for plants in a specific trial, it may not be appropriate for all plants. These authors used processed plant images captured by a colour camera at a fixed height, and the colour content was determined by scanning each pixel. However, no information was provided as to whether the camera lens was calibrated to eliminate any fish eye or distortion. It is possible that ambient light may have affected colour readings, although the green readings were used broadly to identify plants rather than for an exact colour. It is therefore possible that images captured showed overlapping plant leaf material which made the leaves

larger than what they actually were. Overall this paper provides methodology applicable to measuring a plant surface in a 2D plan by identifying the colour of pixels. This work could be extended by automating real-time image capturing to identify changes which could be used to better manage plant cultivation tasks which in turn could lead to greater time and cost efficiencies for farmers.

Robots are increasingly being used to sort, grade, package and even pick fruit and vegetables. For example, an autonomous wheeled robot has been developed to pick kiwifruit from orchard vines using vision to identify both fruit quality and as a means of navigation [10]. The harvester is intended to offset high labour costs associated with harvesting. An apple picker, similar to the kiwifruit picker, has also been developed [11]. The results have indicated a promising future for the machine which has the potential to replace six workers per machine. Automated greenhouse cucumber harvesting using vision for detection of the presence of fruit and to assess fruit ripeness and quality, has been developed to overcome labour shortages and reduce costs in the Netherlands [12].

A vision guided robotic manipulator for packing horticulture products such as tomatoes has been developed [13] with the results showing that the robot was 97.5% accurate in packing loosely placed tomatoes. The robotic manipulator was able to be modified to pack other horticulture products. Again, this system made use of vision to successfully identify the positioning and orientation of the fruit before packing [14]. The current status of different manipulators in the horticulture industry for numerous applications was also reviewed [13].

As well as the automation of fruit harvesting, a number of publications introduced methods of grading fruits using machine vision. Machines for grading fruit and vegetables in the horticulture industry have the potential to save millions of dollars in labour costs and wasted products. One paper in particular [15] introduced machine vision for grading and sorting of apples on a moving conveyer belt. Although only a prototype, the results showed that the system could successfully identify blemishes in the fruit and package the apples, rejecting any blemished fruit.

A recent article showed that machine vision had been used to grade strawberries by identifying size, colour, shape and maturity to ultimately increase the commercial value of them [16]. The paper outlined the use of a camera to capture images to carry out analysis. However, when compared to current systems, it made use of a closed image box to ensure all readings were taken under constant ambient light levels. The system was unique in that the strawberries moved along a conveyer system which had the ability to move back and forth to ensure correct images were taken.

Automation has also been applied to other grading systems for produce. For example, eggplants differ from conventional vegetables in that they have a glossy surface, and in general the more gloss a plant has, the fresher and softer (or healthier) it is and in this example, cameras were able to identify the gloss [17]. This paper also discussed the identification and grading of leeks (up to 60cm in length). Leeks often possess sandy roots, making them hard to identify and grade and the removal of the root system at a fixed length is discussed.

Machine-vision application over a range of fruit products, (apples, citrus, cherries, dates, tomatoes and olives), seeds and grains as well as numerous meat products (including fish, beef, chicken and lamb) has been reported [18] focussing on current methodologies and the potential short falls each of the different methods presents. The authors point out that more recent publications tend to have more accurate results, however they acknowledged these methods are not always the best. Similar machine vision methods to identify and monitor numerous agri-science products have also been reported.

Robotic techniques have been used in other horticultural applications with a report on the identification of seedlings on a large scale using machine-vision [19]. These authors report on interpreting captured black and white and colour photographs, with results showing that the black and white images were sufficient to identify and distinguish between different seed lines.

Automation of plant propagation systems using micro-dosing for the precise application of chemicals has gained the attention of researchers. For example, the first autonomous robot has been developed to automate the labour-intensive and often tedious work associated with micropropagation, [20]. In this

system the robotic arm autonomously clones plant material in a sterile environment with the final products potentially worth millions of dollars to the industry.

A system that combined plant recognition, micro-dosing and autonomous robotics [21] is also described. This system was capable of identifying plant material using machine-vision, with the entire system mounted on a moving autonomous platform with a micro dosing system capable of applying very small doses of chemicals to the targeted area. Strategies for automated plant production in harsh climates has been reviewed in [22].

Investigations into non-destructive methods to measure the health status of plants using colour is gaining popularity. Colour can be used as a means of identifying very subtle changes in a plant's leaf colour with this information providing a measure of the current plant health status. Although colour has been used extensively with machine-vision as a means of assessing quality, limited work has been carried out in real-time using colour for monitoring changes in plant material over time.

Investigations with microalgae [23] identified Chlorophyll *a* and Lipid content using processed digital images. The Chlorophyll *a* and Lipid content provided information on the growth status during cultivation. Chlorophyll was measured by identifying and evaluating the three primary colours, resulting in two linear correlation functions. The results obtained were compared to those determined by a standard method. The tedious and labour-intensive manual process of identifying Chlorophyll *a* and Lipid were replaced by these imaging techniques.

The authors of the above paper stated that this is the first rapid and reliable method for simultaneous determination of Chlorophyll *a* and Lipids. However, their method uses trichromatic theory, which states that any known colour can be made of the three primary colour (R, G and B). The paper reports that procedures were in place to ensure readings were constant, such as using a constant background colour. Selected images were analysed with custom software created in Visual Basic 6. RGB values recorded for each pixel were averaged to determine a mean RGB value for the entire picture. The study involved seven experiments with variances of 0.05 and 0.07 respectively for the

Chlorophyll *a* and Lipids. Having determined the values of Chlorophyll *a* and Lipids using the standard method, results showed the use of images to be accurate with an R^2 value of 0.99. The results were successfully applied to real microalgae cultivations.

The process described still required human input to capture the images which were captured at fixed heights. Although only set up for use on microalgae, the results could be applied to other plant material.

While the authors determined the RGB based on pixel values from an image, other methods such as the use of a colour sensor may render faster and more accurate results.

As discussed by Su *et al* chlorophyll *a* and Lipid contents are identified using processed digital images, similar to this a non-invasive and non-destructive method of determining the chlorophyll content of micropropagated potato plants by reviewing the RGB content of the plant leaves has recently been reported [24]. The methodology describes how potato plant leaves (approx $1cm^2$) were subject to a SPAD (Soil Plant Analysis Development) which gave information on the chlorophyll content. Chlorophyll traps light using energy to drive photosynthesis essential for the growth of plant.

Potato plant leaves were scanned into a computer providing images with 330ppi (pixels per inch).A fixed number of pixels (4 x 4) were selected from the median portion of the images and processed to determine the pixel colours in an RGB colour space. The RGB values were linked to the chlorophyll content from the SPAD experiments. The results showed correlation between the RGB content and the chlorophyll content of the leaf. One issue identified was the need to normalise the RGB values to eliminate any variance from ambient light.

Although a correlation existed, it was somewhat poor with a maximum correlation value of 0.77. This however shows there is potential for work to be conducted using images, as micropropagated potato plants are grown in Petri dishes, where the SPAD chlorophyll meter cannot directly access the plants. The work demonstrated potential, however there is a requirement for a significant human input with images still needing to be captured. The paper

made interesting discoveries by normalising the RGB values to remove any ambient light fluctuations. An improvement would be to remove the need to scan the images into the computer and use a more conventional approach using a digital camera.

Similar to [22-23] the paper made no comment on calibration of the camera or colours recorded, this in turn shows results only being amenable to their system. Again, an alternative method of capturing the colour of the plants material is possible, as the leaves are only small (approx 1cm^2) and again the RGB values are averaged over a central portion of the leaf. A sensor has been developed [25] to determine the chlorophyll content using a normalised difference vegetation index (NDVI) – a similar approach to using a RGB colour space.

Colour has also been used in other areas in agri-science industries for grading specific fruit and vegetables, and [26] explains how colour has been used as a means to estimate, from a sample of apples, the redness of the remaining apples on orchard trees (this did however rely on human interpretation as to the colour of the apples).

The use of algorithms to determine a food product's colour using an image segmentation approach, i.e. not including background colour when viewing the food product on a manufacturing line, has been investigated [27]. The results showed the algorithm to work successfully when used with potato chips and apricots, and would be ideal for colour quality measurements involving multiple objects. A simple method for identifying the colour of food products using a digital camera was introduced providing the user with information on the colour distribution and average colour of the food products [28]. This system used the image captured from the camera and determined the colour content using RGB, CMYK and *L*a*b** colour spaces. The authors noted that the system would not replace sophisticated systems currently available, but could provide an attractive alternative due to its simplicity in determining food colour.

A computer based system was described whereby a system was developed to monitor the colour of tea while it ferments [29]. The system used a digital image taken continuously over the fermentation period (between 1 and 3 hours) to

determine the RGB colour content as the tea fermented under optimum conditions, with the intention that it can be used to create a quality control system for the tea industry.

Colour sensing has also been used in other research areas including evaluation of bakery products using a machine vision system [30], designed to determine and separate the difference between dark and light coloured muffins on an automated conveyer system. Results showed the system to be 96% accurate with the pre-graded muffins and 79% accurate with the un-graded muffins. Again, this system made use of the RGB colour space for identification of muffin colour.

Colour has also been used in the health sector to determine blood glucose levels [31] where a colour meter with a RGB output is used to convert this to a *hue* and *saturation* colour space to identify glucose content present in a plasma sample of blood. This particular paper introduces the theory of how to convert between RGB and *hue* and *saturation* (and XYZ) utilising the 1931 XY chromaticity diagram.

An alternative means of measuring plants and crop health is to utilise Near Infrared Reflectance (NIR). Recent work detailed how near infrared sensors have become increasingly relied upon as they are cheaper, more portable and better adapted to hostile environments [32]. Three commercially available NIR sensors, different only in terms of measurement and wavelength, were tested and compared using apples. Work has been reported whereby a NIR sensor has been used to estimate nutritional status of many leaves [33]. This paper explained that the nutrient status of citrus plants was evaluated using NIR spectral reflectance, with this replacing the destructive method of chemical analysis.

A review has been completed of NIR spectroscopy used in the horticulture industry involving numerous spectrophotometer designs and novel techniques to estimate light absorption in vegetables [34]. The paper looked at calibration factors between devices while focusing on the non-destructive measurements of fruit solids. Attempts to use NIR to determine fruit quality and make predictions on the harvest dates have been detailed [35]. Using NIR, the

authors are able to determine the internal parameters of the fruit, which would otherwise have to be destructively assessed in a laboratory. This information is then used to predict harvest dates based on the status of the fruit quality. Further readings, including [36], [37], [38], [39], [40], [41], [42] all discuss the use of NIR spectroscopy and its application as a means of non-destructive measurements for quality and identifications.

2.2. Robotic Systems

A review has been conducted on the six main types of robotic systems currently available [43]. All standard robotic systems are based on the following designs, with customised robotic systems being based on one or more of these designs.

Cartesian Robot: A Cartesian robot consists of three prismatic joints, whose axes follow a cartesian coordinate system (Figure 4). There are many applications for a robot of this type, and generally are used where a basic XYZ movement is required.

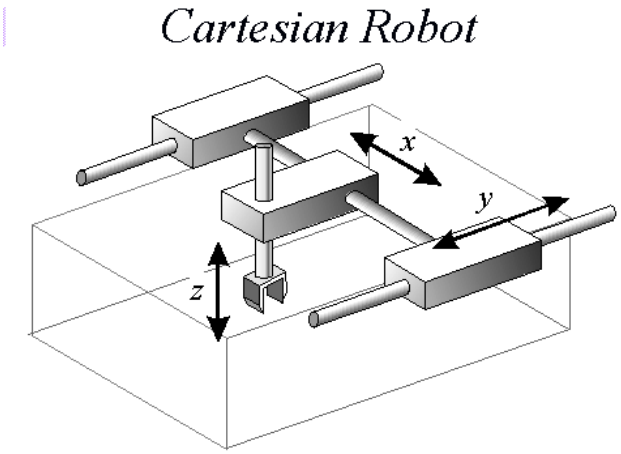


Figure 4: Cartesian robot design. (http://prime.jsc.nasa.gov/ROV/types.html)

Cylindrical Robot: Cylindrical robots are robotic systems whose workspace can be defined within a cylindrical system, often used on manufacturing lines, to move items between conveyer systems (Figure 5). Cylindrical robots are constructed using 2 prismatic joints and 1 revolute joint to provide movement in the X, Y and Z directions.

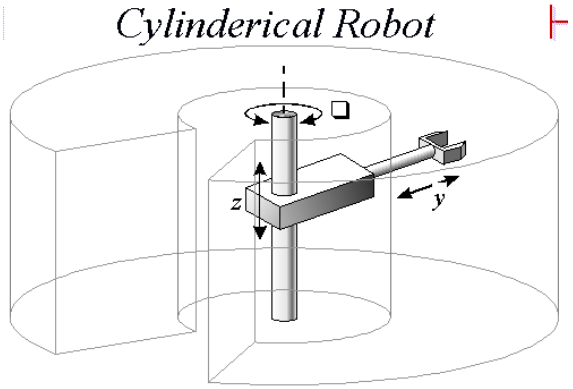


Figure 5: Cylindrical robot design. (http://prime.jsc.nasa.gov/ROV/types.html)

Spherical Robot: Spherical robots are similar in operation to the cylindrical robotic systems, however they have one prismatic joint and two revolute joints, as shown in Figure 6, giving more freedom in the z axis. These results in the workspace following polar coordinate systems giving it a 7 hemispherical like workspace.

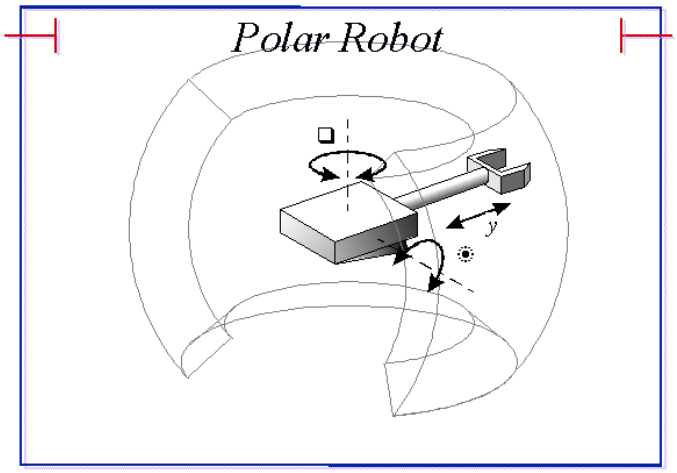


Figure 6: Polar robot design. (http://prime.jsc.nasa.gov/ROV/types.html)

SCARA (Selective Compliant Assembly Robot Arm or Selective Compliant Articulated Robot Arm)

In these systems movements compose three parallel-axis revolute joints, allowing them to move to any X-Y-Z coordinate within their working environment. Often a fourth axis of motion is available by means of a revolute joint at the end-effector (theta-Z). (Figure 7).

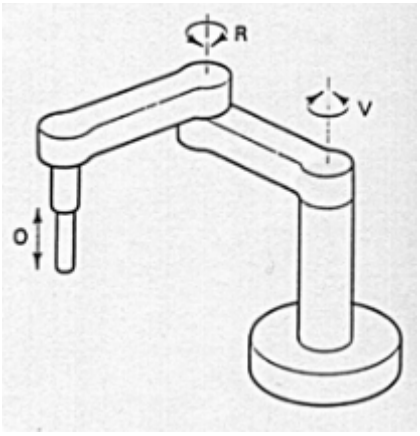


Figure 7: SCARA (Selective Complaint Assembly/Articulated Robotic Arm). (http://prime.jsc.nasa.gov/ROV/types.html)

Articulated Robot: Articulated robotic systems (Figure 8) use multiple rotary joints in a daisy chain arrangement to allow the end-effector to reach anywhere within the working environment. Articulated robots may have up to 7° of freedom and are often used to access difficult-to-reach areas. The end-effector may also have a revolute joint.

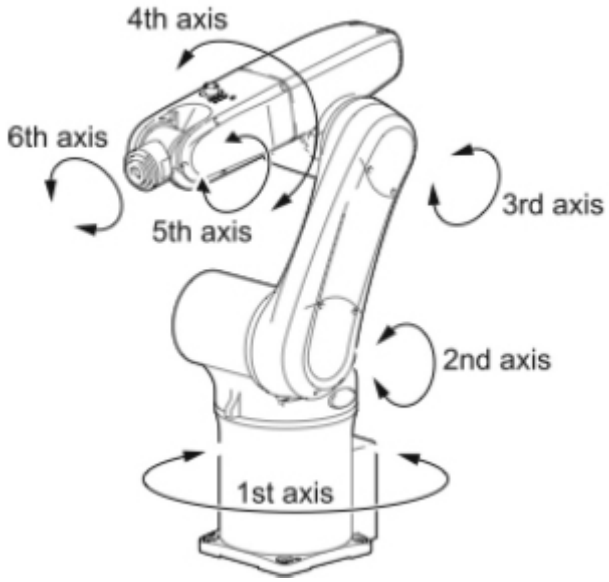


Figure 8: Articulated robot design. (http://www.processonline.com.au/articles/36410-Packaging-automation-trends-using-small-assembly-robots-in-upstream-packaging-processes)

Parallel Robot: A parallel robot uses two or more prismatic operated “legs” connected to the end-effector, to enable them to pick and place objects (Figure 9). Other uses are in simulators, whereby the end-effector is removed, connecting the legs directly to a base. Moving the legs independently provides motion on the top platform.

Figure 9: Parallel robot design. (http://iccrobotics.com/robot_parallel.html)

After evaluating different types of robotic systems currently available it was decided to design and build our own robotic system. Cost was a major factor with most commercially available robots costing $5,000 (for a very basic system) to $100,000+ for an advanced system. It also gave us the ability to customise the operation using custom software, as opposed to using proprietary software. The initial idea was to use a Cartesian robotic system mounted to the underside of a shelf to allow the end-effector to move anywhere within a tray (X and Y axis) with the ability to move the end-effector in the Z axis. However a Cartesian robot would only allow operation to be conducted on one shelf. To overcome this evaluations of different robotic arm configurations and types were undertaken with a modified SCARA/cylindrical robot (using the principle of a Cartesian robot) selected.

An anthropomorphic robotic arm and hand for use in the horticulture industry to grasp apples, with the capability of developing it further to allow for automated harvesting and placement operations has been reported [44]. The authors point out that the design of the robotic arm is critical due to the unstructured environment in which they are intended to work, necessitating a custom robotic arm to be designed. The design of a robotic system with a gripper for harvesting tomatoes has been described making use of a Programmable Logic Controller (PLC) and sensors to control the movement and picking of the tomato fruit [45].

A robotic harvester has been described for harvesting radicchio [46]. This system used a custom designed robotic arm with vision to identify the radicchio which enabled the custom designed end-effector to harvest it. The design and construction of a hybrid mechanism was described in [47] whereby the locomotion platform and manipulator arm were designed as one entity to support both locomotion and manipulation interchangeably. The design required a number of simulations to be conducted to ensure the custom design worked. Although not intended for the horticulture industry, the paper points out the requirements to design a robotic system noting the cost and efficiency in programming as opposed to buying an off-the-shelf product to modify.

Currently there is much research into control theory of robotic and automated systems. Detailed work outlining advancements in robotic control theory has been discussed and reported in both [48] and [49]. Robotic control systems can become very advanced, especially when dealing with multiple inputs and feedbacks. The control theory chosen for this project made use of inverse kinematics algorithms, using a basic feedback system as outlined by others [50], [51], [52].

2.3. Summary

Following discussions with Plant & Food Research, a clear idea was reached as to how to undertake this project. From the literature, machine-vision appeared to be a common and relatively straightforward method of measuring items in horticultural applications, including automatic weed control systems [3-6, 19], automatic fruit harvesting [10-11], and the quality control of fruit or vegetables on a production line [15-17]. Machine-vision has also been used to measure the colour of objects, to provide information on the health status of plants by linking colour to chlorophyll content. It has been used to assess the quality of fruit, vegetables and bakery products and in the health sector to determine blood glucose levels. Plant health status is able to be identified simply by viewing the colour of plant material with recent publications have also used NIR instead of visual colours as a means of health status [32-34, 40].

Following the literature review it was proposed to use a simple colour sensor to monitor the changes in plant material colour over time. Machine-vision was to

also be incorporated, however it would only be used as a means of identifying plant growth rates by capturing images over time, and through algorithms, determining plant growth. Colour sensors were used instead of captured images to determine the plant colour, as a number of problems had been reported where insufficient plant materials lighting caused false colour readings to be taken (Figure 34). RGB sensors were selected over other colour sensors with different colour spaces due to a number of papers presenting relationships between colour and chlorophyll content. The RGB sensor would be implemented to take colour readings of plant material over set time periods, with the results being used as a means of identifying the status of plants health. Any changes in plant colour would allow researchers to monitor and routinely adjust media levels in response to these changes.

A number of other sensors were also required. Research showed that images needed to be taken at fixed heights in order to make comparisons and analysis between images. These lead to a proximity sensor being required and in addition temperature, relative humidity and CO_2 sensors were included, although they were not key requirements.

In order to automate the process, a robotic arm needed to be constructed. Investigations into the different types of robotic systems currently on the market yielded some good ideas. The robotic arm needed to be custom built, allowing the end-effector housing the camera and colour sensor to move over multiple shelving units and over multiple trays on each of the shelves.

The basic concept of this project required an autonomous anthropomorphic robotic arm to be designed and fabricated, which contained a range of sensors including RGB colour, temperature and proximity sensors as well as a compact colour camera. Custom software created in Microsoft Visual Studio (VB.net) allowed for a completely automated operation with very little human input required. A systems engineering approach was employed taking the robotic system from a concept to a reality, making use of engineering techniques and integrating standard components to make the final product. The work detailed in this Thesis focuses on the design and development of an anthropomorphic

robotic arm for routinely monitoring the growth of plantlets and the immediate growing environment.

The specifications for the arm included the ability to move over multiple 600mm x 600mm trays of tissue on a shelving unit with a high resolution zoom camera and colour sensors mounted at the end of the arm. Images would be captured and plant leaf colour monitored during various growth stages.

3. Robotic Arm Design

A modified SCARA/cylindrical robot using the principle of a Cartesian robot was designed in the 3D CAD package SolidWorks (2010) using the knowledge and features of the six main types of robotics. Essentially, an anthropomorphic robotic arm, (meaning human-like), was designed using three revolute joints, in the X and Y axis, with one prismatic joint in the Z axis. This design would allow for the end-effector, which houses the colour sensor and camera, to move anywhere around the shelf of plant material (X and Y axis) and up and down over multiple shelves (Z axis).

3.1. Required Parts

A number of key mechanical parts were required to allow for a custom modified SCARA/cylindrical robot to be constructed; the use of a linear motion in the Z axis, bearings to allow for smooth operations of joint movements and a gearing system allowing for more accurate positioning of the various joints. Electromechanical components (motors) were also required; these were used to provide the various joint movements.

3.1.1. Z-axis Motion – (Linear Ball Shaft)

Because the arm was required to move up and down over multiple shelves in the Z axis, a number of options were available to give the arm vertical movement. Initial ideas included using a linear actuator however, linear actuators become relatively expensive once a vertical height of 200mm is exceeded. Research into what other current robotic systems use revealed the use of a linear ball screw and shaft assembly. A linear ball screw houses a number of recalculating bearings rotating around a linear shaft. When the shaft is being rotated and the screw is fixed, it results in the screw and its attachment being driven up or down (depending on the direction the shaft is rotating).

A number of manufacturer currently produce ball screw and shaft assemblies, all with different intended applications, sizes, accuracy and price. Initial research into SKF Bearings' line of ball shafts and screws yielded a number of

very accurate, yet very expensive assemblies. Further investigation revealed Industrial Automation Services Limited NZ (Industrial Automation Services Limited, Auckland, NZ) to stock Hiwin Ball Screw and shaft assemblies.

Specifications for the ball shaft used, a Hiwin 16-5T3FS, had a nominal diameter of 16mm, root diameter of 14.4mm, dynamic load of 664 kgf, static load of 1196 kgf, 0.07mm axial play and an accuracy of 0.1mm/300mm. The shaft has a 5mm lead which when combined with a microstep driver, moved the arm vertically by increments of less than one mm. This is essential to ensure all sensor readings are taken at “fixed heights”. To drive the shaft, initial designs used a stepper motor driving a 3:1 gear ratio, this gave the ball screw and shaft assembly even more resolution.

The ball nut, a Hiwin 16-5T3 standard lead ball nut, contains 3 internally rotating circuits containing approx 80 x 3.175mm ball bearings. Each of the three circuits allows the ball bearings to make one revolution around the screw spindle (of the shaft), before reticulating the bearings back via a return cap that moves the bearings under the adjacent ball track as shown in Figure 10.

Figure 10: Cut away of how a ball screw operates.
Source: Hiwin Data Sheet

Theoretically, movements as small as 0.0000651mm (or 65nm) are possible, this is achieved with a 25600 microstep, combined with a 3:1 gear ration and a 5mm lead. However in reality many factors would influence this, including accuracy of the microstep driver, slippage within the gears, accuracy of the ball shaft, and accuracy of proximity sensor to confirm minute movements.

3.1.2. Stepper Motors

The robotic system required a total of five stepper motors. Stepper motors convert electrical pulses into very precise and repeatable movements by dividing a full rotation into a large number of steps. Stepper motors were selected to provide the required movements over other types of motors and were selected due to their high torque, low cost and accuracy. As a result of having high accuracy, no encoders were required to measure motor position via a feedback interface.

A stepper motor contains a permanent magnet which rotates as coils energise in a sequence. Bipolar motors differ to Unipolar motors in that they require two power sources (or switchable polarity power source). Although their operation is simpler, they require more complex circuitry. Further information on the stepper motors used can be found in section: 3.2.4.

3.1.3. Bearings

Bearings are used to provide the various joints with smooth operation reducing friction that may occur. Bearings have been used for each of the moving joints, and have been used to support and guide the ball shaft to provide smooth operation when running on the linear shaft (design 2 only). Three types of bearings were used; deep groove roller bearings for the various joint movements, a thrust bearing to take the axial load of the ball shaft and a linear bearing running along the linear shaft used in design 3 (providing additional support).

Each of the bearings were sourced from SKF Bearings (SKF Bearings, Palmerston North, NZ) (except the linear bearing which was sourced from Linear Motion) and were sourced once an initial design had been completed. The bearings used in the first design have been retained and used for the remaining designs.

3.1.4. Gearing System

A gearing system was initially integrated into the system to provide each joint with more accuracy and to move the movement of the joint of the axis away

from the motor. When using a gear ratio, a smaller movement of a joint is possible as well as providing an increase in torque.

3.1.5. Prototypes

All purchased products, including stepper motors, bearings and gears were first drawn as 3D models, dimensioned as per the data sheets (or, alternatively downloaded from respective websites). This allowed ideas to be designed and test fitted within Solidworks.

Once an idea of how the robotic system would function (Figure 11), a number of designs were drawn up:

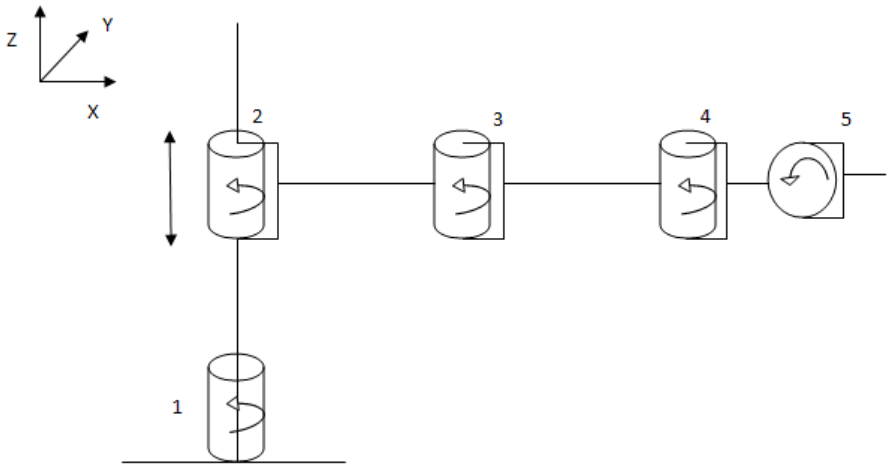


Figure 11: Concept drawing of robot, showing joint 1 to be prismatic, joint 2, 3, 4 and 5 to be revolute.

The aluminium fabricated robotic arm went through a number of design phases, with each design an improvement over previous designs. Design 1 followed the initial concept (Figure 11), but with the intention of allowing each of the robotic arms to be extended, which would allow each arm length to be changed with ease. A sleeve approach was taken to allow this to happen so that each motor and related machined part fitted into the sleeve (Figure 12). To make the movements more accurate, a gearing system was investigated. Design 1 made use of spur gears to provide the various gear ratios (a 3:1 gear ratio for motor 1, and a 2:1 gear ratio for motors 2 – 4), sourced from Ronson Gears (Ronson Gears, Melbourne, Australia). Design 1 made use of 3 MOT-122 stepper motors for motors 1 – 3 and 2 MOT-122 for motors 4 and 5.

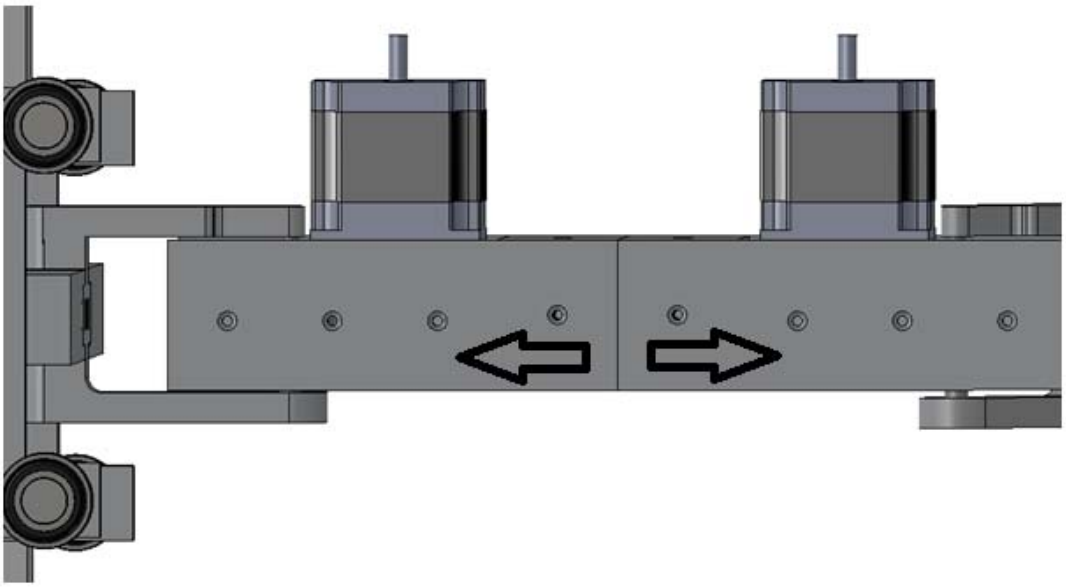


Figure 12: Design 1, showing how the arm was able to extended (or retract) using a sleeve approach.

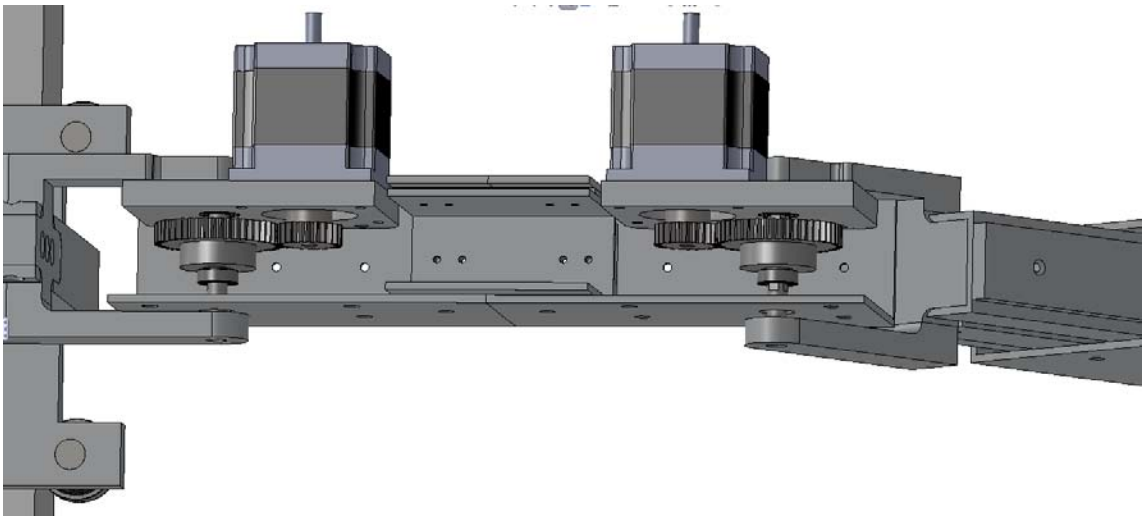


Figure 13: Cut away of deign 1, showing gearing system used.

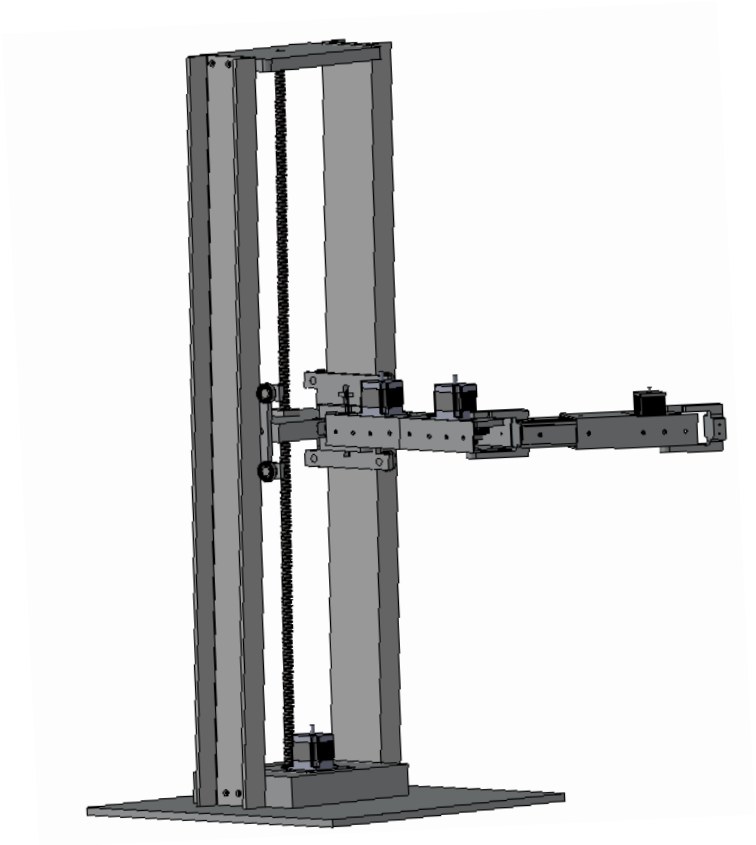


Figure 14: Design 1 complete (less the pan and tilt system).

Having machined and constructed the first design (Figure 14), it was noted that the arm was extremely heavy, and did not provide the necessary joint arc movements as expected. Excessive aluminium was required to be removed to reduce the weight, while a new joint connection was required to provide the required movements. With a new joint connection designed, there was little need to allow each arm to have the ability to extend (or retract), since the arm could now reach any desirable position with the extended joint angle movements (Figure 15). A redesign using the same parts was completed in Solidworks, utilising Finite Element Analysis (FEA) (This ensured the required modifications would still allow the arm to have sufficient strength) refer Figure.

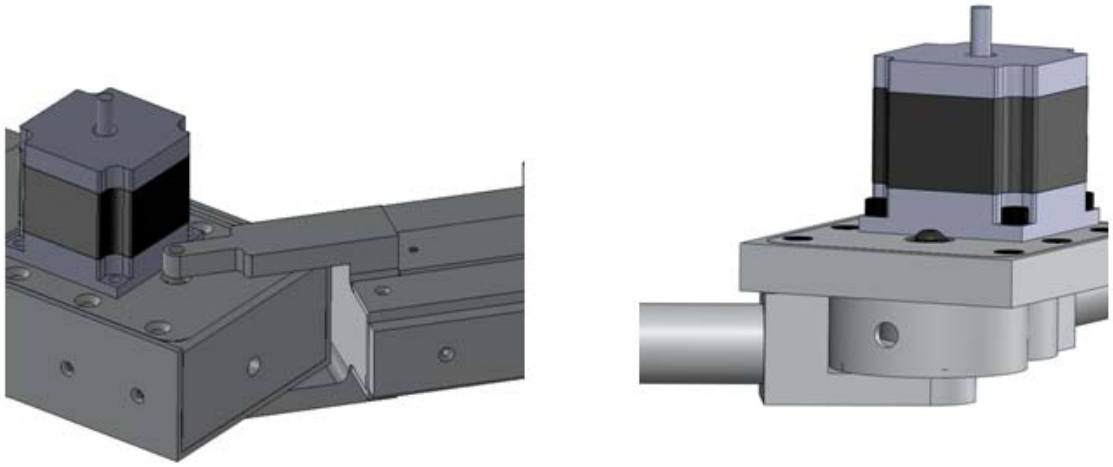


Figure 15: Screenshot of original joint (left) and the redesigned joint (right) allowing for more freedom.

Figure 16: Solidworks Images of redesigned parts to reduce excess weight.

Once excessive aluminium was removed, a new joint connection designed and the removal of the sleeve approach for arm extensions, Design 2 was complete (Figure 17). Testing was then undertaken with this a much lighter arm.

Figure 17: Complete robotic arm (design 2).

However when testing, it was noted that in homing the robotic arm, there were problems with the gears having slight movement (or backlash) which occurred between the teeth as they mated (Figure 18). Although initially not a significant issue this would later lead to problems with homing the robot in the X-direction (refer page 98). The use of a 3:1 gear ratio to drive the ball shaft up and down also provided some homing issues when homing the Z-axis. Although providing vary accurate movements, homing the robot in the Z-axis became almost impossible, since the system made such minute movements that were smaller than the slippage (or backlash) between the gears.

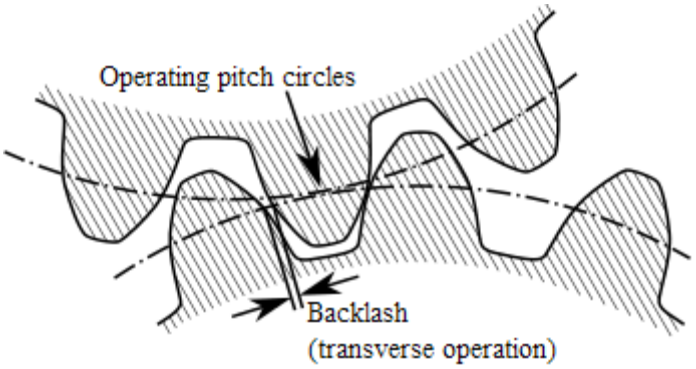


Figure 18: Gear Backlash example.
(http://en.wikipedia.org/wiki/Backlash_%28engineering%29)

To overcome the issue of slippage in the gearing system, a belt drive system was proposed. A belt drive system allows power to be transformed from one source to another with the use of a pulley and belt set up. Although this would not entirely eliminate slippage, it would reduce it sufficiently to overcome homing problems. The decision was made to make a third design, making the system even lighter, and incorporating belt drive systems. Smaller motors (MOT-110) were purchased for motors 4 and 5, with motors 2 and 3 being replaced with the MOT – 120. A 1:1 gear ratio was incorporated into motor 1, replacing the 3:1 gear previously used (an MOT-122 motor remained). To make the system even lighter, a number of aluminium parts were replaced with light-weight acrylic. Acrylic was used as it is extremely light weight, yet durable which made the robotic system more aesthetically pleasing.

Figure 19: Image of Final design 4.

All components were fabricated in-house using lathes and mills, with more complex and precise components being machined on an Acerman 500 3-Axis Computer Numerical Control (CNC) machine. After a number of design phases, with input from Massey University machining technicians, the final Anthropomorphic Robotic Arm was designed.

3.1.6. Machining

Raw aluminium was sourced, in various sizes, cut down to manageable sizes, and clamped into the Acerman CNC machine (Figure 20). Setting up the CNC machine first required homing the machine, followed by setting the tool to the appropriate datum (as per the Solidworks dimensioned drawing). G-Code generated through SolidCam was sent to the machine to provide instructions and directions for the appropriate tool selection and movement.

Figure 20: CNC machine in use machining a component out.

Each fabricated part machined on the CNC machine required G-code. This was generated using SolidCam software (as shown in Figure 21) with code based on the SolidWorks drawing and dimensions. The CNC machine was then able to cut very precise (to approx 0.001mm) cuts and profiles using a range of cutting tools.

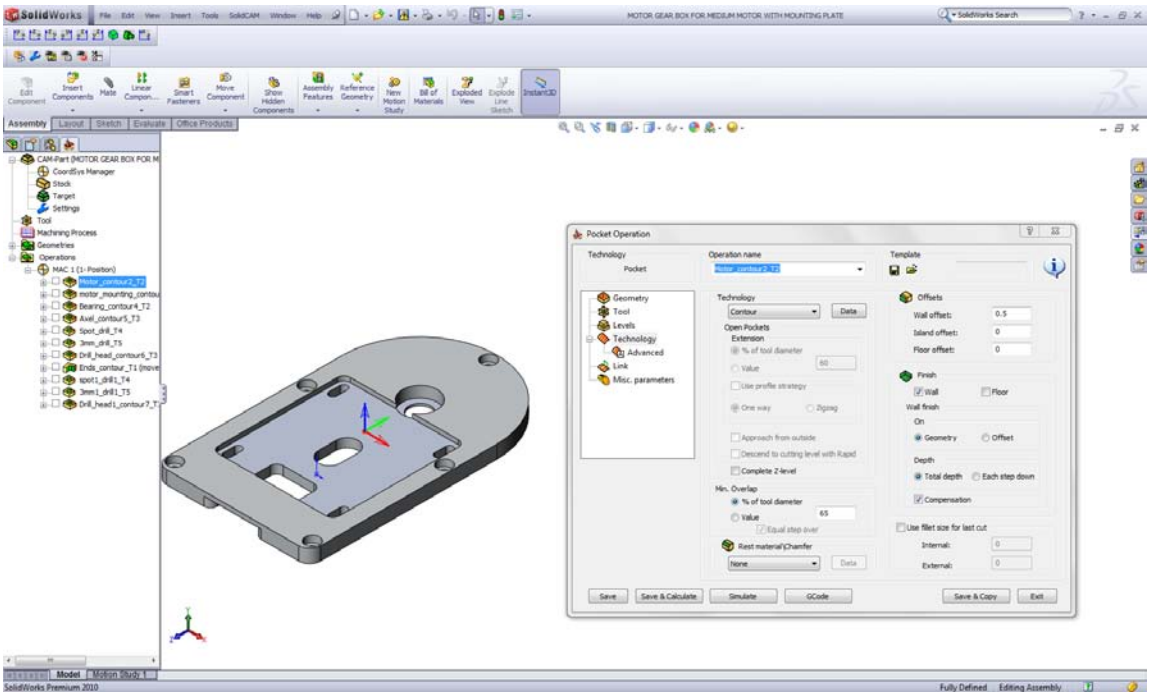
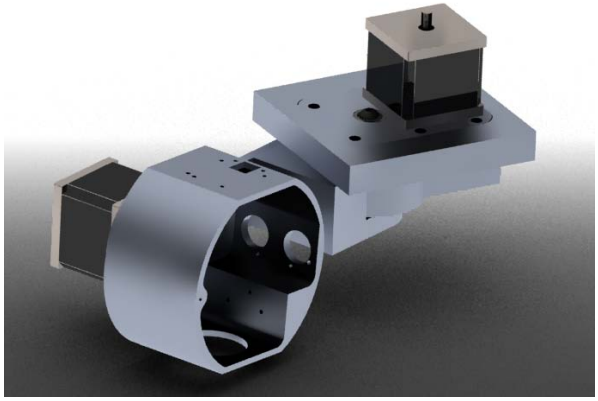
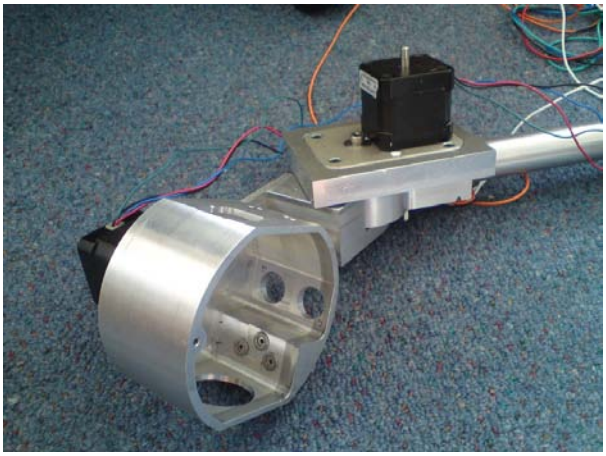


Figure 21: Screenshot of SolidCam interface, setting up a profile cut.





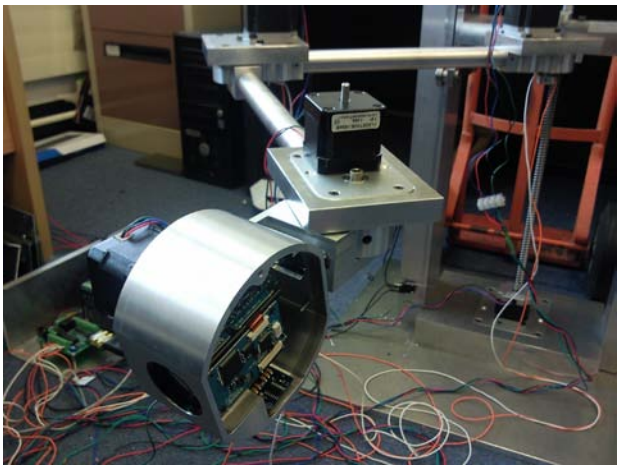


Figure 22: Pan& tilt system in Solidworks (top), after machining (middle) and with camera and sensors (bottom) – Design 2.

Once complete each part was identical to that of the SolidWorks dimensioned drawing (Figure 22). A dimensioned part drawing can be seen on page 129 with the G code generated for that part generated in SolidCam shown on page 130

3.2. Automation & Control

The following section describes the automation components used in this project, with detailed overview of how each of the electronic components are used and integrated into the system.

3.2.1. Overview

To allow movement of the robotic arm, a number of electronic components were required. Stepper motors were selected to provide the numerous joint movements. A motor controller was required to interface these motors to a computer allowing for custom software to be developed to control the stepper motors. Much research was conducted trying to source the appropriate motors, and controllers. Ocean Controls (Ocean Controls, Vic, Aus) provided all the required components at a reasonable price.

3.2.2. Motor Controller

Figure 23: Ocean Control KTA-190 Serial Stepper Motor Controller. (www.oceancontrols.com.au)

The KTA-190 serial stepper motor controller (Figure 23) from Ocean Controls provides an easy interface for controlling bipolar stepper motors from a personal computer. With inexpensive USB to serial converters it is able to be controlled through a USB port (alternatively a KTA-205 parallel port interface can be used for a parallel connection). The KTA-190 allows for a total of four stepper motors to be controlled independently (or collectively) off the one board, with the ability to connect up to four serial stepper motors controller boards together allowing the user individual control of a up to16 motors. 2 KTA-190 motor controllers were required to provide communication to the 5 motors.

The KTA-190 controller requires a 12V DC power source and has an ATMega 168 microcontroller on-board, which is connected via a MAX232 chip (ensuring the incoming signals are in the microcontroller range of 0 – 5V). Limit switch inputs have pull up resistors, with step and direction outputs connected through a 74HC245 buffer chip. Dip switch SW1 identifies which board is being addressed when more than one serial stepper board is in use. The controller

has a 9 pin female D-connector, with a 10 pin IDC header onboard giving an interface for additional controllers to be connected.

Custom software (alternatively serial terminal software) is used to send commands to the controller board, via a serial port (or USB port when using a USB to serial adaptor). A board rate of 9600, 8 data bits, 1 start, 1 stop bit and no parity (9600, 8, N, 1) was required for communication. ASCII commands are sent to the controller board in the format:

@AA CMND XXXXCR

Where: @AA is the address of the motor required to be controlled (between 01 and 16)
CMND is the command required to be executed (refer Table 1)
XXXX is the parameters of the CMND (refer Table 1)
CR is the carriage return bite (0x0D)

Alternatively, a single command can be sent to the desired controller board to control all four motors simultaneously using the following format:

@AA CMND WWWW XXXX YYYY ZZZZCR

Where: @AA is the address of the controller board (01, 05, 09, 13)
CMND is the command required to be executed (refer Table 1)
WWWW is the parameters of motor 1 (refer Table 1)
XXXX is the parameters of motor 2 (refer Table 1)
YYYY is the parameters of motor 3 (refer Table 1)
ZZZZ is the parameters of motor 4 (refer Table 1)
CR is the carriage return bite (0x0D)

Command (CMND)	Description
POSN	Set the position that motor AA is currently at to be XXXX where XXXX is between -99,999,999 and 99,999,999
PSTT	Returns the position of motor AA
AMOV	Move motor AA to the absolute position XXXX where XXXX is between -99,999,999 and 99,999,999
RMOV	Move motor AA relatively from the current position by XXXX where XXXX is between -99,999,999 and 99,999,999
STOP	Stop motor AA immediately
STAT	Get the status of the motors see “Status Command Detail”
ACCN	Set the maximum stepping rate of motor AA to XXXX where XXXX is between 0 and 9999 see ”Acceleration”. If the value for ACCN is 0 or less than RATE then no acceleration or deceleration occurs
ACCI	Set the Acceleration interval of motor AA to XXXX where XXXX is between 1 and 9999 see ”Acceleration”
RATE	Set the minimum stepping rate of motor AA to XXXX where XXXX is between 1 and 9999 see ”Acceleration”
DRON	Turn Direction output (D1-D4) associated with AA on for XXXX * 0.131mS. If XXXX is -1 output will be on until a ‘DROF’ command is received
DROF	Turn Direction output (D1-D4) associated with AA off immediately from ‘DRON’ command
DRST	Returns time remaining of ‘DRON’ command in time intervals, or -1 if it is on permanently
OPTN	Bit 1 of the parameter turns Verbose mode on and off, bit 2 turns Checksum mode on and off. Valid addresses are 01, 05, 09 and 13. See the section on the Option Command
ACCF	When set to 1 the alternate Acceleration Curve is used, when set to 0 the standard Acceleration Curve is used.
SAVE	Saves RATE, ACCI, ACCN, ACCF and OPTN parameters to EEPROM, which are then automatically loaded on the next power up. Valid addresses are 01, 05, 09 and 13.
REL1	Turns Relay 1 ON when parameter is 1, Turns Relay 1 OFF for all other parameters. Valid addresses are 01, 05, 09 and 13.
REL2	Turns Relay 2 ON when parameter is 1, Turns Relay 2 OFF for all other parameters. Valid addresses are 01, 05, 09 and 13.

Table 1: Commands and Parameters for KTA-190 Stepper Motor Controller.

KTA-190 controller has a status command (which is sent in the same way as any other command), and responds with an integer, which when converted to 12 bit binary, providing the real-time status of the controller board being investigated.

@AA STAT

MSB	11	10	9	8	7	6	5	4	3	2	LSB
L4	L3	L2	L1	D4	D3	D2	D1	M4	M3	M2	M1

Table 2: Representation of the binary number returned from the status command.

Where: M represents the movement of the motor (1 = moving, 0 = stopped)

D represents the Direction of the motor (1 = forward, 0 = reverse)

L represents the status of the limit switch (1 = closed, 0 = open)

In the event a binary number is less than 12 digits, 0 are added to the beginning. i.e. 1001110 = 000001001110.

Each motor connected to the KTA-190 has the ability to have a limit switch input, to ensure the motors do not travel out of range. Since the limit switches have a pull up resistor they only require a ground connection to activate, thus ensuring they will work even if power is cut to the entire system. Each motor can have multiple limit switches, provided they are connected in parallel. When a switch is activated, the corresponding motor will stop, and only move one step at a time until the switch is deactivated. This ensures no damage is done to the application and allows the motors to be homed.

The ability to control the speed and acceleration of any stepper motor is important. The KTA-190 has the ability to control both, by manipulating the acceleration settings by sending the appropriate command. The acceleration and final speed is determined by the commands ACCN, ACCI, and RATE. The motor starts stepping one step every ACCN interval, decreasing by the ACCI interval every step until RATE, with the opposite occurring when decelerating. By default the controller has the following settings: ACCN = 50, ACCI = 2, and RATE = 10.

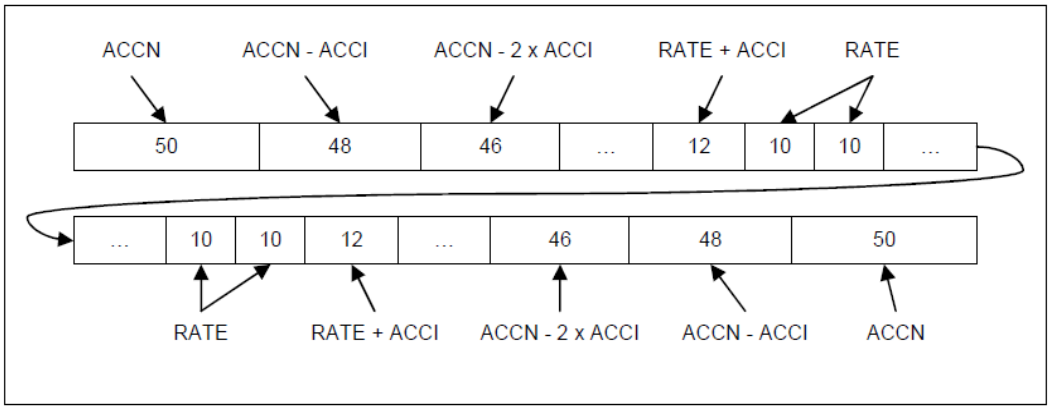


Figure 24: KTA-190 Accleration example.
Source: www.oceancontrols.com.au

3.2.3. Microstep Driver

A microstepper driver is required to interface the stepper motor to the motor controller, giving the user the ability to increase the motors steps per revolution, resulting in a finer resolution. The Ocean Controls M542 microstepper driver uses advanced technologies to microstep NEMA (National Electrical Manufacturers' Association) 17 to 34 sized bipolar, 2-phase or 4-phase stepper motors.

The M542 Microstepper (Figure 25) works by using an advanced bipolar constant – current chopping technique, giving it the ability to output more speed and torque from the same motor, when compared to other equivalent drivers. The M542 provides microsteps from 2 to 128 (refer Table 3) and allows current settings of between 1.00 and 4.20 A (peak) by adjusting dip switch (SW1 – SW8).

Both a pulse signal and direction signal are required for connecting a 4 wire stepper motor to the driver, with speed and torque depending on winding inductance. Alternatively, a 6 wire motor can be connected, making use of an enabled signal.

An M542 driver is required for each stepper motor being used providing the power to each stepper motor. The driver itself requires either a regulated or unregulated input voltage of between 20V and 50V. An unregulated power supply is recommended, as it can withstand current surges and generally produce higher current output. Multiple drivers can be connected to one power

source in parallel provided the power source has sufficient current output. In general, the current drawn by the stepper motors determines the torque (particularly at lower speeds) while the supply voltage determines the high speed performance.

Figure 25: Ocean Controls M542 microstepper driver. (www.oceancontrols.com.au)

MicroStep	Steps/rev	SW5	SW6	SW7	SW8
2	400	OFF	ON	ON	ON
4	800	ON	OFF	ON	ON
8	1600	OFF	OFF	ON	ON
16	3200	ON	ON	OFF	ON
32	6400	OFF	ON	OFF	ON
64	12800	ON	OFF	OFF	ON
128	25600	OFF	OFF	OFF	ON
5	1000	ON	ON	ON	OFF
10	2000	OFF	ON	ON	OFF
20	4000	ON	OFF	ON	OFF
25	5000	OFF	OFF	ON	OFF
40	8000	ON	ON	OFF	OFF
50	10000	OFF	ON	OFF	OFF
100	20000	ON	OFF	OFF	OFF
125	25000	OFF	OFF	OFF	OFF

Table 3: M542 Microstep resolutions and configuration.

3.2.4. Stepper Motors

Currently there are a variety of different motors available for different applications in robotics, with most being controllable by a PC via an interface card of some description. Stepper motors were selected over other conventional

motors, including DC (direct current) motors and servo motors as they do not require an encoder or feedback system to identify where the motors are located. Stepper motors produce a much high torque (both holding and moving torque) than DC and servo motors, for these reasons stepper motors were selected.

Different types of stepper motors exist for different applications. In this study the stepper motors were bipolar and sourced from Ocean Controls. OneFL57STH56-2804B (MOT-122) high torque steeper motors had a 1.8° step angle and a holding torque of 12.6kg/cm (Figure 26). This was combined with a further twoFL42STH38-1684B (MOT-120) stepper motors, with a 1.8°step angle, and holding torques of 3.6 kg/cm were used. Two FL35ST36-0504B (MOT-110) stepper motors were used for the pan and tilt system, these MOT-110 motors had a holding torque of 1.0kg/cm and weighed just 180 grams. All motors are compatible with the microstepper drivers, but the maximum current setting simply needs to be adjusted accordingly.

After design 3, motor 1 was with a 1:1 gear ratio to provide the necessary movements and accuracy to drive the entire system up and down (Z axis) via a Hiwin ball shaft and screw assembly. Motors 2 and 3 utilised a 2:1 gear ratio to provide movement in the X and Y plane while motors 4 and 5 (both MOT-110) provide movement of the pan and tilt system, both utilising a 1:1 gear ratio.

Figure 26: Ocean Controls MOT-122 hybrid stepper motor. (www.oceancontrols.com.au)

The motor controller, microstep drivers and the motors wiring diagram is shown in Figure 27.

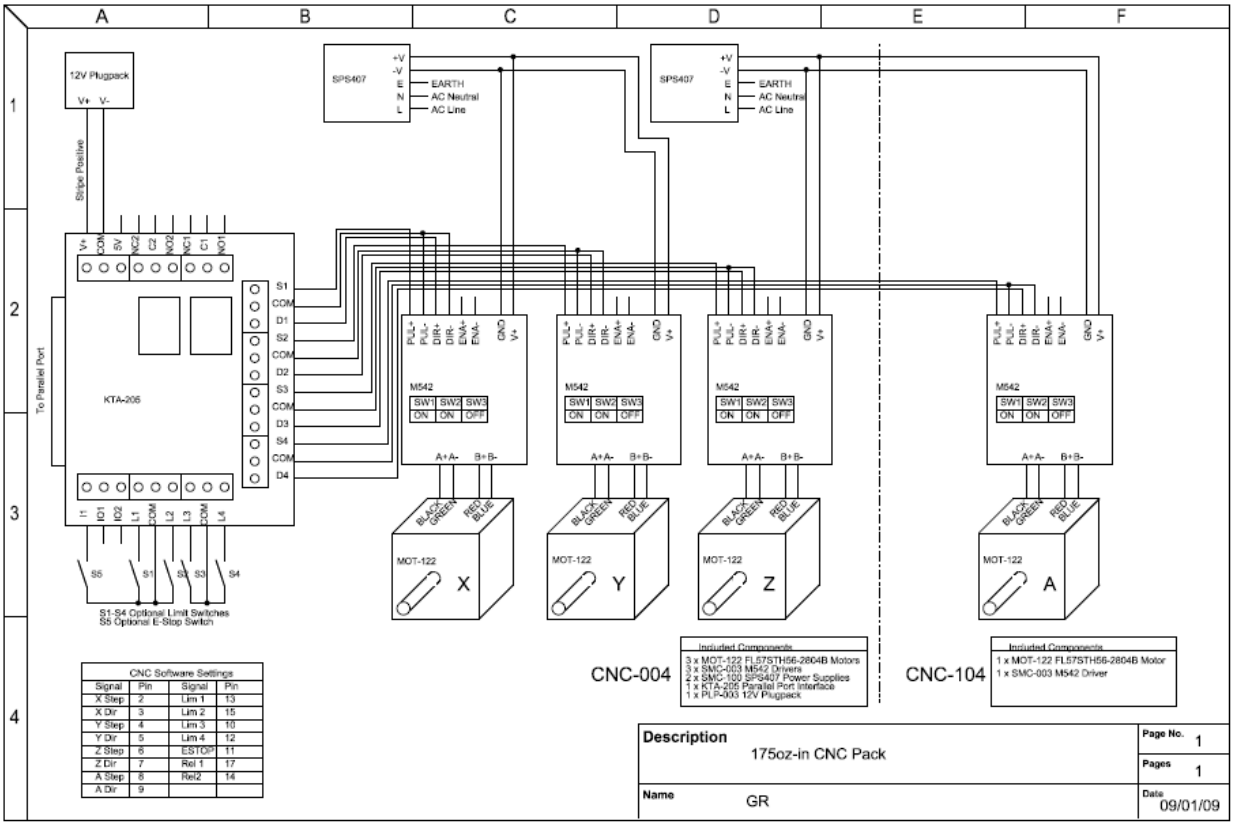


Figure 27: Ocean Controls CNC Wiring diagram. (www.oceancontrols.com.au)

3.2.5. Control Theory (Inverse Kinematics)

Background: Inverse kinematics is the process used to determine joint angles of a robotic system so the end-effector reaches a known point in space. It is the complete opposite to forward kinematics in which the robotic joint angles are known but not the end-effector position. Humans undertake inverse kinematics every day. For example – if we want to pick up a glass of water, our brain calculates the required angles of our arm joints with ease. Solving inverse kinematics for robotic systems requires a number of unknown equations to be solved.

Inverse kinematics is used since each of the plant trays are in a known position from the robotic system. The user enters in the details of where the end-effector is required to be and the software implements algorithms to run through the mathematics to solve the joint angles.

This solution follows an algebraic solution to solving Inverse Kinematics of a three-link planar manipulator as outlined in the book: Introduction to Robotics : Mechanics & Control by John J. Craig [53] with guidance from [49], [54], [48], [55].

We have four non-linear equations that are required to be solved for θ_1 θ_2 θ_3

$$\cos_{\phi} = \cos_{123} \tag{3.1}$$

$$\sin_{\phi} = \sin_{123} \tag{3.2}$$

$$x = l_1 \cos\theta_1 + l_2 \cos\theta_{12} \tag{3.3}$$

$$y = l_1 \sin\theta_1 + l_2 \sin\theta_{12} \tag{3.4}$$

By squaring and adding equations (3.3) and (3.4) we obtain

$$x^2 + y^2 = l_1^{\,2} + l_2^{\,2} + 2l_1 l_2 \cos\theta_2 \tag{3.5}$$

Making use of equations:

$$\cos\theta_{12} = \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2 \tag{3.6}$$

$$\sin\theta_{12} = \cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2 \tag{3.7}$$

Solving equation(3.5) for cos θ_2

$$\cos_2 = \frac{x^2 + y^2 - l_1^{\,2} - l_2^{\,2}}{2l_1 l_2} \tag{3.8}$$

To ensure the solution or desired position actually exists the right hand side of the equation must lie between -1 & 1

Assuming the answer lies in between -1 & 1 then we solve for sin θ_2

$$s^2 = \pm\sqrt{1 - \cos\theta_2^{\,2}} \tag{3.9}$$

The ± corresponds to the multiple solutions in which the elbow can be “up” or “down”

Finally we solve for $\theta_2$using the two-argument arctangent routine.

$$\theta_2 = A\tan 2(s_2, c_2) \tag{3.10}$$

Having solved θ_2 we can solve θ_1 using equations(3.1) and(3.2) in the form of:

$$x = k_1 \cos\theta_1 - k_2 \sin\theta_1 \tag{3.11}$$

$$y = k_1 \sin\theta_1 - k_2 \cos\theta_1 \tag{3.12}$$

Where

$$k_1 = l_1 + l_2 \cos\theta_2 \tag{3.13}$$

$$k_2 = l_2 \sin\theta_2 \tag{3.14}$$

To solve an equation of this form, a change of variables is required, essentially changing the way we write the constants k_1 and k_2

If:

$$r = +\sqrt{{k_1}^2 + {k_2}^2} \tag{3.15}$$

And

$$\gamma = A\tan 2(k_2, k_1) \tag{3.16}$$

Then

$$k_1 = r\cos\gamma \tag{3.17}$$

$$k_2 = r\sin\theta_\gamma \tag{3.18}$$

Equations (3.11)&(3.12) can be written as:

$$\frac{x}{r} = \cos\gamma\cos\theta_1 - \sin\gamma\sin\theta_1 \tag{3.19}$$

$$\frac{y}{r} = \cos\gamma\cos\theta_1 - \sin\gamma\sin\theta_1 \tag{3.20}$$

Or

$$\cos(\gamma + \theta_1) = \frac{x}{r} \tag{3.21}$$

$$\sin(\gamma + \theta_1) = \frac{y}{r} \tag{3.22}$$

Using the two-argument arctangent we get:

$$\gamma + \theta_1 = A\tan 2(\frac{y}{r}, \frac{x}{r}) = A\tan 2(y, x) \tag{3.23}$$

And so

$$\theta_1 = A\tan 2(y, x) - A\tan 2(k_1, k_2) \tag{3.24}$$

We can finally solve for θ_1 using:

$$\theta_1 + \theta_2 + \theta_3 = A\tan 2(\sin_\phi, \cos_\phi) = \phi \tag{3.25}$$

Solving the three unknown angles, to allow the end-effector to reach a desired point in space, algorithms then determine where to move the end-effector based on the number of readings required to be taken. The Z axis is simply calculated by determining the shelf number being used.

Shelf 1 = 100 mm

Shelf 2 = 400 mm

Shelf 3 = 700 mm

Shelf 4 = 1000 mm

4. Sensors

This section discusses the sensors used in the project, along with some background as to why the particular sensors were selected. The sensors' operations are presented along with the results from testing and calibration of the sensors.

4.1. Colour

To monitor the colour of plant material, and in particular determine the RGB colour content of plant material, a colour sensor that can interpret the colour of an object was required to be implemented

4.1.1. Background to Colour

Visible colour is the portion of the electromagnetic spectrum which the human eye can detect, with the electromagnetic spectrum being the range of all electromagnetic frequencies known to man, ranging in frequency from 1 KHz to 2.4 x 10^23 KHz (Figure 28).

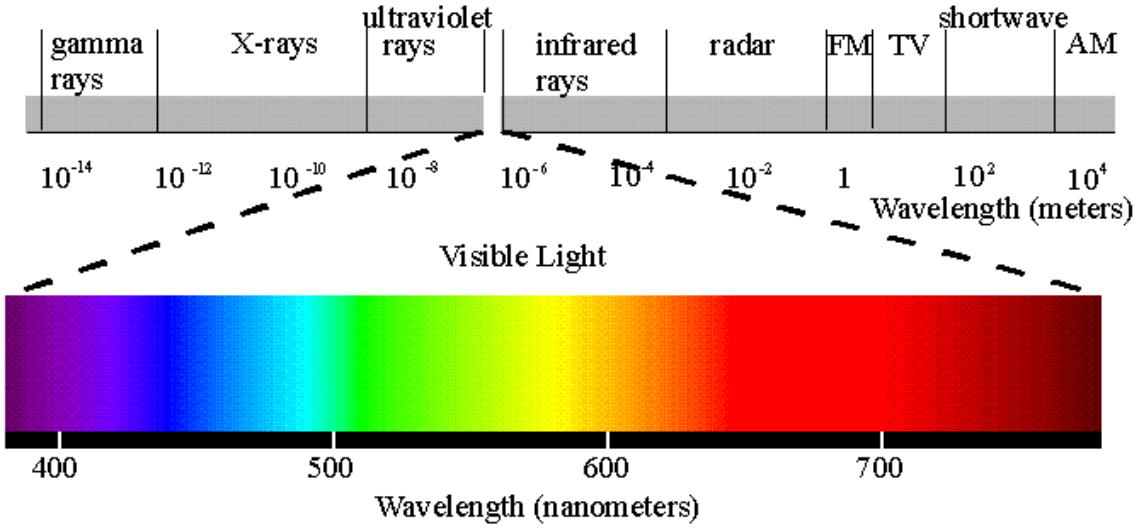


Figure 28: Representation of the Electromagnetic spectrum. (http://www.yorku.ca/eyes/spectu.htm)

The wavelength of a wave is inversely proportional to the frequency, $\lambda = v/f$; this generally implies that as the frequency increases the wavelength decreases (provided the wave is travelling through the same media). The human eye can only identify a small portion of the electromagnetic spectrum, known as the visible light spectrum, ranging from blue (380nm - 400nm - shortest

wavelength), through green (middle wavelength) to red (760nm - 780nm - longest wavelength), with all other colours being made up of a combination of these red, green and blue wavelengths. This leads us to the common colour space of RGB; RGB produces a cubic representation of all the possible colours.

Coloured objects that exhibit a particular colour, as seen by humans, absorb certain wavelengths and reflect others back to the observer. Therefore a red object would absorb all the green and blue wavelengths, but reflect back only the red wavelengths, thus giving its red appearance to the human. The range of colours produced by the primary lights of a RGB space is called its “garmut”.

As well as RGB there are a number of other colour spaces currently used around the world for different applications. These include CYMK (cyan, magenta, yellow and key (black)) – used as a subtractive colour model (as opposed to RGB which uses an additive colour space), HSV (Hue, Saturation, Luminance) also known as HSB (Hue, Saturation, Brightness) which represents colour as a cylindrical representation (Figure 29).

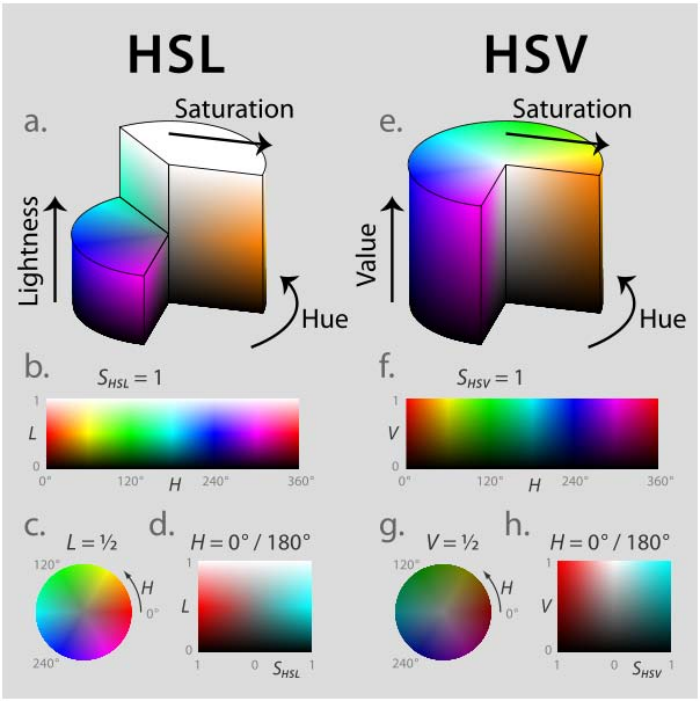


Figure 29: HSL and HSV graphical representation. (http://en.wikipedia.org/wiki/File:Hsl-hsv_models.svg)

Where hue represents the angle around the centre vertical axis, saturation represents the distance from the centre axis while the distance along the axis represents the luminance (or brightness).

The colour space, XYZ is another commonly used colour space, The Commission Internationale de l'éclairage (French for International Commission on Illumination and often abbreviated as CIE) developed the CIE XYZ colour space in 1931, which was one of the first mathematically defined colour spaces, resulting in the 1931 CIE XY chromaticity diagram (Figure 30). This was the first international standard way of defining colour, and is still often referred to as the "gold standard" (http://www.colorbasics.com/CIESystem/). Often spectrophotometers will produce colour readings in a range of colour spaces. It should be noted that when two colour spaces have different gamuts, there will be colours that one space can produce and the other cannot.

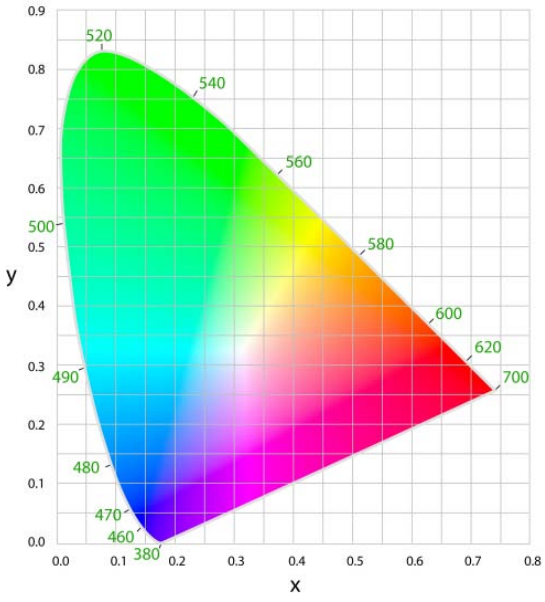


Figure 30: 1931 XY Chromaticity diagram. (http://www.search.com/reference/CIE_1931_color_space)

4.1.2. Why an RGB Colour Space was Used

Research outlined in the literature review has shown that there is a correlation between colour and the chlorophyll content of plant leaf material, with details of how a computer was used to take an image, captured via a camera, to obtain

the RGB content of each pixel. This process made use of the RGB colour space; it therefore made sense to also use the RGB colour space (instead of CYMK, HSV/HSB, XYZ).

To monitor the colour of plant leaf material, and in particular determine the RGB colour content of plant material a basic RGB colour sensor was required.

Currently there are a number of colour sensors available globally from numerous suppliers. Initial investigations revealed colour sensors ranged dramatically in price, size, sophistication and the colour space they used. RGB colour space was the preferred colour space to use. A number of sensors were unpractical for this project due to their cost and size restrictions. TAOS (Texas Advanced Optoelectronic Solutions) have a number of colour sensors available, including the TAOS TCS3404CS which outputs colour using the chromaticity and luminance (intensity) of ambient light. TAOS also produce a number of RGB colour sensors, including the TAOS TSL13T light to voltage chip and the TAOS TCS3200D light to frequency chip. These sensors were preferred as they output colour using the RGB colour space, cost only a few dollars and are extremely compact. Programming the chips required the chips to be mounted on a PCB (printed circuit board), and connected to a microcontroller. Interface to the microcontroller is achieved using an I^2C interface.

However, because of the size of the chips, it was impracticable to mount the chips on a PCB. This lead to further investigations of a simpler solution, fortunately Parallax (Parallax Inc, CA, USA) produce two colour sensors that utilise the TAOS TSL13T and the TAOS TCS3200D chips, pre-mounted onto a board that connects directly to a Parallax Basic Stamp Microcontroller. The Parallax ColorPAL and Parallax TCS3200, gave fast readings of the RGB content of an object, with the ColorPAL (Figure 31) using a light-to-voltage sensor while the TSC3200 (Figure 32) using a light-to-frequency sensor. Both the ColorPAL and TCS 3200 colour sensors are provided with source code making them amenable to integrating into our customised system.

4.1.3. ColorPal Colour Sensor

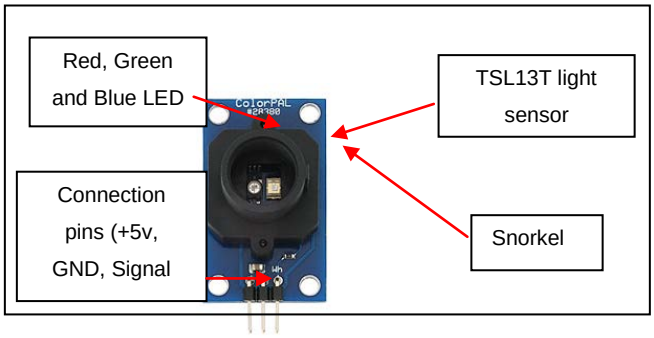


Figure 31: Parallax ColorPAL RGB Sensor. (www.parallax.com)

The ColourPAL sensor illuminates a sample using an inbuilt red, green and blue LED light source (one colour at a time) and records the quantity of light reflected back from the object. The ColorPal makes use of a TAOS TSL13Tlight to voltage chip. When light is reflected back, the voltage level, which is proportional to the light reflected, is used to determine the sample's R, G and B colour content. The ColorPAL requires the sample to be illuminated using each of the red, green and blue LEDs, with a 'snorkel' to shield possible interfering light. This requires the ColourPAL to be in direct content with the object for an optimum reading with minimal interference.

Key Features:

- Detects a full range of colours, and outputs data as RGB (Red/Green/Blue) components.
- Detects broad-spectrum ambient light with sensitivity down to 44µW/cm2 per lsb.
- Generates 24-bit colour using onboard RGB LED.
- Plugs into servo headers (with optional cable) or wireless breadboards.
- Single-pin interface uses a simple serial protocol to define and initiate colour detection and generation.
- Colour detection and generation details handled by onboard microcontroller.
- Onboard EEPROM for saving custom colour detection and generation programs.

- Autorun feature permits running a pre-designated EEPROM program with only a power supply.

4.1.4. TSC3200 Colour Sensors

Figure 32: Parallax TCS3200-DB RGB colour sensor.

The TCS3200 Colour sensor makes use of a TAOS TCS3200 RGB light-to-frequency chip. The TCS3200 colour sensor operates by illuminating the object with two white LEDs, while an array of photo detectors (each with a red, green, blue and clear filter located evenly throughout the array to eliminate any bias) interpret the colour being reflected by means of a square wave output whose frequency is proportional to the light reflected. The TSC3200 Colour sensor has a 5.6mm lens, which is positioned to allow an area approx 3.5mm^2 to be viewed.

The TCS3200 accounts for any light interference, (from surrounding ambient light) by gating the two LEDs on and off, allowing for ambient light subtractions to be calculated.

Key Features:

- High-Resolution conversion of light intensity to frequency.
- Programmable colour and full-scale output frequency.
- Power down feature.
- Communicates directly to microcontroller.
- Single-supply operation (2.7v to 5.5v).
- Stable 200 ppm/deg C temperature coefficient.

- Low-profile surface mount package.
- RoHS compliant.

Feature or Capability	ColorPAL	TCS3200-DB
Price	★★★★★	★★★
Color detection accuracy with normal reflective subjects	★★★★	★★★★★
Color detection accuracy with fluorescent (e.g. Day-Glo, Astrobrite) reflective subjects	★★	★★★★★
Color detection on very glossy surfaces	★★★	★★★★★
Color detection of radiant subjects (e.g. LEDs, CRTs)	Not possible	★★★★★
Color sensor output	Serial I/O	Frequency
Pins needed for interface (including +5V and ground)	3	6
Color detection resolution (per RGB component)	Up to 8 bits, using onboard 10-bit ADC	8 bits or more, limited only by sample time
Compatibility with Parallax's PC color-matching software	★★★★★	★★★★★
Accurate color detection requires ambient light correction	Automatic	Yes
Accurate color detection requires white balance	Yes	Yes
Accurate color detection requires black balance	Maybe	No
Color sensing averaged over spot diameter of:	0.47″ (12mm)	0.14″ (3.5mm)
Color generation	★★★★★	Not possible
On-board programmable sensing and generation sequences	★★★★★	None

Table 4: Parallax comparison between ColourPAL and TCS3200 RGB colour sensor – (the more stars the better) (www.parallax.com)

4.1.5. Selection and Calibration of Colour Sensors

To ensure the Parallax colour sensors gave accurate readings, a number of tests were required to be completed. Firstly, it was determined which of the two sensors were best suited to the application, followed by calibrating the best sensor to a known and theoretical true value.

Initial testing was required to be conducted on the Parallax TCS3200 to determine at which height the sensor produced the largest quantity of light. To complete this, a USB4000 spectrometer (Ocean Optics Inc, FL, USA) was used to find the height at which the greatest intensity of light occurred when the RGB sensor was placed above a sample. As the two white LEDs are directed down at an angle, there is a point where the light intensity was the greatest. This position is 20mm above the surface of the sample, as shown in Figure 33.

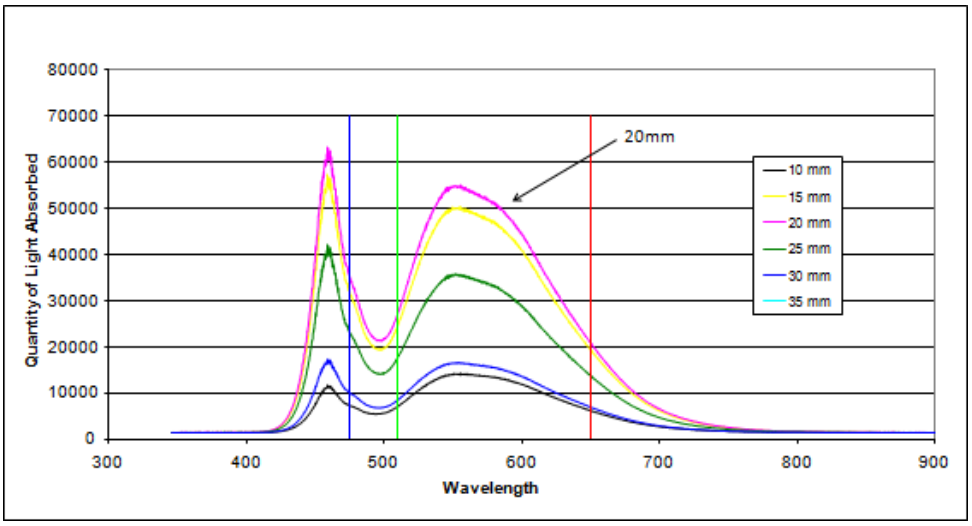


Figure 33: Light absorbed from TCS3200 across the white LED light spectrum when the sensor is positioned at 6 different heights. Results obtained using a USB4000 Spectrometer.

Since the TCS3200 is mounted 20 mm above the sample, and therefore not in direct contact with the sample, it was more suited for our application than the full contact required by the ColorPAL sensor.

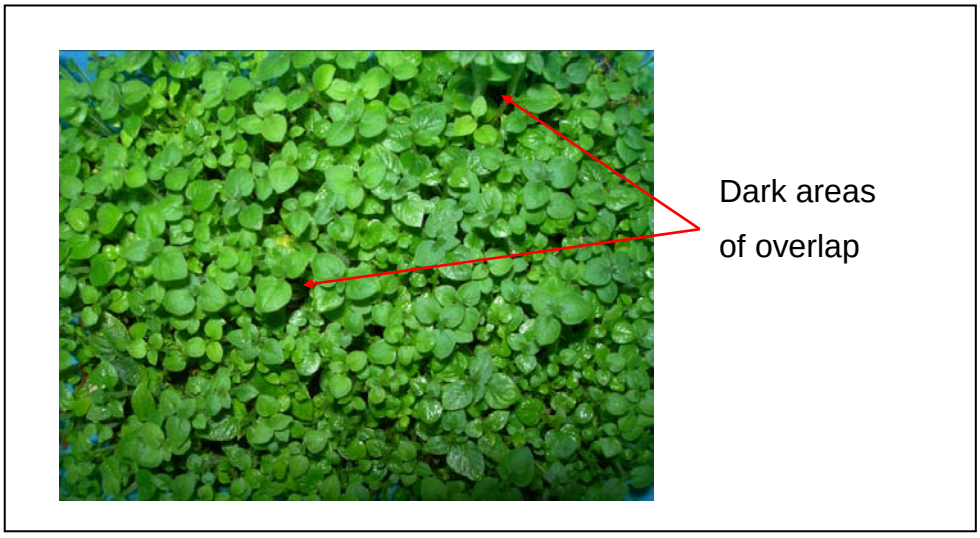


Figure 34: Captured image showing areas of ”darkness” caused by overlapping which would lead to false colour readings.

An alternate method of determining plant leaf colour is to use an image captured by a camera and through software, determine the colour of individual pixels. However, because plant leaves can overlap (Figure 34), shadows are created leading to false colour readings from the image. Since the TCS3200 colour sensor uses its own light source to illuminate the sample surface, it eliminates any potential shadowing.

A Konica Minolta CM-700D Spectrophotometer (Konica Minolta Sensing Americas, Inc, NJ, USA) was used to validate and calibrate the RGB sensors. For accurate measurements, CM-700D was firstly calibrated by taking both “white readings” by sampling a supplied white object and “black readings” by sampling a supplied black object.

Since the CM-700D produced colour values using the XYZ colour space, and as the RGB colour sensors from Parallax need to be calibrated to the CM-700D, a colour conversion matrix was required.

The CM-700D gives colour in the XYZ colour space, as well as L*a*b*, L*C*h, Hunter Lab, Yxy and Munsell. A linear transformation matrix was required to transform data from the XYZ colour space to the RGB colour space for comparisons with the Parallax sensor.

The 700D uses the Konica Minolta software – SpectraMagic NX (Figure 35), which interfaces to the calorimeter, providing the user with the ability to select colour spaces, with further information pertaining to the colour by means of graphical images.

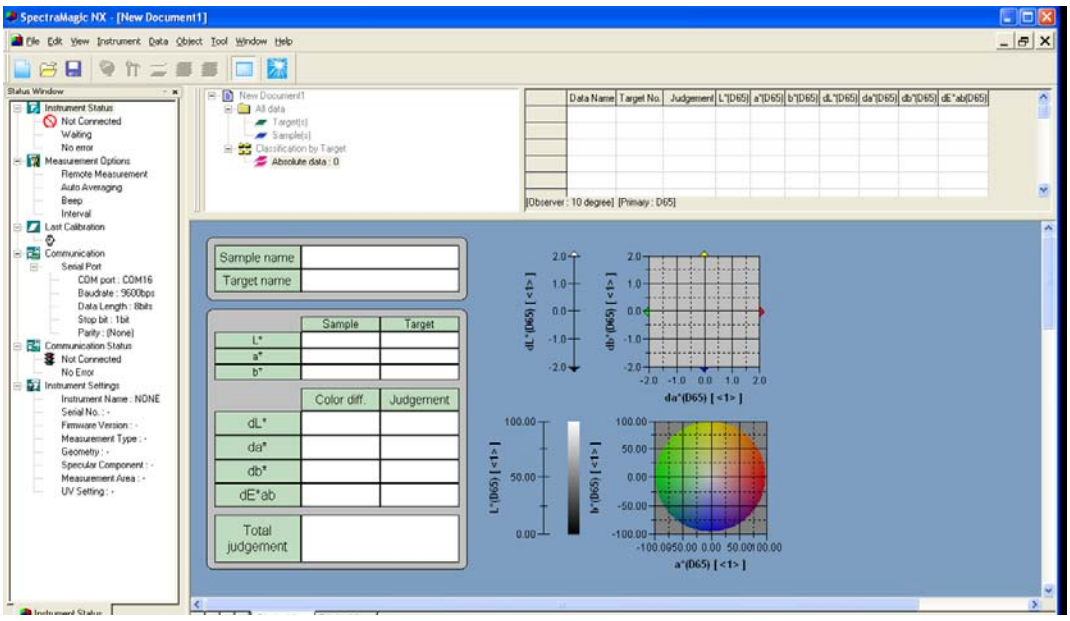


Figure 35: SpectraMagic NX software interface.

The linear transformation equations used to convert the XYZ data to RGB are:

$$x = \frac{X}{X + Y + Z} \tag{4.1}$$

$$y = \frac{Y}{X + Y + Z} \tag{4.2}$$

$$z = \frac{Z}{X + Y + Z} \tag{4.3}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{4.4}$$

The equations (4.1)-(4.4) combined with the standard 1931 xy chromaticity diagram (Figure 30) provided the foundation for the linear transformation. This transformation converted the XYZ data to an RGB colour space, with the chromaticity values of x, y and z shown in

Table 5 being standard [56].

Converting to XYZ to sRGB

Colour	x	y	z
Red	0.64	0.33	0.212656
Green	0.30	0.60	0.715158
Blue	0.15	0.06	0.072186

Table 5: X, Y and Z chromaticity values of red, green and blue for an XYZ to sRGB conversion.

From the x, y and z chromaticity values, the transformation matrix, M, is calculated(4.5) to solve equation(4.4).

$$M \approx \begin{pmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9503041 \end{pmatrix} \tag{4.5}$$

To calculate the R, G and B values the inverse is taken(4.6).

$$M^{-1} \approx \begin{pmatrix} 3.2404542 & -1.5371385 & -0.4985314 \\ -0.9692660 & 1.8760108 & 0.0415560 \\ 0.0556434 & -0.2040259 & 1.0572252 \end{pmatrix} \tag{4.6}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = M^{-1} \times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{4.7}$$

In order to validate the TCS3200 colour sensor, it was necessary to calibrate it to the CM-700D. To calibrate the TCS3200 a number of samples had to be interpreted by both the sensors. This involved taking 200 RGB colour readings with the TCS3200 and averaging for these different coloured samples. The same samples were then measured, 20 times each with the CM-700D, and averaged. These tests were all completed in a constant temperature dark room. As the CM-700D uses the XYZ colour space, the linear transformation matrix was required to convert the XYZ values to RGB ((4.1) - (4.7)). For full calculations refer to page 133.

The TCS3200 was first calibrated through software by modifying the integration time to allow a white object to have a RGB value as close as possible to 255,255,255.

In order to calculate a calibration factor the following equation was used:

$$R_N^{'} = R_N^{\gamma} \tag{4.8}$$

where:

$R_N^{'} = \text{CM} - 700\text{D}\ (\text{desired RGB value})$

$R_N = \text{TCS3200 RGB}\ (\text{Un} - \text{calibrated sensor data})$

$\gamma = \text{Gamma}\ (\text{required calibration factor})$

First the sensor data was scaled to ensure all values are offset so that the white reading is exactly 255 for each of R, G and B. This was required, since although the integration time had been modified, each of the R, G and B components were not exactly 255.

$$R_N = R \times \frac{255}{R_{\max}} \tag{4.9}$$

$$G_N = G \times \frac{255}{G_{\max}} \tag{4.10}$$

$$B_N = B \times \frac{255}{B_{\max}} \tag{4.11}$$

Where $R_{\max}, G_{\max}, B_{\max}$ represent the maximum R, G and B value of a white object.

The calibration factors (γ) for each colour are calculated using normalized data.

$$\gamma_R = \frac{\log(R_N^{'} / 255)}{\log(R_N / 255)} \tag{4.12}$$

$$\gamma_G = \frac{\log(G_N^{'} / 255)}{\log(G_N / 255)} \tag{4.13}$$

$$\gamma_B = \frac{\log(B_N^{'} / 255)}{\log(B_N / 255)} \tag{4.14}$$

For each colour sample measured, the calibration factor was calculated and averaged using a geometric mean (as opposed to the more general arithmetic mean function [57]), thus providing the calibration factor for R, G and B individually. The (desired) calibrated values were then obtained using equation(4.8). This gamma calibration transformation matrix converts the XYZ to sRGB.

$$R_{N(calibrated)}^{'} = (R_N / 255)^{\gamma} \times 255 \tag{4.15}$$

For a range of seven colours, measurements were carried out using the TCS3200 RGB sensor and the CM-700D Spectrophotometer. The gamma calibration factors calculated are:

$$(\text{Red})\ \gamma_{\text{R}} = 0.88,\ (\text{Green})\ \gamma_{\text{G}} = 0.46,\ (\text{Blue})\ \gamma_{\text{B}} = 0.68$$

Table 6 summarises the raw data for the for un-calibrated and calibrated RGB sensor data compared to CM-700D spectrophotometer. Table 7 summarises the average error, error percentage and the standard deviation with a graphical representation of Table 6 shown in Figure 36 for red, green and blue.

ID	TCS-3200						CMD-700			Output		
	Gain Adjusted			White Adjusted			RGB Equivalent (sRGB)			Calibrated Result		
	R	G	B	R	G	B	R	G	B	R	G	B
Red	172	45	41	160	45	39	232	91	77	170	114	71
Green	101	128	62	94	127	59	14	197	76	106	184	94
Light Blue	137	180	211	128	179	200	156	217	214	139	216	216
Light Green	180	197	161	168	196	153	195	231	171	177	225	181
Dark Blue	42	107	169	39	106	160	1	168	200	49	170	186
White	266	251	259	248	250	246	248	250	246	249	253	249
Black	20	20	22	19	19	21	59	58	55	26	77	48

Table 6: Results obtained comparing the TCS3200 colour sensor (calibrated and uncalibrated) to the CM-700D over a range of 7 diverse colours.

	TCS3200 (un calibrated)			TCS3200 (calibrated)		
	R	G	B	R	G	B
Error	39.756	41.16	20.174	38.799	9.328	8.693
Error %	15.591	16.141	7.911	15.215	3.658	3.409
S.D	26.195	21.771	13.035	31.136	8.899	5.744

Table 7: Average Error (0-255), percentage error and standard deviation for red, green and blue measurements of the TCS3200 colour sensor, calibrated and uncalibrated, when compared to CM-700D results across a range of colours.

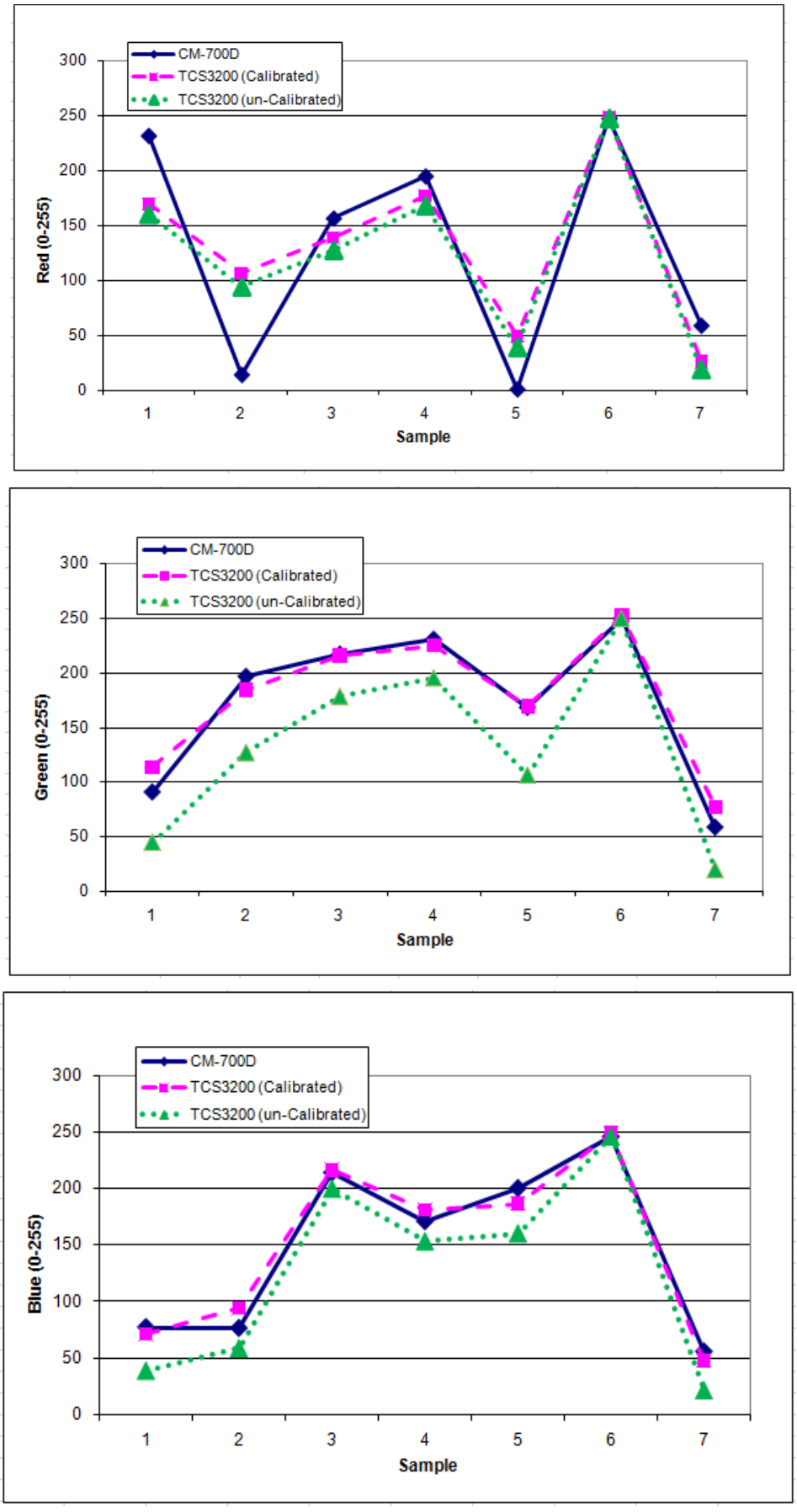


Figure 36: TCS3200 sensor RGB readings, calibrated and un-calibrated, compared to the CM-700D readings of: Red (A); Green (B); Blue (C).

As the colour sensor is intended to measure the colour of plant leaves, there is no requirement to calibrate it across the full range of colours. The sensor was therefore calibrated for a range of green – yellow colours only. 15 RHS (Royal Horticulture Society, London, UK) colour charts, designed for growers to identifying horticulture colours, were used. The measured data is shown in Table 8.

Two hundred colour readings were taken by the TCS3200 colour sensor and averaged, followed by 20 readings by the CM-700D (again averaged) for each colour chart.

The gamma calibration factors calculated for the 15 RHS colours are:

$$(\text{Red})\,\lambda_{\text{R}} = 0.50,\ (\text{Green})\,\lambda_{\text{G}} = 0.38,\ (\text{Blue})\,\lambda_{\text{B}} = 0.59$$

Table 8 summarises the average error, error percentage and the standard deviation for un-calibrated and calibrated RGB sensor data compared to CM-700D spectrophotometer outputs for the 15 colours.

Table 9 summarises the average error, percentage error and standard deviation for these 15 colours with graphs of the raw data contained in Figure 37.

ID	TCS-3200						CMD-700			Output		
	Gain Adjusted			White Adjusted			RGB Equivalent (sRGB)			Calibrated Result		
	R	G	B	R	G	B	R	G	B	R	G	B
123A	99	148	167	99	150	148	138	208	194	160	206	188
127C	38	79	75	38	80	66	55	166	134	99	160	120
129C	99	152	137	99	154	121	143	217	173	160	208	168
131C	25	41	35	25	42	31	45	120	84	81	123	79
133C	62	88	85	62	89	75	122	166	144	127	167	129
135C	42	51	35	42	52	31	102	156	97	104	134	79
137C	42	51	35	39	49	27	115	130	85	101	132	68
139C	68	82	58	68	83	51	141	162	108	133	163	104
141C	57	80	45	57	81	40	129	164	87	121	161	91
143C	71	88	48	71	89	42	149	166	81	135	167	94
145C	171	168	122	171	171	108	224	224	151	209	217	158
147C	84	86	62	84	87	55	170	154	112	147	166	108
149C	174	183	114	174	186	101	221	237	134	211	225	152
155D	255	249	258	255	253	228	255	253	228	255	254	240
202A	17	17	20	17	17	18	62	62	61	67	86	58

Table 8: **Results obtained comparing the TCS3200 colour sensor (calibrated and uncalibrated) to the CM-700D over a range of 15 colours using a sRGB transformation matrix.**

	TCS3200 (un calibrated)			TCS3200 (calibrated)		
	R	G	B	R	G	B
Error	11.4	9.343	13.072	10.289	6.117	5.683
Error %	4.47	3.664	5.126	4.035	2.399	2.229
S.D	11.854	7.529	9.149	6.562	4.739	3.357

Table 9: **Average Error (0-255), percentage error and standard deviation for red, green and blue measurements of the TCS3200 colour sensor, calibrated and uncalibrated, when compared to CM-700D results across a range of 15 RHS colour chart samples using sRGB transformation).**

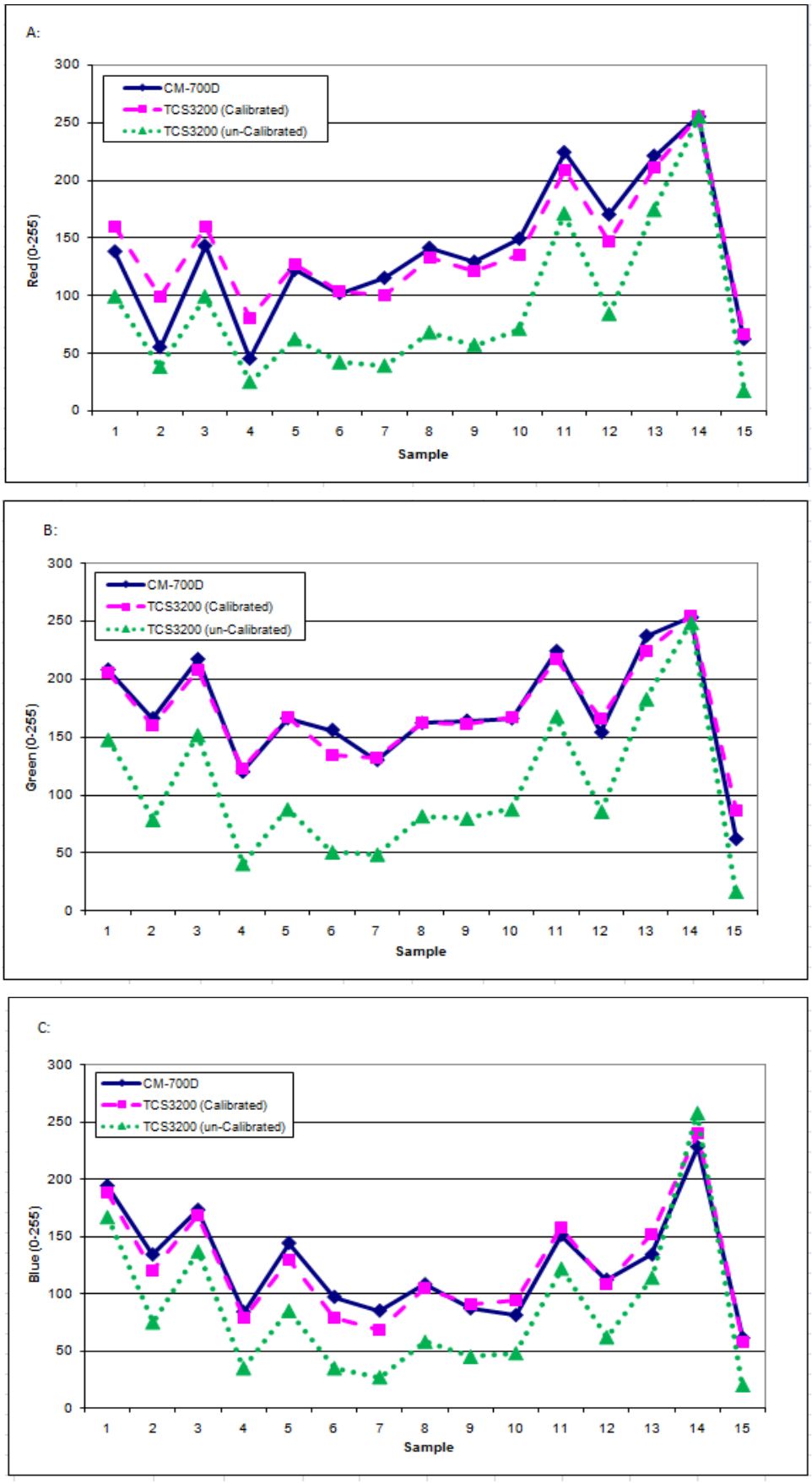


Figure 37: TCS3200 sensor RGB readings, calibrated and uncalibrated, compared to the CM-700D readings of: Red (A); Green (B); Blue (C) for a range of 15 RHS green/yellow colours using an sRGB transformation matrix.

TCS3200 (uncalibrated)	TCS3200 (calibrated)	CM-700D Spectrophotometer
RGB = 57,81,40	RGB = 121,161,91	RGB = 129,164,87
		

Table 10: Visual results showing the RGB colour interpreted by the CM-700D and TCS3200 colour sensor, before and after calibration (RHS 141C).

Further investigation into colour transformation revealed the standard RGB colour space used for transformations was to sRGB. sRGB is a colour space derived by HP (Hewlett Packard) and Microsoft as a standard for producing colour on monitors, printers and the internet. However, converting colour to a more suitable RGB colour space, in particular CIE RGB, would yield more accurate results. Investigations and testing proved this to be the case.

Although calculating a gamma value on the calculated sRGB values to determine a calibration factor would (in theory) remove any discrepancy between different RGB colour spaces, results showed the calibration factor improved when calculating a gamma calibration factor for the CIE RGB. The CIE RGB colour space uses a different transformation matrix (equations (4.16) and(4.17)) with different x, y and z chromaticity values Table 11.

Colour	x	y	z
Red	0.7350	0.2650	0.176204
Green	0.2740	0.7170	0.812985
Blue	0.1670	0.0090	0.010811

Table 11: X, Y and Z chromaticity values of red, green and blue for an XYZ to CIE RGB conversion.

$$M \approx \begin{pmatrix} 0.4887180 & 0.3106803 & 0.2006017 \\ 0.1762044 & 0.8129847 & 0.0108109 \\ 0.0000000 & 0.0102048 & 0.9897952 \end{pmatrix} \tag{4.16}$$

$$M^{-1} \approx \begin{pmatrix} 2.3706743 & -0.9000405 & -0.4706338 \\ -0.5138850 & 1.4253036 & 0.0885814 \\ 0.0052982 & 0.0052982 & 1.0093968 \end{pmatrix} \tag{4.17}$$

CIE RGB calibration factors

$$(\text{Red})\,\lambda_R = 1.05,\ (\text{Green})\,\lambda_G = 0.92,\ (\text{Blue})\,\lambda_B = 1.00$$

Table 12 shows the results obtained when using the CIE RGB transformation matrix, with Figure 38 showing the results in a graph format. Table 13 displays the error, percentage error and standard deviation for the results shown in Table 12.

ID	TCS-3200						CMD-700			Output		
	Gain Adjusted			White Adjusted			RGB Equivalent (CIE RGB)			Calibrated Result		
	R	G	B	R	G	B	R	G	B	R	G	B
123A	99	148	167	88	152	144	62	155	152	85	158	143
127C	38	79	75	34	81	65	17	89	69	31	89	64
129C	99	152	137	88	156	118	71	166	123	85	162	117
131C	25	41	35	22	42	30	10	43	27	20	49	29
133C	62	88	85	55	90	73	47	93	80	52	98	72
135C	42	51	35	37	52	30	36	78	39	35	60	29
137C	42	51	35	37	52	30	40	54	30	35	60	29
139C	68	82	58	61	84	50	63	88	48	57	92	49
141C	57	80	45	51	82	39	55	87	35	48	90	38
143C	71	88	48	63	90	41	72	91	32	60	98	41
145C	171	168	122	152	172	105	169	185	101	149	178	104
147C	84	86	62	75	88	53	84	91	51	71	96	53
149C	174	183	114	155	187	98	170	206	86	152	192	97
155D	255	249	258	227	255	222	227	255	222	226	255	222
202A	17	17	20	15	17	17	10	13	13	14	22	17

Table 12: Results obtained comparing the TCS3200 colour sensor (calibrated and uncalibrated) to the CM-700D over a range of 15 colours using a CIE RGB transformation matrix.

	TCS3200 (un calibrated)			TCS3200 (calibrated)		
Error	9.691	6.806	5.107	10.161	6.162	4.966
Error %	3.800	2.669	2.003	3.985	2.416	1.947
S.D	7.423	7.298	3.485	6.631	4.757	3.699

Table 13: Average Error (0-255), percentage error and standard deviation for red, green and blue measurements of the TCS3200 colour sensor, calibrated and uncalibrated, when compared to CM-700D results across a range of 15 RHS colour chart samples using CIE RGB transformation.

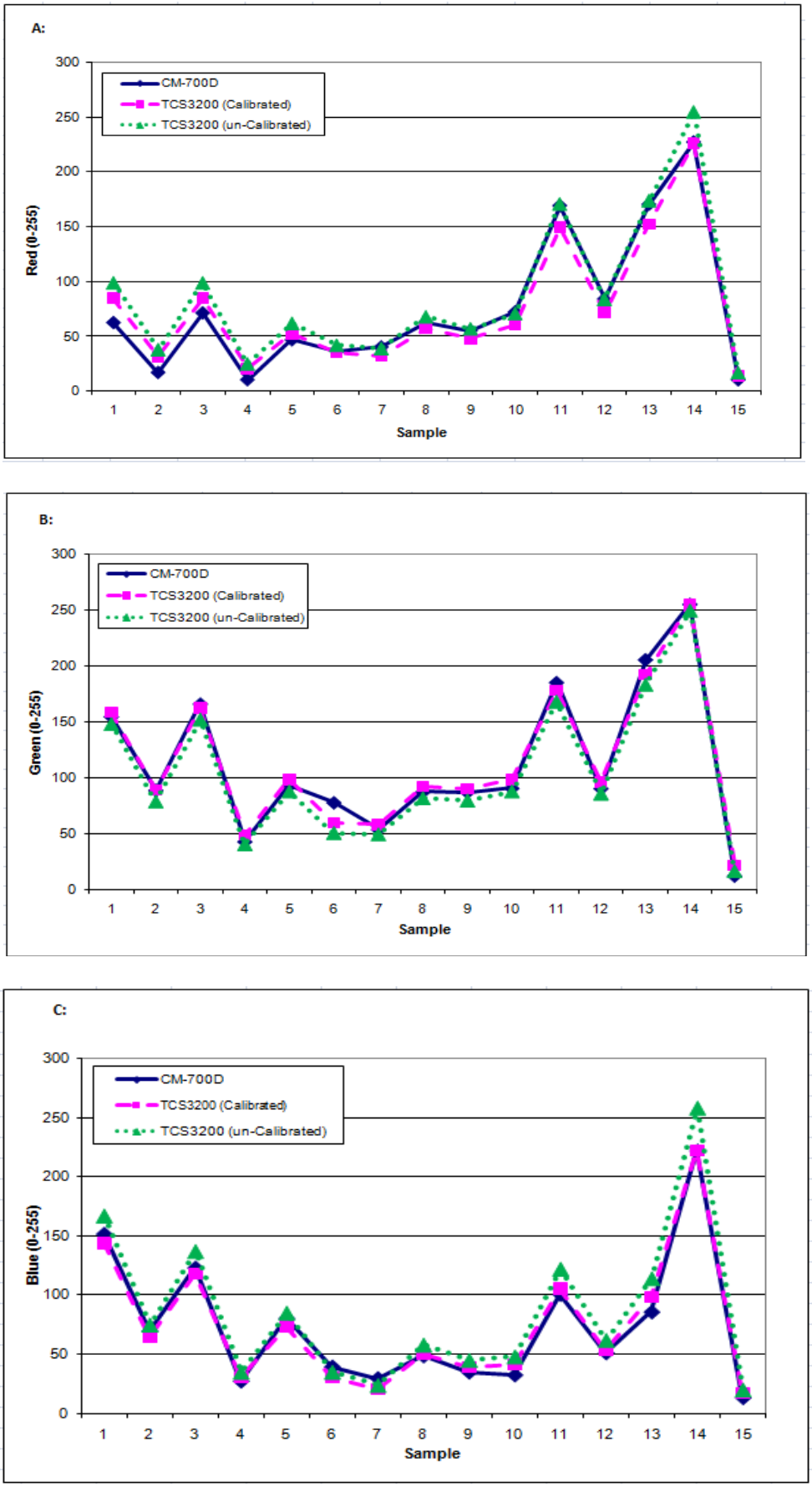


Figure 38: TCS3200 sensor RGB readings, calibrated and uncalibrated, compared to the CM-700D readings of: Red (A); Green (B); Blue (C), using a CIE RGB transformation matrix.

TCS3200 (uncalibrated)	TCS3200 (calibrated)	CM-700DSpectrophotometer
RGB = 63,90,41	RGB = 60,98,41	RGB = 72,91,32
		

Table 14: An example of a green colour interpreted by the CM-700D and TCS3200 colour sensor before and after a calibration factor (143C).

Results showed when calibrating the CM-700D XYZ values to CIE RGB instead of sRGB the calibration, results improved as shown in Table 15 to have a much smaller error for R, G and B.

	CIE RGB			sRGB		
Error	10.289	6.117	5.683	14.777	7.055	9.564
Error %	4.035	2.399	2.229	5.795	2.767	3.751
S.D	6.562	4.739	3.357	12.314	7.540	5.772

Table 15: Comparisons between CIE RGB and sRGB transformation matrix, showing the CIE RGB results to be more accurate than the sRGB.

A least squares regression was undertaken as an alternative method to calibrate the sensor. This takes into account the error between the actual ideal results and the un-calibrated results to determine a gamma value for each of the R, G and B values (Eq (4.18) – (4.22)). A table of results is shown (Table 16) with the associated errors, percentage errors and standard deviations shown in Table 17 and graphs shown in Figure 39.

Formula for calculating the least squares regression:

$$Q = I^{\gamma} \tag{4.18}$$

$$\ln Q = \gamma \times \ln I \tag{4.19}$$

$$e^2 = \sum_i 2(\ln Q - \gamma \ln I)^2 \tag{4.20}$$

$$\frac{de^2}{d\gamma} = \sum_i 2(\ln Q_i - \gamma \ln I_i)(\ln I_i) = 0 \tag{4.21}$$

$$\sum (\ln Q \ln I) = \gamma \sum (\ln I)^2 \tag{4.22}$$

ID	TCS-3200						CMD-700			Output		
	Gain Adjusted			White Adjusted			RGB Equivalent (CIE RGB)			Calibrated Result		
	R	G	B	R	G	B	R	G	B	R	G	B
123A	99	148	167	88	152	144	62	155	152	87	159	145
125C	28	67	67	25	69	58	2	76	56	25	78	59
127C	38	79	75	34	81	65	17	89	69	33	90	66
129C	99	152	137	88	156	118	71	166	123	87	163	119
131C	25	41	35	22	42	30	10	43	27	22	50	31
133C	62	88	85	55	90	73	47	93	80	55	99	74
135C	42	51	35	37	52	30	36	78	39	37	61	31
137C	39	50	24	35	51	21	40	54	30	34	59	21
139C	68	82	58	61	84	50	63	88	48	60	93	51
141C	57	80	45	51	82	39	55	87	35	50	91	40
143C	71	88	48	63	90	41	72	91	32	63	99	42
145C	171	168	122	152	172	105	169	185	101	152	179	106
147C	84	86	62	75	88	53	84	91	51	74	97	55
149C	174	183	114	155	187	98	170	206	86	154	193	99
155D	255	249	258	227	255	222	227	255	222	227	255	222
202A	17	17	20	15	17	17	10	13	13	15	22	18

Table 16: Results obtained comparing the TCS3200 colour sensor (calibrated and uncalibrated) to the CM-700D over a range of 16 colours using a least squares regression analysis on CIE RGB transformation matrix.

	TCS3200 (un calibrated)			TCS3200 (calibrated)		
	R	G	B	R	G	B
Error	12.3148	9.2944	12.9194	10.6967	6.2728	5.5981
Error %	4.8293	3.6449	5.0664	4.1948	2.4599	2.1953
S.D	12.0232	7.2764	8.8594	7.5769	4.3531	3.3053

Table 17: Average Error (0-255), percentage error and standard deviation for red, green and blue measurements of the TCS3200 colour sensor, calibrated and un calibrated, when compared to CM-700D results across a range of colours using a least squares regression (when using CIE RGB).

Gamma values obtained using a least squares regression

$$(\mathrm{Red})\lambda_{\mathrm{R}} = 1.00,\ (\mathrm{Green})\lambda_{\mathrm{G}} = 0.91,\ (\mathrm{Blue})\lambda_{\mathrm{B}} = 0.99$$

Figure 39: TCS3200 sensor RGB readings, calibrated and uncalibrated, compared to the CM-700D readings of: Red (A); Green (B); Blue (C).for a range of RHS green colours. (Colour samples are as given in Table 18

Comparisons between the standard method of determining the gamma function using $R_N' = R_N^Y$ and the least squared regression $\sum(\ln Q \ln I) = \gamma \sum(\ln I)^2$ are shown in Table 18. Results showed both the methodologies to be very similar to each other in terms of percentage errors and standard deviations. Further testing across a larger range of colours is required to determine which methodology is the "best".

	Original Gamma Method			Least Squares Regression		
	R	G	B	R	G	B
Error	10.289	6.117	5.683	10.6967	6.2728	5.5981
Error %	4.035	2.399	2.229	4.1948	2.4599	2.1953
S.D	6.562	4.739	3.357	7.5769	4.3531	3.3053

Table 18: Comparison between Gamma calculation and least squares regression.

With the calibration factor determined, the TCS3200 was able to identify colour of selected plant leaf material. The sensor has been integrated with a proximity sensor to ensure all readings are taken at a fixed height above the plant leaves. As the colour sensor is attached to a robotic arm, the sensor is able to autonomously return to the same position to record the colour, making it possible over time to identify subtle changes in the colour of the plant leaves. These changes in plant material provide the potential for plant nutrient concentrations to be adjusted in order to achieve optimum growth.

Application of colour sensor:

Having successfully tested and calibrated the colour sensors, an experiment was conducted to monitor the colour of a sensing leaf. The leaf was placed in a dark room with the colour sensor mounted 20mm above the leaf. Readings were taken every 2.5 minutes over a 48 hour period with the result showing the leaf to change colour over the time period. The R, G and B values were plotted in a 3 dimensional graph using a scale of 0 – 255 for each axis (representing the entire colour space possibilities - see Figure 39). The results show the colour of the leaf to be only a very small portion within the colour space available.

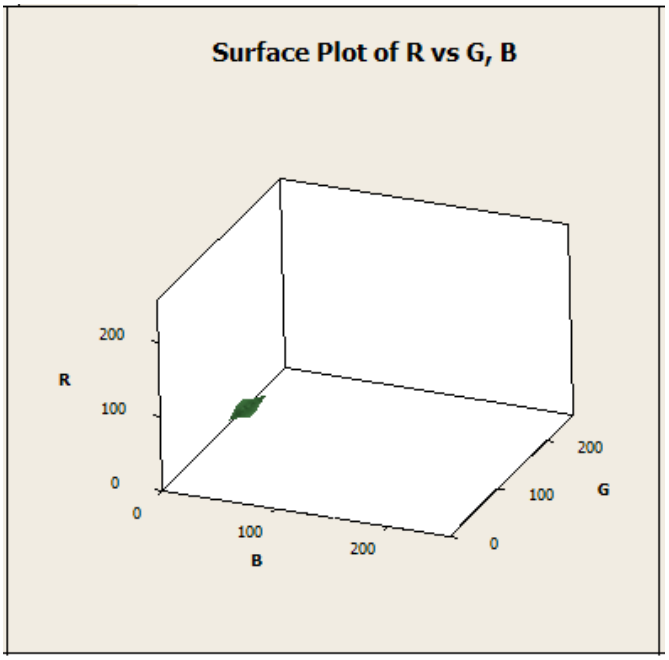


Figure 40: 3D representation monitoring the colour of a leaf (scale 0 - 255).

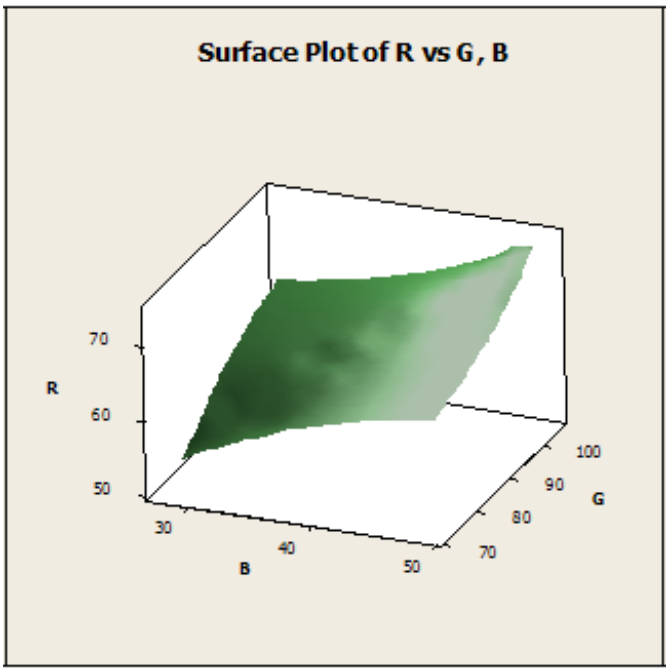


Figure 41: 3D representation of RGB monitoring for the colour on a plant on a reduced scale.

Following adjusting the scale of the data, it can be seen that the colour of the plant leaf does vary substantially over the 48 hour period. It should be noted that the graph does not inform us as to when the colours were achieved, i.e. in which order the plant colours were achieved. Initial and further testing was conducted on plants located in the laboratory.

An issue with colour readings:

Surrounding ambient light can have an effect on colour readings taken by the TCS3200 when in the laboratory which could lead to false colour readings. To eliminate any ambient light, the TCS3200 can take an initial ambient light reading. This is achieved by turning off the two LEDs on the TCS3200 then simply taking a reading of the working environment. This ambient colour reading is then subtracted from the actual colour reading of the plant tissue, eliminating any possible ambient light effects. Testing however showed that ambient light to have little effect when the colour sensor illuminated the object, regardless of whether it was in a light or dark environment.

4.2. Proximity Sensor

Figure 42: Parallax Ping)))™ ultrasound proximity sensor. (www.parallax.com)

Because the TCS3200 colour sensor was required to be at 20mm above the plant material a Parallax Ping)))™ proximity sensor (Figure 42) was used, the sensor uses ultrasound as a means to measure the distance between itself and an object. The sensor measures the distance using sonar, or ultrasound (at a frequency of 40 KHz) by sending a pulse then waiting for the echo to bounce back from the object being measured.

The time elapsed from when the pulse is sent to the time the echo is returned provides information on how far away the object is. From this time interval, along with a calculation for the speed sound travelling through air, one can determine the object distance.

Example: Temperature = 26.5°, time for echo to be received 321.6 microseconds. (This is the time for the pulse to be sent there and back, so need to half)

$$t = 0.5 \text{ x } (t_1 - t_2) \quad (4.23)$$

$$t = 160.8 \text{ microseconds}$$

Where: t = time for pulse to travel in one direction
t1 = time pulse is initially sent
t2 = time pulse returns back

The speed of sound in air at 26.5°C

$$V_{air} = 331.5 + (0.6 \text{ x } T_c) \text{ m/s} \quad (4.24)$$

$$V_{air} = 347.4 \text{ ms}^{-1}$$

$$V_{air} = 34740 \text{ cms}^{-1}$$

$$V_{cm} = 2.8785 * 10^{-5} seconds cm^{-1} \quad (4.25)$$

$$\text{Distance} = t * (V_{cm} \text{ x } 1000000) \quad (4.26)$$

$$\text{Distance} = 5.586192 \text{ cm}$$

$$\text{Distance} = 55.86192 \text{ mm}$$

Where: V_{air} = the speed of sound through air at a given temperature
T_c = temperature of the air
V_{cm} = the time in seconds to cover 1cm (V_{air}^{-1})
Distance = the distance the object is away from the sensor

The speed that sound travels through air varies with temperature. Results from Parallax show over a temperature range of 0° to 70° the percentage error ranges from 11% - 12% in magnitude. Therefore a temperature sensor was also required.

Proximity sensor testing involved mounting the proximity sensor directly above a flat surface. A temperature reading is taken before taking ten proximity readings at the same height. These ten readings are then averaged. Electronic vainer callipers were used to find the actual distance, and then compared to the

proximity readings. The absolute errors were then recorded and an average error was applied the proximity senor readings of 11.39mm (Table 19 and Figure 43).The sensor was only calibrated over the heights from 10 to (approx) 80 mm since the sensor will only be required to identity distances over this range.

Actual Height (mm)	Average Proximity Height (mm)	Error (mm)
73.81	86.06	12.25
62.32	73.61	11.29
47.32	58.31	10.99
34.02	44.73	10.71
20.91	32.65	11.74
10.98	22.31	11.33
	Average Error	11.39

Table 19: Comparisons between actual height measured and proximity height.

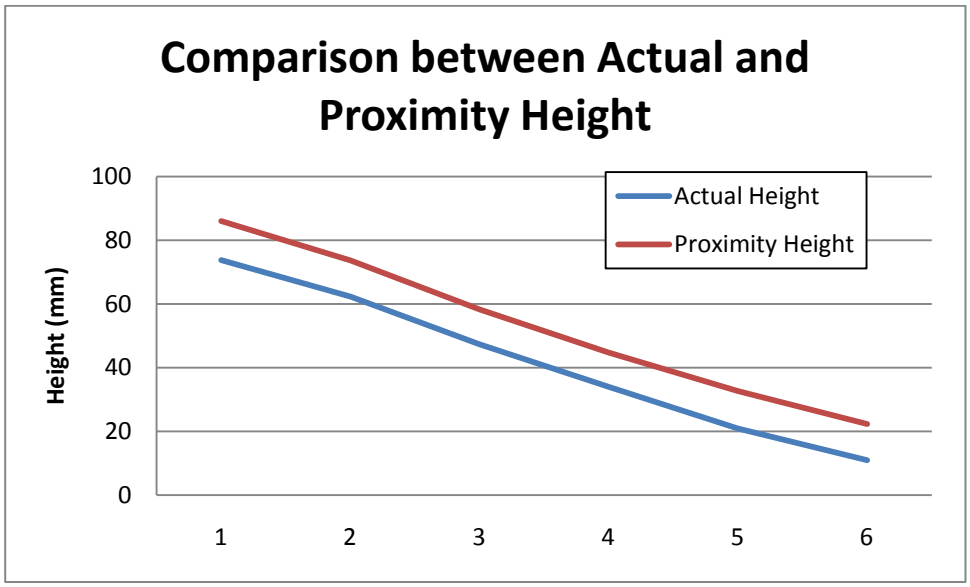


Figure 43: Graph of actual measured height v proximity height.

4.3. Temperature and Humidity Sensor

A Sensirion SHT11sensor module (Figure 44) from Parallax provides temperature and humidity measurements, allowing for accurate proximity readings to be calculated as well as providing basic environmental conditions for the user. The Sensirion SHT11 has a 8 pin-dip module for interfacing to a microcontroller, to sense temperatures ranging from -40^{o}C to 123.8^{o}C with an

accuracy of ±0.5^{o}C at 25^{o}C and relative humidity (RH) in the range of 0 to 100% with an accuracy of ± 3.5% RH.

Figure 44: Parallax Sensirion SHT11 Temperature and humidity sensor. (www.parallax.com)

The SHT11 interfaces to the microcontroller using a 2 wire configuration, a SCK (serial clock input) is used to synchronise communication between the sensor and the microcontroller, while the DATA (serial data) pin transmits the data to and from the sensor. An onboard heating element can be turned on, increasing the temperature of the SHT11 by approx 5^{o}C.

The sensor is programmed to take 20 temperature readings, and averaging these to provide an “average” reading of the ambient temperature. This is followed by 20 humidity readings, again averaging these results. The temperature sensor was tested and calibrated against a Vernier GO!™ Temp USB temperature probe (Vernier Software and Technology, OR, USA) over a range of temperatures while in a temperature controlled room. Results showed the Sensirion SHT11 to be accurate to within 2% over a range of temperatures from 15^{o}C to 30^{o}C. Results showed the Sensirion SHT11 temperature sensor to be on average incorrect by an absolute value of 0.26 ^{o}C as shown in Table 20. Graphing the results showed there to be little difference as can be seen in Figure 45.

GO!™ Temp	Sensirion SHT11	Error (Abs)
10.2	9.8	0.4
11.4	11.6	0.2
12.8	12.5	0.3
13.1	13.1	0
15.2	14.9	0.3
16.3	15.9	0.4
17.2	17.3	0.1
18.2	18.2	0
19.5	19.1	0.4
20.4	20.2	0.2
21.3	22	0.7
22.9	22.8	0.1
24.5	24.4	0.1
25.1	25.5	0.4
	Average Error:	0.257142857

Table 20: Comparisons between Vernier GO!™ Temp USB temperature probe and Sensirion SHT11 Temperature sensor.

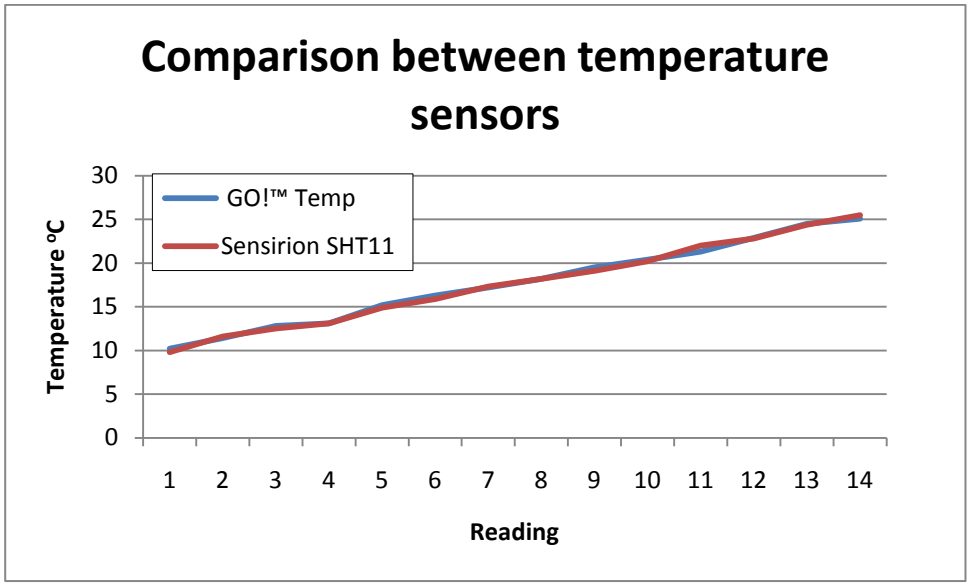


Figure 45: Comparisons between Vernier GO!™ Temp USB temperature probe and Sensirion SHT11 Temperature sensor.

Further testing is required to ensure the humidity sensor is working accurately to simply provide information pertaining to the working environment.

4.4. Carbon Dioxide Sensor

Figure 46: Parallax CO_2 gas sensor module. (www.parallax.com)

The Parallax CO_2 sensor, (Figure 46), allows CO_2 content of the surroundings to be monitored, again giving basic information on the environment. The sensor was required to be programmed to allow CO_2 readings to be taken, resulting in source code having to be created.

The heart of the CO_2 sensor is a Hanwei Electronics MG811 gas sensor. Once a heating element is activated, the gas sensor responds to the surrounding CO_2 present by generating a voltage proportional to the quantity of CO_2 present. This voltage is then interpreted by the microcontroller.

To test the CO_2 sensor, the sensor was connected to an oscilloscope and CO_2 was identified by breathing onto the sensor, this showed the sensor to react to CO_2. Further testing is required to determine the quantity of CO_2 present at a given time, measured in parts per million (PPM).

4.5. Microcontroller and Development Board

A Parallax Basic Stamp microcontroller was used for communication and simple integration of the Parallax sensors to a PC. Parallax microcontrollers are programmed through PBASIC (current language version 2.5) language which is a variation of the BASIC programming language. The particular model selected for this application was the BS2e microcontroller (Figure 47). The BS2e features a processor speed of 20MHz, giving it the ability to execute 4000 PBASIC instructions/second. The microcontroller features 16 I/O (input and/or output) pins + 2 dedicated serial communication pins (Tx (transmit) and Rx (receive)). With a RAM (Random Access Memory) of 32 bytes, Scratch Pad RAM of 64 Bytes and EEPROM size of 2 x 8 Kbytes (equating to approx 4000 instructions).

Figure 47: Parallax Basic Stamp 2e Microcontroller. (www.parallax.com)

The Basic Stamp Microcontroller is designed to integrate either into a custom designed PCB or into a Parallax Board of Education (BOE) (Figure 48) which readily integrates sensors to the microcontroller. The BOE provides a USB interface to the PC and provides both a 2.1mm centre-positive plug and a 9-volt battery supply connection to power the device. The BOE allows for any of the Parallax Basic Stamp 2 (24 pin) microcontrollers to be used.

The BOE has a regulated 5v (Vdd) and an unregulated input (Vin), as well as ground (Gnd) connection. An onboard breadboard allows for testing and integration of sensors or custom circuitry, by placing the wires into the appropriate sockets on the breadboard. To be able to use the BOE a virtual USB COM port driver needs to be installed, allowing the PC to recognise the connected device.

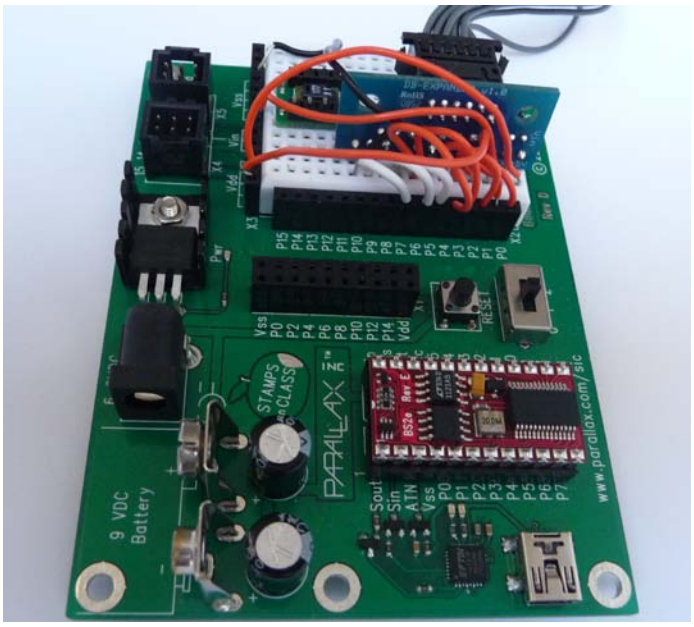


Figure 48: Parallax USB Board of Education (BOE) for prototyping and testing.

4.6. Colour Camera

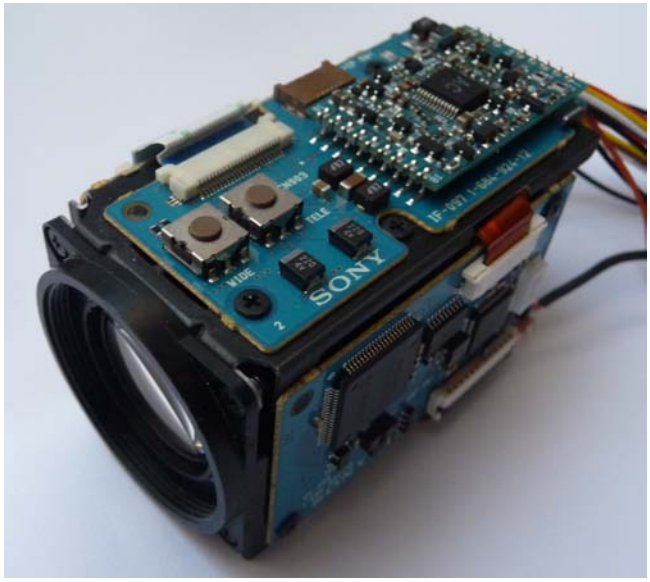
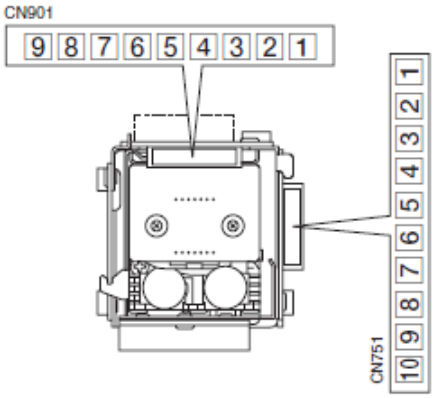


Figure 49: Sony FCB-1X11AP colour box camera.

A Sony colour block camera (model: FCB-IX11AP) was used for this project (Figure 49) this colour camera features a ¼ type Exview HAD™ CCD (charge coupled device) using PAL (Phase Alternating Line) transmission methods [as opposed to an NTSC (National Television System(s) Committee)]. The camera has a 40x zoom ratio (10x optical, 4x digital) which is controllable via Sony's VISCA protocol; the camera's macro capability allows it to capture images as close as 10mm from the subject.

The camera is compact at 65mm x 44.8mm x 39.3mm and weights 95g. It has numerous controllable functions (up to 38,000 different command combinations) including digital and optical zoom, aperture, shutter speed and iris settings. It also has the ability to be interfaced with up to seven additional cameras which can all be controlled through a single controller. The camera requires a stable 6v to 12v (DC) input voltage to operate.

This particular camera was selected over other cameras on the market, not only because of its ability to interface with a PC to control numerous features, but also due to its specifications being suitable for the working environment. The camera has approx 440K pixels [752 (H) x 582 (V)] elements and can operate in light conditions as low as 1.5 lux, and temperatures ranging from 0 to 50°C. The electronic shutter speed is controllable from 1/10 to 1/10000 of a second.



CN751 (for communications)

Pin No.	Name	Level
1	TxD IN	RS-232C level
2	NC	
3	NC	
4	RxD IN	RS-232C level
5	TD	TTL level
6	NC	
7	NC	
8	RD	TTL level
9	GND	
10	AF OUT	

Connector type: JST S10B-ZR-SM3A-TF

CN901 (for DC and video)

Pin No.	Name	Level
1	DC IN	6V to 12V
2	GND (for DC IN)	
3	NC	
4	VBS OUT	Composite video signal
5	GND (For VBS OUT)	
6	Y-Out	
7	GND (For Y)	
8	C-Out	
9	GND (For C)	

Connector type: JST S9B-ZR-SM3A-TF

Figure 50: Sony FCB-IX11AP camera connections and pin allocation. (http://www.pro.sony.eu/biz/lang/en/eu/product/fcbcseries/fcb-ix11ap/overview)

The Sony camera has 2 input connectors (Figure 50); the CN751socket is for communication, allowing for data to be sent to the camera at either a RS232 or TTL level and the CN901 socket is for power and video out. Six pins are required to power, control and receive the image, with the remaining pins not required in this application but do have uses for advanced applications.

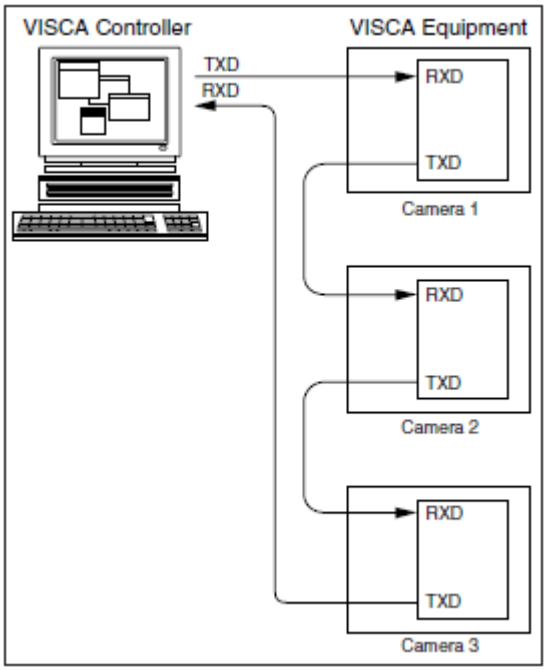


Figure 51: Illustration of how multiple cameras can be connected. (http://www.pro.sony.eu/biz/lang/en/eu/product/fcbcseries/fcb-ix11ap/overview)

Sony's VISCA protocol allows up to seven Sony cameras to be connected in a daisy chain effect (Figure 51). These can all be controlled with one command either simultaneously or individually. This is achieved as the first byte of serial data sent to a camera contains the camera's unique address (hexadecimal 8X). When one camera is in use, the unique address is 1 – therefore, hexadecimal 81 is required to be sent to the camera. If a 2nd camera was used, its address would be hexadecimal 82. A full list of commands can be seen in the appendix – page 140.

4.6.1. Frame Grabber

A USB frame grabber was required to capture the image seen through the lens of the camera and interfaced to the PC. A USB2.0 frame grabber (Jaycar Electronics, NZ) is currently being used. Although a relatively cheap model, it

provides the necessary interface and with a maximum of 29.7 fps (frames per second) offers the ability to record video.

4.7. Overview

The above section explains in details the sensors that were used to make this project complete, along with the integration of the sensors and to allow for the respective readings to be obtained from each sensor. Testing of each sensor has been completed where possible with calibrations being conducted to ensure desired results are achieved. Calibration of the colour sensor required a number of different experiments to be conducted, with results continually improving with the implementation of different methodologies. Eventual results used the original gamma calculation, having calibrated the colour sensor originally over a variation of colours – before conducting the sensor over a range of green-yellow RHS colours. Further investigations identified that a least squares regression may yield more accurate results, however as shown in Table 18 this was not the case. A more in-depth calibration over a wider range of RHS colours could potentially yield more accurate results.

Testing of the proximity sensor revelled the standard program (from Parallax) to have an average error of 11.39 mm over a range of heights ranging from approx 10mm to 80mm. Further testing of the proximity sensor is not required, as the proximity sensor will only be used for over this range (10 – 80mm). Testing of the temperature and humidity sensor also showed the Parallax program to have a slight error however was easily accounted for with an offset (similar to that used for the proximity sensor). Further testing needs to be conducted on the CO2 sensor, to allow the microcontroller to interpret the CO2 present, by interpreting the voltage produced by the sensor when CO2 is present.

Control and integration of the microcontroller, board of education and camera were conducted with little hindrance. Overall all sensors were integrated and tested, to allow results to be obtained. Further testing is not essential but would provide more accurate results for most of the sensors. However further testing is required for the CO2 sensor.

Implementation of the colour sensor has been tested to identify the colour change of a plant dying overtime, as shown in Figure 40 and Figure 41.These results will allow research to be conducted to identify plant health status and in the future identify chlorophyll contents of selected plant material.

5. Software Implementation

Automation of this project required a large amount of program code to be written in Visual Studio 2005 was the selected language due to its ease of use and prior work being completed using the software. Visual Studio makes use of the VB.net programming language, with the more free, strip down version, Visual Studio Express Edition, being popular with programming enthusiasts. Considerable support is available on the internet. Being a Microsoft product, integration into other Microsoft products is done with ease, and support from MSDN (Microsoft Development Network) provides users with a large amount of information related to the programming capabilities and potential.

5.1. Timer & Serial Port

Visual Studio has an inbuilt timer function that executes a specific method (or sub routine) at a specific time interval as set up by the user. The timer function allows the user to start and stop code at specific times. The timer's interval can be set to almost any time period, with the computer being the limiting factor (i.e. PCs will struggle to execute large amounts of code in a very short time interval).

Visual Studio has an inbuilt serial port class which is used to control the I/O (input/ output) of the serial port and its features. The serial port class supports ASCIIE encoding, (the process of transforming a set of Unicode characters into a sequence of bytes [58]).

5.2. Motor Controller

Communication from the PC to the Ocean Controls KTA-190 motor controller required the use of a serial port (or USB port when using a USB to serial converter) connected to the controller. The specifications of the motor controller require a baud rate of 9600, 8 data bits, 1 start bit, 1 stop bit and no parity (9600, 8, N, 1). The KTA-190 motor controller requires an array of ASCII command to be sent. Implementation of this requires an array to be declared, with each of the indexes of the array containing part of the code.

```
Public Motor_1(18) AsByte            'move motor 1 to postion
      Motor_1(0) = Asc("@")
        Motor_1(1) = Asc("0")
        Motor_1(2) = Asc("1")
        Motor_1(3) = Asc(" ")
        Motor_1(4) = Asc("R")
        Motor_1(5) = Asc("M")
        Motor_1(6) = Asc("O")
        Motor_1(7) = Asc("V")
        Motor_1(8) = Asc(" ")
        Motor_1(9) = Asc("1")
        Motor_1(10) = Asc("0")
        Motor_1(11) = Asc("0")
        Motor_1(12) = Asc("0")
        Motor_1(13) = &HD
```

By setting all indices of the array (declared as Motor_1) to the required ASCII command (as per the KTA-190 motor controller data sheet) this command would move motor one a relative movement of 1000 steps.

Once the array is full with the required commands, the array is then sent via the serial port to the controller, making use of the serial ports I/O features. Because all commands sent to the controller are required to be in the same format, (9600, 8, N, 1) there is no requirement to program the serial port's features into the required format (instead the required format is hard programmed in the serial port properties)

Ensuring the serial port is open before attempting to send a command is essential as failure to ensure this will cause the program to crash in the event it is closed. This procedure is achieved by checking to see if the port is currently open, and if not, opening it using a basic if-then-else statement. If the port is already open the if-then-else statement is ignored.

Sending the array requires the "write" method to be called upon, specifying the array to send, an offset, if required and finally the number of the indices contained within the array required to be sent.

To ensure the program runs efficiently, the serial port is then closed.

```
      IfMe.SerialPort_Motor.IsOpen = FalseThen
Me.SerialPort_Motor.Open()
EndIf

Me.SerialPort_Motor.Write(Motor_M1, 0, 14)

IfMe.SerialPort_Motor.IsOpen = TrueThen
Me.SerialPort_Motor.Close()
EndIf
```

5.3. Microcontroller

The Basic Stamp microcontroller from Parallax is programmed through Basic Stamp Editor/Development System © software available from parallax (Figure 52). The microcontroller is programmed via the PBASIC language, which is a modified version of the original BASIC language specifically designed for the programming of the Basic Stamp microcontroller. The microcontroller has two serial communication pins, (one input and one output pin).

A Board of Education (BOE) was used to initially implement the sensors and program the microcontroller, providing the ability to simply plug the microcontroller into the board and begin programming (once power and serial communication have been established). To set up a program on the microcontroller simply involves selecting the language version to use (in this case, version 2.5) then begin programming. Parallax provides manuals relating to the programming language and the user interface.

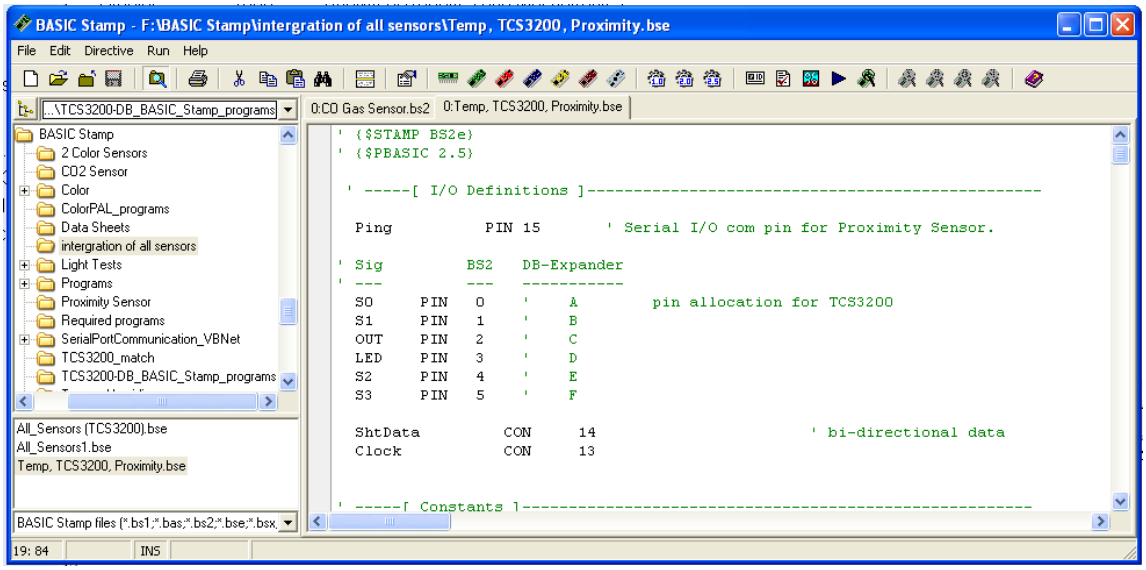


Figure 52: BASIC Stamp software interface.

As with any programming language, it is preferable to declare variables and constraints at the beginning of the program. When programming the Basic Stamp. The following steps were necessary:

1. Specified the programming language: Version 2.5.
2. Specified the Basic Stamp Microcontroller being used: BS2e (This is essential as each microcontroller has a different processor speed and capabilities).
3. Declared all I/O definitions and assigned a variable name to each of the required pins.
4. Declared any constraints required. declared variables and assigned a constant value to that variable.
5. Declared the name and type of variable used. Data types range from Bit: 0 or 1; Nib: 0 to 15; byte: 0 to 255; Word: 0 to 65535. Ideally a variable should be declared using the smallest variable type to save memory.
6. Initialised the program to be executed by turning on appropriate pins. Then calling on the appropriate subroutines when required.
7. The subroutines hold the program for each of the sensors and these are executed when called upon.

Integration of the temperature, CO_2, proximity and colour sensors was required involving grouping each of the I/O definitions, constants, initialisation and subroutines into the one program. To call on each of the sensors individually, a case statement was created in the initialization stage. Once integrated with custom software, this allowed the user to select which sensor to take a reading by simply clicking a button on a GUI. This works by sending a command to the microcontroller (via the serial port) and using the case statement to call on the appropriate sub-routine. After successfully integrating the sensors using buttons, automation involved simply replacing the button with a timer.

```
GetInput:                              ' Receives input from some external source
  SERIN INPUT_PIN, baud, [Command]
  GOSUB TakeAction
RETURN


TakeAction:  ' Determines what to do based on result of GetInput
  SELECT Command

    CASE "Prox"
    GOSUB Task1
    RETURN

    CASE "TCS32"
    GOSUB Task2
    RETURN

    CASE "Temp"
    GOSUB Task3
    RETURN


  ENDSELECT
```

5.4. Sensor Programming

Parallax sensors come with example downloadable source code from Parallax. This code is useful to get started, but more detailed manuals accompany the sensors for more advanced programming options.

The TCS3200 colour sensor source code allows the user to plug the sensor into the microcontroller to start displaying the RGB colour content values on screen and with the use of the data sheet, customised software is able to be created. A colour sensor program was written to take 20 RGB readings and average these 20 readings and send this information back via the serial port to the PC for it to be stored and displayed on the main program GUI. The source code was also used to calibrate the sensor by adjusting the integration time of the on-board TAOS chip. This is achieved by altering the time interval for readings that are taken to achieve a white colour of 255, 255, 255 (or as close as possible).

Source code was downloaded for use with the temperature sensor, then modified. This source code provided temperature readings on a regular basis in both degrees Centigrade (˚C) and Fahrenheit (˚F) although temperature readings were only required to be in degrees Centigrade (°C). Testing revealed that the temperature sensor, when using the source code, was accurate to within an average of 0.26°, therefore the code was modified as part of the

calibration process to take into account this error. The code was written to allow the temperature readings to be taken when an input was received, so that the sensors were not continually taking temperature readings but only taking them when required. The temperature sensor has an on-board heater to ensure no condensation builds up, which may result in false readings. The sensor requires approx 15 seconds to take one reading.

The Parallax proximity sensor source code gave good information on the distance the sensor was from a given object, however, the microcontroller was unable to deal with decimal points accurately, with distances only to the nearest cm. To overcome this issue, the code was modified to only record the raw data and send this information back to the PC on request. This allowed for more accurate (within 1mm) and faster proximity distance calculations to be made. (achieved because a PC can calculate complex calculations including decimal values). The proximity sensor integrated with the temperature sensor, recording the temperature of the ambient surroundings before calculating the distance. This is achieved by calling the sub-routine of the temperature sensor from within the proximity sensor's routine. This is essential, as the temperature of air can affect the speed at which sound travels through it.

Due to the nature of the CO_2 sensor, Parallax does not provide any source code examples on these sensors, although data sheets provide some information regarding how to test and calibrate the sensor. It should be noted that the sensor should only be used as an indication for the presence of CO_2 and should not be used as an actual alarm for monitoring of CO_2. For the intended application this was sufficient as the sensor was only used to indicate CO_2 levels for use in statistical data on the environment. The code that was developed allowed the sensor to indicate if CO_2 was present in the surroundings, by providing a 0 – 3.3v microcontroller voltage, proportional to the quantity of CO_2.

5.5. Camera Control through Software

The Sony FCB-1x11AP camera uses the VISCA (Video System Control Architecture) command protocol for communications with the PC. The basic units of VISCA communication are 3 to 16 byte packets. The first byte of the packet is the header comprising of the sender's and header's addresses, the remainder of the packet holds the commands (or message) to be sent to the camera (Figure 53). The packet's final byte comprises a terminator signifying the end of the packet and usually has the value FF (or 11111111), as shown in Figure 53.

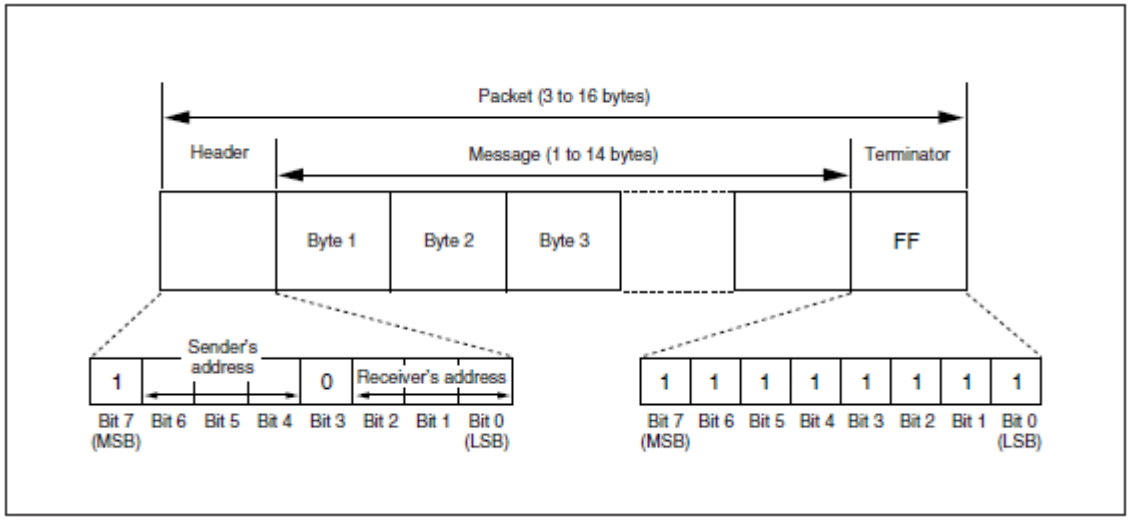


Figure 53: Example of serial command data required to communicate with camera. (http://www.pro.sony.eu/biz/lang/en/eu/product/fcbcseries/fcb-ix11ap/overview)

In order for any of the camera features to be controlled, hexadecimal commands (as serial data) were sent to the camera. On the PC side pin 3 (XTX) and pin 2 (XRX) of the serial port were connected directly to the camera's RxD (at RS232 level) and TxD (at RS232 level) inputs respectively.

Commands were sent to the camera using a serial port communication protocol of 9600, 8, N, 1. Again, there is no requirement to program the serial port's features into the required format (instead the required format is hard programmed in the serial port's properties).

Programming of the camera was completed in a similar manner to that of the KTA-190 motor controller with commands sent to the camera via a separate serial port (similar to that of the controller board). Hexadecimal commands

(representing the numerous controls) were programmed into the indices of an array and sent via the serial port, to the camera as an array.

Executing the zoom functions of the camera (or any other function) required an array to be declared, followed by the indices of that array being filled with the appropriate hexadecimal command. The various functions of the camera had multiple values that could be executed (for example the zoom function commands are essentially the same, except indices 4-7 of the array differ depending on the zoom function to be executed). A case statement was implemented in the code. As a slide bar on the GUI (Figure 54) is moved, a case statement is activated within the software which synchronises the slide bar position with the appropriate zoom command. This replaced the corresponding indices of the array being used, which was then sent to the camera. This approach of using a case statement is used where functions have multiple values that are possible; this saves on programming execution time, making the program run more efficiently.

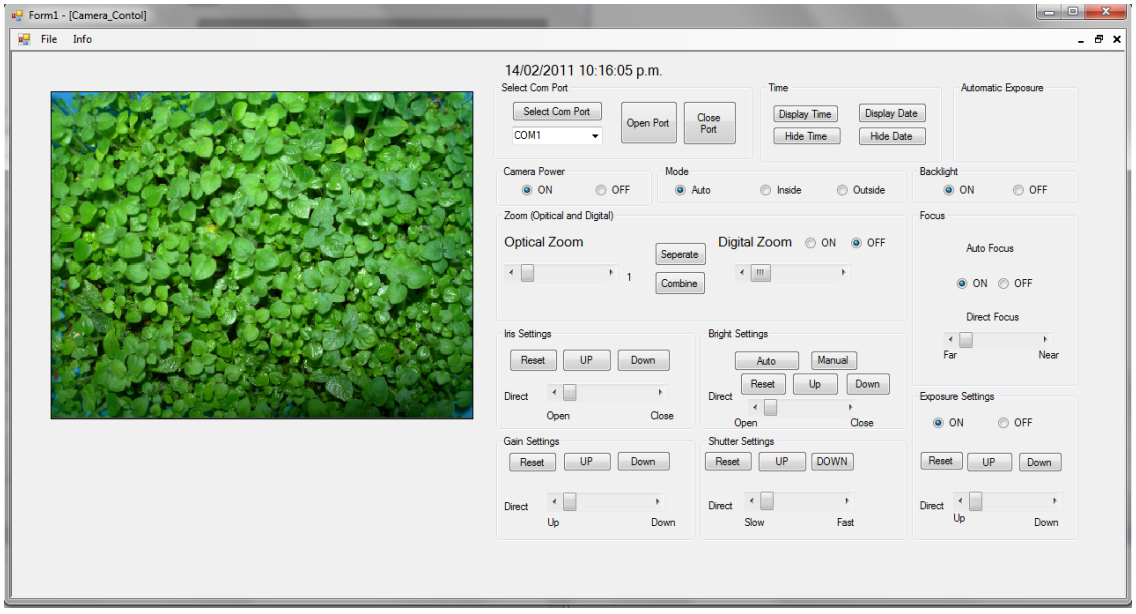


Figure 54: GUI of camera control form, showing camera controls and image as seen by camera.

An example of a case statement is shown in Figure 55 where the zoom function was declared with the indices of the array being set to the corresponding commands. The program then called on the Zoom Camera sub-routine and set the remainder of the array to the corresponding values, determined by the slide bar value, before exiting the routine and sending the array through the serial

port. Any change in value of the slide bar executed the routine again, sending the corresponding zoom value to the camera.

```
Zoom(0) = &H81
Zoom(1) = &H1
Zoom(2) = &H4
Zoom(3) = &H47
Zoom(8) = &HFF

ZoomCamera(ZoomPostion)


If Me.SerialPort1.IsOpen = False Then
    Me.SerialPort1.Open()
End If

SerialPort1.Write(Zoom, 0, 9)

    Private Sub ZoomCamera(ByVal ZoomPostion As Double)

        Select Case ZoomPostion
            Case 1
                Zoom(4) = &H0
                Zoom(5) = &H0
                Zoom(6) = &H0
                Zoom(7) = &H0

            Case 1.1
                Zoom(4) = &H0
                Zoom(5) = &H2
                Zoom(6) = &H7
                Zoom(7) = &H4
```

Figure 55: Complete code for zoom function of camera.

5.6. Integration of a Database

Information in this section is from http://www.homeandlearn.co.uk/net/nets12p5.html

To store large amounts of data including the positions of plants, motor coordinates, colour readings, images etc, the user needs a simple effective way of storing this information in a logical order such that it is able to be retrieved when required. Visual Studio, being a Microsoft product, integrates with Microsoft Access with ease. Setting up the Access database required some knowledge of database design. First tables were set up with an appropriate field names and data types. Multiple tables were then integrated using queries. Once created, the Access database was integrated with Microsoft Visual Studio by creating a data source within Visual Studio to provide the link between the two programs.

To open, read or modify the data contained in the Access database from within Visual Studio, a connection object must be created. In general, connecting to an Access database requires an OLE DB (Object Linking and Embedding) connection which allows connection to the data source. There are a number of OLE DB objects (known as data providers), but a common one to use is “Jet” (other include SQL Server and Oracle). A variable is created to hold the connection string (in this case ‘con’);

```
Dim con As New OleDb.OleDbConnection
```

The provider of the database technology used and its location on the computer is required to be declared by firstly declaring the variable, dbProvider and dbSource:

```
Dim dbProvider As String
Dim dbSource As String

dbProvider = "PROVIDER=Microsoft.Jet.OLEDB.4.0;"
dbSource = "Data Source = C:/Database.mdb"

con.ConnectionString = dbProvider & dbSource
```

The above code sets the connection technology to ‘Jet’ and then declarers where the database is located (i.e. on the C drive in this instance). This information is then stored in the database connection variable “con”. To open the connection, the open function is called:

```
Con.open()
```

And to close: `con.close()`

Since we wanted to manipulate the data within the Access database, a DataSet is required. Which is simply a copy of the information contained within the database. The Connection object and the DataSet cannot communicate directly, therefore a Data Adaptor was required, allowing for information contained within the DataSet to be manipulated and the database to be updated.

```
Dim ds As New DataSet
Dim da As OleDb.OleDbDataAdapter

da = New OleDb.OleDbDataAdapter(sql, con)
```

Since the Data Adaptor is acting as a go between it was necessary to declare how and what is to be manipulated by the Data Adapter. SQl (Structured Query Language) allows this communication to happen. To select all the tables from the Database, we use:

```
sql = "SELECT * FROM Colour"
```

The above code would select all the data held in the ‘Colour’ database and store it in the Data Set; more specifically one can select individual tables within the database with:

```
sql = "SELECT tbl_R_Colour FROM Colour"
```

The above line would select all the data held in the tbl_R_Colour table only from the database and store it in the Data Set

Once the information from the database is stored in the Data Adapter, the Data Set can be filled:

```
da.Fill(ds, "Colour")
```

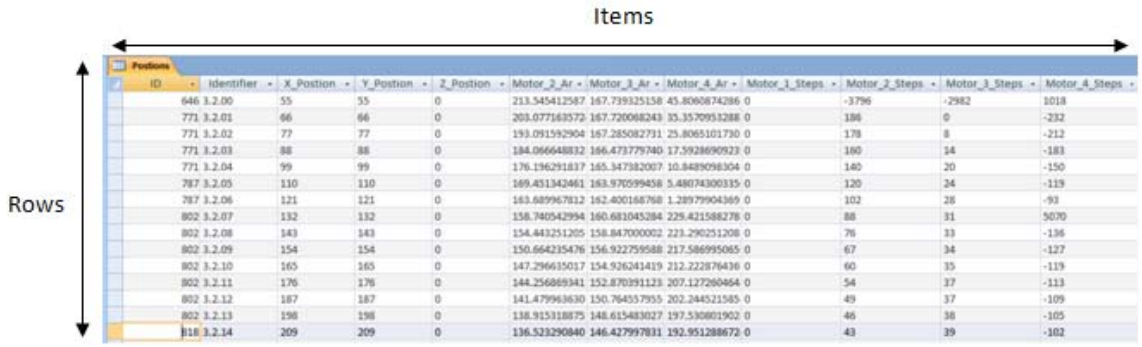
To update a record then the information contained in the text box (“txtbox_R_Colour” in this case) or the information contained in a variable to the Data Set, specifying where in the Data Set to store the data:

```
ds.Tables("Colour").Rows(inc).Item(1) = txtbox_R_Colour.Text
```

It should be noted that any changes to the Data Set does not change the under lying database until a command builder has been specified. The Command Builder works similar to the Data Adaptor in that it communicates between the Data Set and the Database to update it.

Displaying information on the GUI in Visual Studio involved selecting the information that is in the Data Set needed to be displayed:

```
Txtbox_R_Colour.Text = ds.Tables("Colour").Rows(1).Item(1)
```



ID	Identifier	X_Postion	Y_Postion	Z_Postion	Motor_2_Ar	Motor_3_Ar	Motor_4_Ar	Motor_1_Steps	Motor_2_Steps	Motor_3_Steps	Motor_4_Steps
646	3.2.00	55	55	0	213.545412587	167.739325158	45.8060874286	0	-3796	-2982	1018
771	3.2.01	66	66	0	203.077163572	167.720068243	35.3570953288	0	186	0	-232
771	3.2.02	77	77	0	193.091592904	167.285082731	25.8065101730	0	178	8	-212
771	3.2.03	88	88	0	184.066648832	166.473779740	17.5928690923	0	160	14	-183
771	3.2.04	99	99	0	176.196291837	165.347382007	10.8489098304	0	140	20	-150
787	3.2.05	110	110	0	169.451342461	163.970599458	5.48074300335	0	120	24	-119
787	3.2.06	121	121	0	163.689967812	162.400168768	1.28979904369	0	102	28	-93
802	3.2.07	132	132	0	158.740542994	160.681045284	229.421588278	0	88	31	5070
802	3.2.08	143	143	0	154.443251205	158.847000002	223.290251208	0	76	33	-136
802	3.2.09	154	154	0	150.664235476	156.922759588	217.586995065	0	67	34	-127
802	3.2.10	165	165	0	147.296635017	154.926241419	212.222876436	0	60	35	-119
802	3.2.11	176	176	0	144.256869341	152.870391123	207.127260464	0	54	37	-113
802	3.2.12	187	187	0	141.479963630	150.764557955	202.244521585	0	49	37	-109
802	3.2.13	198	198	0	138.915318875	148.615483027	197.530801902	0	46	38	-105
818	3.2.14	209	209	0	136.523290840	146.427997831	192.951288672	0	43	39	-102

Figure 56: Example of a database table of the motor positions showing rows and columns.

To navigate through records for reviewing the data required changing the row and item to be viewed by having an incremental variable (Figure 56).

Once completed the database stored all information from the sensors, motor positions, plant and tray numbers, images as well as time and dates the information was collected.

5.7. Metrology methodology for plant size calculation

Images captured via the camera were imported into VB.net, where algorithms were applied to determine the quantity of plant material present. Each pixel on the image was scanned, with the R, G and B values being interoperated, first by being normalised, then compared to pre determined threshold values. Normalisation puts the R, G and B values in the range of 0 - 1 (as opposed to 0 - 255) this “normal” data made it easier to identify subtle changes in colour – refer equations: (5.1) - (5.3).

As the colour of the tray the plant material is growing in is known, and is a fixed colour, a threshold value for R, G and B was required to be determined to identify the colour of that tray. A program was written to allow these threshold value to be determined using a trial an error approach.

The program allows the user to import a picture in VB.net. By clicking the mouse anywhere on the picture, the program displays the pixel clicked, the normalised value of the RGB content of that pixel, and then displays a sample of that colour on the screen (Figure 57). The user can then set the threshold values by entering values into a textbox. Once threshold values have been determined, the software can automatically run through the image and determine what is plant material and what is the tray. When the software runs

through the algorithm, each pixel and its corresponding RGB values are stored in a data grid view with the total number of pixels representing plant material being displayed (Figure 58).

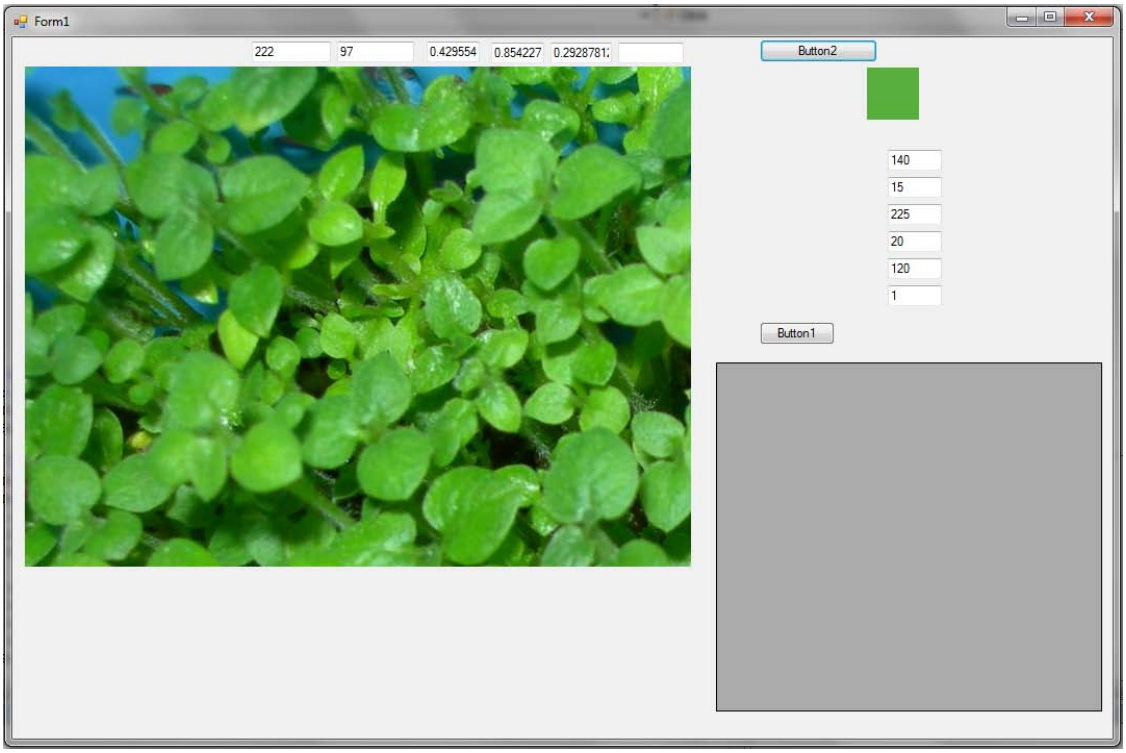


Figure 57: Screenshot of a program designed to identify thresholds for plant growth calculations. It allows a mouse down event to display current pixel and RGB value.

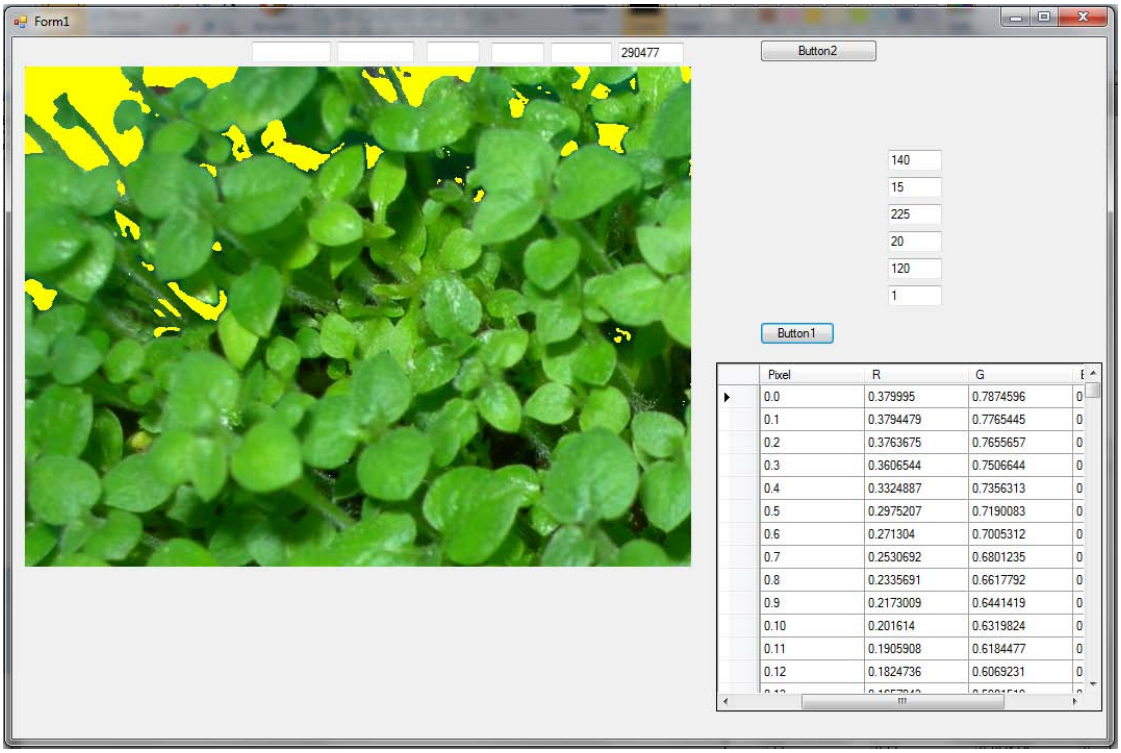


Figure 58: Screenshot of program designed to identify thresholds for plant growth calculations, showing all pixel values in a grid view.

Normalised values of R, G and B are calculated as:

$$R_{Norm} = \frac{R}{\sqrt{R^2 + G^2 + B^2}} \quad (5.1)$$

$$G_{Norm} = \frac{G}{\sqrt{R^2 + G^2 + B^2}} \quad (5.2)$$

$$B_{Norm} = \frac{B}{\sqrt{R^2 + G^2 + B^2}} \quad (5.3)$$

For the image shown in Figure 58, the threshold was determined as:

R < 0.42, G < 0.85, B > 0.5

```
If R1(i, j) < 0.42 And G1(i, j) < 0.85 And B1(i, j) > 0.5
Then
        s.SetPixel(i, j, Drawing.Color.Yellow)
        Yellow = Yellow + 1
EndIf
```

Pixels that fell within the threshold were therefore non-plant material; these pixels were counted and at the conclusion of scanning the entire image subtracted from the total number of pixels. During testing, to ensure the algorithms were correct, each pixel identified as non-plant material was changed to a fixed colour (yellow in this case) as shown in Figure 59 and Figure 60.

Plant material = Total Pixels – non-plant material pixels.

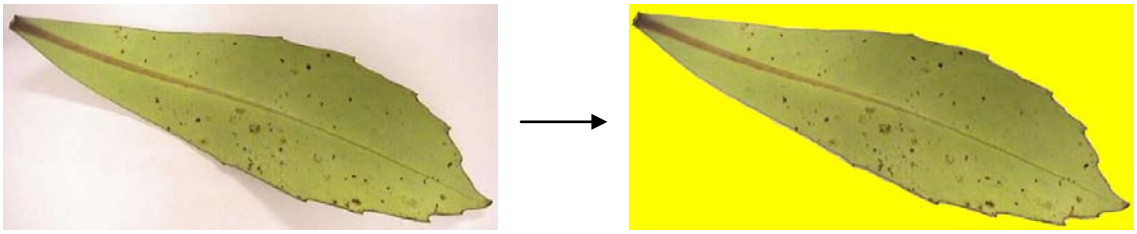


Figure 59: Initial testing of the image analysis algorithm to determine quantity of pixels that make up a leaf on a white background.

Figure 60: Effect of an improved image analysis algorithm to determine quantity of pixels that make up a multiple leafs on a blue background.

This approach gave the user the ability to determine and calculate, in a 2D plane, the surface area occupied by plant material (in total number of pixels). By monitoring the surface area over time it is possible to identify how fast the plant material is growing, as plant material pixels increase (or non-plant material decreases). This can be expressed as a percentage.

As the plants are growing in a modified plant micropropagation system, it is possible that the plant material will move as liquid plant media is pumped in and out of the container, making it difficult to monitor the growth of plant material over time

5.8. Integration of all Parts

Having calibrated and programmed each of the components discussed above, the entire system needed to be integrated into a complete system as shown in the flow chart (Figure 61).

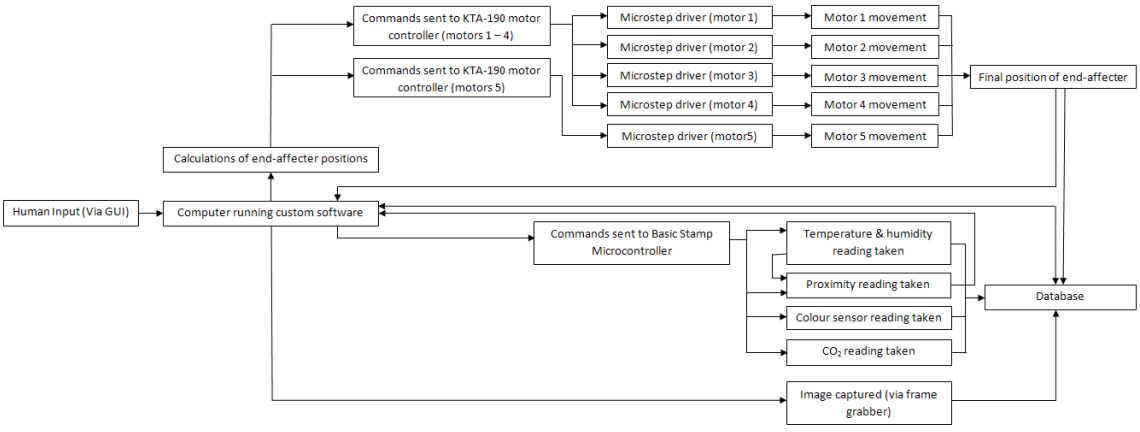


Figure 61: Complete system showing how each part integrates into a complete system.

5.9. Automation of the Entire System

In order to automate the entire system, custom software was implemented that made use of timers, and serial communication to communicate with the various devices outlined above. By making use of the “STAT” command on the KTA-190 motor controller, it was possible to home the robot before implementing movement commands.

5.9.1. Homing the Robotic Arm

Homing the robotic arm requires the user to simply click the “Home Robot” button on the GUI. This assumes the robotic arm has been left in a position where it can be homed, and if not, requires the user to move the arm manually to a suitable position. Once in a suitable position, the first joint of the arm rotates at a very slow speed until it hits a limit switch, at which point the program sends appropriate commands to back the motor off the limit switch, one step at a time. Once the motor is backed off, the motor’s position is reset to 0 via the command ‘POSN’. The software then rotates the motor back in the opposite direction until the opposite limit switch is activated. The arm is then backed off the second limit switch (again one step at a time). Once the arm has been backed off the second limit switch the position of the motor is determined using the ‘PSTT’ command. The arm is then moved a relative movement ‘RMOV’ of half the position (exactly between the two limit switches). Bounds are then set up to ensure the motor cannot get to within 5 steps of the limit switches. This is then repeated for the remaining 2 joints. Homing of the ball shaft and screw initially used the same method described above, however the time taken was inefficient. An alternative method was to use a proximity sensor to determine the position of the arm in the Z-axis. Homing of the pan and tilt mechanism was completed by rotating the mechanism until the proximity sensor’s reading was at minimum, and once this was achieved, a relative movement of 90°ensured the Pan and Tilt mechanism was in a known place. A flow diagram describing the homing operation is presented in Figure 62.

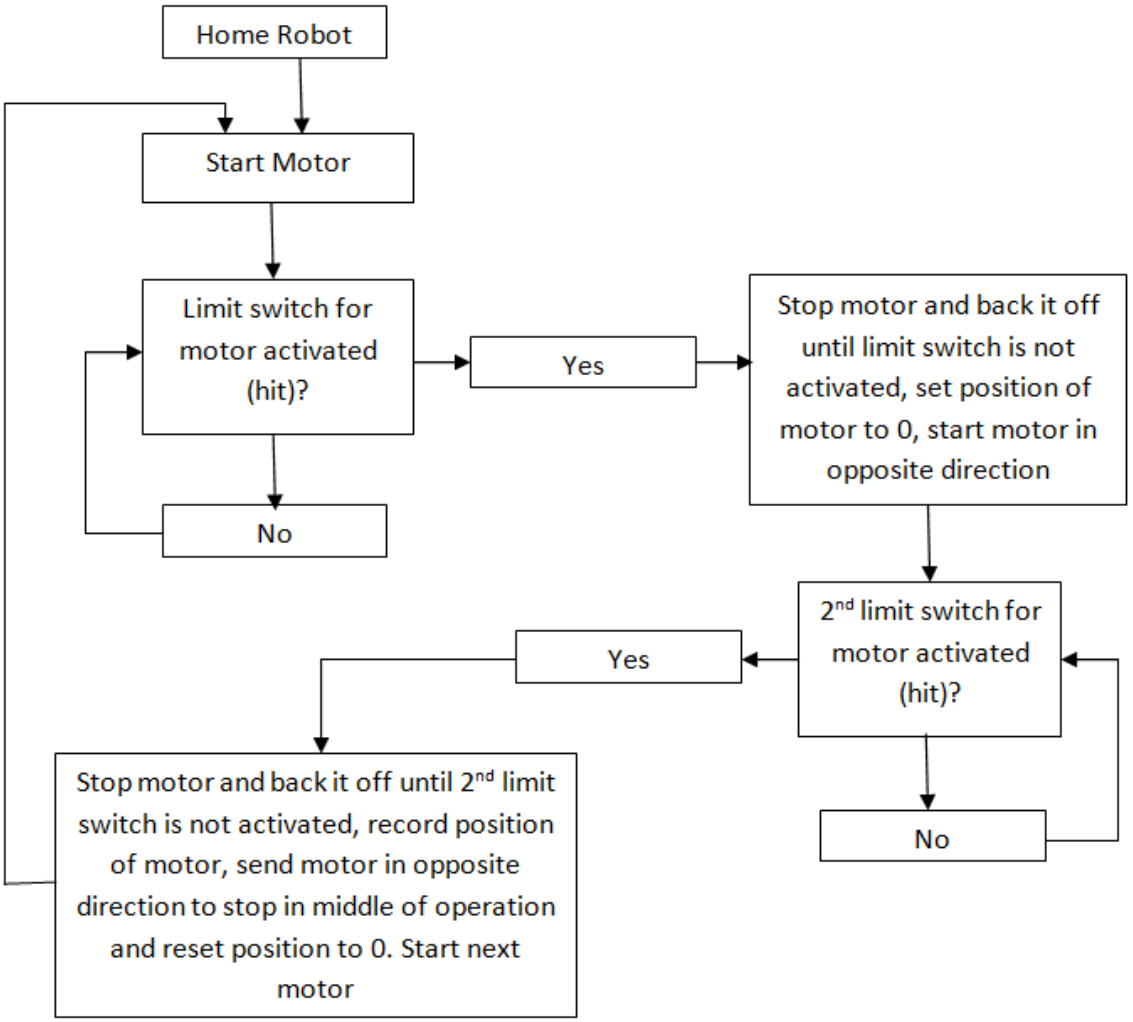


Figure 62: Flow diagram showing method of homing robot (joints 2, 3 & 4).

5.9.2. Manual Motor Control (manual control of the robot)

The user has the ability to move each of the motors manually by selecting how many steps and in which direction to move the motors. The user has further control by manually setting the acceleration settings of each of the motors. In the event a position is entered by the user, or an acceleration setting is set that is not possible, the user is prompted with a warning message before the program ignores the requested command. The user has the ability to stop all five motors at once (Emergency Stop), or can stop each motor independently for any reason. When moving a motor, a timer is activated that continually updates the current position of the motor, therefore if the motor is stopped for any reason, the user will always know where it is in terms of steps from when it was last homed.

5.9.3. Automatic Motor Control (Autonomous control of the robot)

Having successfully homed the robotic arm, the user can set up the robotic arm to autonomously move to predefined positions. Currently, the user is required to select which shelf to send the arm to, followed by which tray to monitor. The user is then prompted to select how many readings are to be taken within that tray. An algorithm then determines where to take the readings from to give an even spread over the tray. The user then selects how frequent readings are required. This information is interpreted and all joint angles and positions that the arm is required to move to, are calculated with the aid of Inverse Kinematic algorithms. Algorithms are implemented to determine how many steps each motor is required to move for each reading. This information is displayed in a datagrid view and the user can review and add further trays of shelves and again review the data. Once the user is satisfied, the information is loaded into the Microsoft Access database. On clicking the “begin automation” the program retrieves the information stored in the database and moves the robotic arm to the corresponding positions. Once in position, the proximity sensor determines how high the arm is above the plant material and adjusts the height of the arm to ensure it is at 20 mm above the plant material. The pan and tilt system then rotates to expose the camera lens to the plant material to be viewed. An image is captured and processed using algorithms to determine plant material quantity present. The pan and tilt system then rotates the opposite way presenting the colour sensor to the plant material, where a colour reading is taken. The information gathered is then transferred back to the database. The arm then repeats the process for the next reading, before moving to the next tray. Once the desired time interval has passed the robotic arm returns back to the original positions by again reading data from the database, and takes another image and colour reading before saving this into the database.

The software has a number of in-built safety features that ensure efficient and safe operation. These include ensuring the time interval between readings is not too short (i.e. the arm has time to take the required number of readings before having to return to a particular point). The acceleration has been programmed to be fixed to ensure the user cannot accidently change the acceleration and

speed. Acceleration can be adjusted by adjusting the program code however, it cannot be adjusted via the GUI.

5.9.4. Overview

The programming of the robotic arm and all the sensors has been a major component of this study. Although the software is working well, in communicating with all the sensors, as discussed, there will always be room for improvement. For the future it would be ideal to be able to display graphs on the GUI showing the colour change of plant material over time, with the user being able to select the particular tray and particular time period, giving the ability to make comparisons between selected trays. Descriptive mathematics could be displayed on screen, such as colour variations overtime and standard deviations of the sensor readings. This would allow the user to review the images captured, displaying them on the GUI, and allowing them to specify a particular tray at a particular time would be an enhancement. In addition, allowing for a video sequence to be created to provide a time lapse video of a selected tray, to be viewed on screen, would be useful thereby allowing the user to watch images captured over a long period of time to be viewed over a short time period. Implementation of the information discussed above would allow the system to be more user friendly, although would require some time to compete.

Once the system is working as desired, more sensors, such as NIR sensors could be programmed and integrated into the system, with information being stored into the database. Further testing of the current sensors would also lead to continual improvement with this being achieved both programmatically and with actual testing.

6. Summary and Future Research

In conclusion this project has required a large amount of work to be completed. The robotic arm has been through a number of design phases, with each design making appropriate changes. The final design has made use of acrylic to ensure the arm is light weight and belt drive system to ensure there is no slippage or backlash as was identified with the use of gears. The arm has been designed to satisfy the requirements set out by Plant & Food Research to be able to monitor plant material growing in trays on a multiple level shelving unit, by hosing a camera, colour sensor and proximity sensor in the end-effecter. Components of the robotic arm were designed in the cad software – SolidWorks, with components being machined in house on a CNC machine.

Investigation into current methodologies used to monitor the plant health revealed a simple method of using colour. Colour has been used in many areas with work detailing the use of colour sensors to identify chlorophyll content of plant. To allow colour readings of plant material a colour sensor was purchased after investigations into various colour sensors. The Colour sensor used, the TCS3200 was calibrated to the CMD-700 spectrophotometer, with different calibration methods used, with the most accurate making use of a gamma calibration, with results of $(\text{Red})\,\lambda_{\text{R}} = 1.05$, $(\text{Green})\,\lambda_{\text{G}} = 0.92$, $(\text{Blue})\,\lambda_{\text{B}} = 1.00$. Alternative methods were investigated including that of a least squares regression. The colour sensor was integrated with a number of other sensors, to ensure correct readings were taken, including a proximity sensor and temperature sensor. The colour sensors have been successfully tested with the monitoring of a leaf dying over time, with further testing required by scientist to determine what colour is deemed a “healthy” plant and what is not. A Sony Camera has been integrated, to allow for images to be captured. The user has the ability to control most features of the camera, through the software, with algorithms being implemented to allow plant material calculations to be calculated. This will allow a method of calculating plant material quantity and monitoring this over time.

Custom software has been design to allow custom operation of the robotic arm, allowing for the arm to be programmed by an operator to get end-effecter to move to specified trays on specified shelf's, to capture relevant data. The arms joint angles are calculated using Inverse Kinematics, with all information pertaining to positions recoded and stored into a Microsoft Access database allowing with the data recorded by the sensors. Programming of the sensors has allowed calibrations calculations to be taken, ensuring the sensors (proximity, temperature and humidity) are providing accurate results. The sensors have been integrated into the microcontroller, which has allowed for communication between the computer and the sensors.

Overall the project has delivered to Plant & Food Research a robotic system, capable of monitoring plant material growth using colour and images. This will allow scientists to conduct research into plant health status to identify subtle changes in colour. The system has been entirely automated allowing for round the clock monitoring to be conducted with little human input.

Future work will involve further calibration of the colour sensors, using the same methodology described in this report, however extending it over a large range of colours to possibly achieve a greater accuracy. Once NIR sensors are investigated further, one could integrate an NIR sensor into this project, allowing for readings to be conducted autonomously. Integration of a TCP/IP connection allowing for monitoring of the system to be completed anywhere in the world using the internet. This would allow for real time information of the system to be available to anywhere in the world – allowing for multiple systems to be integrated into different environments worldwide. Making the robotic arm utilise vision, to identify where the greatest collection of plant material exists in a tray and allowing the colour sensor to only take readings of this collection of plant material. Finally allowing the robotic system to be able to move along the ground, this would allow multiple shelf's to be linked together allowing for more trays to be monitored.

7. References

1. M. Seelye. G. Sen Gupta. J. Seelye & S.C. Mukhopadhyay. (2010). Camera-in-hand Robotic System for Remote Monitoring of Plant Growth in a Laboratory. *Proceedings of IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2010)*, 809-814.
2. IOS. [cited 2010 10 March]; Available from: http://www.iso.org/iso/home.html.
3. D.C. Slaughter. D.K. Giles & D. Downey. (2008). Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture, 61*(1), 63-78.
4. M.J. Aitkenhead. I.A. Dalgetty. C.E. Mullins. A.J.S. McDonald & N.J.C. Strachan. (2003). Weed and crop discrimination using image analysis and artificial intelligence methods. *Computers and Electronics in Agriculture, 39*(3), 157-171.
5. B. Astrand & A.J. Baerveldt. (2002). An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous Robots, 13*(1), 21-35.
6. J. Hemming & T. Rath. (2001). Computer-Vision-based Weed Identication under Field Conditions using Controlled Lighting. *Journal of Agriculture and Engineering Resources., 78*(3), 233-243
7. G.J. Garcia. J. Pomares & F. Torres. (2009). Automatic robotic tasks in unstructured environments using an image path tracker. *Control Engineering Practice, 17*(5), 597-608.
8. R. Kelly. R. Carelli. O. Nasisi. B. Kuchen & F. Reyes. (2000). Stable Visual Servoing of Camera-in-Hand Robotic Systems. *Itransactions On Mechatronics, 5*(1), 39 - 48.
9. T. Kataoka. T. Kaneko. H. Okamoto. S. Hata & I. Ieee. (2003). Crop growth estimation system using machine vision. *International Conference on Advanced Intelligent Mechatronics*, 1079-1083.
10. A.J. Scarfe. R.C. Flemmer. H.H. Bakker & C.L. Flemmer. (2009). Development of An Autonomous Kiwifruit Picking Robot. *Proceedings of the Fourth International Conference on Autonomous Robots and Agents*, 639-643.
11. J. Baeten. K. Donne. S. Boedrij. W. Beckers & E. Claesen. (2008). Autonomous fruit picking machine: A robotic apple harvester. *Field and Service Robotics: Results of the 6th International Conference, 42*, 531-539.
12. E.J. van Henten. J. Hemming. B.A.J. van Tuijl. J.G. Kornet. J. Meuleman. E.A. van Os, et al. (2002). An autonomous robot for harvesting cucumbers in greenhouses. [Article]. *Autonomous Robots, 13*(3), 241-258.
13. N.D. Tillett. (1993). Robotic Manipulators in Horticulture, A review. *Journal of Agricultural Engineering Research, 55*(2), 89-105.
14. N.D. Tillett. W. He & R.D. Tillett. (1995). Development Of A Vision-Guided Robot Manipulator For Packing Horticultural Produce. *Journal of Agricultural Engineering Research, 61*(3), 145-154.
15. M. Fattal. A.E. Turk. M. Dahrouj & A. Badereddin. Automatic Apple Sorting and Grading using Machine Vision.
16. X. Liming & Z. Yanchao. (2010). Automated strawberry grading system based on image processing. [Proceedings Paper]. *Computers and Electronics in Agriculture, 71*, S32-S39.
17. N. Kondo. (2010). Automation on fruit and vegetable grading system and food traceability. [Review]. *Trends in Food Science & Technology, 21*(3), 145-152.
18. E.R. Davies. (2009). The application of machine vision to food and agriculture: a review. *The Imaging Science Journal, 57*(4), 197-217.

19. P.M. Granitto. P.F. Verdes & H.A. Ceccatto. (2005). Large-scale investigation of weed seed identification by machine vision. [doi: DOI: 10.1016/j.compag.2004.10.003]. *Computers and Electronics in Agriculture, 47*(1), 15-24.
20. P.J. Sobey. B. Harter & A. Hinsch. (1997). Automated micro-propagation of plant material. *Fourth Annual Conference on Mechatronics and Machine Vision in Practice*, 60-65.
21. H.T. Sogaard & I. Lund. (2007). Application accuracy of a machine vision-controlled robotic micro-dosing system. *biosystems engineering, 96*(3), 315-322.
22. M. Bamsey. A. Berinstain. T. Graham. P. Neron. R. Giroux. S. Braham, et al. (2009). Developing strategies for automated remote plant production systems: Environmental control and monitoring of the Arthur Clarke Mars Greenhouse in the Canadian High Arctic. *Advances in Space Research, 44*(12), 1367-1381.
23. C.H. Su. C.C. Fu. Y.C. Chang. G.R. Nair. J.L. Ye. I.M. Chu, et al. (2008). Simultaneous estimation of chlorophyll a and lipid contents in microalgae by three-color analysis. [Article]. *Biotechnology and Bioengineering, 99*(4), 1034-1039.
24. S.P. Yadav. Y. Ibaraki & S.D. Gupta. (2010). Estimation of the chlorophyll content of micropropagated potato plants using RGB based image analysis. *Plant Cell Tissue and Organ Culture, 100*(2), 183-188.
25. D. Cui. M.Z. Li & Q. Zhang. (2009). Development of an optical sensor for crop leaf chlorophyll content detection. *Computers and Electronics in Agriculture, 69*(2), 171-176.
26. C. Miranda. T. Girard & P.E. Lauri. (2007). Random sample estimates of tree mean for fruit size and colour in apple. *Scientia Horticulturae, 112*, 33-41.
27. S.P. Kang & H.T. Sabarez. (2009). Simple colour image segmentation of bicolour food products for quality measurement. *Journal of Food Engineering, 94*, 21-25.
28. K.L. Yam & S.E. Papadakis. (2004). A simple digital imaging method for measuring and analyzing color of food surfaces. *Journal of Food Engineering, 61*, 137-142.
29. S. Borah & M. Bhuyan. (2005). A computer based system for matching colours during the monitoring of tea fermentation. *International Journal of Food Science and Technology, 40*, 675-682.
30. M.Z. Abdullah. S.A. Aziz' & A.M.D. Mohamed. (2000). Quality Inspection Of Bakery Products Using A Color-Based Machine Vision System. *Journal of Food Quality, 23*, 39-50.
31. A.S. Raja & K. Sankaranarayanan. (2006). Use of RGB Color Sensor in Colorimeter for better clinical measurments of blood Glucose. *BIME Journal 6*(1), 23 - 28.
32. P. Paz. M.ı.-T. S´anchez. D. P´erez-Mar´ın. J.e.-E. Guerrerob & A. Garrido-Varob. (2009). Evaluating NIR instruments for quantitative and qualitative assessment of intact apple quality. *inWiley Interscience, 89*, 781-790.
33. P. Menesatti. F. Antonucci. F. Pallottino. G. Roccuzzo. M. Allegra. F. Stagno, et al. (2010). Estimation of plant nutritional status by Vis–NIR spectrophotometric analysis on orange leaves. *Biosystems Engineering, 105*, 448-454.
34. B.M. Nicola¨ı. K. Beullens. E. Bobelyn. A. Peirs. W. Saeys. K.I. Theron, et al. (2007). Nondestructive measurement of fruit and vegetable quality by means of NIR spectroscopy: A review. *Postharvest Biology and Technology, 46*, 99-118.
35. G. Costa. G. Fiori & M. Noferini. (2006). Using NIRs to Determine Intrinsic Fruit Quality and Harvest Date. *Proc. 6th Intl. Peach Symposium*, 435-440.
36. T.-T. Guo. B. Zhang. L. Guo. D.-S. Li. Y. Wu. J.-J. Wu, et al. (2010). Classification of Plant Leaves by Near-infrared Spectroscopy using ANN and Wavelet. *Second International Workshop on Education Technology and Computer Science*.
37. T.-t. Guo. L. Guo. X.-h. Wang & M. Li. (2009). Application of NIR Spectroscopy in Classification of Plant Species. *International Workshop on Education Technology and Computer Science, 1*, 879-883.

38. H. Lin & Y. Ying. (2009). Theory and application of near infrared spectroscopy in assessment of fruit quality: a review. *Sensers & Instrumentation in Food Quality, 3*, 130-141.
39. C. Camps & D. Christen. (2009). Non-destructive assessment of apricot fruit quality by portable visible-near infrared spectroscopy. *Food Science and Technology, 42*, 1125-1131.
40. X. Hu. Y. He. A.G. Pereira & A.H. Gómez. (2005). Nondestructive Determination Method of Fruit Quantity Detection Based on Vis/NIR Spectroscopy Technique. *Engineering in Medicine and Biology 2*, 1956-1959.
41. M.N. Merzlyak. A.A. Gitelson. O.B. Chivkunova. A.E. Solovchenko & S.I. Pogosyan. (2003). Application of reflectance spectroscopy for analysis of higher plant pigments. *Russian Journal of Plant Physiology, 50*(5), 704-710.
42. D.-J. Lee. R. Schoenberger. J. Archibald & S. McCollum. (2008). Development of a machine vision system for automatic date grading using digital reflective near-infrared imaging. *Journal of Food Engineering, 86*, 388-398.
43. R. Shelton & S.L. Smith. *NASA - JSC Learning Technologies*. 2009 [cited 2010 4 April]; Available from: http://prime.jsc.nasa.gov/ROV/types.html.
44. G. Figliolini & P. Rea. (2006). Overall design of Ca.U.M.Ha. robotic hand for harvesting horticulture products. [Article]. *Robotica, 24*, 329-331.
45. M. Ceccarelli. G. Figliolini. E. Ottaviano. A.S. Mata & E.J. Criado. (2000). Designing a robotic gripper for harvesting horticulture products. [Article]. *Robotica, 18*, 105-111.
46. M.M. Foglia & G. Reina. (2006). Agricultural robot radicchio harvesting. [Article]. *Journal of Field Robotics, 23*(6-7), 363-377.
47. P. Ben-Tzvi. A.A. Goldenberg & J.W. Zu. (2010). Articulated hybrid mobile robot mechanism with compounded mobility and manipulation and on-board wireless sensor/actuator control interfaces. [Article]. *Mechatronics, 20*(6), 627-639.
48. R. Manseur, *Robot Modeling and Kinematics*. 2006, Boston, Massachuseetts: Charles River Media.
49. H. Choset. K. Lynch. S. Hutchinson. G. Kantor. W. Burgard. L. Kavraki, et al., *Principles of Robot Motion: Theory, Algorithms, and Implementation*. 2005, Cambridge, Massachusetts: MIT Press.
50. A.A. Goldenberg. (1985). A Complete Generalized Solution to the Inverse Kinematics of Robots. *Robotics And Automation, 1*(1), 14-20.
51. J.Q. Gan. E. Oyama. E.M. Rosales & H.S. Hu. (2005). A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm. *Robotica, 23*, 123-129.
52. J. T.Wunderlich. (2004). Simulating a Robotic Arm in a Box: Redundant Kinematics, Path Planning, and Rapid Prototyping for Enclosed Spaces. *The Society for Modeling and Simulation International, 80*(6), 301-316.
53. J.J. Craing, *Introduction to Robotics: Mechanics & Control*. 1 ed. 1986, Palo Alto: Addison-Wesley Publishing Comapany.
54. J. Lenarcic & C. Galletti, *On Advances in Robot Kinematics*. 2004, Dordrecht, The Netherlands: Kluwer Academic Publishers.
55. W. Stadler, *Analytical Robotics and Mechatronics*. 1995, San Francisco: McGraw-Hill Companies.
56. B.J. Lindbloom. *BruceLindbloom*. 2010 15/8/2010]; Available from: www.BruceLinbloo.com.
57. P.J. Fleming & J.J. Wallace. (1986). How not to lie with statistics: The correct way to summarize the benchmark results. *Communications of the ACM, 29*(3).
58. Microsoft. *http://msdn.microsoft.com/en-us/vbasic/default*. 2011 [cited 2010 - 2011.
59. R. Juckett. *RGB color space conversion - Linear transformation of color*. 2010 14/8/2010]; Available from: http://www.ryanjuckett.com.

8. Appendix

8.1. Publications to Date

Camera-in-hand Robotic System for Remote Monitoring of Plant Growth in a Laboratory

Mark Seelye[1], Gourab Sen Gupta[1], John Seelye[2], S. C. Mukhopadhyay[1]
[1]School of Engineering and Advanced Technology (SEAT),
MasseyUniversity, Palmerston North, New Zealand
[2]The New Zealand Institute of Plant & Food Research Limited
Private Bag 11 600, Palmerston North, New Zealand
Email: m.a.s.01@hotmail.com, G.SenGupta@massey.ac.nz

***Abstract*–A remote controlled robotic arm has been designed and fabricated for the purpose of remote monitoring of plant tissue growth in research projects conducted by Plant & Food Research Limited, New Zealand. The robotic arm consists of a light weight aluminum tubing with low voltage, high torque servo motors providing the movements for the various joints. The arm can swivel 180 degrees on its base. A compact colour zoom camera, on a pan and tilt mount, is fixed to the end of the arm. Customised Visual Basic software has been developed to control the arm through a servo control board. The software can capture the plant images, including video sequences. The position of the arm and all camera operations can be remotely controlled, either from the GUIon the PC screen, or through a joystick. The arm can be programmed to move to predefined positions at set times and capture images automatically without the need for human input. The entire system has been made wireless using Zigbee transceivers. The system has been successfully tested in a laboratory environment.**

Keywords – Robotic arm, plant growth monitoring, Zigbee, wireless communication, remote control

I. INTRODUCTION

Today robots are becoming more and more popular and are increasingly relied upon to undertake routine, often repetitive tasks which are expensive to do using a highly paid labor force. The International Organization for Standardization defines a robot in ISO 8373 as: "an automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications".

The use of robotics and automated machinery has been relatively limited in the horticulture industry, although as technology moves forward more possibilities are becoming realties. Several research articles have recently appeared in literature; the current status of the core technologies required for the successful development of a robotic system for weed control has been reviewed in [1]. A system for automatic harvesting of cut flowers grown in greenhouses has been developed and reported in [2]. Advancement in electronics has empowered engineers to build robots that are capable of operating in unstructured environments [3]. Camera-in-hand robotic systems are becoming increasingly popular wherein a camera is mounted on the robot, usually at the hand, which provides an image of the objects located in the robot's workspace [4]. More and more robots are being used to sort, grade and package fruits. In [5], the system identifies and classifies fruits moving on a conveyor belt. Robotic techniques for production of seedlings have been presented in [6]. The work reported in [6] has identified a need to add machine vision system to detect irregularities in seed trays and provide supplemental sowing using a 5-arm robot.

Micro-propagation for rapid multiplication of plants and micro-dosing precise application of chemicals, such as herbicides, is gaining the attention of researchers. A system has been detailed in [7] which combines plant recognition, micro-dosing and autonomous robotics. An autonomous robotic system has been described in [8] for micro-propagation of a grass species which replaces the cost-intensive and tedious manual process.

Two very important topics of research and development in the domain of horticulture is vision processing and wireless communication. All the systems reported in [2-8] use some aspects of imaging and vision processing. A wireless solution for intelligent field irrigation system has been elaborated in detail in [9].

Apart from data collected, such as ph levels, temperatures, moisture content and light levels, images of plants are becoming increasingly more common to identify possible changes over time for new plants so that corrections can be made to optimize growth. The work detailed in this paper is an effort to design and develop an anthropomorphic robotic arm for the New Zealand Institute of Plant & Food Research to monitor plant or tissue growth. The system was required to have wireless capabilities to capture and transmit images on a regular basis using a camera. The specifications for the arm included the ability to move it anywhere around a 400mm x 600mm tray of tissue with a high resolution camera (mounted at the end of the arm) ideally with zoom capability to capture images of the plant during various stages of growth. A user friendly control of the arm using a joystick was desirable and making the system wireless would enable viewing of the tissue to done away from the growing and testing environment of the laboratory.

The rest of the paper is organized as follows- Section II gives the system overview and architecture. The hardware is detailed in section III while the

software implementation details are in Section IV. The modular testing and systems integration procedures are presented in Section V. The paper ends with a discussion and enumeration of future work in section VI.

II. SYSTEM OVERVIEW

The basic concept of this project, from human input to the control of the arm and camera, is outlined in the functional block diagram shown in figure 1. A systems engineering approach was employed to take the robotic arm from concept to reality, making use of standard components freely available and integrating them together to make the final product.

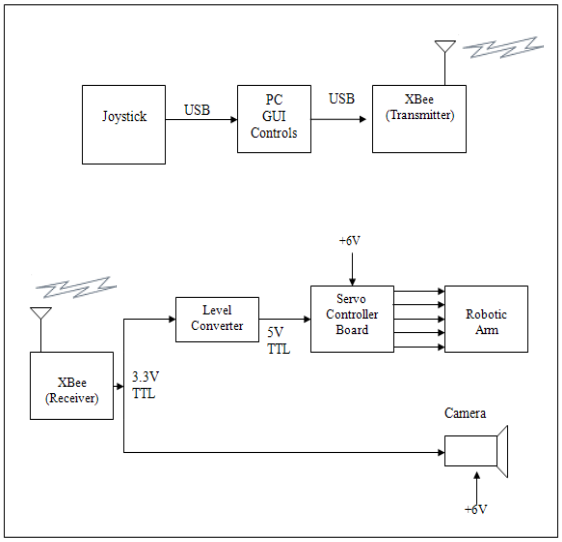


Fig. 1. Functional block diagram of the robotic system

The operator uses a joystick to control the movement of the robotic arm. This joystick connects to the PC via a USB interface. Movements of the joystick, made by the operator, vary the slide bars on the Graphical User Interface (GUI) running on the PC and at the same time control the movement of the joints of the arm. Serial data is then sent via the USB to the wireless transmitter module which transmits the data to the wireless receiver module.

The wireless receiver module receives the data from the transmitter. The received data is then sent to the camera and the servo controller board. The servo controller board requires a level converter to raise the 3.3V TTL serial signal to 5V TTL serial signal.

In the following sections, the implementations of the robotic system are presented with detailed descriptions of the different hardware components used and the software implementation.

III. SYSTEM HARDWARE

The complete system consists of several modules, each of which is described in the following sub-sections.

A. Computer

A PC running Microsoft Windows XP, with the winmml.dll driver installed, and at least one COM port (or USB port) available is necessary with Microsoft Visual Studio Express Edition 2005 (VB.net 2005) providing the programming software.

B. Joystick Input

In order to give the operator full control over the position of the arm and camera controls, two options were pursued. In the first option, a Microsoft gaming joystick was used; this allows the user to move the arm left, right, forward, back, allows the pan and tilt mechanism on the arm, and camera functions to be controlled.

Once the arm has been positioned with the joystick, pressing button 1 locks the arm in place, allowing only the pan and tilt mechanism to finely position the camera. Pressing button 2 locks the pan and tilt mechanism, allowing a still image to be viewed on screen. This button also records and stores the position of the servos in an array for subsequent use. The throttle function of the controller provides control of the camera's zoom function and a third button captures the image. A forth button then releases the arm in order to repeat the steps. The user can also pre-program the arm to move to predefined positions allowing the arm to automatically return to those positions for time-lapsed image capturing. This is achieved by recording the position of all six servos, storing these position values and then running through a loop setting the corresponding servo's position to that of the stored position.

A second option for controlling the arm requires turning off the joystick and simply moving the on-screen slide (scroll) bars to manually set the position of individual servo motors.

C. ZigBee Wireless Transceivers

ZigBee RF modules are designed to provide wireless transmission of data in low power and low cost applications. These modules use IEEE 802.15.4 physical layer radio protocols and operate in the unlicensed band worldwide at a frequency of 2.4 GHz. ZigBee RF modules have been used and shown to be effective in a wireless low-cost intelligent irrigation system [9]. Researchers have also used ZigBee for wireless control of mobile robots [10].

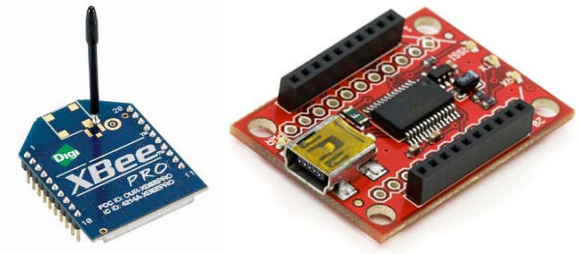


Fig. 2.Digi International's XBee wireless module and USB Explorer board

In order to implement a wireless control of the robot arm and camera, a Digi International XBee Pro series 2 RF (radio frequency) module, shown in figure 2, was interfaced between the controlling PC and the robotic arm (and camera). The XBee RF modules require a 3.3V power source and produce a TX current of 295mA and an RX current of 45mA.

To setup a wireless mesh network, each wireless module was programmed using Digi's X-CTU software. Each module can be configured via a USB explorer board (figure 2) which facilitates simple programming of the XBee module. The series 2 modules used have a XB24-B modem type and can be configured as a Coordinator, Router or End device. The Coordinator can transmit and receive RF data but cannot assist in routing data through the mesh network. The Router can also transmit and receive RF data, and can route data packets through the network. An End device can transmit or receive RF data but can't assist in routing data through the network. XBee modules have a claimed indoor/urban range of up to 90 meters (with a maximum line-of-sight range of 1.6km) with a RF data transfer rate of up to 250kbps.

For this application the transmitting module is programmed as the ZigBee CoordinatorAT, using software version 1047. The receiver module is programmed as ZNET 2.5End Device AT using software version 1247. The XBee modules support both transparent (referred to as AT programming) and API (Application Programming Interface) serial interfaces. For this application the transparent (AT) interface has been used.

The Xbee modules convert the command sent through the serial port into serial data at a TTL level transmitting it wirelessly. The router/end device receives this serial data. Using the USB Explorer board allows for easy access to the pins of the XBee modules. The XBee modules feature a UART In (pin DI) and UART Out (pin DO) line (TX and RX lines) to send and receive data at TTL level.

Data enters the XBee module through the DI pin as an asynchronous serial signal. The signal should idle high when no data is being transmitted. Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). Figure 3 illustrates the serial bit pattern of data passing through the module.

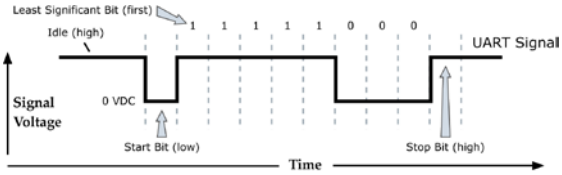


Fig. 3. Serial bit pattern of transmitted data

D. Camera

A Sony color camera (model: FCB-IX11AP), shown in figure 4, was used. It features a ¼" CCD (charge coupled device) image sensor using PAL (Phase Alternating Line) encoding system. The camera has a 40x zoom ratio (10x optical, 4x digital) which is controllable from a PC via Sony's VISCA command protocol. The camera's macro capability allows it to capture images as close as 10mm from the subject.

The camera is extremely compact at 65mm x 44.8mm x 39.3mm, weights 95g and produces high quality images. It has numerous controllable functions (up to 38,000 different command combinations) including digital and optical zoom, aperture setting, shutter speed settings and iris settings. The camera requires a stable 6V to 12V DC input voltage to operate.

Fig.4. Sony FCB-IX11AP Color camera

The camera can operate in light conditions as low as 1.5 lux and temperatures ranging from 0 to 50 degrees centigrade. The electronic shutter speed is controllable from 1/10 to 1/10000 of a second allowing for clarity in photos.

When connected wirelessly using XBee modules, TTL outputs, DO and DI pins, on the XBee are connected to the camera's TTL TD and TTL RD inputs respectively. In order for the camera's zoom function to be controlled via the throttle feature of the joystick, hexadecimal commands (as serial data) are sent to the camera. The serial commands sent to the camera (figure 5) require 8 data bits, 1 start bit, 1 (or 2) stop bit, no parity, and have a communication speed of 9.6kbps, 19.2 kbps or 38.4kbps. These specifications are contained in the packet which is sent to the camera to control the zoom.

E. Servo Controller

A 16-servo controller board (Pololu, USA), shown in figure 5, provides the communication between the PC and the servos. The servos are programmed by sending commands through the serial port at a baud rate of 9600. The commands contain 8-bitdata with no parity and one stop bit. The servo board has direct USB interface and also UART (TTL) interface with the ability to independently control the speed and range of each of the 16 servos while delivering 0.5 microsecond resolution.

Fig. 5. Pololu USB 16-servo controller board

F. Robotic Arm

Standard analogue servos (Hitech, Japan) commonly used in remote controlled applications (e.g. remote controlled cars and planes) were used in conjunction with standard servo components (Lynxmotion, USA). To keep the arm lightweight,

tubular aluminum sections have been used. The servos are plugged into the controller board and powered through an 18VA DC power supply.

Fig. 6. Servo motor and erector parts

Two analogue servo models were used for the arm;servos 1, 4, 5 and 6 utilized the Hitech HS-475HB Super pro servo, which produced 5.5kg/cm torque (when operating at 6V) and an operating speed of 60 degrees/0.18 sec. Servos 2 and 3 required a greater torque to move the entire arm. These comprised Hitech super torque servo (HS-985 MG) which produced 12.4 kg/cm torque (when operating at 6V) and an operating speed of 60 degrees/0.13 sec.

Standard aluminum components comprising servo brackets and connectors, tube connector hubs and 150 mm tubing segments were sourced from Lynxmotion, USA. To assist the servos, a series of springs (Century Spring Corp., USA) provided additional support to the lower arm servos. A servo motor and erector parts are shown in figure 6. Figure 7 shows the CAD drawing of the robotic arm.

Fig. 7. Solidworks CAD drawing of robotic arm

IV. SOFTWARE IMPLEMENTATION

Microsoft Visual Basic 2005 Express Edition was the language selected for this application due to its ease of use. The program runs in a unique way in which it makes use of a timer application. This allows the program to be continuously running every 10th of a second, allowing for smooth movements of both the arm and the camera zoom by repeatedly sending commands to either the servo controller or the camera. The GUI of the application is shown in figure 8. The joystick may be turned on or off from the GUI. When turned off, the robot joint movements can be controlled by the slide bars.

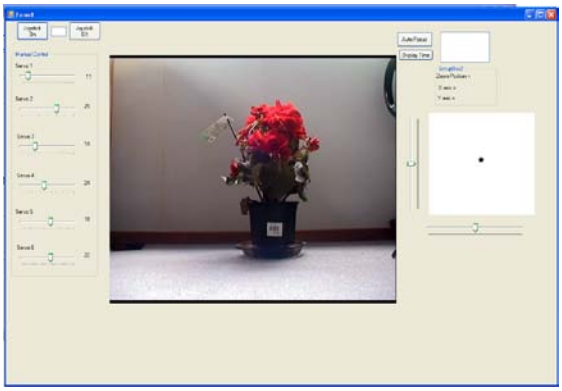


Fig.8.Graphical User Interface (GUI) of the remote control application (camera zoom 1x)

G. Joystick Control

The joystick control relies on the winmm.dll driver being installed. This Windows Multimedia API driver contains low-level audio and joystick functions. Once the driver is loaded a number of functions are possible through VB.Net using the following subroutines:

- joyGetDevCaps - queries the connected joystick to determine its capabilities and determine the number of joystick devices supported by the driver.
- joyGetNemDevs - determines whether a given joystick is physically attached to the system and returns an error if the specified joystick is not connected
- joyGetPos - queries a joystick for its position and button status allowing the application to make use of any movement and or button press events.

With the above subroutines imported into the program, coordinates resulting from the movement of the joystick, joystick throttle and the current button number pressed (resulting from which button on the joystick is pressed) allow for the arm to be programmed.

H. Camera control

The camera is controlled using Sony's VISCA command protocol. The basic unit of VISCA communication is a packet of 3 to 16 bytes. The first byte of the packet is the header consisting of the sender's and header's addresses; the remainder of the packet holds the commands (or message) to be sent to the camera (figure 9). The packet's final byte comprises a terminator signifying the end of the packet and usually has the hexadecimal value FF (or 11111111 binary).

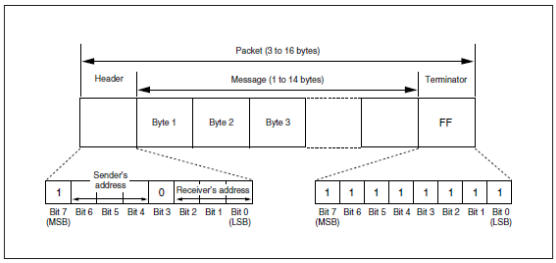


Fig. 9. Serial command data packet layout required to communicate with camera

I. Servo Control

In order to set the servo position, three bytes are required to be sent to the servo controller board. The first byte contains a synchronization value that is fixed at 255, while byte 2 contains information relating to the servo number (a value between 0 – 255) and byte 3 contains the position which the servo moves to (again a value between 0 – 255).

To control the arm using a joystick, the position of the corresponding servos are incremented as the joystick is pushed in the +X direction (moving the arm away) or decremented if the joystick is pulled back in the -X direction (moving the arm closer). A similar approach is used to rotate the arm left and right in the + and – Y direction through the base. With the arm locked, the cameras pan and tilt mechanism is controlled in the same manner by incrementing or decrementing the position byte.

V. TESTING AND INTEGRATION

Testing was done in parts, a modular approach. The sub-systems that required to be programmed, such as the servo-controller, Joystick and camera, were tested separately. The testing setup and procedure are explained in this section.

RS232/TTL Control of Camera

The camera can be programmed and controlled using TTL or RS232C signal level serial interface. To test the camera features, it was directly wired to the PC using the RS232C interface via a USB-to-RS232 converter as shown in figure 10. The video output signal from the camera was fed to a frame grabber/digitizer which is interfaced to the PC using USB. The image captured is shown on the application GUI.

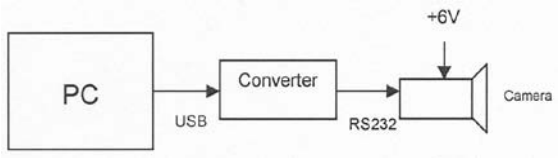


Fig. 10. Block diagram showing wired connection of PC to camera's RS232 inputs

Using a standard communication program (Terminal v1.9b), commands were sent to the camera to familiarize with the VISCA command structure and to test the various camera functions, especially the programming commands for controlling the zoom. In the actual system, however, the camera is controlled using TTL level signals. To test the TTL interface, the system shown in figure 11 was developed. IC ST232 was used to convert the RS232 level signals to 5V TTL. The camera functions were successfully tested. Figure 12 shows the image with camera set to 8x zoom.

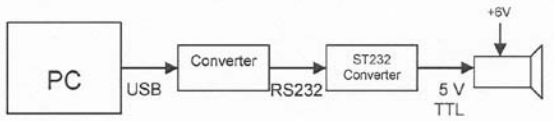


Fig. 11. Block diagram showing wired connection of PC to camera's TTL inputs

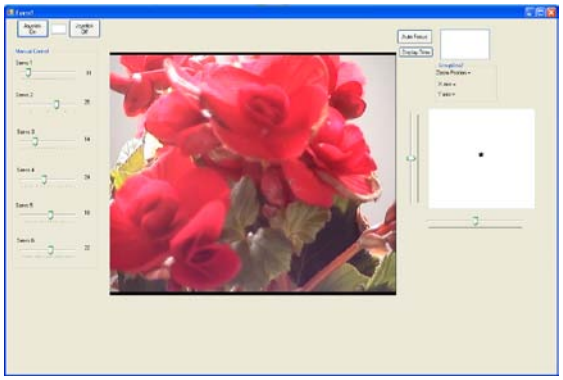


Fig.12.Graphical User Interface (GUI) of the remote control application (camera zoom 8x)

Servo Controller and Joystick

The servo controller board also can take two types of signals – USB and 5V TTL. The controller board, together with the joystick, was tested using the connection diagram shown in figure 13.

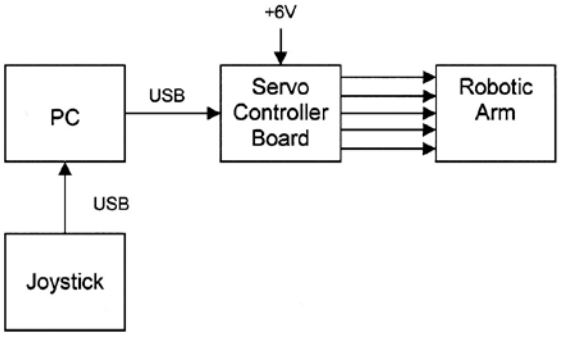


Fig. 13. Block diagram showing connection between PC, Servo Controller Board and Joystick

In the first instance a simple program was written in Visual Basic, allowing one servo motor to be controlled by clicking a button. This action sent the corresponding command to the servo board. The next step was to control the servo by implementing a slide bar. This allowed the operator to slide the bar, which incremented or decremented the position value, allowing for simple movement based on the position byte. On successfully implementing one servo, multiple servos could then be controlled in the same manner.

Finally the entire system was integrated into one final "product". Once all the individual component issues were sorted, the integration to the final product was straight forward.

VI. DISCUSSIONS AND FUTURE DEVELOPMENT

The system has been successfully completed and has been tested in its intended environment. Although a number of issues have arisen during implementation and testing, they have all been resolved in one way or another and have allowed for Plant & Food Research to continue their research into plant growth monitoring. The completed robotic system, with plant trays, is shown in figure 14.

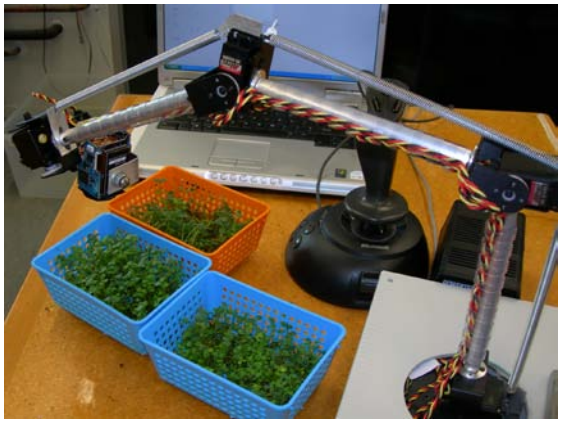


Fig. 14. The completed robotic arm with camera, joystick and plant trays

Future work includes mounting the arm on a movable platform to allow the entire arm to move up and down rows of trays; transmitting the captured images from the laboratory to a remote monitoring station using RF transceivers; and undertaking image analysis of the captured images to automatically measure plant or tissue size over time which can be used to calculate growth rates.

VII. REFERENCES

[1] Slaughter, DC; Giles, DK;Downey, D., "Autonomous robotic weed control systems: A review", Computers And Electronics In Agriculture", Vol.61, 2008, pp. 63-78

[2] Kawollek, M; Rath, T., "Robotic Harvest of Cut Flowers Based on Image Processing by Using Gerbera jamesonii as Model Plant", Proceedings of the International Symposium On High Technology for Greenhouse System Management (Greensys 2007), OCT 04-06, 2007, Naples, ITALY, pp.557-563

[3] Garcia, GJ; Pomares, J; Torres, F, "Automatic robotic tasks in unstructured environments using an image path tracker", Control Engineering Practice, Vol. 17, 2009, pp. 597-608

[4] Kelly, R; Carelli, R; Nasisi, O; Kuchen, B; Reyes, F, "Stable visual servoing of camera-in-hand robotic systems", IEEE-ASME Transactions On Mechatronics, Vol. 5, 2000, pp. 39-48

[5] Reyes, JF; Chiang, LE., "Location And Classification Of Moving Fruits In Real Time With A Single Color Camera", Chilean Journal Of Agricultural Research, Vol. 69, 2009, pp. 179-187

[6] Wang HonYong; Cao QiXin; Masateru, N.; Bao JianYue, "Image processing and robotic techniques in plug seedling production", Transactions of the Chinese Society of Agricultural Machinery, Vol. 30, 1999, pp. 57-62

[7] Sogaard, H. T.; Lund,I., "Application accuracy of a machine vision-controlled robotic micro-dosing system.", Biosystems Engineering, Vol. 96, 2007, pp. 315-322

[8] Otte, C; Schwanke, J; Jensch, P, "Automatic micropropagation of plants", Conference on Optics in Agriculture, Forestry, and Biological Processing, Boston, USA, Nov 19-20, 1996, pp. 80-87

[9] Zhou, Yiming; Yang, Xianglong; Wang, Liren; Ying, Yibin, "A wireless design of low-cost irrigation system using ZigBee technology", International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, China, April 25-26, 2009, pp.572-575

[10] Sutter, G; Todorovich, E; DeCastro, A; Boemo, E, "Implementation of a wireless control system with self timed activation for mobile robots", Proceedings of the 4th Southern Conference on Programmable Logic,March 26-28, 2008, pp. 205-208

Autonomous Anthropomorphic Robotic Arm using Vision and Colour to Monitor Plant Growth in a Laboratory

Mark Seelye[1], Gourab Sen Gupta[1], John Seelye[2]

[1]School of Engineering and Advanced Technology (SEAT),
Massey University, Palmerston North, New Zealand
[2] The New Zealand Institute for Plant & Food Research Limited
Private Bag 11 600, Palmerston North, New Zealand
Email: marksnz@xtra.co.nz, G.SenGupta@massey.ac.nz

Abstract: An autonomous anthropomorphic robotic arm has been designed and fabricated for the automated monitoring of plant tissue growing in a modified clonal propagation system being developed by Plant & Food Research. The custom fabricated aluminium arm uses a vertical linear ball shaft and high speed stepper motors to provide the various arm joint movements enabling the arm to swivel 180 degrees horizontally. Sensors located at the end of the arm are used to monitor plant growth and the immediate growing environment. These include a compact colour zoom camera on a pan and tilt mount for image capturing, RGB (red, green and blue) colour sensors to monitor leaf colour as well as temperature, relative humidity and carbon dioxide sensors. The robotic arm is capable of reaching anywhere over multiple trays (600mm x 600mm) of plantlets. Captured plant tissue images are processed using innovative algorithms to determine tissue, or whole plant, growth rates over specified time periods. Leaf colour sensors provide information on tissue health status when compared to predetermined optimum values. Custom software fully automates the operation of the arm and the captured data, allowing the arm to return to specified sites (i.e. individual plantlets) at set time intervals to identify subtle changes in growth rates and leaf colour. This will allow plant nutrient levels and the immediate environment to be routinely adjusted in response to this continuous sensing resulting in optimised rapid growth with minimal human input.

Keywords: Micropropagation, Non-destructive sensing, Plant growth monitoring, Remote sensing, RGB, Robotic arm

I. INTRODUCTION

In the modern world robotics and automated products are becoming increasingly common in all economic sectors. The International Organization for Standardization defines a robot in ISO 8373 as: "an automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications".

In the past decade there has been a push towards more automation in the horticulture industry, and it is only now, as robots become more sophisticated and reliable, that we are beginning to see them used to undertake routine, often repetitive tasks, which are expensive to do using a highly paid labour force. As this technology moves forward more and more applications are becoming realties.

Recent published research includes: the current status of the core technologies required for the successful development of a robotic system for weed control [1]; a system for automatic harvesting of numerous agri-science products such as cutting greenhouse grown flowers [2] and automating cucumber harvesting in greenhouses [3]. Advancement in electronics has empowered engineers to build robots that are capable of operating in unstructured environments [4]. Camera-in-hand robotic systems are becoming popular wherein a camera is mounted on the robot, usually at the hand, to provide images of objects located in the robot's workspace [5]. In the horticulture industry robots are increasingly being used to sort, grade, package and even pick fruits. Fruit can be identified and classified on a continuously moving conveyer belt [6]. Kiwifruit can be picked from orchard vines using an autonomous wheeled robot [7]. Robotic techniques for production of seedlings have been developed [8] identifying a need to add a machine vision system to detect irregularities in seed trays using a 5-arm robot.

Automation of micropropagation for the rapid multiplication of plants and micro-dosing for the precise application of chemicals, such as herbicides, have also gained the attention of researchers. For example, an autonomous robotic system has been described for the micropropagation of a grass species replacing the costly and tedious manual process [9] and a system has been detailed which combines plant recognition, micro-dosing and autonomous robotics [10].

Investigations into non-destructive methods to measure the health status of plants using colour sensors is gaining popularity. Colour sensors can identify subtle leaf colour changes to assist as a measure of plant health. Although limited work has been carried out in real-time, current plant based systems have been used on images captured via a digital camera and scanned to identify the colour of selected pixels [11]. Colour sensors are also gaining popularity in other research areas including

evaluating bakery products using a colour based machine vision [12] and in the health sector to determine blood glucose levels [13], as well as in the agri-science industry for grading specific fruits and vegetables [14,15,16].

Apart from numerical data collected, such as pH levels, temperatures, moisture content and light levels, images of plants are becoming increasingly more common to identify possible changes (or irregularities) over time. This allows plant growers to optimize growth and make modifications to plant liquid media. The work detailed in this paper focuses on the design and development of an anthropomorphic robotic arm for routinely monitoring the growth of plantlets and the immediate growing environment.

The specifications for the arm included the ability to move over multiple 600mm x 600mm trays of tissue on a shelving unit with a high resolution zoom camera and colour sensors mounted at the end of the arm. Images will be captured and plant leaf colour monitored during various growth stages.

Section II of this paper gives the system overview, with the hardware detailed in section III and the software in section IV. Robot testing and implementation is detailed in Section V. The discussion and enumeration of future work is presented in section VI, followed by the list of cited references in section VII.

II. SYSTEM OVERVIEW

The basic concept of this project required an autonomous anthropomorphic robotic arm to be designed and fabricated, which contained a number of sensors including RGB colour, temperature and proximity sensors as well as a compact colour camera. Custom software created in Microsoft Visual Studio (VB.net) allowed for a completely automated operation with very little human input required. A systems engineering approach was employed taking the robotic system from a concept to a reality, making use of engineering techniques and integrating standard components to make the final product.

In the following sections the implementation of the robotic system is presented with descriptions of the different hardware components used and the software.

III. SYSTEM HARDWARE

The complete system consists of several modules, each of which is described here.

A. Computer

A PC with three COM port (or USB port) running Microsoft Windows operating system (XP, Vista or 7) is required and was used. The programming software was Microsoft Visual Studio 2005 (VB.net 2005.

B. Robotic Arm

The anthropomorphic robotic arm was designed in the 3D CAD package SolidWorks (2010), with design input from Massey University machining technicians. All components were fabricated in-house using lathes and mills, with more complex and precise components being machined on an Acerman 500 3-Axis Computer Numerical Control (CNC) machine.

Each fabricated part machined on the CNC machine required G-code. This was generated using SolidCAM software (SolidCAM Inc., PA, USA) with the code based on the SolidWorks drawing and dimensions. The CNC machine was then able to cut very precise (to approx 0.001mm) cuts and profiles using a range of cutting tools.

To allow the robotic arm to move vertically, a ball screw and shaft assembly is incorporated converting rotational motion into vertical movement. The ball shaft has a 5mm lead which when combined with a microstep driver, can move the arm vertically by fractions of a mm. This is essential to ensure all sensor readings are taken at “fixed heights”. The robotic arm has been through a number of design changes, i.e. to make the arm lighter, more robust and to give more movement, with designs incorporating Finite Element Analysis (FEA) on the selected parts to ensure safety factors are sufficient for the intended operation.

Fig. 1.SolidWorks rendered photo of completed robotic Arm.

C. Motor Controller

Fig. 2. The Ocean Controls motor controller (KTA – 190) [18]

KTA-190 motor controllers (, Ocean Controls, VC, Aus) provide an interface between the computer and up to 4 stepper motors, giving the ability to control each motor independently or collectively.. Utilizing a RS-232 9600, 8N1 ASCII serial communication protocol up to 4 controller boards can be linked, giving control of up to 16 stepper motors. A motor is controlled by a simple address, followed by the appropriate ASCII commands.

The KTA-190 controller has as interface to allow limit switches to be used to prevent the motors from travelling out of range. With a total of 17 commands it is possible to tailor the operation and movement of the motors. Commands include: setting the position of the motors, return the current positions of the motors, move the motors by a relative or absolute amount and acceleration settings. A status command returns a 12 bit binary representation on the status of the controller board at any given time, providing information on the movement, direction and status of the limit switch respectively.

D. Stepper Motors

The robotic arm required a total of five stepper motors sourced from Ocean Controls. Three FL57STH56-2804B (MOT-122) high torque steeper motors have been used, each with a 1.8o step angle and a holding torque of 12.6kg/cm. This provides the necessary movements of the first 2 joints and vertical movement. The motors moving the arm to the required positions are used in conjunction with a 2:1 gear ratio, while the motor being used to move the arm vertically uses a 3:1 gear ratio, thereby giving the user very precise and fine movement. The remaining two joints utilize FL42STH38-1684B (MOT-120) stepper motors, again with a 1.8o step angle, and holding torque of 3.6 kg/cm.

Fig.3 High torque steeper motorsFL57STH56-2804B (MOT-122 [18]

E. Camera

A Sony colour camera (model: FCB-IX11AP), (figure 4), was used. It features a ¼" CCD (charge coupled device) image sensor using PAL (Phase Alternating Line) encoding system. The camera has a 40x zoom ratio (10x optical, 4x digital) which is controllable from a PC via Sony's VISCA command protocol. The camera's macro capability allows it to capture images as close as 10mm from the subject.

This compact camera (65mm x 44.8mm x 39.3mm), weighing 95g produces high quality images. It has numerous controllable functions (up to 38,000 different command combinations) including digital and optical zoom, aperture setting, shutter speed settings and iris settings. The camera operates on a 6V to 12V DC input.

Fig. 4. Sony FCB-IX11AP colour camera

The camera can operate in light conditions as low as 1.5 lux and temperatures ranging from 0^{o}C to 50^{o}C. The electronic shutter speed is controllable from 1/10 to 1/10000 of a second allowing for clarity in photographs. In order for the camera's functions to be controlled, hexadecimal commands (as serial data) are sent to the camera. These serial commands require 8 data bits, 1 start bit, 1 (or 2) stop bit, no parity, and have a communication speed of 9.6kbps, 19.2 kbps or 38.4kbps. These specifications are contained in the data packet sent to the camera to control the zoom.

F. Sensors

To monitor the colour of the plant material, software can be designed to determine the colour of each pixel on an image; however a faster way of obtaining the RGB colour content of the plant material is to utilize a basic RGB colour sensor. Parallax (Parallax Inc. CA, USA) have two such sensors (ColorPAL and TSC3200) that interface directly to their BASIC Stamp microcontroller, giving quick readings of the RGB values. Both the sensors incorporate a programmable TAOS (Texas Advanced Optoelectronic Solutions, TX, USA) chip, with the ColorPAL using a light-to-voltage sensor while the TSC3200 utilizes a light-to-frequency sensor.

The TCS3200 colour sensor operates by illuminating the object with 2 white LEDs, while an array of photo detectors (each with a red, green, blue and clear filter) interpret the colour being reflected by means of a square wave output whose frequency is proportional to the light reflected. The TSC3200 Colour sensor has a 5.6mm lens, which is positioned to allow an area approx 3.5mm2 to be viewed.

Since the TCS3200 is mounted 20 mm above a sample surface, and therefore not in direct contact with the sample, it was more suited forour application than the full contact required by the ColorPAL sensor.

Fig. 5. Parallax ColorPAL (left) and Parallax TSC3200 (right) colour sensors [17]

Other sensors in use include an ultrasound proximity sensor (Parallax Ping))) ™ (#28015)) to ensure the robotic arm is always at a fixed height above the plant material to ensure colour readings are taken from a fixed point, A temperature and humidity sensor (Parallax Sensirion SHT11 Sensor Module (#28018)) and CO_2 sensor (Parallax CO_2 Gas Sensor Module (#27929)) are also incorporated for constant monitoring of the immediate growing environment. The Parallax sensor modules readily integrated into the Parallax Basic Stamp microcontroller.e.

IV. SOFTWARE IMPLEMENTATION

Microsoft Visual Basic 2005 was the selected for this application due to its ease of use and prior utilisation of this software. Our program makes use of multiple timer application, which enables the program to run continuously repeatedly sending commands and allowing for smooth movements of the arm, camera control and sensors.

The program contains a number of GUI pages giving the user a vast array of options, including the ability to manually move the arm to any position, or alternatively, to specify start and end positions of a tray. The software calculates positions using an inverse kinematic algorithm to determine the required joint angles and appropriate movements of each motor to allow the appropriate sensor to reach the required (selected) position. The user has full control over the camera settings, so that optimal values can be set to get the "best" image. All information pertaining to the position, motor coordinates and sensor readings are stored in an Access database, giving the user the ability to review the collated data on screen or export it to an Excel spreadsheet. There is also the option to watch time lapsed videos of a selected plant growing.

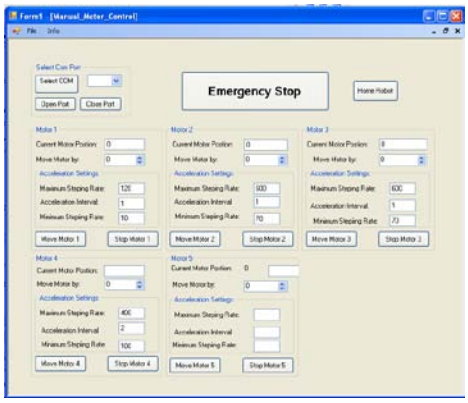


Fig.6. GUI giving the user manual control of the arm.

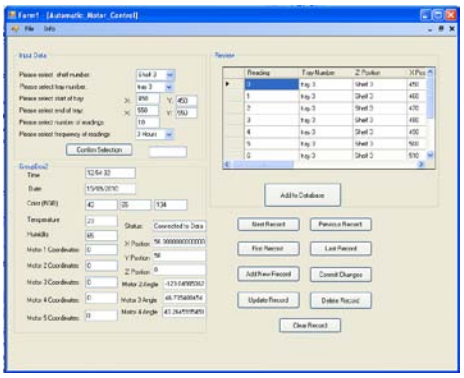


Fig.7. GUI giving the user automatic control of the arm and reviewing collated data.

A. Motor Control

The Ocean Controls serial stepper motor controllers are programmed by sending ASCII commands from a serial port. The command contains the motor being used (@AA), a command required to be executed (CMND), followed by the parameters of the command (XXXX) and finally a carriage return byte (CR).

@AA CMND XXXXCR

It is possible to simultaneously communicate and control all 4 motors on a controller board by sending one command, @AA CMND WWWW XXXX YYYY ZZZZCR where @AA refers to the controller board being used, CMND the command to send to the controller, WWWW parameter of motor 1, XXXX parameter of motor 2, YYYY parameter of motor 3, ZZZZ parameter of motor 4, followed by the carriage return byte (CR).

The information is sent from the serial port by setting up a byte array using the required protocol. Below is an example to move motor 1 a relative movement of 1000 steps (using a microstep of 3600, and 2:1 gear ration, this would move the motor 50 degree).

```
Motor_1(0) = Asc("@")
Motor_1(1) = Asc("0")
Motor_1(2) = Asc("1")
Motor_1(3) = Asc(" ")
Motor_1(4) = Asc("R")
Motor_1(5) = Asc("M")
Motor_1(6) = Asc("O")
Motor_1(7) = Asc("V")
Motor_1(8) = Asc(" ")
Motor_1(9) = Asc("1")
Motor_1(10) = Asc("0")
Motor_1(11) = Asc("0")
Motor_1(12) = Asc("0")
Motor_1(13) = &HD
```

All commands are sent to the controller in the same format, with the exception of the "position" and "status" command which do not require parameters. When the motor controller receives the status

command it replies with an integer number which, when converted represents a 12 bit binary representation of the controller. The bits that make up the binary number show which of the motors are moving, in which direction and the status of the limit switch.

B. Camera control

The camera is controlled using Sony's VISCA command protocol. The basic unit of VISCA communication is a packet of 3 to 16 bytes. The first byte of the packet is the header consisting of the sender and header addresses; the remainder of the packet holds the command (or message) to be sent to the camera (figure 8). The final byte comprises a terminator signifying the end of the packet and usually has the hexadecimal value FF (or 11111111 binary).

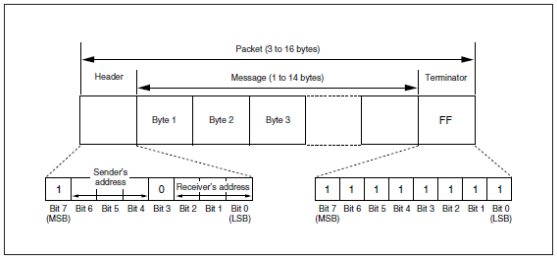


Fig.8. Serial command data packet layout required to communicate with camera

Each of the commands required for the camera controls is sent via the serial port to the camera, in a byte format of the appropriate hexadecimal values. Each command has its own unique command, which is required to be sent to the camera each time. A timer function allows each command to be executed almost instantaneously by the camera.

C. Sensors

Parallax design and manufacture sensors for a large range of applications. These sensors easily integrate into the Parallax Basic Stamp microcontroller, with most sensors coming with some source code. The Basic Stamp 2e microcontroller uses PBASIC language with a processor speed of 20 MHz. The microcontroller integrates to the PC using a serial input and serial output pin; however a Board of Education (#28850) was used which connects the microcontroller to the PC through a USB connection.

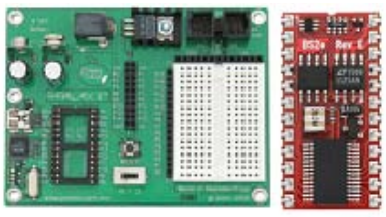


Fig. 9. Parallax Board of Education (left), BASIC Stamp 2e microcontroller (right) [17]

Having implemented the sensors to the microcontroller it became obvious that the microcontroller was unable to perform some of the calculations quickly, therefore more complex calculations are solved within the software on the PC.

V. TESTING AND INTEGRATION

Testing was done in parts using a modular approach. The sub-systems that required to be programmed, the construction of the anthropomorphic robotic arm, control of the motor through the motor controller, implementation of all sensors and control of the camera, were all tested separately. The testing setup and procedure are described below.

A. Anthropomorphic Robotic Arm

The robotic arm, designed in SolidWorks, was tested as parts were created but it was soon discovered that initial designs were too heavy. A quick solution was implemented utilizing finite element analysis (FEA) to ensure minimal material was used on specific components while still retaining sufficient strength.

Fig. 10. Original motor mount (left), redesigned motor mount with excess material removed (right)

Finally the entire system was integrated into one final "product".

B. Motor Controller

In the first instance a simple program was written in Visual Studio, allowing one stepper motor to be controlled by clicking a button. This action sent the corresponding command to the motor controller board. The next step was to control the motor by selecting the required steps from a numerical up/down box, which increased or decreased the relative movement position, allowing for simple movement based on the position byte. Once one stepper motor was successfully implemented, multiple motors could then be controlled in the same manner. Finally a number of algorithms were set up to allow the motors to move to a specific position calculated via inverse kinematics algorithms.

To ensure the robotic arm was functioning correctly, the arm was set up in the working environment, with a number of different shaped objects temporarily replacing the plants. The robotic arm was then programmed to move to the same coordinates autonomously and capture images which, once reviewed confirmed the arm had returned to the same positions each time, with all images showing the correct object located in the same position.

C. Control of Camera

The camera can be programmed and controlled using TTL or RS232C signal level serial interface. To test the camera features it was directly wired to the PC using the RS232C interface via a USB-to-RS232 converter as shown in figure 11. The video output signal from the camera was fed to a frame grabber/digitizer which is interfaced to the PC via USB.

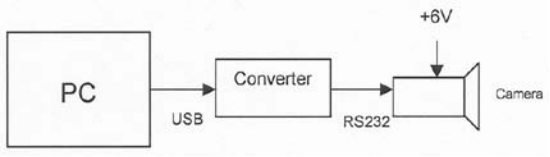


Fig. 11. Block diagram showing wired connection of PC to camera's RS232 inputs

Using a standard communication program (Terminal v1.9b), commands were sent to the camera to familiarize with the VISCA command structure and to test the various camera functions, especially the programming commands for controlling the zoom. Once successfully mastered, a more complex program was created.

D. Sensors

Parallax sensors come with source code allowing for straight forward integration and testing. Once the sensors were implemented, custom software in Visual Studio was created to communicate with the microcontroller and sensors.

Custom software allowed the proximity sensor to be integrated with the temperature sensor, this was vital since the speed of sound in air is affected by the ambient temperature C_{air} = 331.5 + (0.6 x T_c) m/s. Results from Parallax show over a temperature range of 0^o to $70^o$the error percentage is in the magnitude of 11% - 12%.

E. Colour Sensor Calibration

Testing the TCS3200 sensor involved comparing the RGB values obtained to a CM-700d Spectrophotometer (Konica Minolta Sensing Americas, Inc, NJ, USA),which gives colour in the XYZ colour space, using RHS colour charts (Royal Horticulture Society, London, UK), for a range of green colours. To make the comparisons, a linear transformation matrix was required to convert the XYZ colour to an RGB colour space,

XYZ to RGB transformation matrix equation (alternatively the inverse can be taken to solve for RGB):

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 1.436603368435593 & -0.118534007680723 & -0.226698746676605 \\ -0.681279626157456 & 1.618476401223029 & 0.026671792203898 \\ 0.042919641610377 & -0.102206221605993 & 0.974614036323184 \end{pmatrix} \times \begin{pmatrix} G \\ G \\ B \end{pmatrix}$$

In order to validate the results a calibration factor (gamma) was required to minimise any differences between ideal results (from the spectrophotometer) and actual results (from TCS3200).

$$R_N{}' = R_N{}^{\gamma}$$

Where: RN' = Spectrophotometer (desired RGB, known)

RN= TCS3200 RGB (Not Calibrated, known)

γ= Gamma (required calibration factor)

To solve the equations the data is required to be, white calibrated, which involves offsetting the data to ensure the white reading has an RGB value of 255,255,255. The following equation is then solved:

$$\gamma_R = \frac{\log(R_N{}'/255)}{\log(R_N/255)}, \gamma_G = \frac{\log(G_N{}'/255)}{\log(G_N/255)}, \gamma_B = \frac{\log(B_N{}'/255)}{\log(B_N/255)}$$

Data is required to be normalised (i.e. put in a range from 0-1). From this the gamma values can be solved for the corresponding red, green and blue values.

Results showed that over 16 random samples of green, from the RHS colour chart, the gamma was calculated to be ((Red) γR = 0.50, (Green) γG = 0.38, (Blue) γB = 0.59) this lead to an absolute error of; red 5.67%, green 3.01% and blue 3.98% showing the TCS3200 sensor to be accurate to within an average error of 4.22% across the 3 colours.

F. Image Analysis

Images captured via the camera were imported into VB.net, where algorithms were applied to determine the quantity of plant material present. Pixels the non-plant material in the image were counted and replaced with a contrasting colour pixel. Initial leaf testing making the background yellow showed that the algorithm worked correctly (figure 10).

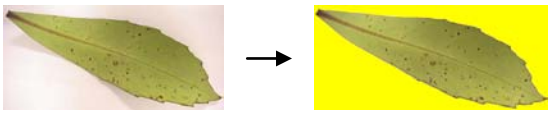


Fig. 12. Initial testing of the image analysis algorithm to determine quantity of pixels that make up the plant material content

VI. DISCUSSIONS AND FUTURE DEVELOPMENT

The system is currently being tested in the working environment. A number of issues have arisen during implementation and testing and are being addressed. Future work involves completing the entire robotic system and fully integrating the sensors. On completion, investigations based on the continuous information obtained from the sensors and processed through algorithms will be used to optimise the growth of plants by manipulating liquid nutrient solutions and the immediate growing environment.

Opportunities exist for TCP/IP connections so that monitoring could be accessed from anywhere in the world.

VII. REFERENCES

1. Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. Computers and Electronics in Agriculture, 61(1), 63-78.
2. Kawollek, M., & Rath, T. (2008). Robotic Harvest of Cut Flowers Based on Image Processing by Using Gerbera jamesonii as Model Plant. Proceedings of the International Symposium on High Technology for Greenhouse System Management, Vols 1 and 2(801), 557-563.
3. van Henten, E. J., Hemming, J., van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., et al. (2002). An autonomous robot for harvesting cucumbers in greenhouses. [Article]. Autonomous Robots, 13(3), 241-258.
4. Garcia, G. J., Pomares, J., & Torres, F. (2009). Automatic robotic tasks in unstructured environments using an image path tracker. Control Engineering Practice, 17(5), 597-608.
5. Kelly, R., Carelli, R., Nasisi, O., Kuchen, B., & Reyes, F. (2000). Stable Visual Servoing of Camera-in-Hand Robotic Systems. ITRANSACTIONS ON MECHATRONICS, 5(1), 39 - 48.
6. Reyes, JF; Chiang, LE.,(2009). Location And Classification Of Moving Fruits In Real Time With A Single Colour Camera, Chilean Journal Of Agricultural Research, Vol. 69, , 179-187.
7. Scarfe, A. J., Flemmer, R. C., Bakker, H. H., & Flemmer, C. L. (2009). Development of An Autonomous Kiwifruit Picking Robot. Proceedings of the Fourth International Conference on Autonomous Robots and Agents, 639-643.
8. Wang HonYong; Cao QiXin; Masateru, N.; Bao JianYue, (1999) Image processing and robotic techniques in plug seedling production, Transactions of the Chinese Society of Agricultural Machinery, Vol. 30, 1999,. 57-62.
9. Otte, C., Schwanke, J., & Jensch, P. (1996). Automatic micropropagation of plants. Optics in Agriculture, Forestry, and Biological Processing, 2907, 80-87.
10. Sogaard, H. T., & Lund, I. (2007). Application accuracy of a machine vision-controlled robotic micro-dosing system. biosystems engineering, 96(3), 315-322.
11. Yadav, S. P., Ibaraki, Y., & Gupta, S. D. (2010). Estimation of the chlorophyll content of micropropagated potato plants using RGB based image analysis. Plant Cell Tissue and Organ Culture, 100(2), 183-188.
12. Abdullah, m. Z., aziz', s. A., & mohamed, a. M. D. (2000). Quality inspection of bakery products using a color-based machine vision system. Journal of food quality, 23.
13. Raja, A. S., & Sankaranarayanan, K. (2006). Use of RGB Color Sensor in Colorimeter for better clinical measurements of blood Glucose. BIME Journal 6(1), 23-28.
14. Miranda, C., Girard, T., & Lauri, P. E. (2007). Random sample estimates of tree mean for fruit size and colour in apple. Scientia Horticulturae, 112, 33-41.
15. Kang, S. P., & Sabarez, H. T. (2009). Simple colour image segmentation of bicolour food products for quality measurement. Journal of Food Engineering, 94, 21-25.
16. Yam, K. L., & Papadakis, S. E. (2004). A simple digital imaging method for measuring and analyzing color of food surfaces. Journal of Food Engineering, 61, 137-142.
17. Parallax Home. Retrieved 05/07/2010, from www.Parallax.com
18. OceanControls. Retrieved 12/04/2010, from www.oceancontrols.com.au

Low Cost Colour Sensors for Monitoring Plant Growth in a Laboratory

Mark Seelye, Gourab Sen Gupta, Donald Bailey
School of Engineering and Advanced Technology (SEAT)
Massey University, Palmerston North, New Zealand
Email: marksnz@xtra.co.nz, G.SenGupta@massey.ac.nz

John Seelye
New Zealand Institute for Plant & Food Research Limited
Private Bag 11 600, Palmerston North, New Zealand

Abstract—**An automated system for measuring plant leaf colour, as an indicator of plant health status, has been developed for plantlets growing in a modified micropropagation system. Using a custom built robotic arm, sensors located on a pan and tilt system at the end of the arm monitor plant growth and the ambient growing environment. Sensors include a compact colour zoom camera, RGB (red, green and blue) colour sensors, as well as environmental sensors. Leaf colour sensors provide information, in a non-destructive manner, on the health status of tissue by comparing the sensor outputs to predetermined optimum values. These low-cost colour sensors can be incorporated into a continuous automated system for monitoring leaf colour of growing plants. Subtle colour changes can be an early indication of stress from less than optimum nutrient levels. When this is combined with automated image sensing for growth analysis, and environmental sensing (RH, CO_2 and temperature) in a controlled environment, optimised rapid growth with minimal human input can be achieved using a modified micropropagation system. In this paper we detail the calibration technique for a RGB sensor and compare it with a high end spectrophotometer.**

Keywords - Micropropagation, plant growth monitoring, colour sensor calibration, RGB colour sensor

I. INTRODUCTION

Robotic and automated systems are becoming increasingly common in all economic sectors. In the past decade there has been a push towards more automation in the horticulture industry, and it is only now, as robots become more sophisticated and reliable, that we are beginning to see them used to undertake routine, often repetitive tasks, which are expensive to do using a highly paid labour force. With rapid strides in technological advancement, more and more applications have become possible. This includes the development of a robotic system for weed control [1], a system for automatic harvesting of numerous agri-science products such as cutting flowers grown in greenhouses [2] and automating cucumber harvesting in greenhouses [3]. Advancement in electronics has empowered engineers to build robots that are capable of operating in unstructured environments [4]. Camera-in-hand robotic systems are becoming increasingly popular wherein a camera is mounted on the robot, usually at the hand, to provide an image of the objects located in the robot's workspace [5]. Increasingly, robots are being used to sort, grade, pack and even pick fruits. Fruits can be identified and classified on a continual moving conveyer belt [6]. An autonomous wheeled robot has been developed to pick kiwifruit from orchid vines [7]. Robotic techniques for production of seedlings have been developed identifying a need to add a machine vision system to detect irregularities in seed trays and to provide supplementary sowing using a 5-arm robot [8].

Automation of micropropagation for the rapid multiplication of plants has been described for the micropropagation of a grass species replacing the costly and tedious manual process [9]. A system has also been developed which combines plant recognition and chemical micro-dosing using autonomous robotics [10].

Colour as a means of assessing quality is also gaining popularity amongst researches. These include evaluating bakery products using a colour based machine vision [11], monitoring tea during fermentation [12], grading specific fruits and vegetables [13 - 15] and in the health sector to determine blood glucose levels [16]. Near infrared (NIR) sensors are also gaining popularity as non-destructive means of assessing fruit and plant material, including use as a measure of plant nutrient status [17] as well as testing of fruit quality [18 - 20].

An investigation into non-destructive methods to measure the health status of plants is a relatively new area of research. Subtle leaf colour changes can be used as a measure of plant health. Although limited work has been carried out in real-time, a recent micropropagation based system used potato tissue images captured via a digital camera and scanned to identify the colour of selected pixels [21]. Spectral reflectance, using a range of spectral bands, has been used as a non-destructive measure of leaf chlorophyll content in a range of species [22]. Alternative methods make use of spectroscopic systems using a fixed light source to record colour reflectance of multiple samples [23].

This paper focuses on the use of low cost colour sensors for monitoring leaf colour of plant tissues growing in a modified micropropagation system. The calibration method of these sensors is described and its accuracy evaluated.

OVERVIEW OF SYSTEM HARDWARE

The reported system uses an autonomous robotic arm containing RGB colour, environmental and proximity sensors as well as a compact colour camera. Custom software created in Microsoft Visual Studio (VB.net) allow for a completely automated operation which requires minimal human input.

A. ROBOTIC ARM

The robotic arm, shown in figure 1, uses 5 stepper motors that are controlled through a motor controller and microstep driver [24]. To allow the robotic arm to move vertically, a ball screw and shaft assembly is incorporated, converting rotational motion into vertical movement. The arm contains a pan and tilt system at the end of the arm which houses a camera [25], colour and proximity sensor [26]. The operation of the arm is completely automated, continually gathering information from the sensors and captured images for assessment and analysis.

The arm uses bipolar, high torque stepper motors, which provide a maximum torque of 12.3kg/cm. and have a step angle of 1.8^{o}. The use of a micro step driver allows the user to select an even finer resolution (i.e. more steps per revolution). The motors are controlled through a motor driver board that allows commands to be sent from the PC via a USB port to the controller to control the movement.

Figure 1. SolidWorks rendered photo of the robotic Arm.

B. COLOUR SENSORS

Currently there are a number of colour sensors on the market with prices ranging from low-cost light-to-frequency chips to sophisticated and very expensive spectrophotometers.

Parallax Inc has 2 colour sensors which integrate seamlessly with their Basic Stamp microcontroller. Both the ColorPAL and TCS 3200 colour sensors are provided with some source code making them amenable to integrating into our customised system.

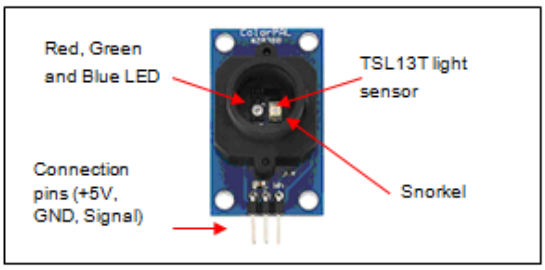


Figure 2. Parallax ColorPAL colour sensor.

The ColourPAL sensor (Figure 2) illuminates a sample using in-built red, green and blue LED light sources (one colour at a time) and records the quantity of light reflected back from the object. The ColorPal makes use of a TAOS(Texas Advanced Optoelectronic Solutions) light-to-voltage chip. When light is reflected back, the voltage level, which is proportional to the light reflected, is used to determine the sample's R, G and B colour contents. The ColorPAL requires the sample to be illuminated using each of the red, green and blue LED's, with a 'snorkel' to shield possible interference from external light sources. This requires the ColorPAL to be in direct contact with the object for an optimum reading without interference.

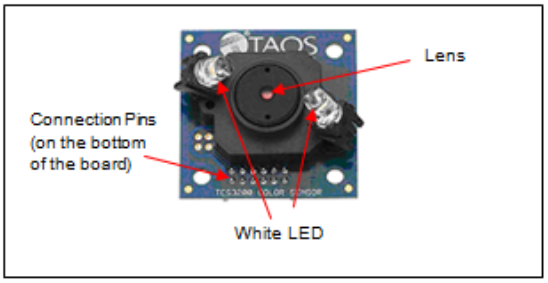


Figure 3. Parallax TCS3200 colour sensor.

The TCS3200 Colour sensor (Figure 3) makes use of a TAOS TCS3200 RGB light-to-frequency chip. The TCS3200 colour sensor operates by illuminating the object with 2 white LEDs, while an array of photo detectors (each with a red, green, blue and clear filter) interpret the colour being reflected by means of a square wave output whose frequency is proportional to the light reflected. The TSC3200 Colour sensor has a 5.6mm lens, which is positioned to allow an area of 3.5mm^{2} to be viewed.

A USB4000 spectrometer (Ocean Optics Inc, FL, USA) was used to find the height at which the greatest intensity of light occurred when the RGB sensor was placed above a sample. As the 2 white LEDs are directed down at an angle, there is a point where the light intensity was the greatest. This position is 20mm above the surface of the sample as shown in figure 4.

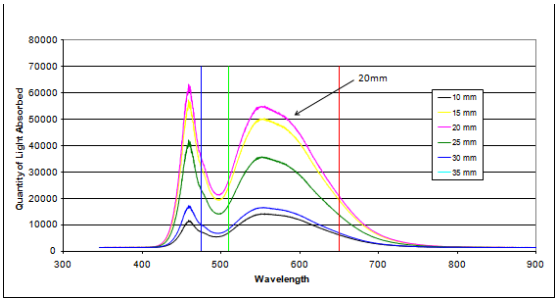


Figure 4. Light absorbed from TCS3200 across the white LED light spectrum when the sensor is positioned at 6 different heights.

Since the TCS3200 is mounted 20 mm above the sample, and therefore not in direct contact with the sample, it was more suited for our application than the full contact required by the ColorPAL sensor.

An alternate method of determining plant leaf colour is to use an image captured by a camera and through software determine the colour of the pixels. However, because plant leaves can overlap (see figure 5), shadows are created leading to false colour

readings from the image. Since the TCS3200 colour sensor uses its own light source to illuminate the sample surface, it eliminates any potential shadowing.

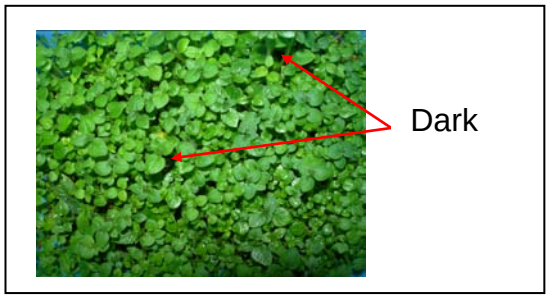


Figure 5. Captured image showing areas of "darkness" caused by overlapping which would lead to false colour readings

A Konica Minolta CM-700D Spectrophotometer (Konica Minolta Sensing Americas, Inc, NJ, USA)was used to validate and calibrate the RGB sensors. For accurate measurements, CM-700D was calibrated for "white readings" by sampling a supplied white object.

The CM-700D gives colour in the XYZ colour space, as well as L*a*b*, L*C*h, Hunter Lab, Yxy and Munsell. A linear transformation matrix was required to transform data from the XYZ colour space to the RGB colour space for comparisons with the Parallax sensor. The linear transformation equations used [27] are:

$$x = \frac{X}{X+Y+Z} \tag{1}$$

$$y = \frac{Y}{X+Y+Z} \tag{2}$$

$$z = \frac{Z}{X+Y+Z} \tag{3}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = M \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{4}$$

Equations (1 – 4) combined with the standard 1931 xy chromaticity diagram provided the foundation for the linear transformation. This transformation converted the *XYZ* data to sRGB colour space, (where sRGB is an equivalent colour space to RGB but makes use of a gamma correction) with the chromaticity values of *x*, *y* and *z* shown in Table I being standard [28].

TABLE I. X, Y, AND Z CHROMATICITY VALUES OF RED, GREEN AND BLUE

Colour	*x*	*y*	*z*
Red	0.64	0.33	0.212656
Green	0.30	0.60	0.715158
Blue	0.15	0.06	0.072186

From the *x*, *y* and *z* chromaticity values, the transformation matrix, *M*, is calculated (Eq. 5)

$$M \approx \begin{pmatrix} 0.721144 & 0.063298 & 0.166008 \\ 0.303556 & 0.643443 & 0.052999 \\ 0.000076 & 0.064689 & 1.024294 \end{pmatrix} \tag{5}$$

To calculate the *R*, *G* and *B* values the inverse is taken (Eq. 5-6).

$$M^{-1} \approx \begin{pmatrix} 1.436603 & -0.118534 & -0.226698 \\ -0.681279 & 1.618476 & 0.026671 \\ 0.042919 & -0.102206 & 0.974614 \end{pmatrix} \tag{6}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = M^{-1} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{7}$$

II. EXPERIMENTAL RESULTS AND DISCUSSION

In order to validate the TCS3200 colour sensor, it was necessary to calibrate and test it using the CM-700D.

This involved taking 200 RGB readings with the TCS3200 using different coloured samples and averaging them. The same samples were measured, each 20 times, with the CM-700D and averaged. These tests were all completed in a constant temperature dark room. As the CM-700D uses the XYZ colour space, the linear transformation matrix was required to convert the XYZ values to RGB (Eqs. 1-4 and Eq. 7).

The TCS3200 was firstly calibrated through software by modifying the integration time, to allow a white object to have a RGB value as close as possible to 255,255,255.

In order to calculate a *calibration factor* the following equation was used:

$$R_N^{'} = R_N^{\gamma} \tag{8}$$

where $R_N^{'}$ = CM-700D (desired RGB value)

R_N = TCS3200 RGB (Un-calibrated sensor data)

γ = Gamma (required *calibration factor*)

First the sensor data was scaled to ensure all values are offset so that the white reading is 255 each for *R*, *G* and *B* (Eq. 9)

$$R_N = R \times \frac{255}{R_{\max}}, G_N = G \times \frac{255}{G_{\max}}, B_N = B \times \frac{255}{B_{\max}} \tag{9}$$

where $R_{\max}, G_{\max}, B_{\max}$ represent the maximum *R*, *G* and *B* value of a white object.

The *calibration factors* (γ) for each colour are calculated using normalized data. (Eq. 10)

$$\gamma_R = \frac{\log(R_N^{'}/255)}{\log(R_N/255)}, \gamma_G = \frac{\log(G_N^{'}/255)}{\log(G_N/255)}, \gamma_B = \frac{\log(B_N^{'}/255)}{\log(B_N/255)} \tag{10}$$

For each colour sample measured, the *calibration factor* is calculated and averaged using a geometric mean (as opposed to the more general arithmetic mean function [29]), thus providing the γ factor for *R*, *G* and *B* individually. The (desired) calibrated values

are then obtained using equation 11. This gamma calibration eliminates any disparity in using a transformation matrix to sRGB (and not directly to RGB)

$$R'_{N(calibrated)} = (R_N / 255)^{\gamma} \times 255 \qquad (11)$$

For a range of 7 colours, measurements were done using the TCS3200 RGB sensor and the CM-700D Spectrophotometer. (Table II). The gamma *calibration factors* calculated are-

(Red) $\gamma_R = 0.91$, (Green) $\gamma_G = 0.48$, (Blue) $\gamma_B = 0.72$

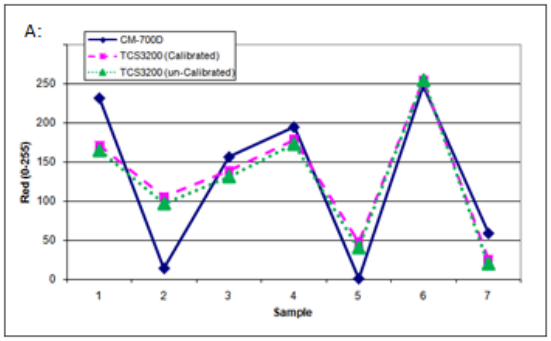
Table III summarises the average error, error percentage and the standard deviation for un-calibrated and calibrated RGB sensor data compared to CM-700D spectrophotometer outputs.

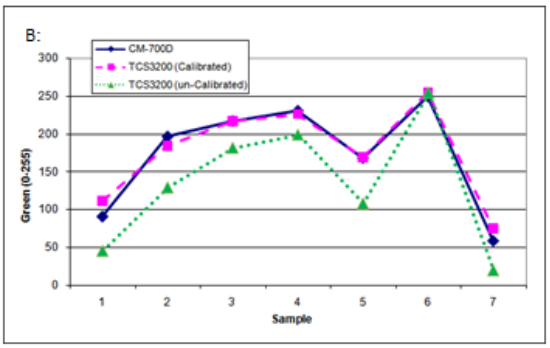
TABLE II. RESULTS OBTAINED COMPARING THE TCS3200 COLOUR SENSOR (CALIBRATED AND UN CALIBRATED) TO THE CM-700D OVER A RANGE OF 7 DIVERSE COLOURS

Colour	TSC3200 (un calibrated)			TCS3200 (Calibrated)			CM-700D Spectro-photometer		
	R_N	G_N	B_N	R_N'	G_N'	B_N'	*R*	*G*	*B*
Red	172	45	41	171	112	67	232	91	77
Green	101	128	62	106	184	91	14	197	76
Light Blue	137	180	211	139	217	220	156	217	214
Light Green	180	197	161	179	227	181	195	231	171
Dark Blue	42	107	169	47	169	187	1	168	200
White	266	251	259	255	255	255	248	250	246
Black	20	20	22	24	75	44	59	58	55

TABLE III. AVERAGE ERROR (0-255), PERCENTAGEERROR AND STANDARD DEVIATION FOR RED, GREEN AND BLUE MEASUREMENTS OF THE TCS3200 COLOUR SENSOR, CALIBRATED AND UN CALIBRATED, WHEN COMPARED TO CM-700D RESULTS ACROSS A RANGE OF COLOURS

	TCS3200 (uncalibrated)			TCS3200 (calibrated)		
Colour	*R*	*G*	*B*	*R*	*G*	*B*
Ave Error	40.339	40.304	21.128	38.951	8.663	10.571
Error %	15.819	15.805	8.286	15.275	3.397	4.145
σ	26.285	20.329	13.055	29.668	8.081	2.852





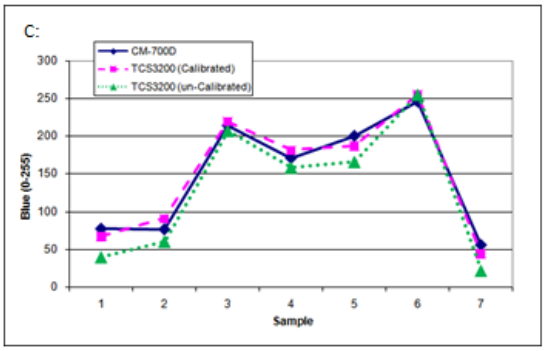


Figure 6. TCS3200 sensor *RGB* readings, calibrated and un-calibrated, compared to the CM-700D readings of: Red (A); Green (B); Blue (C). (Colour samples are as given in Table II)

TABLE IV. AN EXAMPLE OF A GREEN COLOUR INTERPRETED BY THE CM-700D AND TCS3200 COLOUR SENSOR BEFORE AND AFTER A CALIBRATION FACTOR

TCS3200 (uncalibrated)	TCS3200 (calibrated)	CM-700D Spectrophotometer
RGB = 101,128,62	*RGB* = 106,184,91	*RGB* = 14,197,76

As the colour sensor is intended to measure the colour of plant leaves, there is no requirement to calibrate it across the full range of colours. The sensor was therefore calibrated for a range of green - yellow colours only. 15 RHS (Royal Horticulture Society, London, UK) colour charts, designed for growers to identifying plant colours, were used and the measured data is shown in Table V. 200 colour readings were taken by the TCS3200 colour sensor, followed by 20 readings by the CM-700D for each colour chart.

The gamma calibration factors calculated are

(Red) $\gamma_R = 0.50$, (Green) $\gamma_G = 0.38$, (Blue) $\gamma_B = 0.59$

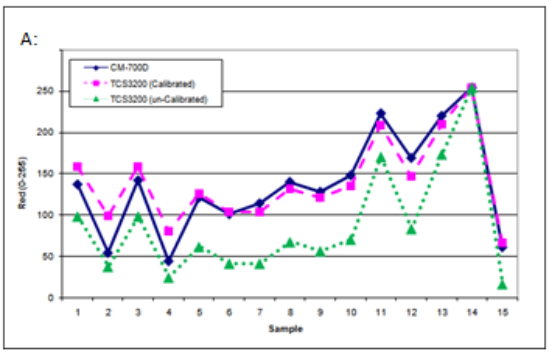
Table VI summarises the average error, error percentage and the standard deviation for un-calibrated and calibrated RGB sensor data compared to CM-700D spectrophotometer outputs for the 15 colours. There is a vast improvement across the 3 colour components (RGB), with the average red error improving to 5.69%, green 3.19% and blue 3.92%.

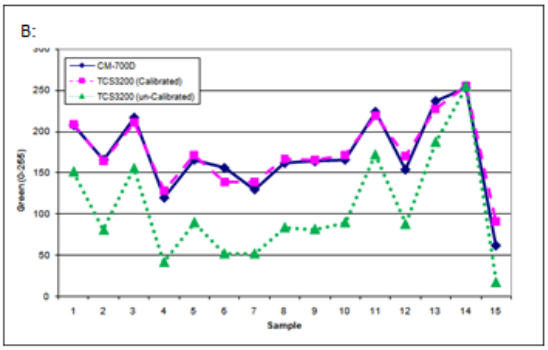
TABLE V. RESULTS OBTAINED COMPARING THE TCS3200 COLOUR SENSOR (CALIBRATED AND UN CALIBRATED) WITH THE CM-700D OVER A RANGE OF 15 COLOURS

Colour		TCS3200 (uncalibrated)			TCS3200 (Calibrated)			CM-700D		
RHS Colour Group	ID	R_N	G_N	R_N	R_N'	G_N'	B_N'	*R*	*G*	*B*
Blue-Green	123A	99	150	148	159	206	188	138	208	194
Green	127C	38	80	66	99	160	120	55	166	134
Green	129C	99	154	121	159	208	168	143	217	173
Green	131C	25	42	31	80	123	78	45	120	84
Green	133C	62	89	75	126	167	128	122	166	144
Green	135C	42	52	31	104	134	78	102	156	97
Green	137C	42	52	31	104	134	78	115	130	85
Green	139C	68	83	51	132	162	104	141	162	108
Green	141C	57	81	40	121	161	90	129	164	87
Green	143C	71	89	42	135	167	93	149	166	81
Yellow-Green	145C	171	171	108	209	217	157	224	224	151
Yellow-Green	147C	84	87	55	147	166	107	170	154	112
Yellow-Green	149C	174	186	101	211	224	151	221	237	134
White	155D	255	253	228	255	254	239	255	253	228
Black	202A	17	17	18	66	86	57	62	62	61

TABLE VI. AVERAGE ERROR (0-255), PERCENTAGE ERROR AND STANDARD DEVIATION FOR RED, GREEN AND BLUE MEASUREMENTS OF THE TCS3200 COLOUR SENSOR, CALIBRATED AND UN CALIBRATED, COMPARED WITH CM-700D RESULTS ACROSS A RANGE OF COLOURS

	TCS3200 (uncalibrated)			TCS3200 (calibrated)		
Colour	*R*	*G*	*B*	*R*	*G*	*B*
Ave Error	51.133	66.459	48.513	14.509	7.179	9.114
Error %	20.052	26.063	19.025	5.690	2.815	3.574
σ	24.628	24.030	16.998	12.303	7.473	5.374





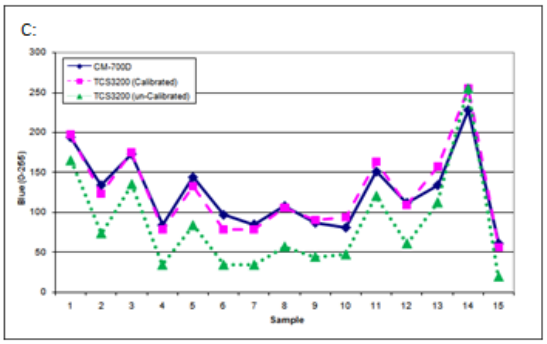


Figure 7. TCS3200 sensor RGB readings , calibrated and un calibrated, compared to the CM-700D readings of: Red (A); Green (B); Blue (C).for a range of RHS green colours. (Colour samples are as given in Table V)

TABLE VII. VISUAL RESULTS SHOWING THE RGB COLOUR INTERPRETED BY THE CM-700D AND TCS3200 COLOUR SENSOR, BEFORE AND AFTER CALIBRATION (RHS 141C)

TCS3200 (uncalibrated)	TCS3200 (calibrated)	CM-700D Spectrophotometer
RGB = 57,82,44	RGB = 121,165,91	RGB = 129,164,87

With the calibration factor determined, the TCS3200 was able to identify colour of selected plant material. The sensor has been integrated with a proximity sensor to ensure all readings are taken at a fixed height above the plant leaves. As the colour sensor is attached to a robotic arm, the sensor is able to autonomously return to the same position to record the colour, making it possible over time to identify subtle changes in the colour of the plant leaves. These changes in plant material will allow plant nutrient concentrations to be adjusted in order to achieve optimum growth.

III. CONCLUSION AND FUTURE WORK

Experimental results show that the Parallax TCS3200 is a useful low cost colour sensor, which, following calibration, can provide accurate RGB readings. It is therefore a useful component for integrating into an automated monitoring system such as a robotic arm, with various other sensors, for the monitoring and control of plants growing in a modified plant micropropagation system. Future work involves completing the entire robotic system with fully integrated sensors. This will allow investigations to be carried out to optimise plant growth based on the information obtained from the colour sensors by relating colour information to plant quality. This will be further enhanced by additional growth data obtained from captured plant images. The use of NIR sensors offers further potential for non destructive assessments of plant tissue structure.

Elaborate and extensive test results will be presented in an extended version of this paper.

References

[1] Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. Computers and Electronics in Agriculture, 61(1), pp. 63-78.

[2] Kawollek, M., & Rath, T. (2008). Robotic Harvest of Cut Flowers Based on Image Processing by Using Gerbera jamesonii as Model Plant. Proceedings of the International Symposium on High Technology for Greenhouse System Management, Vols 1 and 2(801), pp. 557-563.

[3] van Henten, E. J., Hemming, J., van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., et al. (2002). An autonomous robot for harvesting cucumbers in greenhouses. [Article]. Autonomous Robots, 13(3), pp. 241-258.

[4] Garcia, G. J., Pomares, J., & Torres, F. (2009). Automatic robotic tasks in unstructured environments using an image path tracker. Control Engineering Practice, 17(5), pp. 597-608.

[5] Kelly, R., Carelli, R., Nasisi, O., Kuchen, B., & Reyes, F. (2000). Stable Visual Servoing of Camera-in-Hand Robotic Systems. ITRANSACTIONS ON MECHATRONICS, 5(1), pp. 39 - 48.

[6] Reyes, JF; Chiang, LE., "Location And Classification Of Moving Fruits In Real Time With A Single Colour Camera", Chilean Journal Of Agricultural Research, Vol. 69, 2009, pp. 179-187

[7] Scarfe, A. J., Flemmer, R. C., Bakker, H. H., & Flemmer, C. L. (2009). Development of An Autonomous Kiwifruit Picking Robot. Proceedings of the Fourth International Conference on Autonomous Robots and Agents, pp. 639-643.

[8] Wang HonYong; Cao QiXin; Masateru, N.; Bao JianYue, "Image processing and robotic techniques in plug seedling production", Transactions of the Chinese Society of Agricultural Machinery, Vol. 30, 1999, pp. 57-62

[9] Otte, C., Schwanke, J., & Jensch, P. (1996). Automatic micropropagation of plants. Optics in Agriculture, Forestry, and Biological Processing, 2907, pp. 80-87.

[10] Sogaard, H. T., & Lund, I. (2007). Application accuracy of a machine vision-controlled robotic micro-dosing system. biosystems engineering, 96(3), pp. 315-322.

[11] Abdullah, m. Z., aziz', s. A., & mohamed, a. M. D. (2000). Quality inspection of bakery products using a color-based machine vision system. Journal of food quality, pp. 23.

[12] Borah, S., & Bhuyan, M. (2005). A computer based system for matching colours during the monitoring of tea fermentation International Journal of Food Science and Technology, pp. 40.

[13] Omar, A. F. B., & MatJafri, M. Z. B. (2009). Optical Sensor in the Measurement of Fruits Quality: A Review on an Innovative Approach. International Journal of Computer and Electrical Engineering, 1(5).

[14] Miranda, C., Girard, T., & Lauri, P. E. (2007). Random sample estimates of tree mean for fruit size and colour in apple. Scientia Horticulturae, 112, pp. 33-41.

[15] Kang, S. P., & Sabarez, H. T. (2009). Simple colour image segmentation of bicolour food products for quality measurement. Journal of Food Engineering, 94, pp. 21-25.

[16] Raja, A. S., & Sankaranarayanan, K. (2006). Use of RGB Color Sensor in Colorimeter for better clinical measurements of blood Glucose. BIME Journal 6(1), pp. 23 - 28.

[17] Menesatti, P., Antonucci, F., Pallottino, F., Roccuzzo, G., Allegra, M., Stagno, F., et al. (2010). Estimation of plant nutritional status by Vis–NIR spectrophotometric analysis on orange leaves. biosystems engineering, pp. 105.

[18] Hu, X., He, Y., Pereira, A. G., & Gómez, A. H. (2005). Nondestructive Determination Method of Fruit Quantity Detection Based on Vis/NIR Spectroscopy Technique. Engineering in Medicine and Biology

[19] Paz, P., S´anchez, M. T., P´erez-Mar´ın, D., Guerrerob, J. e.-E., & Garrido-Varob, A. (2009). Evaluating NIR instruments for quantitative and qualitative assessment of intact apple quality. InWiley Interscience.

[20] Nicola¨ı, B. M., Beullens, K., Bobelyn, E., Peirs, A., Saeys, W., Theron, K. I., et al. (2007). Nondestructive measurement of fruit and vegetable quality by means of NIR spectroscopy: A review. Postharvest Biology and Technology, pp. 46.

[21] Yadav, S. P., Ibaraki, Y., & Gupta, S. D. (2010). Estimation of the chlorophyll content of micropropagated potato plants using RGB based image analysis. Plant Cell Tissue and Organ Culture, 100(2), pp. 183-188.

[22] Gitelson, A. A., Gritz, Y., & Merzlyak, M. N. (2003). Relationships between leaf chlorophyll content and spectral reflectance and algorithms for non-destructive chlorophyll assessment in higher plant leaves. Journal of Plant Physiology, 160(3), pp. 271-282.

[23] Yam, K. L., & Papadakis, S. E. (2004). A simple digital imaging method for measuring and analyzing color of food surfaces. Journal of Food Engineering, 61, pp. 137-142.

[24] OceanControls. Retrieved 12/04/2010, from www.oceancontrols.com.au

[25] Sony Professional. Retrieved 05/06/2009, from www.pro.sony.eu

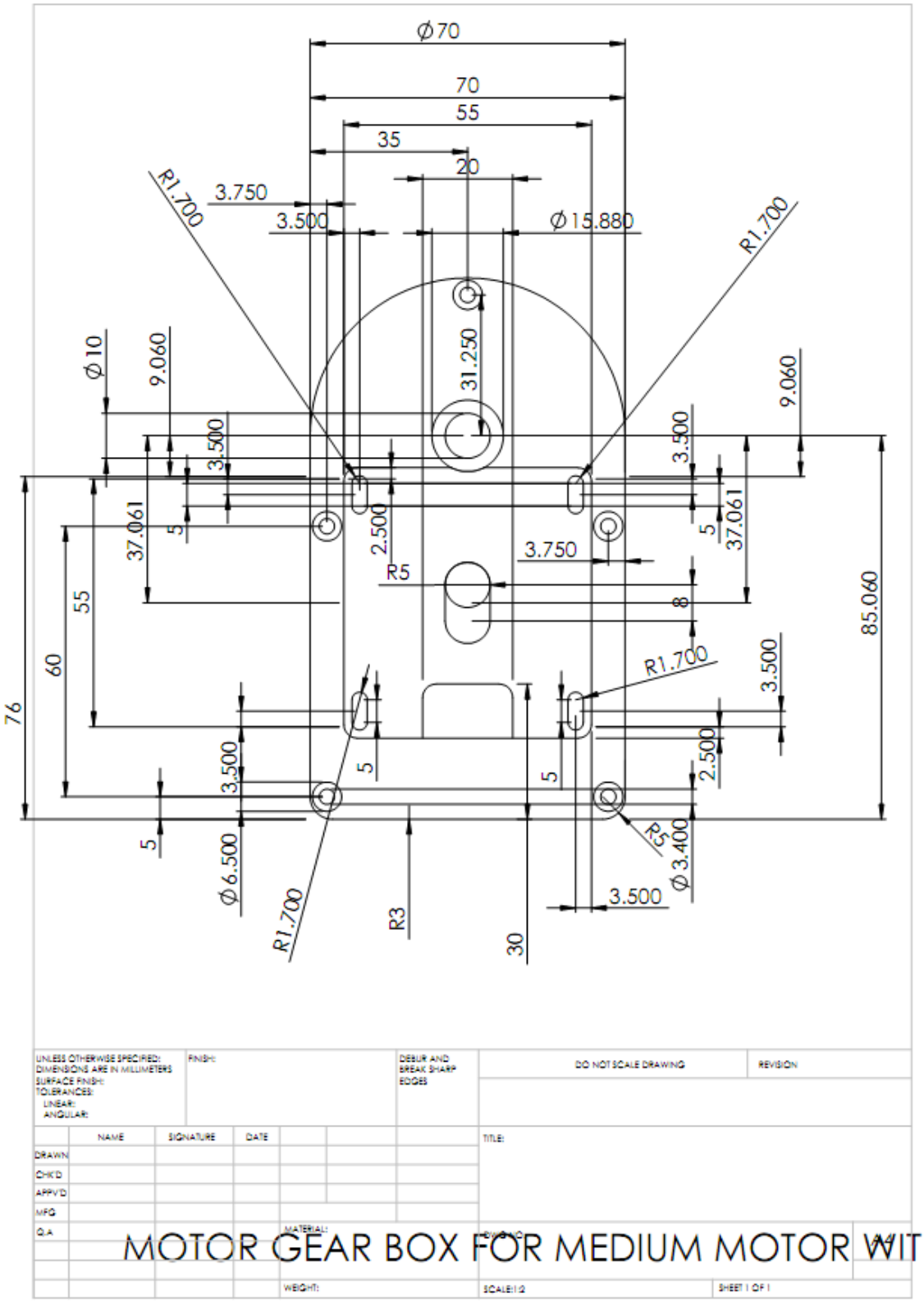
[26] Parallax Home. Retrieved 05/07/2010, from www.Parallax.com

[27] Juckett, R. RGB color space conversion - Linear transformation of color. Retrieved 14/8/2010, from http://www.ryanjuckett.com

[28] Lindbloom, B. J. (2010).Retrieved 15/8/2010, from http://www.brucelindbloom.com/

[29] Fleming, P. J., & Wallace, J. J. (1986). How not to lie with statistics: The correct way to summarize the benchmark results. Communications of the ACM, 29(3).

8.2. Detailed SolidWorks Drawing Showing all Dimensions



8.3. G code Generated in SolidCam of Part Shown Above

```
%
O5000 /* O1111.CNC */
/* 02-FEB-2011*/
G90 G17
G80 G49 G40
G54
G91 G28 Z0
G90
M01
N1 M6 T1
T2
/* TOOL -1- MILL DIA 10.0 R0. MM
*/
G90 G00 G40 G54
G43 H1 D1 G0 X-50.5 Y-55.03
Z70. S4800 M3
M8
/*----------------------------*/
/*SIDES-CONTOUR1-T1 -
PROFILE*/
/*----------------------------*/
X-50.5 Y-55.03 Z50.
  Z2.
G1 Z-3. F100
G41 G1 Y-70.03 F720
G3 X-35.5 Y-55.03 R15.
G1 Y25.03
G3 X-50.5 Y40.03 R15.
G40 G1 Y25.03
G0 Z50.
  Y-55.03
Z-1.
G1 Z-6. F100
G41 G1 Y-70.03 F720
G3 X-35.5 Y-55.03 R15.
G1 Y25.03
G3 X-50.5 Y40.03 R15.
G40 G1 Y25.03
G0 Z50.
  Y-55.03
Z-4.
G1 Z-6.2 F100
G41 G1 Y-70.03 F720
G3 X-35.5 Y-55.03 R15.
G1 Y25.03
G3 X-50.5 Y40.03 R15.
G40 G1 Y25.03
G0 Z50.
  Y-55.03
  Z2.
G1 Z-6.2 F100
G41 G1 Y-70.53 F720
G3 X-35. Y-55.03 R15.5
G1 Y25.03
G3 X-50.5 Y40.53 R15.5
G40 G1 Y25.03
G0 Z50.
  X50.5
  Z2.
G1 Z-3. F100
G41 G1 Y40.03 F720
G3 X35.5 Y25.03 R15.
G1 Y-55.03
G3 X50.5 Y-70.03 R15.
G40 G1 Y-55.03
G0 Z50.
  Y25.03
Z-1.
G1 Z-6. F100
G41 G1 Y40.03 F720
G3 X35.5 Y25.03 R15.
G1 Y-55.03
G3 X50.5 Y-70.03 R15.
G40 G1 Y-55.03
G0 Z50.
  Y25.03
Z-4.
G1 Z-6.2 F100
G41 G1 Y40.03 F720
G3 X35.5 Y25.03 R15.
G1 Y-55.03
G3 X50.5 Y-70.03 R15.
G40 G1 Y-55.03
G0 Z50.
  Y25.03
  Z2.
G1 Z-6.2 F100
G41 G1 Y40.53 F720
G3 X35. Y25.03 R15.5
G1 Y-55.03
G3 X50.5 Y-70.53 R15.5
G40 G1 Y-55.03
G0 Z50.
M9
G91 G28 Z0
G90
M01
N2 M6 T2
T3
/* TOOL -2- MILL DIA 6.0 R0. MM
*/
G90 G00 G40 G54
G43 H2 D2 G0 X0.89 Y-12.031
Z70. S4500 M3
M8
/*----------------------------*/
/*MOTOR-CONTOUR2-T2 -
POCKET*/
/*----------------------------*/
X0.89 Y-12.031 Z50.
  Z2.
G1 Z-2. F80
  Y-9.509 F510
G2 X-0.89 Y-9.509 R34.55
G1 Y-15.421
  X0.89
  Y-12.031
  X2.99
  Y-7.282
G2 X-2.99 Y-7.282 R32.45
G1 Y-17.521
  X2.99
  Y-12.031
  X5.09
  Y-4.89
G2 X-5.09 Y-4.89 R30.35
G1 Y-19.621
  X5.09
  Y-12.031
  X7.19
  Y-2.341
  X6.992
G2 X-6.992 Y-2.341 R28.25
G1 X-7.19
  Y-21.721
  X7.19
  Y-12.031
  X9.29
  Y-0.241
  X6.723
G2 X-6.723 Y-0.241 R26.15
G1 X-9.29
  Y-23.821
  X9.29
  Y-12.031
  X11.39
  Y1.859
  X6.443
G2 X-6.443 Y1.859 R24.05
G1 X-11.39
  Y-25.921
  X11.39
  Y-12.031
  X13.49
  Y3.959
  X6.149
G2 X-6.149 Y3.959 R21.95
G1 X-13.49
  Y-28.021
  X13.49
  Y-12.031
  X15.59
  Y6.059
  X5.842
G2 X-5.842 Y6.059 R19.85
G1 X-15.59
  Y-30.121
  X15.59
  Y-12.031
  X17.69
  Y8.159
  X5.517
G2 X-5.517 Y8.159 R17.75
G1 X-17.69
  Y-32.221
  X17.69
  Y-12.031
  X19.79
  Y10.259
  X5.171
G2 X-5.171 Y10.259 R15.65
G1 X-19.79
  Y-34.321
  X19.79
  Y-12.031
  X21.89
  Y12.359
  X4.801
G2 X-4.801 Y12.359 R13.55
G1 X-21.89
  Y-36.421
  X21.89
  Y-12.031
  X23.99
  Y14.459
  X4.4
G2 X-4.4 Y14.459 R11.45
G1 X-23.99
  Y-38.521
  X23.99
  Y-12.031
  X22.49
G41 G1 Y-17.041 F510
G3 X27.5 Y-12.031 R5.01
G1 Y17.969
  X3.631
G2 X-3.631 Y17.969 R7.94
G1 X-27.5
  Y-42.031
  X27.5
```

```
   Y-12.031
G3 X22.49 Y-7.021 R5.01
G40 G1 Y-12.031
G0 Z50.
M9
G91 G28 Z0
G90
M01
N3 M6 T3
T2
/* TOOL -3- MILL DIA 3.0 R0. MM
*/
G90 G00 G40 G54
G43 H3 D3 G0 X0. Y-7.141 Z70.
S8000 M3
M8
/*-----------------------------------*/
/*MOTOR-MOUNTING-
CONTOUR3-T3 - POCKET*/
/*-----------------------------------*/
   X0. Y-7.141 Z50.
   Z0.
G1 Z-3.5 F100
G3 X-0.89 Y-8.031 R0.89 F500
G1 Y-16.031
G3 X0.89 Y-16.031 R-0.89
G1 Y-8.031
G3 X0. Y-7.141 R0.89
G1 Y-6.091
G3 X-1.94 Y-8.031 R1.94
G1 Y-16.031
G3 X1.94 Y-16.031 R-1.94
G1 Y-8.031
G3 X0. Y-6.091 R1.94
G1 Y-5.041
G3 X-2.99 Y-8.031 R2.99
G1 Y-16.031
G3 X2.99 Y-16.031 R-2.99
G1 Y-8.031
G3 X0. Y-5.041 R2.99
G0 Z50.
   Y-7.141
   Z-1.5
G1 Z-5. F100
G3 X-0.89 Y-8.031 R0.89 F500
G1 Y-16.031
G3 X0.89 Y-16.031 R-0.89
G1 Y-8.031
G3 X0. Y-7.141 R0.89
G1 Y-6.091
G3 X-1.94 Y-8.031 R1.94
G1 Y-16.031
G3 X1.94 Y-16.031 R-1.94
G1 Y-8.031
G3 X0. Y-6.091 R1.94
G1 Y-5.041
G3 X-2.99 Y-8.031 R2.99
G1 Y-16.031
G3 X2.99 Y-16.031 R-2.99
G1 Y-8.031
G3 X0. Y-5.041 R2.99
G0 Z50.
   Y-7.141
Z-3.
G1 Z-6.2 F100
G3 X-0.89 Y-8.031 R0.89 F500
G1 Y-16.031
G3 X0.89 Y-16.031 R-0.89
G1 Y-8.031
G3 X0. Y-7.141 R0.89
G1 Y-6.091
G3 X-1.94 Y-8.031 R1.94
G1 Y-16.031
G3 X1.94 Y-16.031 R-1.94
G1 Y-8.031
G3 X0. Y-6.091 R1.94
G1 Y-5.041
G3 X-2.99 Y-8.031 R2.99
G1 Y-16.031
G3 X2.99 Y-16.031 R-2.99
G1 Y-8.031
G3 X0. Y-5.041 R2.99
G1 Y-6.541 F500
G41 G1 X3.51 F500
G3 X0. Y-3.031 R3.51
X-5. Y-8.031 R5.
G1 Y-16.031
G3 X5.Y-16.031 R-5.
G1 Y-8.031
G3 X0. Y-3.031 R5.
   X-3.51 Y-6.541 R3.51
G40 G1 X0.
G0 Z50.
M9
G91 G28 Z0
G90
M01
N4 M6 T2
T3
/* TOOL -2- MILL DIA 6.0 R0. MM
*/
G90 G00 G40 G54
G43 H2 D2 G0 X0.Y25.46 Z70.
S4500 M3
M8
/*-----------------------------*/
/*BEARING-CONTOUR4-T2 -
POCKET*/
/*-----------------------------*/
   X0. Y25.46 Z50.
   Z2.
G1 Z-3. F80
G3 X0.Y25.46 I0. J-0.43 F510
G1 Y27.56
G3 X0.Y27.56 I0. J-2.53
G1 Y29.66
G3 X0.Y29.66 I0. J-4.63
G0 Z50.
   Y25.46
Z-1.
G1 Z-5. F80
G3 X0.Y25.46 I0. J-0.43 F510
G1 Y27.56
G3 X0.Y27.56 I0. J-2.53
G1 Y29.66
G3 X0.Y29.66 I0. J-4.63
G1 Y27.96 F510
G41 G1 X5.01 F510
G3 X0. Y32.97 R5.01
   X0. Y32.97 I0. J-7.94
   X-5.01 Y27.96 R5.01
G40 G1 X0.
G0 Z50.
M9
G91 G28 Z0
G90
M01
N5 M6 T3
T4
/* TOOL -3- MILL DIA 3.0 R0. MM
*/
G90 G00 G40 G54
G43 H3 D3 G0 X0.Y24.89 Z70.
S8000 M3
M8
/*-------------------------*/
/*AXEL-CONTOUR5-T3 -
POCKET*/
/*-------------------------*/
   X0. Y24.89 Z50.
Z-3.
G1 Z-6.1 F100
G3 X0.Y24.89 I0. J0.14 F500
G1 Y23.84
G3 X0.Y23.84 I0. J1.19
G1 Y22.79
G3 X0.Y22.79 I0. J2.24
G1 Y21.74
G3 X0.Y21.74 I0. J3.29
G0 Z50.
   Y24.89
   Z-4.1
G1 Z-6.2 F100
G3 X0.Y24.89 I0. J0.14 F500
G1 Y23.84
G3 X0.Y23.84 I0. J1.19
G1 Y22.79
G3 X0.Y22.79 I0. J2.24
G1 Y21.74
G3 X0.Y21.74 I0. J3.29
G1 Y23.54 F500
G41 G1 X-3.51 F500
G3 X0. Y20.03 R3.51
   X0. Y20.03 I0. J5.
   X3.51 Y23.54 R3.51
G40 G1 X0.
G0 Z50.
M9
G91 G28 Z0
G90
M01
N6 M6 T4
T5
G90 G00 G40 G54
G43 H4 D4 G0 X-31.25 Y-55.03
Z70. S2500 M3
M8
/*---------------------*/
/*SPOT-DRILL-T4 - DRILL*/
/*---------------------*/
X-31.25 Y-55.03 Z50.
G98 G81 Z-1. R2. F100
   Y4.97
   X31.25
   Y-55.03
G80
M9
G91 G28 Z0
G90
M01
N7 M6 T5
T3
/* TOOL -5- DRILL DIA 3.4 MM */
G90 G00 G40 G54
G43 H5 D5 G0 X-31.25 Y-55.03
Z70. S2200 M3
M8
/*--------------------*/
/*3MM-DRILL-T5 - DRILL*/
/*--------------------*/
X-31.25 Y-55.03 Z50.
G98 G83 Z-6.4 R2. Q2. F80
   Y4.97
   X31.25
   Y-55.03
G80
M9
G91 G28 Z0
G90
M01
N8 M6 T3
T1
/* TOOL -3- MILL DIA 3.0 R0. MM
*/
G90 G00 G40 G54
G43 H3 D3 G0 X-31.25 Y-55.71
Z70. S8000 M3
M8
```

```
/*---------------------------------*/
/*DRILL-HEAD-CONTOUR6-T3 -
POCKET*/
/*---------------------------------*/
X-31.25 Y-55.71 Z50.
  Z2.
G1 Z-1.5 F100
G3 X-31.25 Y-55.71 I0. J0.68 F500
G1 Y-56.76
G3 X-31.25 Y-56.76 I0. J1.73
G0 Z50.
  Y-55.71
  Z0.5
G1 Z-3. F100
G3 X-31.25 Y-55.71 I0. J0.68 F500
G1 Y-56.76
G3 X-31.25 Y-56.76 I0. J1.73
G1 Y-55.77 F500
G41 G1 X-33.76 F500
G3 X-31.25 Y-58.28 R2.51
X-31.25 Y-58.28 I0. J3.25
  X-28.74 Y-55.77 R2.51
G40 G1 X-31.25
G0 Z50.
  Y4.29
  Z2.
G1 Z-1.5 F100
G3 X-31.25 Y4.29 I0. J0.68 F500
G1 Y3.24
G3 X-31.25 Y3.24 I0. J1.73
G0 Z50.
  Y4.29
  Z0.5
G1 Z-3. F100
G3 X-31.25 Y4.29 I0. J0.68 F500
G1 Y3.24
G3 X-31.25 Y3.24 I0. J1.73
G1 Y4.23 F500
G41 G1 X-33.76 F500
G3 X-31.25 Y1.72 R2.51
X-31.25 Y1.72 I0. J3.25
  X-28.74 Y4.23 R2.51
G40 G1 X-31.25
G0 Z50.
  X31.25 Y4.29
  Z2.
G1 Z-1.5 F100
G3 X31.25 Y4.29 I0. J0.68 F500
G1 Y3.24
G3 X31.25 Y3.24 I0. J1.73
G0 Z50.
  Y4.29
  Z0.5
G1 Z-3. F100
G3 X31.25 Y4.29 I0. J0.68 F500
G1 Y3.24
G3 X31.25 Y3.24 I0. J1.73
G1 Y4.23 F500
G41 G1 X28.74 F500
G3 X31.25 Y1.72 R2.51
X31.25 Y1.72 I0. J3.25
  X33.76 Y4.23 R2.51
G40 G1 X31.25
G0 Z50.
  Y-55.71
  Z2.
G1 Z-1.5 F100
G3 X31.25 Y-55.71 I0. J0.68 F500
G1 Y-56.76
G3 X31.25 Y-56.76 I0. J1.73
G0 Z50.
  Y-55.71
  Z0.5
G1 Z-3. F100
G3 X31.25 Y-55.71 I0. J0.68 F500
G1 Y-56.76
G3 X31.25 Y-56.76 I0. J1.73
G1 Y-55.77 F500
G41 G1 X28.74 F500
G3 X31.25 Y-58.28 R2.51
X31.25 Y-58.28 I0. J3.25
  X33.76 Y-55.77 R2.51
G40 G1 X31.25
G0 Z50.
M9
G91 G28 Y0 Z0
M30
G90 G00 G40 G54
G43 H6 D6 G0 X-50.5 Y-55.03
Z70. S2400 M3
M8
/*-----------------------*/
/*FINISH-BEARING-CONTOUR1-
T6 - PROFILE*/
/*-----------------------*/
X-50.5 Y-55.03 Z50.
  Z2.
G1 Z-3. F100
G41 G1 Y-70.03 F360
G3 X-35.5 Y-55.03 R15.
G1 Y25.03
G3 X-50.5 Y40.03 R15.
G40 G1 Y25.03
G0 Z50.
  Y-55.03
Z-1.
G1 Z-6. F100
G41 G1 Y-70.03 F360
G3 X-35.5 Y-55.03 R15.
G1 Y25.03
G3 X-50.5 Y40.03 R15.
G40 G1 Y25.03
G0 Z50.
  Y-55.03
Z-4.
G1 Z-6.2 F100
G41 G1 Y-70.03 F360
G3 X-35.5 Y-55.03 R15.
G1 Y25.03
G3 X-50.5 Y40.03 R15.
G40 G1 Y25.03
G0 Z50.
  Y-55.03
  Z2.
G1 Z-6.2 F100
G41 G1 Y-70.53 F360
G3 X-35. Y-55.03 R15.5
G1 Y25.03
G3 X-50.5 Y40.53 R15.5
G40 G1 Y25.03
G0 Z50.
  X50.5
  Z2.
G1 Z-3. F100
G41 G1 Y40.03 F360
G3 X35.5 Y25.03 R15.
G1 Y-55.03
G3 X50.5 Y-70.03 R15.
G40 G1 Y-55.03
G0 Z50.
  Y25.03
Z-1.
G1 Z-6. F100
G41 G1 Y40.03 F360
G3 X35.5 Y25.03 R15.
G1 Y-55.03
G3 X50.5 Y-70.03 R15.
G40 G1 Y-55.03
G0 Z50.
  Y25.03
Z-4.
G1 Z-6.2 F100
G41 G1 Y40.03 F360
G3 X35.5 Y25.03 R15.
G1 Y-55.03
G3 X50.5 Y-70.03 R15.
G40 G1 Y-55.03
G0 Z50.
  Y25.03
  Z2.
G1 Z-6.2 F100
G41 G1 Y40.53 F360
G3 X35. Y25.03 R15.5
G1 Y-55.03
G3 X50.5 Y-70.53 R15.5
G40 G1 Y-55.03
G0 Z50.
M9
G91 G28 Y0 Z0
M30
%
```

8.4. Methodology / Information on linear transformation from XYZ to sRGB

Methodology / Information on linear transformation from XYZ to sRGB obtained to allow for a transformation matrix to be calculated. This information is from [59]

White point chromaticity coordinates (0.3127, 0.3290). This white point is also known as D65 which is an estimation of the white coloor produced by mid-day sunlight

$$x = \frac{X}{X+Y+Z} \tag{7.1}$$

$$y = \frac{Y}{X+Y+Z} \tag{7.2}$$

$$z = \frac{Z}{X+Y+Z} \tag{7.3}$$

$$\therefore x + y + z = 1 \tag{7.4}$$

$$\frac{X}{X+Y+Z} + \frac{Y}{X+Y+Z} = \frac{Z}{X+Y+Z} = \frac{X+Y+Z}{X+Y+Z} = 1 \tag{7.5}$$

$$\therefore z = 1 - x - y \tag{7.6}$$

using a generalized approach

$$red = \begin{pmatrix} 0.64 \\ 0.33 \end{pmatrix}, green = \begin{pmatrix} 0.30 \\ 0.60 \end{pmatrix}, blue = \begin{pmatrix} 0.15 \\ 0.06 \end{pmatrix}$$

$$i = red = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, j = green = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, k = blue = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\therefore yellow = i + j = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

We can therefore describe colour as

$$Color = ri + gj + bk \tag{7.7}$$

Let *l* be the red primary in XYZ space, *m* be the green primary in XYZ space and *n* be the blue primary in XYZ space.

$$Colour_{XYZ} = r_l + g_m + b_n \tag{7.8}$$

We can take this whole process one step further by recognizing that we are really multiplying a vector by a transformation matrix to convert it from linear RGB space to XYZ space. In the previous example we were using the XYZ vectors l, m and n to build the transformation matrix. Let's specify the three vectors in terms of their components.

$$l = \begin{pmatrix} l_X \\ l_Y \\ l_Z \end{pmatrix}, m = \begin{pmatrix} m_X \\ m_Y \\ m_Z \end{pmatrix}, n = \begin{pmatrix} n_X \\ n_Y \\ n_Z \end{pmatrix} \tag{7.9}$$

By combing these three column vectors into a column major transformation matrix, we can rewrite our color transformation equation as follows.

$$Color_{XYZ=} \begin{pmatrix} l_X & m_X & n_X \\ l_Y & m_Y & n_Y \\ l_Z & m_Z & n_Z \end{pmatrix} \times \begin{pmatrix} r \\ g \\ b \end{pmatrix} \tag{7.10}$$

The goal is to solve for a column-major transformation matrix, M, that will convert from linear RGB space to XYZ space. The first step is to convert all xy chromaticity coordinates to xyz chromaticity coordinates by using the previously discussed equation:

$$z = 1 - x - y \tag{7.11}$$

$$r_{xyz} = \begin{pmatrix} r_x \\ r_y \\ 1 - r_x - r_y \end{pmatrix}, g_{xyz} = \begin{pmatrix} g_x \\ g_y \\ 1 - g_x - g_y \end{pmatrix}, b_{xyz} = \begin{pmatrix} b_x \\ b_y \\ 1 - b_x - b_y \end{pmatrix}, w_{xyz} = \begin{pmatrix} w_x \\ w_y \\ 1 - w_x - w_y \end{pmatrix} \tag{7.12}$$

The next step is to define an equation for converting from xyz space to XYZ space. Using the XYZ to xyz equations $x = \frac{X}{X+Y+Z}$, $y = \frac{Y}{X+Y+Z}$ and $z = \frac{Z}{X+Y+Z}$, let's define an XYZ vector v_{XYZ} in terms of its chromaticity coordinates v_{xyz}

$$v_{xyz} = \frac{v_X}{v_X + v_Y + v_Z}, \frac{v_Y}{v_X + v_Y + v_Z}, \frac{v_Z}{v_X + v_Y + v_Z} \tag{7.13}$$

$$v_{xyz} = \frac{1}{v_X + v_Y + v_Z}(v_X, v_Y, v_Z) = \frac{1}{v_X + v_Y + v_Z} v_{XYZ} \tag{7.14}$$

$$v_{XYZ} = (v_X + v_Y + v_Z) v_{xyz} \tag{7.15}$$

Now we can solve for w_{XYZ} by using our known value of w_Y and the above conversion from xyz space to XYZ space.

$$w_y = \frac{w_Y}{w_X + w_Y + w_Z} \tag{7.16}$$

$$(w_X + w_y + w_z) = \frac{w_y}{w_y} = \frac{1}{w_y} \tag{7.17}$$

$$w_{XYZ} = (w_X + w_Y + w_Z) w_{xyz} = (\frac{1}{w_y}) w_{xyz} \tag{7.18}$$

Now we can work towards solving for M. To do this we first need to recall that the columns of M are the red, green, and blue primaries in XYZ space. Let's define M using our xyz to XYZ conversion,

$$M = [r_{XYZ} g_{XYZ} b_{XYZ}] \tag{7.19}$$

$$M = [(r_X + r_Y + r_Z) r_{xyz} + (g_X + g_Y + g_Z) g_{xyz}] + (b_X + b_Y + b_Z) b_{xyz}] \tag{7.20}$$

Expanding our column vector notation we get:

$$M = \begin{pmatrix} r_X + r_Y + r_Z) r_x & g_X + g_Y + g_Z) g_x & b_X + b_Y + b_Z) b_x \\ r_X + r_Y + r_Z) r_y & g_X + g_Y + g_Z) g_y & b_X + b_Y + b_Z) b_y \\ r_X + r_Y + r_Z) r_z & g_X + g_Y + g_Z) g_y & b_X + b_Y + b_Z) b_z \end{pmatrix} \tag{7.21}$$

Noticing that each column of M is a scalar multiple of the respective xyz primary colour, we can factor a scale matrix out of M like so:

$$M = \begin{pmatrix} r_x & g_x & b_x \\ r_y & g_y & b_y \\ r_z & g_z & b_z \end{pmatrix} \begin{pmatrix} (r_X + r_Y + r_Z) & 0 & 0 \\ 0 & (g_X + g_Y + g_Z) & 0 \\ 0 & 0 & (b_X + b_Y + b_Z) \end{pmatrix} \tag{7.22}$$

We know all of xyz primary values and thus know the left matrix in the above equation. Our unknowns are now down to the three scalar values that make up the right matrix. Note that while each of those scalars is written as a sum of X, Y and Z components, we only really care about the summed result and not the individual parts. That is why I say there are only three unknowns instead of nine. In order to solve for the three unknowns, we will use our known RGB and XYZ values of the white point. Because M transforms from RGB to XYZ space, we can declare the following equality:

$$w_{XYZ} = M\,\omega_{RGB} \tag{7.23}$$

$$\omega_{XYZ} = \begin{pmatrix} r_x & g_x & b_x \\ r_y & g_y & b_y \\ r_z & g_z & b_z \end{pmatrix} \begin{pmatrix} (r_X + r_Y + r_Z) & 0 & 0 \\ 0 & (g_X + g_Y + g_Z) & 0 \\ 0 & 0 & (b_X + b_Y + b_Z) \end{pmatrix} \omega_{RGB} \tag{7.24}$$

Let's substitute in our know the value of w_{RGB}.

$$\omega_{XYZ} = \begin{pmatrix} r_x & g_x & b_x \\ r_y & g_y & b_y \\ r_z & g_z & b_z \end{pmatrix} \begin{pmatrix} (r_X + r_Y + r_Z) & 0 & 0 \\ 0 & (g_X + g_Y + g_Z) & 0 \\ 0 & 0 & (b_X + b_Y + b_Z) \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{7.25}$$

Now we can multiply our substituted w_{RGB} vector by the scale matrix.

$$\omega_{XYZ} = \begin{pmatrix} r_x & g_x & b_x \\ r_y & g_y & b_y \\ r_z & g_z & b_z \end{pmatrix} \begin{pmatrix} (r_X + r_Y + r_Z) \\ (g_X + g_Y + g_Z) \\ (b_X + b_Y + b_Z) \end{pmatrix} \tag{7.26}$$

We will now left multiply each side of the equation by the inverse of our remaining 3x3 matrix. This will put all of our known values on the left side of the equation and our unknown values on the right. If you are unfamiliar with calculating the inverse of a 3x3 matrix, I will provide code at the end of the article, but for the sake of sanity won't be writing a full derivation here. Google should bring up a lot of results about the process including this one from Wikipedia and this one from Mathworlds.

$$\begin{pmatrix} r_x & g_x & b_x \\ r_y & g_y & b_y \\ r_z & g_z & b_z \end{pmatrix}^{-1} \omega_{XYZ} = \begin{pmatrix} (r_X + r_Y + r_Z) \\ (g_X + g_Y + g_Z) \\ (b_X + b_Y + b_Z) \end{pmatrix} \tag{7.27}$$

Example:

$$r_{xy} = \begin{pmatrix} 0.703708691 \\ 0.296217118 \end{pmatrix} \tag{7.28}$$

$$g_{xy} = \begin{pmatrix} 0.082053395 \\ 0.834090315 \end{pmatrix} \tag{7.29}$$

$$b_{xy} = \begin{pmatrix} 0.133509341 \\ 0.04269239 \end{pmatrix} \tag{7.30}$$

$$w_{xy} = \begin{pmatrix} 0.3127 \\ 0.3290 \end{pmatrix} \tag{7.31}$$

$$r_{xyz} = \begin{pmatrix} r_x \\ r_y \\ 1 - r_x - r_y \end{pmatrix} = \begin{pmatrix} 0.703708691 \\ 0.296217118 \\ 0.000074191 \end{pmatrix} \tag{7.32}$$

$$g_{xyz} = \begin{pmatrix} g_x \\ g_y \\ 1 - g_x - g_y \end{pmatrix} = \begin{pmatrix} 0.082053395 \\ 0.834090315 \\ 0.08385629 \end{pmatrix} \tag{7.33}$$

$$b_{xyz} = \begin{pmatrix} b_x \\ b_y \\ 1-b_x-b_y \end{pmatrix} = \begin{pmatrix} 0.133509341 \\ 0.04269239 \\ 0.823798269 \end{pmatrix} \tag{7.34}$$

$$w_{xyz} = \begin{pmatrix} w_x \\ w_y \\ 1-w_x-w_y \end{pmatrix} = \begin{pmatrix} 0.3127 \\ 0.3290 \\ 0.3583 \end{pmatrix} \tag{7.35}$$

$$w_{XYZ} = (w_X + w_Y + w_Z)w_{xyz} = (\frac{1}{w_y})w_{xyz} = \frac{1}{0.3290}\begin{pmatrix} 0.3127 \\ 0.3290 \\ 0.3583 \end{pmatrix} = \begin{pmatrix} 0.950452 \\ 1 \\ 1.08906 \end{pmatrix} \tag{7.36}$$

$$\begin{pmatrix} (r_X + r_Y + r_Z) \\ (g_X + g_Y + g_Z) \\ (b_X + b_Y + b_Z) \end{pmatrix} = \begin{pmatrix} r_x g_x b_x \\ r_y g_y b_y \\ r_z g_z b_z \end{pmatrix}^{-1} w_{XYZ} \tag{7.37}$$

$$\begin{pmatrix} (r_X + r_Y + r_Z) \\ (g_X + g_Y + g_Z) \\ (b_X + b_Y + b_Z) \end{pmatrix} = \begin{pmatrix} 0.703708691 & 0.082053395 & 0.133509341 \\ 0.296217118 & 0.834090315 & 0.04269239 \\ 0.000074191 & 0.08385629 & 0.8237698269 \end{pmatrix}^{-1} \begin{pmatrix} 0.950452 \\ 1 \\ 1.08906 \end{pmatrix} \tag{7.38}$$

$$\begin{pmatrix} (r_X + r_Y + r_Z) \\ (g_X + g_Y + g_Z) \\ (b_X + b_Y + b_Z) \end{pmatrix} = \begin{pmatrix} 1.472198696816915 & -0.121470974849569 & -0.232315757264834 \\ -0.525560804187442 & 1.248544248685019 & 0.020575470073666 \\ 0.053367280373085 & -0.127085592508812 & 1.211857755108272 \end{pmatrix} \begin{pmatrix} 0.950452 \\ 1 \\ 1.089060 \end{pmatrix} \tag{7.39}$$

$$\begin{pmatrix} (r_X + r_Y + r_Z) \\ (g_X + g_Y + g_Z) \\ (b_X + b_Y + b_Z) \end{pmatrix} = \begin{pmatrix} 1.024777422330621 \\ 0.771431852661884 \\ 1.243423252634561 \end{pmatrix} \tag{7.40}$$

$$M = \begin{pmatrix} r_x & g_x & b_x \\ r_y & g_y & b_y \\ r_z & g_z & b_z \end{pmatrix} \begin{pmatrix} (r_X + r_Y + r_Z) & 0 & 0 \\ 0 & (g_X + g_Y + g_Z) & 0 \\ 0 & 0 & (b_X + b_Y + b_Z) \end{pmatrix} \tag{7.41}$$

$$M \approx \begin{pmatrix} 0.703708691 & 0.082053395 & 0.133509341 \\ 0.296217118 & 0.834090315 & 0.04269269 \\ 0.000074191 & 0.08385629 & 0.823798269 \end{pmatrix} \begin{pmatrix} 1.024777422330621 & 0 & 0 \\ 0 & 0.771431852661884 & 0 \\ 0 & 0 & 1.243423252634561 \end{pmatrix} \tag{7.42}$$

$$M \approx \begin{pmatrix} 0.721144778434635 & 0.063298602522047 & 0.166008619043317 \\ 0.303556614634245 & 0.643443836987784 & 0.052999548377970 \\ 0.000076029261740 & 0.064689413152052 & 1.024294557586207 \end{pmatrix} \quad (7.43)$$

If you want the reverse transformation from XYZ to linear sRGB space, use the inverse of M.

M^{-1}

$$M^{-1} \approx \begin{pmatrix} 1.436603368435593 & -0.118534007680723 & -0.226698746676605 \\ -0.681279626157456 & 1.618476401223029 & 0.026671792203898 \\ 0.042919641610377 & -0.102206221605993 & 0.974614036323184 \end{pmatrix} \quad (7.44)$$

8.5. Sony VISCA Protocol for the Various Camera Controls

Command Set	Command	Command Packet	Comments
AddressSet	Broadcast	88 30 01 FF	
IF_Clear	Broadcast	88 01 00 01 FF	
CommandCancel		8x 2p FF	p: Socket No.(=1or2)
CAM_Power	On	8x 01 04 00 02 FF	Power ON/OFF
	Off	8x 01 04 00 03 FF	
CAM_Zoom	Stop	8x 01 04 07 00 FF	
	Tele(Standard)	8x 01 04 07 02 FF	
	Wide(Standard)	8x 01 04 07 03 FF	
	Tele(Variable)	8x 01 04 07 2p FF	p=0 (Low) to 7 (High)
	Wide(Variable)	8x 01 04 07 3p FF	
	Direct	8x 01 04 47 0p 0q 0r 0s FF	pqrs: Zoom Position
CAM_DZoom	On	8x 01 04 06 02 FF	Digital zoom ON/OFF
	Off	8x 01 04 06 03 FF	
	Combine Mode	8x 01 04 36 00 FF	Optical/Digital Zoom Combined
	Separate Mode	8x 01 04 36 01 FF	Optical/Digital Zoom Separate
	Stop	8x 01 04 06 00 FF	
	Tele(Variable)	8x 01 04 06 2p FF	p=0 (Low) to 7 (High)
	Wide(Variable)	8x 01 04 06 3p FF	
	x1/Max	8x 01 04 06 10 FF	x1/MAX Magnification Switchover
	Direct	8x 01 04 46 00 00 0p 0q FF	pq: D-Zoom Position
CAM_Focus	Stop	8x 01 04 08 00 FF	
	Far(Standard)	8x 01 04 08 02 FF	
	Near(Standard)	8x 01 04 08 03 FF	
	Far(Variable)	8x 01 04 08 2p FF	p=0 (Low) to 7 (High)
	Near(Variable)	8x 01 04 08 3p FF	
	Direct	8x 01 04 48 0p 0q 0r 0s FF	pqrs: Focus Position
	Auto Focus	8x 01 04 38 02 FF	AF ON/OFF
	Manual Focus	8x 01 04 38 03 FF	
	Auto/Manual	8x 01 04 38 10 FF	
	One Push Trigger	8x 01 04 18 01 FF	One Push AF Trigger
	Infinity	8x 01 04 18 02 FF	Forced infinity
	Near Limit	8x 01 04 28 0p 0q 0r 0s FF	pqrs: Focus Near Limit Position
AF Sensitivity	Normal	8x 01 04 58 02 FF	AF Sensitivity High/Low
	Low	8x 01 04 58 03 FF	
CAM_AFMode	Normal AF	8x 01 04 57 00 FF	AF Movement Mode
	Interval AF	8x 01 04 57 01 FF	
	Zoom Trigger AF	8x 01 04 57 02 FF	
	Active/Interval Time	8x 01 04 27 0p 0q 0r 0s FF	pq: Movement Time, rs: Interval
CAM_ZoomFocus	Direct	8x 01 04 47 0p 0q 0r 0s 0t 0u 0v 0w FF	pqrs: Zoom Position tuvw: Focus Position
CAM_Initialize	Lens	8x 01 04 19 01 FF	Lens Initialization Start
	Comp Scan	8x 01 04 19 02 FF	Start of Fault Correction Movement

Command Set	Command	Command Packet	Comments
CAM_WB	Auto	8x 01 04 35 00 FF	Normal Auto
	Indoor	8x 01 04 35 01 FF	Indoor mode
	Outdoor	8x 01 04 35 02 FF	Outdoor mode
	One Push WB	8x 01 04 35 03 FF	One Push WB mode
	ATW	8x 01 04 35 04 FF	Auto Tracing White Balance
	Manual	8x 01 04 35 05 FF	Manual Control mode
	One Push Trigger	8x 01 04 10 05 FF	One Push WB Trigger
CAM_RGain	Reset	8x 01 04 03 00 FF	Manual Control of R Gain
	Up	8x 01 04 03 02 FF	
	Down	8x 01 04 03 03 FF	
	Direct	8x 01 04 43 00 00 0p 0q FF	pq: R Gain
CAM_BGain	Reset	8x 01 04 04 00 FF	Manual Control of B Gain
	Up	8x 01 04 04 02 FF	
	Down	8x 01 04 04 03 FF	
	Direct	8x 01 04 44 00 00 0p 0q FF	pq: B Gain
CAM_AE	Full Auto	8x 01 04 39 00 FF	Automatic Exposure mode
	Manual	8x 01 04 39 03 FF	Manual Control mode
	Shutter Priority	8x 01 04 39 0A FF	Shutter Priority Automatic Exposure mode
	Iris Priority	8x 01 04 39 0B FF	Iris Priority Automatic Exposure mode
	Bright	8x 01 04 39 0D FF	Bright Mode (Manual control)
CAM_SlowShutter	Auto	8x 01 04 5A 02 FF	Auto Slow Shutter ON/OFF
	Manual	8x 01 04 5A 03 FF	
CAM_Shutter	Reset	8x 01 04 0A 00 FF	Shutter Setting
	Up	8x 01 04 0A 02 FF	
	Down	8x 01 04 0A 03 FF	
	Direct	8x 01 04 4A 00 00 0p 0q FF	pq: Shutter Position
CAM_Iris	Reset	8x 01 04 0B 00 FF	Iris Setting
	Up	8x 01 04 0B 02 FF	
	Down	8x 01 04 0B 03 FF	
	Direct	8x 01 04 4B 00 00 0p 0q FF	pq: Iris Position
CAM_Gain	Reset	8x 01 04 0C 00 FF	Gain Setting
	Up	8x 01 04 0C 02 FF	
	Down	8x 01 04 0C 03 FF	
	Direct	8x 01 04 4C 00 00 0p 0q FF	pq: Gain Position
CAM_Bright	Reset	8x 01 04 0D 00 FF	Bright Setting
	Up	8x 01 04 0D 02 FF	
	Down	8x 01 04 0D 03 FF	
	Direct	8x 01 04 4D 00 00 0p 0q FF	pq: Bright Position
CAM_ExpComp	On	8x 01 04 3E 02 FF	Exposure Compensation ON/OFF
	Off	8x 01 04 3E 03 FF	
	Reset	8x 01 04 0E 00 FF	Exposure Compensation Amount Setting
	Up	8x 01 04 0E 02 FF	
	Down	8x 01 04 0E 03 FF	
	Direct	8x 01 04 4E 00 00 0p 0q FF	pq: ExpComp Position
CAM_Backlight	On	8x 01 04 33 02 FF	Back Light Compensation ON/OFF
	Off	8x 01 04 33 03 FF	
CAM_SpotAE	On	8x 01 04 59 02 FF	Spot Automatic Exposure Setting
	Off	8x 01 04 59 03 FF	
	Position	8x 01 04 29 0p 0q 0r 0s FF	pq: X (0 to F), rs: Y (0 to F)

Command Set	Command	Command Packet	Comments
CAM_Aperture	Reset	8x 01 04 02 00 FF	Aperture Control
	Up	8x 01 04 02 02 FF	
	Down	8x 01 04 02 03 FF	
	Direct	8x 01 04 42 00 00 0p 0q FF	pq: Aperture Gain
CAM_LR_Reverse	On	8x 01 04 61 02 FF	Mirror Image ON/OFF
	Off	8x 01 04 61 03 FF	
CAM_Freeze	On	8x 01 04 62 02 FF	Still Image ON/OFF
	Off	8x 01 04 62 03 FF	
CAM_PictureEffect	Off	8x 01 04 63 00 FF	Picture Effect Setting
	Neg.Art	8x 01 04 63 02 FF	
	B&W	8x 01 04 63 04 FF	
CAM_Memory	Reset	8x 01 04 3F 00 pp FF	p: Memory Number (=0 to 5)
	Set	8x 01 04 3F 01 pp FF	
	Recall	8x 01 04 3F 02 7F FF	
CAM_CUSTOM	Reset	8x 01 04 3F 00 7F FF	Starts in this mode at Power ON.
	Set	8x 01 04 3F 01 7F FF	
	Recall	8x 01 04 3F 02 7F FF	
CAM_Display	On	8x 01 04 15 02 FF (8x 01 06 06 02 FF)	Display ON/OFF
	Off	8x 01 04 15 03 FF (8x 01 06 06 03 FF)	
	On/Off	8x 01 04 15 10 FF (8x 01 06 06 10 FF)	
CAM_Date/TimeSet	Date/TimeSet	8x 01 04 70 0m 0n 0p 0q 0r 0s 0t 0u 0v 0w FF (8x 01 07 29 0m 0n 0p 0q 0r 0s 0t 0u 0v 0w FF)	mn: Year (20mn) pq: Month, rs: Day tu: Hour, vw: Minute
CAM_DateDisplay	On	8x 01 04 71 02 FF (8x 01 07 2A 02 FF)	Date display ON/OFF
	Off	8x 01 04 71 03 FF (8x 01 07 2A 03 FF)	
CAM_TimeDisplay	On	8x 01 04 72 02 FF (8x 01 07 2B 02 FF)	Time display ON/OFF
	Off	8x 01 04 72 03 FF (8x 01 07 2B 03 FF)	
CAM_Title	Title Set1	8x 01 04 73 00 mm nn pp qq 00 00 00 00 00 00 FF	mm: Vposition, nn: Hposition pp: Color, qq: Blink
	Title Set2	8x 01 04 73 01 mm nn pp qq rr ss tt uu vv ww FF	mnpqrstuvw: Setting of Display Characters (1st to 10st Character)
	Title Set3	8x 01 04 73 02 mm nn pp qq rr ss tt uu vv ww FF	mnpqrstuvw: Setting of Display Characters (11th to 20th Character)
	Title Clear	8x 01 04 74 00 FF	Title Setting Clear
	On	8x 01 04 74 02 FF	Title Display ON/OFF
	Off	8x 01 04 74 03 FF	
CAM_Mute	On	8x 01 04 75 02 FF	Mute ON/OFF
	Off	8x 01 04 75 03 FF	
	On/Off	8x 01 04 75 10 FF	
CAM_KEY Lock	Off	8x 01 04 17 00 FF	Camera control on/off
	On	8x 01 04 17 02 FF	
CA_ID Write		8x 01 04 22 0p 0q 0r 0s FF	pqrs: Camera ID (0000~FFFF)

8.6. Basic Stamp Program Example

```
' {$STAMP BS2e}
' {$PBASIC 2.5}

' -----[ I/O Definitions ]--------------------------------------------------

 Ping        PIN 15      ' Serial I/O com pin for Proximity
Sensor.

' Sig       BS2   DB-Expander
' ---       ---   -----------
 S0    PIN  0   '   A       pin allocation for TCS3200
 S1    PIN  1   '   B
 OUT   PIN  2   '   C
 LED   PIN  3   '   D
 S2    PIN  4   '   E
 S3    PIN  5   '   F

 ShtData       CON    14                ' bi-directional data
 Clock         CON    13


' -----[ Constants ]--------------------------------------------------------

pRED    CON  107/10    'Red reading period.
pGREEN  CON  1099/100  'Green reading period.
pBLUE   CON  108/10    'Blue reading period.
INPUT_PIN       CON       16
IsHigh          CON       1                 ' for PULSOUT
IsLow           CON       0
Trigger         CON       5                 ' trigger pulse = 10 uS
Scale           CON       $200              ' raw x 2.00 = uS
baud CON 84 + 32768

ShtTemp       CON    %00011              ' read temperature
ShtHumi       CON    %00101              ' read humidity
ShtStatW      CON    %00110              ' status register write
ShtStatR      CON    %00111              ' status register read
ShtReset      CON    %11110              ' soft reset (wait 11
ms after)

Ack           CON    13
NoAck         CON    14

No            CON    13
Yes           CON    14

MoveTo        CON    2                   ' for DEBUG control
ClrRt         CON    11                  ' clear DEBUG line to
right

DegSym        CON    186                 ' degrees symbol for
DEBUG

' -----[ Variables ]--------------------------------------------------------

RED     VAR  Word  'Red color reading.
GREEN   VAR  Word  'Green color reading.
BLUE    VAR  Word  'Blue color reading.
Counter VAR  Nib

Command  VAR  Byte
rawDist  VAR  Word                ' raw measurement
i        VAR  Nib

ioByte        VAR    Byte           ' data from/to SHT1x
ackBit        VAR    Bit            ' ack/nak from/to SHT1x
toDelay       VAR    Byte           ' timeout delay timer
timeOut       VAR    Bit            ' timeout status

soT           VAR    Word           ' temp counts from
SHT1x
tC            VAR    Word           ' temp - celcius
tF            VAR    Word           ' temp - fahrenheit
soRH          VAR    Word           ' humidity counts from
SHT1x
rhLin         VAR    Word           ' humidity; linearized
rhTrue        VAR    Word           ' humidity; temp
compensated

status        VAR    Byte           ' SHT1x status byte
' -----[ Initialization ]--------------------------------------------------

HIGH  S0        'Maximum output rate.
HIGH  S1        '      "

   GOSUB SHT_Connection_Reset        ' reset device
connection
 PAUSE 250                           ' let DEBUG window open


 'GOTO Main                          ' skip heater demo

 Main3:
DO
  GOSUB GetInput            'gets data from serial port of
computer
 LOOP
 END



GetInput:                   ' Receives input from some external
source
 SERIN INPUT_PIN, baud, [Command]
 GOSUB TakeAction
RETURN


TakeAction:  ' Determines what to do based on result of
GetInput
 SELECT Command

  CASE "Prox"
  GOSUB Task1
  RETURN

  CASE "TCS32"
  GOSUB Task2
  RETURN

  CASE "emp"
  GOSUB Task3
  RETURN


 ENDSELECT


Task1:
  FOR i = 1 TO 10
    GOSUB Get_Sonar                  ' get sensor value
      SEROUT 16,baud,[" &",DEC3 rawDist, CR]
  NEXT
PAUSE 500

RETURN

Task2:

HIGH  LED       'Turn on LED.
PAUSE 200
 FOR i = 0 TO 10
GOSUB  Color   'Get the color data, and output to DEBUG.
```

```
    SEROUT 16, baud,["&", "R", DEC3 RED, " G", DEC3
GREEN, " B", DEC3 BLUE, CR]
  NEXT
PAUSE 200
LOW LED

RETURN

Task3:

Sensor_Demo:
  GOSUB SHT_Measure_Temp
     Heater_On:
status = %00000100                    ' heater bit = On
      GOSUB SHT_Write_Status

PAUSE 2000

     Heater_Off:
status = %00000000                    ' heater bit = Off
     GOSUB SHT_Write_Status

    GOSUB SHT_Measure_Temp

PAUSE 5000

 FOR i = 1 TO 5
  GOSUB SHT_Measure_Temp
  'DEBUG DEC (tC / 10), ".", DEC1 tC, DegSym, ClrRt, CR
  SEROUT 16, baud,["&", DEC (tC / 10), ".", DEC1 tC, CR]
  'GOSUB SHT_Measure_Humidity

  PAUSE 100                           ' minimum delay between
readings
  NEXT

 RETURN


RETURN

' -----[ Subroutines ]-----------------------------------------------------
'Proximity
Get_Sonar:
  Ping = IsLow                        ' make trigger 0-1-0
  PULSOUT Ping, Trigger                ' activate sensor
PULSIN  Ping, IsHigh, rawDist          ' measure echo pulse
rawDist = rawDist */ Scale             ' convert to uS
' remove return trip
  RETURN

'TCS3200
  Color:
LOW  S2                       'Address the red output.
LOW  S3
COUNT  OUT, pRED, RED         'Read the red component.
HIGH  S3                      'Address the blue output.
COUNT  OUT, pBLUE, BLUE       'Read the blue
component.
HIGH  S2                      'Address the green output.
COUNT  OUT, pGREEN, GREEN     'Read the green
component.
  RETURN

'Temp & Humidity

'
SHT_Connection_Reset:
  SHIFTOUT ShtData, Clock, LSBFIRST, [$FFF\9]

' generates SHT1x "start" sequence
'       _____     _____
' ShtData      |_____|
'          ___   ___
' Clock    ___|  |___|  |___
'

SHT_Start:
  INPUT ShtData                       ' let pull-up take line high
  LOW Clock
  HIGH Clock
  LOW ShtData
  LOW Clock
  HIGH Clock
  INPUT ShtData
  LOW Clock
  RETURN


' measure temperature
' -- celcius = soT * 0.01 - 40
' -- fahrenheit = soT * 0.018 - 40
'
SHT_Measure_Temp:
  GOSUB SHT_Start                      ' alert device
ioByte = ShtTemp                      ' temperature command
  GOSUB SHT_Write_Byte                 ' send command
  GOSUB SHT_Wait                       ' wait until
measurement done
ackBit = Ack                          ' another read follows
  GOSUB SHT_Read_Byte                  ' get MSB
  soT.HIGHBYTE = ioByte
ackBit = NoAck                        ' last read
  GOSUB SHT_Read_Byte                  ' get LSB
  soT.LOWBYTE = ioByte

' Note: Conversion factors are multiplied by 10 to return the
  '       temperature values in tenths of degrees

tC = soT / 10 - 400                   ' convert to tenths C
tF = soT ** 11796 - 400               ' convert to tenths F
  RETURN


measure humidity

SHT_Measure_Humidity:
  GOSUB SHT_Start                      ' alert device
ioByte = ShtHumi                      ' humidity command
  GOSUB SHT_Write_Byte                 ' send command
  GOSUB SHT_Wait                       ' wait until
measurement done
ackBit = Ack                          ' another read follows
  GOSUB SHT_Read_Byte                  ' get MSB
  soRH.HIGHBYTE = ioByte
ackBit = NoAck                        ' last read
  GOSUB SHT_Read_Byte                  ' get LSB
  soRH.LOWBYTE = ioByte

linearize humidity
rhLin = (soRH * 0.0405) - (soRH^2 * 0.0000028) - 4

   FOR the BASIC Stamp:
rhLin = (soRH * 0.0405) - (soRH * 0.004 * soRH * 0.0007) - 4

   Conversion factors are multiplied by 10 AND THEN rounded
TO
   RETURN tenths

rhLin = (soRH ** 26542)
rhLin = rhLin - ((soRH ** 3468) * (soRH ** 3468) + 50 / 100)
rhLin = rhLin - 40

   'temperature compensated humidity
rhTrue = (tC - 25) * (soRH * 0.00008 + 0.01) + rhLin

   Conversion factors are multiplied by 100 TO improve
accuracy AND THEN
' rounded off.
```

```
rhTrue = ((tC / 10 - 25) * (soRH ** 524 + 1) + (rhLin * 10)) +          RETURN
5 / 10
  RETURN                                                                 RETURN


sends "status"

SHT_Write_Status:
  GOSUB SHT_Start                       ' alert device
ioByte = ShtStatW                       ' write to status reg
command
  GOSUB SHT_Write_Byte                  ' send command
ioByte = status
  GOSUB SHT_Write_Byte
  RETURN


' returns "status"
'
SHT_Read_Status:
  GOSUB SHT_Start                       ' alert device
ioByte = ShtStatW                       ' write to status reg
command
  GOSUB SHT_Read_Byte                   ' send command
ackBit = NoAck                          ' only one byte to read
  GOSUB SHT_Read_Byte
  RETURN


' sends "ioByte"
' returns "ackBit"
'
SHT_Write_Byte:
  SHIFTOUT ShtData, Clock, MSBFIRST, [ioByte]   ' send
byte
SHIFTIN  ShtData, Clock, LSBPRE, [ackBit\1]   ' get ack bit
  RETURN


' returns "ioByte"
' sends "ackBit"
'
SHT_Read_Byte:
SHIFTIN  ShtData, Clock, MSBPRE, [ioByte]     ' get byte
  SHIFTOUT ShtData, Clock, LSBFIRST, [ackBit\1] ' send ack
bit
  INPUT ShtData                         ' release data line
  RETURN


' wait for device to finish measurement (pulls data line low)
' -- timeout after ~1/4 second
'
SHT_Wait:
  INPUT ShtData                         ' data line is input
  FOR toDelay = 1 TO 250                ' give ~1/4 second to
finish
timeOut = INS.LOWBIT(ShtData)           ' scan data line
    IF (timeOut = No) THEN SHT_Wait_Done        ' if low,
we're done
    PAUSE 1
  NEXT

SHT_Wait_Done:
  RETURN


' reset SHT1x with soft reset
'
SHT_Soft_Reset:
  GOSUB SHT_Connection_Reset            ' reset the
connection
ioByte = ShtReset                       ' reset command
ackBit = NoAck                          ' only one byte to send
  GOSUB SHT_Write_Byte                  ' send it
  PAUSE 11                              ' wait at least 11 ms
```