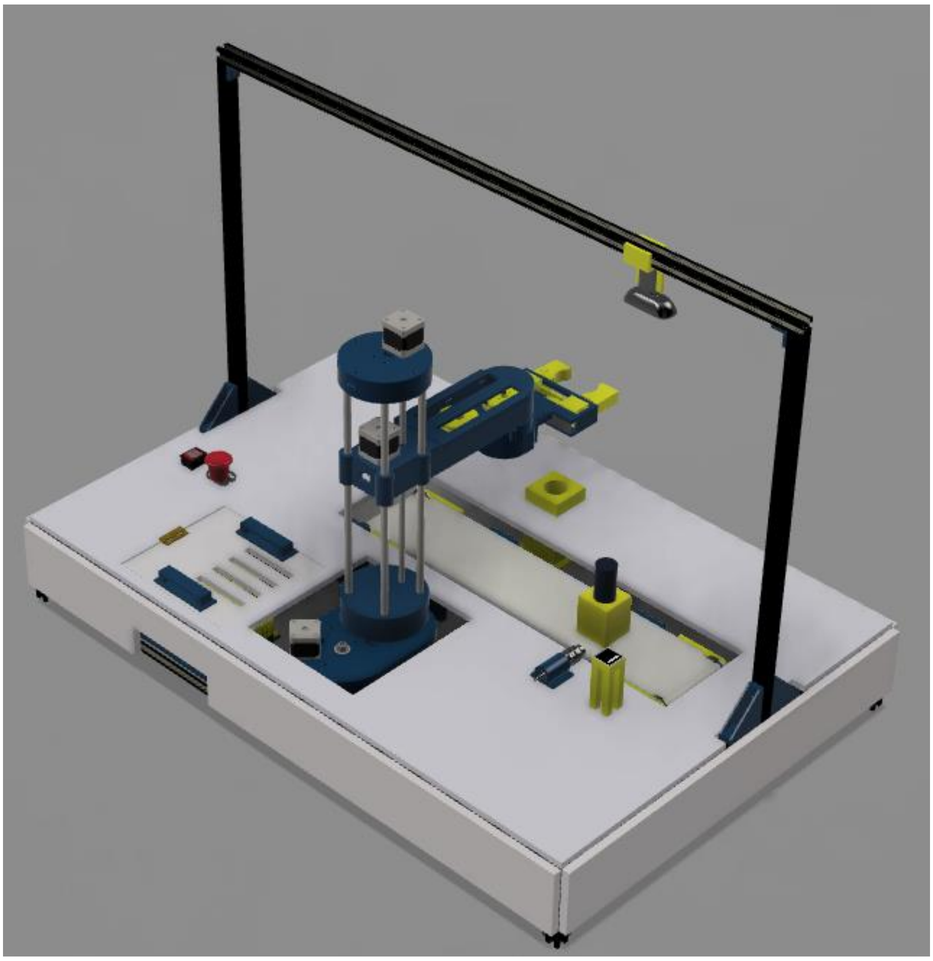
Peg-In-Hole Assembly/Disassembly using at least 3-DOF Robotic Arm and PC Based Machine Vision

R. Le Roux (217833829), *Student*, *NMU*

Conceptual robotic station design.



This project includes the design of a robotic station, which will be used to illustrate the system tracking and control capabilities of integrated machine vision systems.

The engineering methods and resources used in this project, are highly applicable to most automation and manufacturing processes.

Department of Mechatronics Declaration Form

Submission without the completed form will not be accepted.

Surname & Initials:	Le Roux, R
Student Number:	217833829
Supervisor:	Gorlach, Igor (Prof)
Module Code:	EMPV400
Thesis Title:	Robotic Station for Peg-in-hole Assembly
Submission Date:	2022/11/04

I declare that this material submitted for assessment, is entirely my own work and has not been taken from the work of others. The work is based solely on my own study and/or research. I completely understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying.

I have identified and included all sources of all facts, ideas, opinions, and viewpoints of others in the reference list. This thesis, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study, or in any other institutions. I understand that this thesis may undergo electronic detection for plagiarism and an anonymous copy may be retained on the database and used to make comparisons with other work in future.

I have read and understood the NMMU's library's referencing guidelines found at: http://ebeitlibrarian.nmmu.ac.za/Reference-Techniques-(citing sources). I also have read and understood the EBEIT Faculty's plagiarism policy found at: http://ebeit.nmmu.ac.za/ebeit/media/Store/documents/Research Guidelines/Plagiarism and Terminology/plagiarism2.pdf

Student Signature: Roux

Date:2022/11/04

Table of Contents

List of Figures

List of Tables

Peg-In-Hole Assembly/Disassembly using at least 3-DOF Robotic Arm and PC Based Machine Vision

R. Le Roux (217833829), *Student, NMU*

I. ABSTRACT

The objective of the robotic station designed in this report is to perform peg-in-hole assembly/disassembly of parts, off a conveyor. The station will track and execute the process using machine vision. The machine vision system will consist of a camera, which will provide a live feed of the operation, and integrated software, which will process the live camera feed in real time and provide continuous control. The station will interface with the operator through a graphical user interface on a personal computer.

The robotic system consists of three major electro-mechanical components: the at least three degrees of freedom robotic arm, controllable linear conveyor, and station structure/enclosure. A three degrees of freedom SCARA robot design is implemented in this project. A modular design approach was followed for the actuators of the robot. The majority of the robot arm links/components is 3D-printed along with some of the actuator components. The conveyor frame and the station structure/enclosure are built out of aluminium extrusion and white melamine chipboard. A cylindrical part is designed and 3d-printed alongside two assembly parts, which is used to demonstrate the peg-in-hole assembly and disassembly capabilities of the robotic system.

The electrical design used to drive and control the motion of the robot actuators, is also included in this report. The electrical circuit also drives the linear conveyor until the part is in the required detection region of the vision system. The machine vision sequence will then commence. The sequence programs are illustrated in the report and are responsible for identifying and tracking the part on the conveyor, as well as identifying and tracking the position of the station reference point. The station will be semi-automated, as an operator is still required perform basic tasks when interacting with the system.

Keywords – **Robotics, Mechanics, Electronics, Machine Vision, Machine learning**

I. INTRODUCTION

The aim for the “Development of a collaboration robotic cell with a conveyor for vision-based sorting and assembly operations” project, is to design and build a station that can demonstrate the assembly/disassembly of parts. The assembly/disassembly must be done using reprogrammable robotic arm, with at least three degrees of freedom (3-DOF). The arm should operate with point-to-point control capabilities and should also have an appropriate end effector. The part must be conveyed to the robotic arm, using a controllable linear drive/conveyor. Using PC based machine vision and machine learning the station must identify and track the parts in the assembly/disassembly process. This process will require a controller and a human machine interface (HMI), for the operator to intelligently control the process of the assembly.

The purpose of this station will be to demonstrate peg-in-hole assembly of a light component (approximately 100g). The robotic arm must have a positional accuracy of ± 2mm and a repeatability of ± 1mm.

In this project the student combines three major engineering fields. Mechanical, electrical and Information Technology (IT) subsystems need to be developed and integrated, to achieve the end goal. Each one of these subsystems have their own specific requirements and objectives:

Mechanical:

- Design and build robotic arm with at least 3-DOF.
- Design and build a linear conveyor.
- Design and build some sort of station or platform, where all the mechanical parts are installed and integrated.

Electrical:

- Design electrical safety system.
- Design and build a power supply for the system.
- Choose electrical components to control the process (like microcontrollers and motor drivers)
- Choose a camera for vision-based operations.
- Design lighting system for vision-based operations.
- Choose electrical components for HMI.

IT:

- Develop HMI.
- Develop programs to detect and track the object in three-dimensional space, using machine vision and machine learning.
- Develop an integrated application for all the programs of the system.

II.BACKGROUND RESEARCH AND LITERATURE REVIEW

A.*Machine vision*

In the last few years machine vision has become incredibly intuitive and has been integrated into multiple industries [1]. It has been successfully used in automation and various robotic systems to accomplish tasks reliably and accurately, across engineering platforms. Machine vision does not have one clear definition but can basically be defined as the development of autonomous systems to process static images or video feeds to improve required system performance [2]. Machine vision systems can be used as an affordable and very sophisticated method, for robot guidance and can also serve as a source of feedback to control subsequent robotic tasks [3]. Machine Vision systems exist in large variety, but the main components of a machine vision system include lighting, lenses, image sensors, image processing and communications [2].

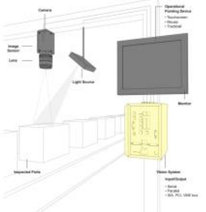


Figure 1: Machine vision system layout example.

Important machine visions aspects/considerations are further discussed and outlined in Appendix A.

B. *Traditional image processing methods*

Background subtraction algorithms, as described in [4], are used to distinguish moving objects (called foreground) from static, or slowly changing, parts (called background). All the pixels of a current frame are compared with a static background frame, to detect zones with significant changes, these zones are then classified as the foreground and would represent a moving object that has entered the frame. The comparison process is called foreground detection and it divides the image into the foreground and the then subsequently subtracted background. This program is called a Static background model, but for this model it is difficult to define everything that should be included in the foreground, which makes it ineffective for most practical situations.

There are various methods used to make these models more sophisticated and functional for practical applications. The simplest techniques include using a static background frame, weighted running average, a first-order low-pass filter, temporal median filtering and then modelling each pixel with a Gaussian [4].

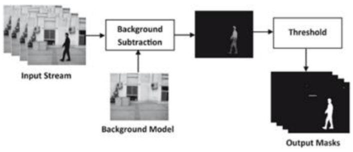


Figure 2:Basic illustration of background subtraction

[5] makes use of a methodology in which they go even further and use a Kalman filter to compensate for any Gaussian noise. [6] also added a frame-level component the pixel-level stage, this is done to detect any rapid changes in the global image and then to change the background frame as needed. In [4] they present a universal method for background subtraction, called ViBe. ViBe was tested against standard improvement techniques and it itself is a combination of multiple innovative techniques, which makes it very capable for practical use.

C. *Color filtering for detection.*

Color segmentation or color filtering is widely applicable and often used for identifying specific objects of a specific color [7]. Digital images are represented in color a format. The images are constructed as a three-dimensional array of pixels. Each layer of the array represents a color channel. The most widely used color space is RGB color space, called an additive color space because the three color shades add up to give color to the image.

To identify an object/region of a specific color, we create masks to separate the different colors. HSV (Hue Saturation Value) color space is more useful for this purpose because the colors in HSV space are much more localized and can thus be easily separated [8] & [9]. After we specify the range of color to be segmented, it is needed to create a mask accordingly and by using it, a particular region of interest can be separated out.

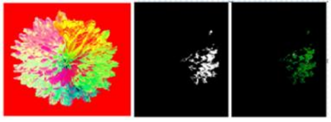


Figure 3: Masking and colour filtering example.

D. *Machine Learning*

Machine Learning (ML) is a category of artificial intelligence (AI), which refers to a process for computers to develop pattern recognition. This enables them to continuously learn from and make predictions based on input data. The computer will then be able to adjust without being specifically programmed to do so [10].

Machine learning is a category of algorithms that can take a data as input and use it to identify patterns and make predictions. Deep learning is a sub-set of machine learning that uses the core functionality of machine learning but moves beyond its capabilities [10].

Machine learning requires some human involvement in that the programmer can review the results returned by the algorithm and make the desired adjustments. Deep learning does not need any form of review from the programmer. Thew deep learning algorithm makes use of its own neural network to check the accuracy of its results and then self-adjusts its learning [10].

Deep learning algorithms have neural networks structures which are layered to replicate the structure of the human brain. Thus, the neural network can learn how to improve over time without the need for programmer feedback [10].

Machine Learning methods, using convolutional neural networks (CNN), usually consist of two stages: feature extraction and pattern classification. Through the analysis of the input image, a feature vector describing the attribute information is designed. This vector is then implemented into a classifier model that is then trained to identify specifics contained in the image [11].

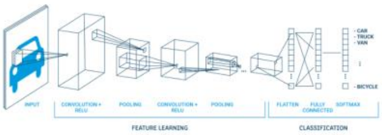


Figure 4: CNN general architecture.

E. *Online resources for machine learning*

Open-source software packages, which provide machine vision and machine learning resources, are freely available to people on the internet. OpenCV (Open-Source Computer Vision Library) is a free open-source Computer Vision (CV) and machine learning software library. There are various algorithms, from traditional CV algorithms to state-of-the art CV and ML, that users can access [12].

TensorFlow is also an opensource platform [13]. TensorFlow offers a lot of community resources for ML, like different pre-trained models and official datasets.

Both OpenCV and TensorFlow, can be integrated into a high-level application programming interface (API). This can be done using most high-level programming languages, especially Python programming language. This is best illustrated at PyTorch, another open-source platform, that also provides resources for ML and the instructions on how to use them alongside Python programming language. PyTorch also provides a description and performance results of their pre-trained models [14].

F. *Object detection and tracking, using Machine Learning models*

[15] looks at methods to tracking multiple objects in a given frame, through the use of CNN. They use a three-stage approach to tracking these multiple objects, which is detecting, identifying and then tracking the objects in particular zones. In this report they make use of YOLOv3 and YOLOv4 detector models to detect and classify the objects. Then the MOT17 dataset was used to evaluate the model performance. To enhance this performance, they made use of a Kalman Filter and SORT (Simple Online Realtime Tracking) algorithm, this was upgraded to DeepSORT. They ultimately produced an MOT-A (Multiple Object Tracking Accuracy) score of 60% accuracy.

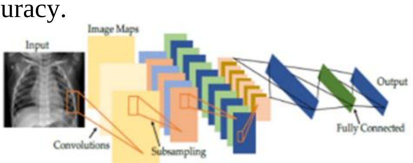


Figure 5: CNN architecture for image processing.

[16] also looks at real time object tracking, and Region Based Convolutional Neural Networks (RCNN). Apart from the previously mentioned YOLO models, the Single Shot Detector (SSD) is also under investigation for its applicability for real time object tracking. Through deep learning, they combine SSD and Mobile Nets to implement detection and tracking in a vision system. The SSD in [16] is based on the VGG-16 architecture. From [17], we can see that VGG-16 is unique in that it has convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2, instead of having a large number of hyper-parameters.

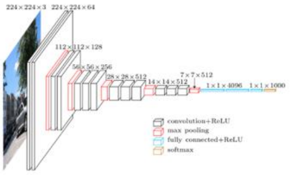


Figure 6: SSD model architecture for image processing.

Shape, color, distinguishable key points are all examples of features that can be tracked by a low-level model. For these models one has to balance performance with computational power requirements. This basically means that a better performing model usually requires more computational power. It is important to look for the most practical and easily distinguishable features of an object and then train the model based on those features. [18] looks at a way to do complex vision-based tasks, but with a method that has lower computational costs than training on conventional features of an object. The solution comes in the form of markers or tags. X-markers and Aruco tags are applied to various fields of optical tracking, because the simplify the identification process. These tags/markers are easily identifiable even without color and thus the model does not need to be as powerful to track them, thus producing high performance accuracy without the need for high computational power [18].

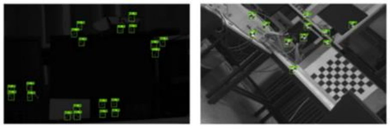


Figure 7: Example of marker detection in bad lighting conditions.

G.*Robotics*

A robot is a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks [19]. Robots have been integrated with almost all modern-day fields, especially in production and industrial sectors/fields [20]. Robots generally refer to the following automated assembled set of components: manipulator, actuators, sensors, controllers, processors, end effector, software and safety [21].

There are certain important characteristics of a robots, like its envelope, accuracy, repeatability and speed. The envelope of a robot refers to the three-dimensional shape which describes the reachable boundaries of the robot, as shown in Appendix B Figure 41. The maximum envelope includes the end effector and workpiece/attachments. This is the specific envelope described when referring to the envelope of a robot. The robot accuracy is a measurement of how close the robot can get to a desired setpoint. Repeatability of a robot refers to the ability to recurrently return to a specified setpoint. Therefore, repeatability is a measure of the variance when repeatedly moving to a setpoint. Maximum speed is the theoretical value, that a robot can reach, under no-load conditions. One would also need to know what the maximum payload is for the robot to still achieve these mentioned characteristics [22].

[23] describes a project that has very similar robotic requirements and is used in the design process for multiple parts of this project. This project requires the student to keep the cost to a minimum while still achieving the required performance. Therefore, it was decided to also review various robots designed and built by hobbyists, because they would usually keep the construction cost to a minimum. The hobby robot with the best performance, was built by [24]. This is a SCARA robot, which was also designed to perform Peg-in-hole assembly and laser engraving.

Important technical aspects/components of robotics are outlined and discussed in Appendix B.

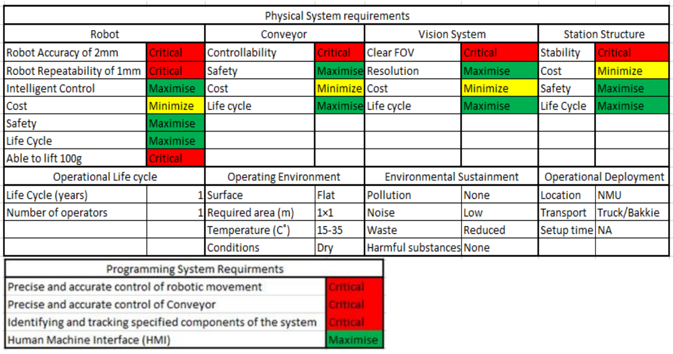
III. SYSTEM REQUIREMENTS

The robotic system is required to convey, detect, and maneuver the designed parts in a controlled and safe operational sequence. The system should utilize a linear conveyor, a robotic arm with at least 3-DOF and a vision system. For practical and safety reasons a station structure will also be designed to house all the different systems. The project is to be environmentally sustainable and cost effective.

Table 1, as shown below, summarizes the top-level system requirements and technical performance measures (TPM's). Some of the considerations in Table 1 do not have numerical specifications but are still required to be either minimized or maximized in the design of the system. There are some considerations that are marked as critical, and these are the main requirements as specified in the problem statement.

Practical and environmental aspects of the project have also been considered and summarized in Table 1.

Table 1: Robotic system requirements



Physical System requirements							
Robot		Conveyor		Vision System		Station Structure	
Robot Accuracy of 2mm	Critical	Controllability	Critical	Clear FOV	Critical	Stability	Critical
Robot Repeatability of 1mm	Critical	Safety	Maximise	Resolution	Maximise	Cost	Minimize
Intelligent Control	Maximise	Cost	Minimize	Cost	Minimize	Safety	Maximise
Cost	Minimize	Life cycle	Maximise	Life cycle	Maximise	Life Cycle	Maximise
Safety	Maximise						
Life Cycle	Maximise						
Able to lift 100g	Critical						
Operational Life cycle		Operating Environment		Environmental Sustainment		Operational Deployment	
Life Cycle (years)	1	Surface	Flat	Pollution	None	Location	NMU
Number of operators	1	Required area (m)	1×1	Noise	Low	Transport	Truck/Bakkie
		Temperature (C˚)	15-35	Waste	Reduced	Setup time	NA
		Conditions	Dry	Harmful substances	None		

Programming System Requirments	
Precise and accurate control of robotic movement	Critical
Precise and accurate control of Conveyor	Critical
Identifying and tracking specified components of the system	Critical
Human Machine Interface (HMI)	Maximise

IV. PROBLEM ANALYSIS AND DESIGN DESCRIPTION

a. *Mechanical*

The mechanical design section covers the following major design sub-sections. These include the conveyor, robotic manipulators and robotic actuators, robotic gripper and lastly the station structure. These systems will be isolated and independently analyzed and subsequently designed to meet the robotic system requirements.

i. *Conveyor*

A simple linear conveyor was design using Inventor's Fusion 360 software, as shown in Figure 8 below. To ensure that the vision system capabilities can be showcased, the conveyor needed to have a relatively large surface area (in comparison to the designed parts that are conveyed). This will enable the student to showcase how the part is detected and subsequently pick-up from a large variety of positions, due to the increased range of where the part can be positioned on the conveyor. The design is straight forward and is made up of the following main components:

- Conveyor belt (White Faux Leather Vinyl Upholstery).
- Two basic conveyor roller assemblies (using radial ball bearings).
- PG30 30X120 Aluminium Extrusion (0.5m)
- 3d-printed stands.

The conveyor is driven by a Nema 17(1.7A, 4kg-cm) stepper motor. The conveyor belt is made from white vinyl upholstery, because the part is so light in weight this conveyor does not need to have a very strong conveyor belt and the vinyl is inexpensive in comparison to almost all conveyor belt materials (see Bill of Materials in Appendix F). The color of the vinyl is white, to assist with the part detection task of the vision system. Calculations related to the design of the conveyor can be seen in Appendix C

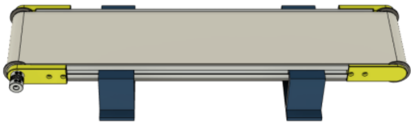


Figure 8: Designed linear conveyor.

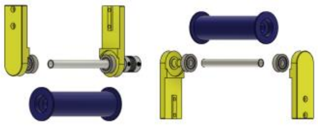


Figure 9: Exploded view of both conveyor roller assemblies.

ii. *Robotic Manipulators and Actuators*

Following the research of the literature review, it was determined that the best robot manipulator configurations for the required task would be either a Cylindrical or SCARA (as they are most suited for automated assembly tasks).

The robot base and linear lifting manipulator is derived from [24] and [23], respectively. The base uses pulleys, GT2 timing belts and a stepper motor to provide rotation capabilities to the robot.

The first pair of pulleys has a ratio of 1:4 and the second pair of pulleys has a ratio of 1:5. The Nema17 stepper motor has a step angle of 1.8 degrees. This means that the theoretical resolution of the base rotation is 0.09 degrees thanks to the combination of the four pulleys (as shown in Appendix C).

The base from [24], had a few issues with stability and was able to tilt when heavier loads are applied, which is not ideal when high accuracy and repeatability is required. The based was modified to resist this tilting effect by designing a smaller lifting tower and by using stacked thrust bearings inside the rotation connector (between the base and the prismatic linear lifting joint). This improves the weight distribution along the Z-axis of the robot.

The robot base uses a mounted limit switch to establish a zero position for the homing sequence, in which the robot will rotate in one direction until the switch is triggered and then moves a certain number of steps to the home position.

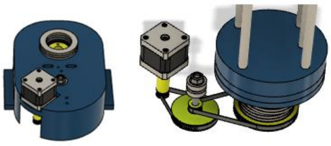


Figure 10: Designed robotic base.

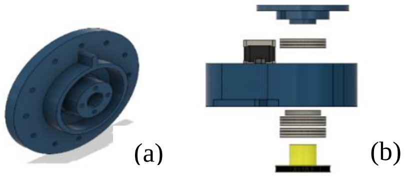


Figure 11:(a) Joint 1 connector plate. (b) Exploded view of joint 1 bearing assembly.

As shown in figures 10 and 11 above the base makes use of both thrust and radial bearings. This enables the base joint (the connection between the last pulley in the base and the connector plate) to be resist both the axial and radial loads, while ensuring smooth and accurate rotation. Large (60mm outside diameter and 40mm inside diameter) thrust ball bearings are used, not because any serious loads are expected to be present in the design, but instead because this increases the contact area and widens the first joint, which will decrease any stress applied to the 3d-printed base or pulleys. This will also increase the overall weight distribution and subsequently increase the robot stability.

The vertical lifting capabilities of the robot is achieved by a prismatic joint, which is made up of a leadscrew, and is driven by a stepper motor and supported by four linear guide rods. This design is very popular for linear actuation and is widely applied for various robotic projects. A limit switch will be used to stop the lifting process when the top limit is reached and will also be used to establish a home position for the Z-axis of the robot.

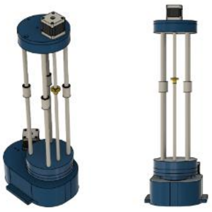


Figure 12: Linear actuator design for movement in the z-axis

From this point three different conceptual designs were made

iii. First conceptual design

The first conceptual design is a cylindrical robot (RPP). This design adds another prismatic actuator to the current base and lifting actuator. The second linear actuator will function the same way as the lifting actuator, only now in the vertical direction.

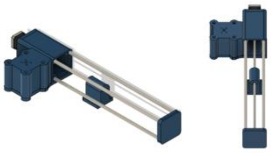


Figure 13: First conceptual actuator design.

The main advantages of using this actuator, is that it is a very sturdy structure and uses basic mechanical components. All the mechanical components should have a very long-life cycle when used only for the lightweight operation of the robotic station. This design would also result in simple robot kinematic control when it comes to the robotic control programming.

However, this design is quite expensive because of the cost of the leadscrew and linear guide rods. The step angle could be lowered using either some specifically designed gearbox or through the limiting capabilities (using half-step connections of the motor drivers). Both these solutions also have their negative aspects. The gearbox adds more complexity to the design and increases the already expensive design. Lowering the step angle through the motor driver, also lowers the current supplied to the motor. The half-step mode means that the motor produces less torque, but half-step mode does provide smoother motion.

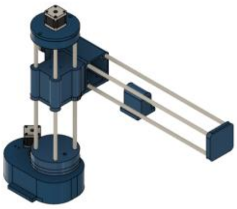


Figure 14: Cylindrical robot conceptual design.

iv. Second conceptual design:

The second conceptual design is a SCARA robot (RPRR) design derived from [24]. The same base is still used but the leadscrew used for the lifting actuator is moved forward to make space for a stepper motor and pulley on the lift component. This design has two rotational joints added that are both driven by their own stepper motor and pulley systems.

This is easily modifiable, seeing that the student could 3d-print different pulley combinations and conduct tests to determine which combinations have the best balance between accuracy, repeatability, and speed (to have the robot perform with high time efficiency). To ensure this is possible, the arms links are wide enough to use at least a 1:5 GT2 pulley ratio. Like [24], the student's design will also use limit switches to limit joint movements and to establish home positions.

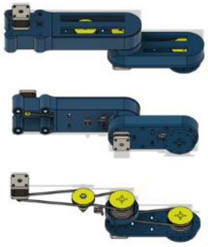


Figure 15: Second conceptual actuator design.

The top covers of the actuators use clips to neatly manage and guide the cables to all the electrical components. The pulleys at the joint centers also have hole through them for the same reason. The bottom covers have slats for tensioner pulleys, to properly tension the timing belts, thus ensuring that no slipping takes place during the actuator motion. Slipping and backlash in these timing belts would decrease the accuracy and repeatability of the actuator. The top covers also have large transparent perspex strips imbedded. This will help with identifying any potential causes of failures within the actuator without having to open the whole casing.

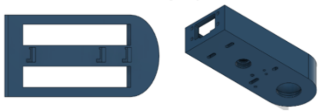


Figure 16: First link cover and casing.

This conceptual design has quite a few advantages over the first conceptual design. The first being that this design will be cheaper to produce, because the body and all the pulleys will be 3d-printed. The design is easily adaptable can be subject to change until the optimal performance is reached, by using tensioner pulleys or changing the actuator pulley ratios. The design is also much more intuitive and maneuverable than the first conceptual design, meaning that the robot can be effectively implemented into more specific/difficult environments and can perform more specific/difficult tasks.

Some of the downsides of this design is that the design is much less sturdy and has potential for backlash, which can affect both the accuracy and repeatability of the robot. The 3d-printed parts will also have a much shorter life cycle than the mechanical parts of the first conceptual design and components like pulleys and drive belts can start to ware and would need to be replaced. This design will also take much longer to manufacture, because of the increase of 3d-printed parts.

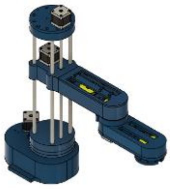


Figure 17: Second conceptual design (SCARA)

v. Third conceptual design:

The third conceptual design is a SCARA robot (RPR) design, this design is and adaptation of the second design, which more closely match this project's requirements. The project requires an at least 3 DOF robotic arm that can perform peg-in-hole assembly with a part of the students choosing. Therefore, the student decided to simply use cylindrical parts for the peg-in-hole operation. This means that the robot would not have to align the TCP at a certain angle to pick up the part and thus the last rotational joint is redundant.

The student then designed a much lighter, but stronger second link to replace the one used in the second conceptual design.

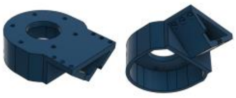


Figure 18: Improved second link for third conceptual design.

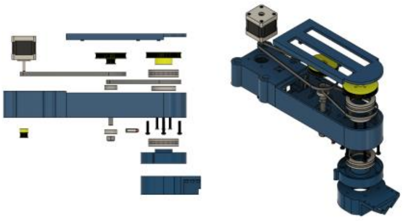


Figure 19: Exploded view of third conceptual design's arm.

This adaptation vastly improves multiple of the robot's attributes. The robot is now lighter and the overall loads on the joints and their connectors have been decreased. The overall cost of the project is decreased, because it requires less filament to print and electronic components to operate (removed a stepper motor and limit switch from the design and not to mention the wires required). The time to manufacture and assemble is also subsequently reduced.

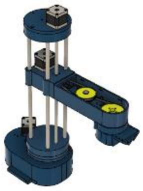


Figure 20: Third conceptual design (SCARA).

vi. Comparison and ranking design criteria

Table 2 represents a weighted decision matrix for selecting the best conceptual design. The criteria are chosen based on the student's estimation of the designs (some of these estimations are based on current circumstances of 2022 like material cost and loadshedding). The criteria are assigned a weighting, based on their relevance to the required operation, and then a rating range of 1-5 is assigned to each conceptual design.

Table 2: Weighted decision matrix for selecting a conceptual design.

		First conceptual design		Second conceptual design		Third conceptual design	
Criteria	Weighting (%)	Rating	Score	Rating	Score	Rating	Score
Cost	25	2	0.5	3	0.75	5	1.25
Mass	25	2	0.5	3	0.75	5	1.25
Robustness	20	5	1	3	0.6	4	0.8
Size/Maneuverability	15	2	0.3	4	0.6	4	0.6
Manufacturability	15	5	0.75	4	0.6	4	0.6
Total			3.05		3.3		4.5

Based on the total scores of the decision matrix, the third conceptual design was chosen by the student for this project.

vii. Robot end effector (gripper):

The student investigated multiple gripper options, including the one showcased in [24]. The final gripper design is a simple parallel motion gripper, designed by the student. The gripper utilizes a servo motor and a rack-and-pinion gear system to produce controlled linear motion.

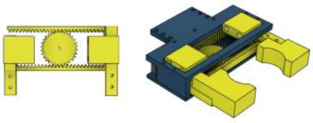


Figure 21: Designed parallel robotic gripper.

Multiple gripper clamps were designed and tested, but ultimately these were the final selection. The clamps have circular inside to match the radius of the part, this is done to ensure that the part is always gripped in the center of the clamps (to improve the accuracy and repeatability of the robot). The clamps are also made wide enough to accommodate any number of visual tracking methods that the student might chose to use for this project. Thus, the final robotic design can be seen in Figure 22 (see Appendix C or related calculations):

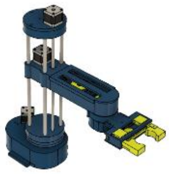


Figure 22: Final conceptual robotic design.

viii. Robotic final outcome:

One thing the student learned about 3d-printing, is that when the PLA+ strands are laid on top of each other in layers, the part expands slightly. This had a large effect on the tolerance of multiple parts, which had to be re-printed with small adjustments made to compensate for these small expansions. PLA+ is a very strong, however it still allows for small amounts of elastic deformation (flex and play), which could affect the overall accuracy and repeatability of the robot. However, the student is very happy with how the robot turned out and feels that it will meet the project requirements.

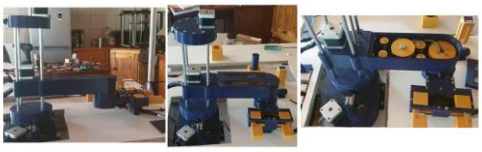


Figure 23:Assembled SCARA robot.

As shown in Figure 23, during the assembly of the SCARA-robot it became apparent that the GT2 timing belts needed more tension. Therefore, idler pulleys as shown in Figure 24, were used to increase the tension in the belts. This improvement was needed, because if there is not a sufficient amount of tension on the timing belts slipping can occur, which would greatly affect the robot performance.

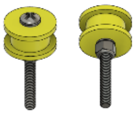


Figure 24: Designed tensioner pulleys.

ix. *Finite Element Analysis (FEA)*

As stated in the project requirements, the robot should be able to handle the load of a 100g part. The weights of the components and the added weight of the100g part was used to analyze the static loads which all joints and connections are subjected to.

The connection between the lifting base and the first link of the arm was identified as the most likely to fail. This connection supports the weight of the entire arm (including the part) and is made entirely out of 3d-printed material bolted together. The total weight, which this connection needs to support is 1.135kg, with the majority of the weight being distributed around the joint at the very end of the first link. This leads to a large torque force being exerted on the connection even if the weight the link body is taken as distributed. To be certain of the connection's strength and ability to handle this load, the student distributed the entire weight of the arm around the joint at the end of first link (thus working with a theoretical maximum torque load of 1.85 Nm).

The Inventor Fusion 360 software produced the following results for the finite element analyses:

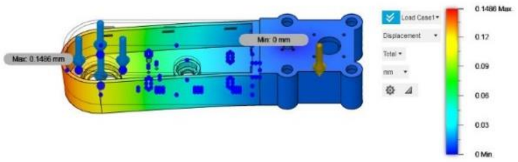


Figure 25: Displacement results

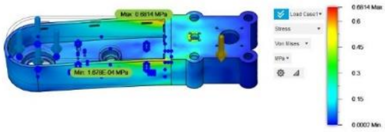


Figure 26: Von Mises stress results

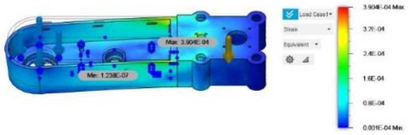


Figure 27: Strain results

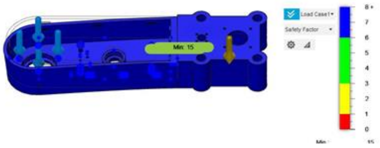


Figure 28: Safety factor

The FEA analysis yielded a maximum Von Mises stress of 0.6814 MPa (Figure 26) and maximum stain of 0.00039 (Figure 27). With PLA+ having a yield strength of 26 MPa, from Fusion material properties, the connection has a safety factor of 15 (Figure 28), which serves as indication that the component might be designed with too much material. However, since this is a 3d-printed component, some defects such as over or under extrusion or non-adhesive layers can occur during the printing process and this could drastically lower the safety factor. Thus, the component was be printed as designed. The maximum displacement of 0.1486mm (Figure 25) is negligible.

The student is not technically required to investigate wear and lifespan components, as there were no specifications outlined on how long the system would have to remain operational. That being said, there are a few mentionable details regarding the operational lifespan of the robot.

All the joints operate smoothly and are very much resistant to the loads exerted by the robot operation, due to the combination of radial and thrust bearings used for each joint. Thus, it is the student's assumption that it would be one of the various load-bearing 3d-printed parts that would wear out first. This is because, even though PLA+ is a very strong filament, its lifespan is very dependent on the environment and circumstances of its use.

x. *Station structure design*

The student designed a strong structure to combine all the different systems. The station serves as a support structure to simplify the transport and setup of the robotic system. The station structure also encases all the electrical parts for safety and an extra layer of ingress protection.

Perhaps the most important function of the station is to serve as uniform (unchanging) environment, with constant lighting conditions. This will be very important for the functionality of the machine vision program in the operational sequence of the system. To realize the machine vision aspects of the project, the station has a single supported beam (aluminium strut) that will elevate the camera to the required height. The strut will also have an RGB light strip imbedded into it, to provide proper lighting conditions.

The station is primarily built out of white chipboard, aluminium struts/extrusion and 3d-printed connectors. See CAD drawings in Appendix G.

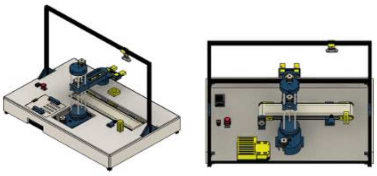


Figure 29: Robotic station design.

The following safety signage will be displayed on the station surface:

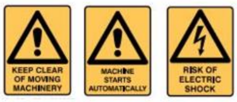


Figure 30: Safety signs to be displayed on the robotic station.

These signs are included for operator safety and represent the main things the operator should be mindful of when using the robotic station.

xi. *Station final outcome*

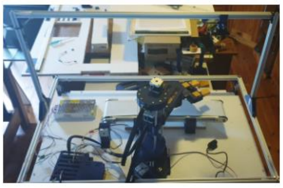


Figure 31: Open Station during assembly process.

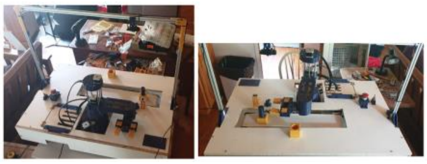


Figure 32: Assembled robotic Station.

b. Electronics

Looking at [23], the author outlined their electrical design with Figure 33 below. Their robot is controlled by an Arduino Mega microcontroller, and they have added a Ramps 1.4 3d-printer shield. The shield comes with a set layout with placement positions for five motor drivers. This really simplifies the electrical design process and is often used by hobbyists, when building a 3D-printer. The problem with these shields is, that they have G-code specifically for 3D-printing operations, so in order to use them for anything else one would have to make use of firmware. Marlin firmware combined with PlatformIO can be used to trick the shield into performing as needed. However, this seems like it could also have various compatibility limitations.

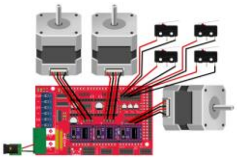


Figure 33: Electric circuit design utilizing the Ramps 1.4 3d-printer shield.

Whilst the circuit from [23] remains an option, the student decided to design a circuit without the Ramps 1.4 shield. This increases the electrical design complexity but does not come with any possible compatibility or integration issues for the IT design. The Ramps 1.4 is not a very expensive electrical component, so the overall electrical design cost, for the student’s design and the [23] design (with the necessary added components), will be very similar

Figure 35 represents the theoretical layout of the student’s designed electrical system and Figure 39 represents the actual schematic circuit diagram (designed using Fritzing software). The student’s design is controlled using an Arduino Mega (as slave microcontroller) and a PC (as Master).

All the Nema17 stepper motors are driven by A4988 motor drivers. The step pin on the driver is used to send a pulse to the stepper motor, which swaps the polarity of the coils. These pulses correspond to the motor steps. The direction pin is used to control the motors direction of rotation. The MS1, MS2 and MS3 pins can be used to adjust the step sizes, but is not used in this project. The A4988 drivers also have potentiometers for adjusting the reference voltage of the motor (specification sheet included in Appendix F).

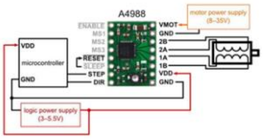


Figure 34: Wiring diagram for A4988 stepper motor driver [25]

The gripper uses a Mg996R 180-degree metal gear servo motor. Because of its built-in control feedback system, servo motors do not need drivers to operate. The Servo motor uses three wires, the live wire, ground wire and a control signal wire. To identify the angular position of the motor, the control signal wire uses a PWM signal. The PWM concept suggests that the motor shaft's angular rotation is regulated by the PWM signal's pulse duration.

The proximity sensor is an E18-D80NK infrared sensor and is an optional addition to the project because the conveyor is controllable. Thus, if a single part is placed on the exact same spot, the conveyor could take a set number of steps to get the part to the required position. However, since we might not know how many parts may come in succession and what their spacing situation is, it is better design practice to include the proximity sensor to improve possible functionality of the station as a whole.

The Arduino Mega is only able to supply 5v or 3.3v. Therefore, a power supply is required for the motors that require a higher input voltage. The system is designed with an 12v-10A power supply, which has to pass through an E-stop (as safety mechanism) and an On/Off switch (for functionality). The station also has a LED light strip imbedded into the aluminium frame to provide ideal circumstances for the vision-based tasks.

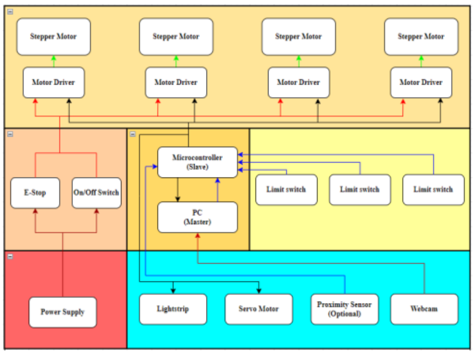


Figure 35: Designed electrical system layout.

To make sure that all the components are functional and that they are able to operate as required, the student created a temporary component housing and connected the whole system to a breadboard for testing (see Figure 36).

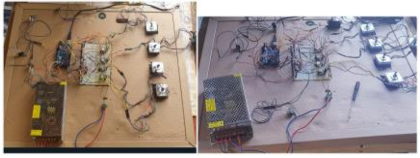


Figure 36: Temporary electrical testing setup.

Once all the components were tested, the Veroboard was designed (using Fritzing) and soldered together.

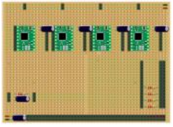


Figure 37: Designed Veroboard on Fritzing.

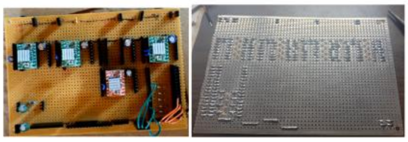


Figure 38: Final soldered Veroboard.

The student decided to alter the board slightly, by moving one of the placement points for the motor driver down. This was done to alleviate some of the electrical interference present at the time. Other methods were also used to remove electrical interference namely: Separating all the wires into twisted pairs; Used shorter wires for all connections (to minimize antenna effect) Only used multi-strand copper core wires (no solid-core wires); Made sure the main power wires were isolated and place as far away from the Veroboard as possible; Used wire sleeves for all the long wires, which connect to the robot and conveyor. Ideally all the wires would be shielded, but small properly shielded wires are hard to come by and more expensive than the Dupont jumper wires.

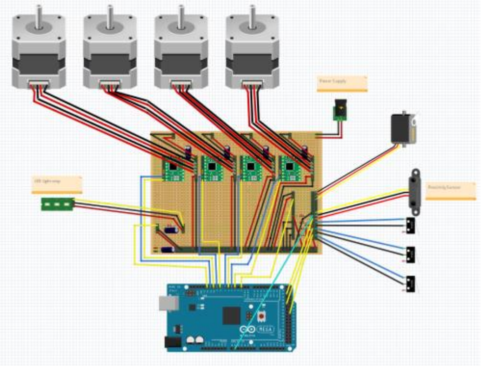


Figure 39: Designed electrical circuit.

To tidy up the electronics and to add an extra level of protection, the student designed an electronics case (see Figure 40). This case was designed to house both the Veroboard and the Arduino Mega and has gaps to allow all the externally connected wires to pass through unhindered. These gaps also allow for airflow to ensure that the motor drivers do not overheat (there is also mounting points for a fan if necessary).

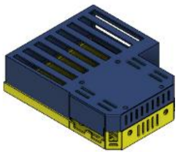


Figure 40: Designed electronics case.

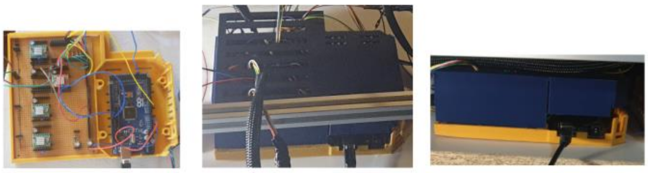


Figure 41: Assembly of electrical case.

c. Programming

i. Overall layout

Originally this project was intended to make use of a Raspberry Pi to function as the master/host controller, but due to the current scarcity and price increase, the project will function using the student's personal laptop. However, the student decided to still create an integrated software application that would be compatible with a Raspberry Pi4, so that the project could be easily updated in the future to utilize one.

The main components of the IT system are illustrated in Figure 42 below. The System host will be the student's laptop (PC) and would function as the master controller. All the integrated software programs on the PC are written in Python programming language, as it is also compatible with Raspberry Pi and is industry standard for vision-based applications. The slave controller or robot controller is the Arduino Mega and is programmed using Arduino IDE. The PC also takes the webcam as a direct input.

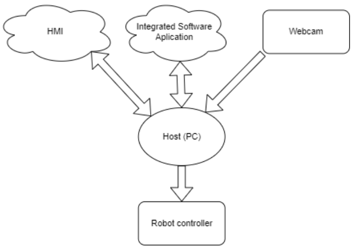


Figure 42: System software communication diagram.

i. Communication and operational sequence

The python application functions as the brain of the station and the Arduino program functions as the muscles. The python program receives all the information from the operating environment (the webcam feed and the user input) and the processes it and performs the necessary computations. The relevant information is then passed through serial communication to the Arduino program, which has all the functions that control the various electrical components.

The robotic station is designed so that the following Peg-in-hole assembly process could operate in both of the following sequences:

Manual sequence:

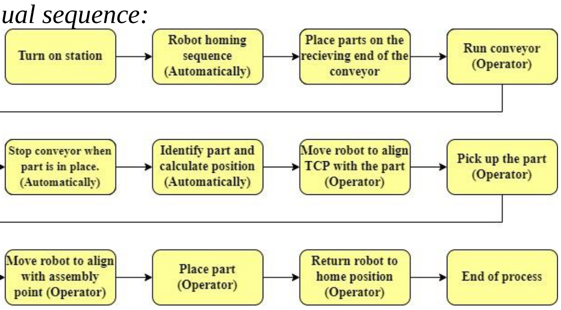

Turn on station
Robot homing sequence (Automatically)
Place parts on the recieving end of the conveyor
Run conveyor (Operator)
Stop conveyor when part is in place. (Automatically)
Identify part and calculate position (Automatically)
Move robot to align TCP with the part (Operator)
Pick up the part (Operator)
Move robot to align with assembly point (Operator)
Place part (Operator)
Return robot to home position (Operator)
End of process

Figure 43: Sequence for manual operation mode.

The manual sequence requires the operator to start each sub-operation by clicking their respective buttons on the on the HMI (hence the inclusion of the "Operator" in the sequence diagram). Even though this sequence of operation does meet the project requirements, it was primarily used to debug whilst programming the application and can also now be used to illustrate the station capabilities separately.

Automated sequence

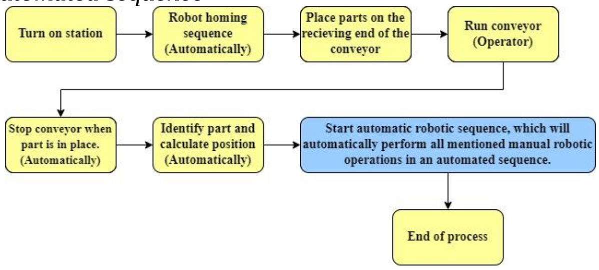

Turn on station
Robot homing sequence (Automatically)
Place parts on the recieving end of the conveyor
Run conveyor (Operator)
Stop conveyor when part is in place. (Automatically)
Identify part and calculate position (Automatically)
Start automatic robotic sequence, which will automatically perform all mentioned manual robotic operations in an automated sequence.
End of process

Figure 44: Sequence for auto operation mode.

The Automated sequence is ideally how the station would run when presenting. This sequence is a good example of a semi-automated system, where some operator input and monitoring are still required but in essence the system runs on its own. The station could easily be made even more automated by having the sequence start automatically when the part is detected in position, but for presenting purposes the student will want to show the part being tracked before the robot starts moving.

ii. Vision-based programming

There are multiple different sections/functions that come together to form the vision-based application of this project.

1. Part detection and tracking

Multiple different methods for object detection and tracking were investigated.

The first option was to train a convolutional neural network to identify and track the part. This is the method that would have the theoretical best performance but would also require the most computational power and was running with quite a decent amount of performance lag. To accomplish this method, the student would take around a hundred pictures of the part (in various different lighting conditions and orientations towards the webcam) and then use Gimp software to create a dataset of about ten thousand images. Then these images would be used to create training, validation and testing datasets and used to train the very popular YOLOv3 pre-trained model.

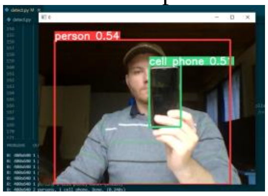


Figure 45: Student using YOLOv3 pre-trained model to track cell phone.

The second option was to use a traditional machine vision technique, which is color detection. For traditional methods to work properly the parts had to be designed with distinct colors or shapes (or both). Originally to showcase the color detection the student used two parts of similar shapes, but with very different colors (the one part will be taken out of the other to perform peg-in-hole disassembly).

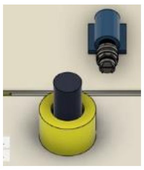
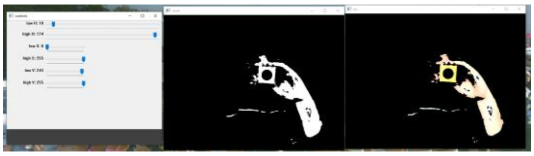


Figure 46: Designed parts for peg-in-hole disassembly

As a start, a program (HSV program) to separate and identify pixels based on their HSV values was required. This HSV based program was then tested on dummy 3d-printed parts, that represent the conveyed parts of the system. The program uses HSV sliders and two displayed screens. Once the color has been isolated, the HSV values can be saved and used in another program. The next program (Pixel detection) will test if these objects can be detected based on the provided HSV values, so detected based on their color.



low H: 18
high H: 174
low S: 0
high S: 255
low V: 243
high V: 255

Figure 47: Example of HSV mask program tuning.

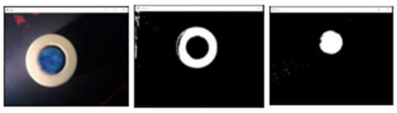


Figure 48: Generated masks for parts, as seen through webcam.

The masks could then be used to track the parts in real time and the center point of the cluster of similar pixels could be used to return the position of the parts.

The third method is also a traditional machine vision method and relies on the shape of the part. This method requires the use of the OpenCV HoughCircles function. The theory behind this function is that a circle is represented mathematically as:

$$(x - x_{centre})^2 + (y - y_{centre})^2 = r^2 \qquad (1)$$

where $(x_{centre}, \ y_{centre})$ is the center of the circle, and “r” is the radius of the circle. From this equation, we can see there are 3 parameters, so a 3D accumulator for hough transform is needed, which would be highly ineffective. However, OpenCV uses a trickier method called **Hough Gradient Method** which uses the gradient information of edges. This method focusses on tracking circles based on nine parameters, which the user has to adjust until satisfied with the performance. This method vastly outperformed the color detection approach, with high accuracy and almost no performance lag.

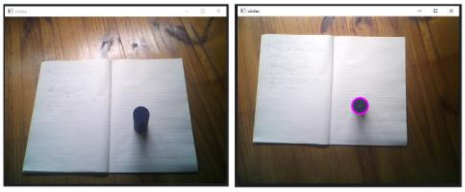


Figure 49: Tracking part using HoughCircles program.

Between all the options, the convolutional neural network would definitely be the most accurate and dependable, but since the student wants to make the project compatible with the Raspberry Pi, it was decided that the third option would be the best choice.

The student wrote a program which searches through a single webcam frame to identify possible circles that match the HoughCircles parameters. The program would then select the best one and draw its boundary and center point on the frame. The center point coordinates (in pixels) is then also returned and displayed in the console. This program would then do this for each incoming frame from the webcam and thus we are able to track the position of the most circular part in the camera frame, in real time. The program was then modified to only look for circles in a specified boundary region, so that the station does not accidentally pick up some other circular objects in the background.

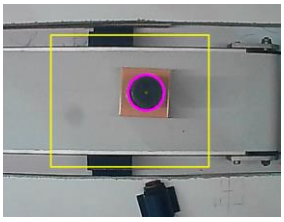


Figure 50: Only tracking part within boundary region.

2. Station reference point tracking

In order to relate the tracked position of the part on the conveyor to its real-world location, we need to relate its pixel position (in the camera frame) to a reference point on the station. This reference point refers to a key feature on the station for which we know the exact location in relation to the robot base frame. The pixel difference between the part on the conveyor and the reference point on the station is then measured and used the to calculate the real-world distance (in mm). The real-world location coordinates of the part are then sent to the robot.

To realize this process, the student makes use of Aruco tags and the built in OpenCv library for detecting and tracking Aruco tags (as mentioned in the literature review section). The main benefit of these markers is that a single marker provides enough correspondences (its four corners) to obtain the camera pose. Also, the inner binary codification makes them especially robust, allowing the possibility of applying error detection and correction techniques.

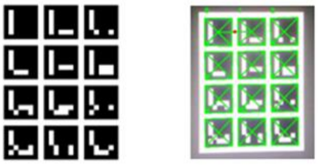


Figure 51: Normal Aruco tag register vs how the register is seen through webcam

The Aruco tag is printed and attached to a small 3d-printed stand (see figure 52), which is attached to the station at a location for which the relative position to the robot base frame is known.

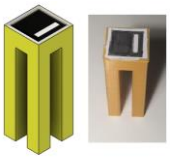


Figure 52: Station reference point.

The student wrote a program to constantly measure the distance between the center point of the tracked part and the center point of the Aruco tag. The pixel coordinates of both the center points are constantly displayed on the HMI as well as the distance between them. The Aruco tags are so effective, that even a hand drawn tag worked perfectly.

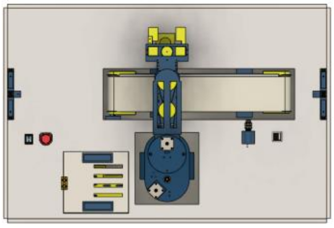


Figure 53: Top view of robotic station

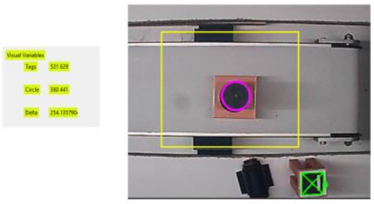


Figure 54: Parts and Reference point as seen on the HMI with all variables being displayed.

The Aruco tags can also be used to track the robot tool center point (TCP), as an extra improvement on the operational sequence. This will enable the robotic system to verify whether the gripper is actually aligned with the part and then make small adjustment if necessary. To realize this, two tags can be attached to the gripper and the student wrote a program to return the center point between two tags, thus continuously tracking the robot TCP. This improvement will only be included if the system does not perform a well in the accuracy testing.

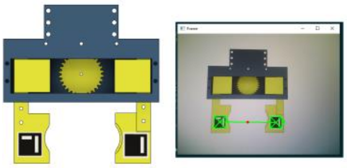


Figure 55: Optional tracking of TCP

The designed station, as shown in Figure 24, needed to be analyzed. This meant that the overall process of operation was broken up into sub operations, that needed to be developed and tested separately, before being combined to create the operation process.

iii. Human Machine Interface (HMI)

The HMI was designed and programmed using a standard python library, called Tkinter. Tkinter has several strengths. It's cross-platform, so the same code works on Windows, macOS, and Linux, thus compatible with Raspberry Pi. Visual elements are rendered using native operating system elements, so applications built with Tkinter looks like it belongs on the platform from where it runs. Although Tkinter is considered the de facto Python GUI framework, it's not without criticism. One notable criticism is that GUIs built with Tkinter look outdated. If you want a shiny, modern interface, then Tkinter may not be what you're looking for. However, Tkinter is lightweight and relatively painless to use compared to other frameworks. This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern sheen is unnecessary, and the top priority is to quickly build something that's functional and cross-platform.

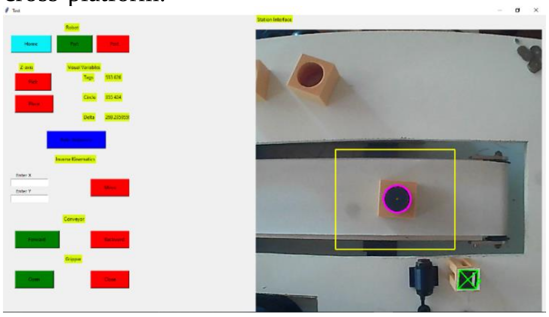


Figure 56: Designed HMI

The first and most notable feature of the HMI is the live display of the webcam feed, so the operator can monitor the performance of the vision programs and the accuracy of the robot movement.

There are four separate labels on the HMI. The first, is the robot label for the section containing the robot functions related to the required project operation, of picking up the part on the conveyor, then placing it in the assembly point and finally to move back to the home position. This section also includes the updating displays of the vision variables, the "Pick" and "Place" buttons used for the manual mode of operation and finally the "Auto sequence" button for the automatic sequence of operation. The inverse kinematics section is also included as an added feature, where the operator can input real-world coordinates (in mm) for the robot to move to. Then the conveyor section also has a “Forward” and “Backward” button, which is used to control the linear conveyor. Lastly, the gripper section has the “Open” and “Close” buttons, for manually controlling the robot gripper.

V. THEORETICAL BACKGROUND

A. *Kinematic modelling*

The robot is made up of multiple different coordinate frames. These frames are used for representing the positions and orientations of rigid objects representing the robot. Homogeneous transformations can be used to represent sequences of rigid motions (rotation and translation). Rigid motions can be represented in matrix form and their composition can be reduced to matrix multiplication. Two types of matrix multiplications are used for the compositions: pre-multiplication and post-multiplication. Pre-multiplication operations multiplies the current frame relative to the inertial or fixed frame. Post multiplication operations multiplies the current frame relative to the frame directly before it.

$R_{x,\theta}$ $R_{y,\theta} R_{z,\theta}$ Represent rotation about the X-axis, Y-axis and Z-axis respectively and can be represented by the following matrices:

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

B. *Designed Robotic station frames*

A Homogeneous Transform is represented by the following notation:

H_2^1, this represents the frame of 2 relative to the frame of 1.

The layout of the station’s coordinate frames is shown in Figure 34 below.

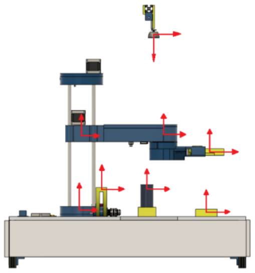


Figure 57:Station coordinate frames.

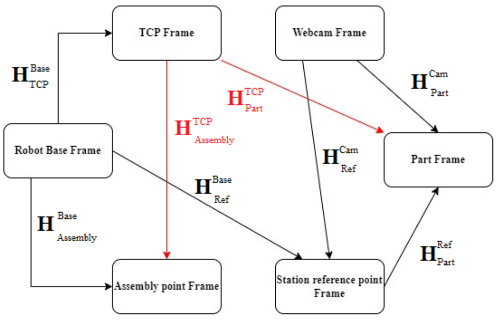


Figure 58: Transform graph for the robotic station.

The red links represents the transforms needed to perform the operational sequence.

Expression for the Part relative to the TCP:

$$H_{Part}^{TCP} = (H_{TCP}^{Base})^{-1} H_{Ref}^{Base} H_{Part}^{Ref}$$

$$H_{Part}^{Ref} = \left(H_{Ref}^{Cam}\right)^{-1} H_{Part}^{Cam}$$

Expression for the Assembly point relative to the TCP:

$$H_{Assembly}^{TCP} = (H_{Assembly}^{Base})^{-1} H_{TCP}^{Base}$$

C. *Forward and Inverse kinematics*

Forward kinematics describes the motion of the robot end effector using the joint variables. The parts and assembly stations are designed with set heights; therefore, the robot will always perform the same set of movements in the Z-axis (moving up or down). This means that the only movements that needs to be calculated is in the XY-plane, as seen from the camera frame.

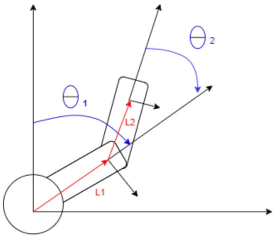


Figure 59: Diagram of manipulators XY-plane projection

The equations for calculating the Flange XY-position with regards to the robot base is as follows:

$$X = L_1 \times \sin\theta_1 + L_2 \times \sin(\theta_1 + \theta_2) \qquad (2)$$

$$Y = L_1 \times \cos(\theta_1) + L_2 \times \cos(\theta_1 + \theta_2) \qquad (3)$$

Inverse kinematics describes the joint variables of the robot through the use of the end effector’s position and orientation. The equations for calculating the joint angles from the Flange XY-position is as follows:

$$\theta_1 = \tan^{-1}\left(\frac{x}{y}\right) - \tan^{-1}\left(\frac{(L_2 \times \sin(\theta_2))}{(L_1 + L_2 \times \cos(\theta_2))}\right) \qquad (4)$$

$$\theta_2 = \cos^{-1}\left(\frac{(x^2 + y^2 - L_1^2 - L_2^2)}{2 \times L_1 \times L_2}\right) \qquad (5)$$

D. *Denavit-Hartenberg method*

Table 3: Denavit-Hartenberg for (RPR) SCARA robot

Denavit-Hartenberg Table				
Frame	ϴ˚	d (mm)	a (mm)	α˚
Base to 1	θ_1^*	d_1^*	0	0
1 to 2	0	0	a_2	0
2 to TCP	θ_3^*	$-d_3$	a_3	0

The H_1^{base}, H_2^1, H_3^2 *and* H_{Flange}^3 matrices can be constructed using the Denavit-Hartenberg table and the following:

$$A_i = Rot_{z,\theta_i}\, Trans_{z,d_i}\, Trans_{x,a_i}\, Rot_{x,\alpha_i} \qquad A_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This, alongside the H_2^{base}, H_3^{base} *and* H_{Flange}^{base} matrices, are derived and shown in the Appendix D. Using these derived matrices from the Appendix D, the Jacobian matrix can be derived as:

Note: $S_x = \sin\theta_x$ *and* $C_x = \cos\theta_x$

$$Jv,w = \begin{bmatrix} -s_1a_3c_3 - c_1a_3s_3 - s_1a_2 & 0 & -s_3c_1a_3 - c_3s_1a_3 \\ c_1a_3c_3 - s_1a_3s_3 + c_1a_2 & 0 & -s_3s_1a_3 + c_3c_1a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

E. *Trajectory planning*

Robotic motion planning is described using paths and trajectories. A path is defined as a sequence of points or poses, which the robot should achieve. A trajectory is defined as a trace, which is a function of time and subsequently also velocity and acceleration. Paths can be specified by an on-line and off-line method. The on-line method is the teaching of a path through a teach pendant or the walk-through method. The off-line method needs an exact map of the robotic cell and involves the programming of trajectory points defining a path.

Common interpolations for point-to-point motion control are joint, linear, and circular. Usually, a robot path consists of five specified points, as shown in Figure 37 below. Joint interpolation is used between points two and four, because this is where constant angular velocity is maintained for all the axes. Multi-segment trajectories (fly-by points) require blending of polynomials. Fly-by points achieve continuous movement by moving past the programmed position. This creates smooth transitions, without unnecessary speed reduction. These polynomials are typically 4-3-4 or 3-5-3 polynomials and have 14 boundary conditions.

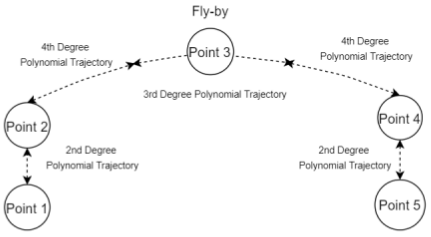


Figure 60: Multi segment robot path

For joint interpolation, the fastest moving axis relative to its maximum velocity is determined first then the velocities of the remaining joints are calculated. This ensures that all the axes reach the destination point at the same time. Polynomials are used to determine trajectories in joint interpolation. The number of constraints dictate the degree of the polynomial. The following matrix expression can be used to derive the equations of motion between two points:

$$\begin{bmatrix} q(t_0) \\ \dot{q}(t_0) \\ \ddot{q}(t_0) \\ q(t_f) \\ \dot{q}(t_f) \\ \ddot{q}(t_f) \end{bmatrix} = \begin{bmatrix} q_0 \\ v_0 \\ a_0 \\ q_f \\ v_f \\ a_f \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$$

Note: The subscripts 0 and f are the initial and final values and variables t, q, v and a are the times, positions, velocities, and accelerations, respectively.

Cartesian space trajectories (paths) are used for linear interpolation. The path is divided into small segments and time increments. The joint angles at each point of the segmented path are calculated using inverse kinematic equations to achieve interpolation.

VI. PERFORMANCE ANALYSIS

A. *Robot performance*

To evaluate the robot's suitability for the required tasks of the project, there are two performance factors to consider (as mentioned in the project requirements section). First is the accuracy, which is a measurement of how close the robot can get to any given target position. The second is the repeatability, which refers to how consistent the robot's movements are when moving to a given position. Figure 61 summarizes these two factors.

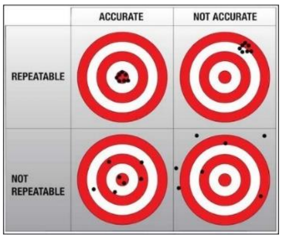


Figure 61: Accuracy and repeatability diagram.

To do this testing, the student printed multiple A3 sheets of millimeter graph paper. Then the graph paper was attached to the robotic station and a coordinate was marked using a blue pen. This coordinate was a point at a measured position, in mm, from the robot base and had to be at a location where both rotational joints of the robot had to be used to reach (so not just on the positive X or Y axis). The coordinate chosen was (200,200), as both rotational joints have to move a decent amount to reach this location. A permanent marker was attached to the gripper of the robot, so that there is no variation in the marker position that would influence the readings.

The test was performed three times and for each test ten runs were recorded. A run consists of:

1. Homing the robot.
2. Then using the inverse kinematics section of the HMI to enter the (200,200) coordinates and clicking the move button.
3. Lowering the z-axis of the robot once it has aligned with the position, to mark the reached position on the graph paper.
4. Taking the measurements of the marked position relative to the target position.

The accuracy of a run is determined by the measured difference (in mm) between the target point and the point marked by the robot. The repeatability is obtained by selecting the marked point closest to the target and then calculating the average distance between it and all the other marks made during the test.

Test 1:

The first test exposed a real problem in the Veroboard circuit. A large amount of electrical interference was in circuitry was causing the stepper motors to vibrate or even move at random, this was obviously a major issue that had to be addressed before testing could continue. Figure 62 shows how the robot was not able to make precise marks on the graph paper, but instead vibrated and drew sporadic lines. The solutions to remove the interference from the circuit are listed in the electrical section of this report.

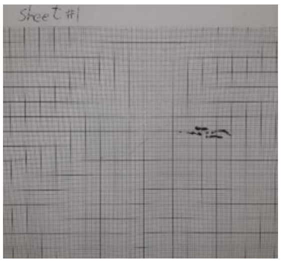


Figure 62: Test 1 marked sheet.

Test 2:
Once the electrical interference was removed from the circuit, the second test was conducted. For the second test the robot was not mounted to the station, which led to a few inconsistencies in the testing data gathered. However, the purpose of this test was simply to test for any other unforeseen electrical or mechanical issues. In this test the motor speeds were also still a too high. High motor speeds can be a problem, because the GT2 pulley belts used in this project are not backlash free, thus it is better to move slowly.

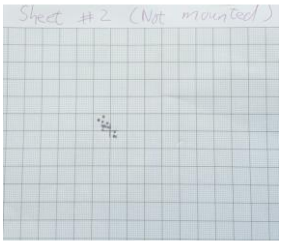


Figure 63: Test 2 marked sheet.

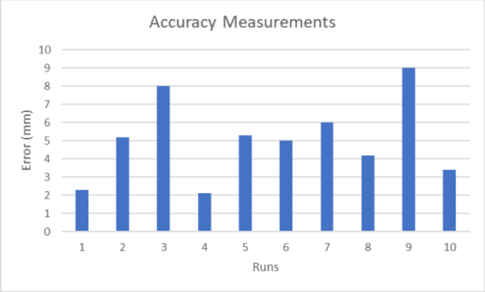


Figure 64: Test 2 Accuracy graph.

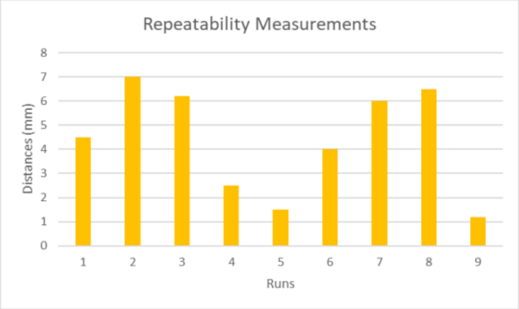


Figure 65: Test 2 Repeatability graph.

The measurement data tables are included in Appendix D. From the measurements, the calculated accuracy and repeatability, for test 2, is 5.05mm and 4.378mm respectively.

Test 3:
In test 3 the robot is fully mounted to the robotic station. The stepper motor speed and acceleration are set to appropriate values as to not cause to much backlash through the pulleys.

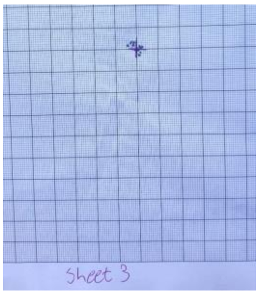


Figure 66: Test 3 marked sheet.

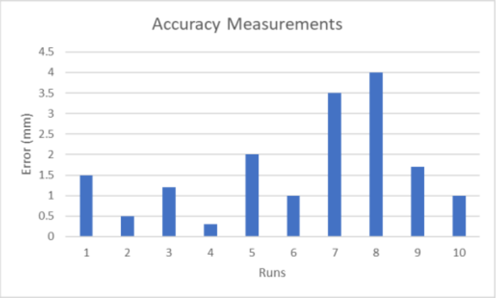


Figure 67: Test 3 Accuracy graph

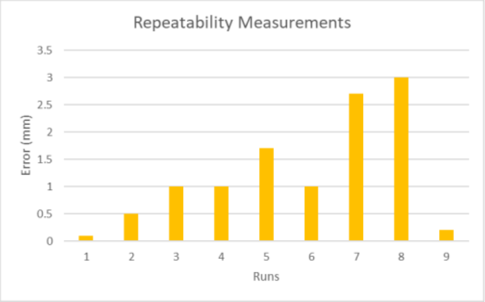


Figure 68: Test 3 Repeatability graph.

From Figure 67, there are still two runs where the accuracy was not too good, but the overall accuracy and repeatability is 1.67mm and 1.24mm respectively (data tables also included in Appendix E). The average accuracy value from this test does satisfy the 2mm accuracy requirement. The average repeatability value does not meet the 1mm repeatability requirement, but is only out by 0.24 mm, which could lead to possible misalignment between the parts and the robot gripper. The overall performance is still very good, when considering that the entire body is 3d-printed and that the GT2 pulleys are not backlash free. The measurements are tested for possible outliers in Appendix E, but none of the measurements qualify to outliers (according to the Chauvenet's Criterion).

B. Vision System performance:

The average accuracy of the vision system also had to be determined, to make sure that linear distance between the part and the reference point, in reality, corresponds with the values returned by the webcam and vision system.

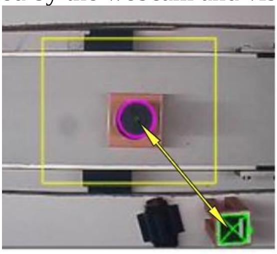
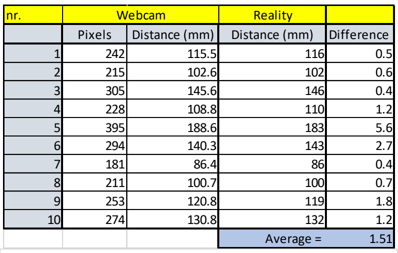


Figure 69: Distance measuring between parts and reference point.

To test the accuracy, the part was placed in multiple positions of the boundary region and then the readings from the vision program was compared to the actual measured distances between the part center and the reference tag center.

Table 4: Comparing the distances returned by vision system with the measured real distances.



nr.	Webcam		Reality	
	Pixels	Distance (mm)	Distance (mm)	Difference
1	242	115.5	116	0.5
2	215	102.6	102	0.6
3	305	145.6	146	0.4
4	228	108.8	110	1.2
5	395	188.6	183	5.6
6	294	140.3	143	2.7
7	181	86.4	86	0.4
8	211	100.7	100	0.7
9	253	120.8	119	1.8
10	274	130.8	132	1.2
			Average =	1.51

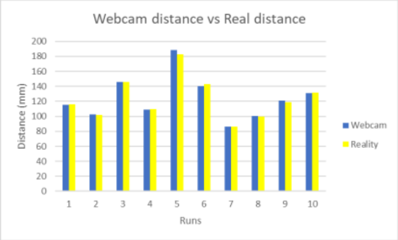


Figure 70: Vision accuracy test graph.

The vision system performed rather well, with an average accuracy of 1.51mm. The error was also perceived to increase as the part moved closer to the edges of the boundary. However, this has more to do with the camera angles than the software, and if it seems like a real issue can be resolved by using the camera calibration (as indicated in Appendix A)

C. Design Stage Uncertainty

Design stage uncertainty represents and estimation of the minimum uncertainty of an implemented system. This could be based on the equipment or chosen methodology [26].

The main components of this robotic design, that contribute to design stage uncertainty, are the Nema 17 stepper motors and the Arduino Mega. Other components, like the servo motor in the gripper, will also have a design stage uncertainty, but this would not contribute to alignment inaccuracies. All the design stage uncertainties of the different components can be combined, using the square root of the sum of the squares (RSS) method, to form the system design stage uncertainty [26].

U_0 is known as the zero-order uncertainty.

$$U_0 = \frac{1}{2}\,Resolution \qquad (6)$$

U_c is known as instrument uncertainty.

$$Design\ uncertainty\ (U_d) = \sqrt{U_0^2 + U_c^2} \qquad (7)$$

For Nema 17 stepper motor:

$Resolution\ (R_m) = 1.8°$

Using equation 6: $U_{0-m} = 0.9°$

$U_{c-m} = \pm 0.05°$

Using equation 7: $U_{d-m} = 0.9014°$

For Arduino Mega microcontroller (Appendix F):

$$Resolution\ (R_c) = \frac{Current\ output\ per\ pin}{Number\ of\ bits} = \frac{40mA}{2^{10}-1}$$

Thus, $R_c = 3.91 \times 10^{-5}$A

Using equation 6: $U_{0-c} = 1.96 \times 10^{-5}$

Using equation 7: $U_{d-c} = 1.96 \times 10^{-5}$

Total system uncertainty:

$$U_{d-total} = \sqrt{{U_{d-m}}^2 + {U_{d-c}}^2} \qquad (8)$$

$U_{d-total} = \pm 0.9014$

The system design stage uncertainty is 0.9014% and is primarily due to the stepper motors. Considering that the resolution of the stepper motors is 1.8 degrees, this is a very small uncertainty. Therefore, it is safe to assume that any inaccuracies in the robot performance, is due to mechanical imperfections, friction, vibration or even bad weight distribution.

D. Discussion

Following the robot and vision system testing, the manual operational sequence was tested on fifteen different runs. The robot was able to perform entire sequence without error (meaning misalignment with either the part or assembly point) 12 times out of the 15 runs, thus having a 80% success rate. For some of these failed attempts, it was noted that fluctuation in the surrounding environment's light could causes the tracked center point of the par to move around. This alongside the robot's repeatability, could account for the misalignment between the part or assembly point, and the robot TCP.

E. Videos of the station's operation modes

Manual:
https://www.youtube.com/watch?v=mJkzlfh3aUs
Automatic:
https://www.youtube.com/watch?v=iQmoW1aUOOc

VII. CONCLUSION

In this report the requirement of developing a robotic system that utilizes an at least three DOF robotic arm, controllable linear conveyor and machine vision, to perform peg-in-hole assembly and disassembly, was investigated, designed, implemented and tested. A systematic approach to solving this presented problem has been followed in the report. The robotic station consists of three major electro-mechanical components: the robotic arm, linear conveyor and station structure. Research has been conducted regarding each aspect of the robotic system, mechanical, electrical and software. Each of these components were broken down into their respective sub-components to be researched, designed, developed, and then subsequently tested.

The final robotic station design consisted of a three DOF SCARA robot, a simple controllable linear belt conveyor and a white chipboard-and-aliminium profile structure, with imbedded lighting features. The conveyor is used to transport the part to the desired location, where traditional machine vision programs are used to identify and track the parts location. The robotic arm is then used to perform the peg-in-hole disassembly and subsequent assembly. The system can perform the sequence of operation in manual mode (with operator input at required each step) or Automatic mode (only requires the operator to start the conveyor and then the automatic sequence).

The performance testing of the robotic system revealed a slight shortcoming in the robot's repeatability, which could lead to slight misalignments between the part and the robot gripper, but overall, the robot is able to perform the required peg-in-hole assembly/disassembly tasks.

Further improvements to the design can be made by improving the weight distribution of the robotic arm. The part identification and tracking aspects can be improved by implementing a machine learning model rather than traditional vision methods (if the processing power is available) The Station could also be made to be an enclosure, thus the lighting conditions could be controlled completely.

VIII. References

[1] L. Calderone, "What is Machine Vision?," Robotics Tomorrow, 17 December 2019. [Online]. Available: https://www.roboticstomorrow.com/article/2019/12/what-is-machine-vision/14548. [Accessed 28 March 2022].

[2] Cognex, "Introduction To Machine Vision," 2016 May 2016. [Online]. Available: https://www.cognex.com/what-is/machine-vision. [Accessed 28 March 2022].

[3] G. Diesing, "How AI and Machine Vision Impact Vision Robotics," Quality Magazine, 1 September 2021. [Online]. Available: https://www.qualitymag.com/articles/96664-how-ai-and-machine-vision-impact-vision-robotics. [Accessed 29 March 2022].

[4] O. B. a. M. V. Droogenbroeck, "ViBe: A Universal Background Subtraction," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. VOL. 20, no. NO. 6, pp. 1709-1724, 2011.

[5] D. R. K. W. M. M. R. P. M. Arief Soeleman, *Tracking Moving Objects based on Background Subtraction using Kalman Filter,* EAI, 2018.

[6] J. K. B. B. B. M. Kentaro Toyama, "Wallflower: Principles and Practice of Background Maintenance," Redmond, 1999.

[7] R. M. M. P. A. Sanket Rege, "2D GEOMETRIC SHAPE AND COLOR," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 2, no. 6, pp. 2479-2487, 2013.

[8] R. P. Salunkhe and A. A. Patil, "Image Processing for Mango Ripening Stage," Sangli, 2015.

[9] Featurepreneur, "Colour Filtering and Colour Pop Effects using OpenCV Python," 2021. [Online]. Available: https://medium.com/featurepreneur/colour-filtering-and-colour-pop-effects-using-opencv-python-3ce7d4576140. [Accessed 3 May 2022].

[10] Hewlett Packard Enterprise, "WHAT IS MACHINE LEARNING?," 2020. [Online]. Available: https://www.hpe.com/za/en/what-is/machine-learning.html?jumpid=ps_1ujsi45bwk_aid-520061736&ef_id=CjwKCAjwj42UBhAAEiwACIhADoEeXL7Qmg0pTer16t1jz989w9aKR33QRBVwOp0C6djgwEPCZWqpMRoCZKIQAvD_BwE:G:s&s_kwcid=AL!13472!3!559817225203!e!!g!!what%20is%20machine%20le. [Accessed 4 May 2022].

[11] V. Kurama, "ML-based Image Processing," 2021. [Online]. Available: https://nanonets.com/blog/machine-learning-image-processing/. [Accessed 4 May 2022].

[12] OpenCV team, "About OpenCV," 2022. [Online]. Available: https://opencv.org/about/. [Accessed 4 May 2022].

[13] TensorFlow, "Models & datasets," 2022. [Online]. Available: https://www.tensorflow.org/resources/models-datasets. [Accessed 4 May 2022].

[14] PyTorch, "From Reasearch to Production," 2022. [Online]. Available: https://pytorch.org/. [Accessed 4 May 2022].

[15] S. Kumar, Vishal, P. Sharma and N. Pal, "Object tracking and counting in a zone using YOLOv4, DeepSORT and TensorFlow," Dehli, 2021.

[16] A. J. H. J. M. Chandan G, "Real Time Object Detection and Tracking Using Deep," Bengaluru, 2018.

[17] R. Thakur, "Step by step VGG16 implementation in Keras for beginners," 2019. [Online]. Available: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c#:~:text=VGG16%20is%20a%20convolution%20neural,vision%20model%20architecture%20till%20date.. [Accessed 4 May 2022].

[18] O. Gieseler, O. Gamal, S. Kumar and H. Roth, "Camera-Based Surgical Navigation System: Evaluation of Classification and Object Detection CNN Models for X-markers Detection," Siegen, 2021.

[19] Robots Done Right, "What is a Robotic Manipulator?," Robots Done Right, January 2022. [Online]. Available: https://robotsdoneright.com/Articles/what-is-a-robotic-manipulator.html. [Accessed 30 March 2022].

[20] M. A. J. P. G. D. S. M. A. ELENA GARCIA, "The Evolution of Robotics Research," IEEE, MRA, 2007.

[21] D. Goodwin, "Mechatronics and Robot Modelling and Control Lectures," 6 December 2012. [Online]. Available: https://warwick.ac.uk/fac/sci/physics/research/condensedmatt/imr_cdt/students/david_goodwin/teaching/mechatronics/. [Accessed 2 April 2022].

[22] S. B. Niku, Introduction to Robotics Analysis, Control, Applications, 3rd ed, John Wiley & Sons, 2020.

[23] B. S. ANTON LABBÉ, "Construction of a Selective Compliance Articulated Robot Arm," KTH, Stockholm, 2021.

[24] Dejan, "SCARA Robot | How To Build Your Own Arduino Based Robot," How to Mechatronics, 2 October 2020. [Online]. Available: https://howtomechatronics.com/projects/scara-robot-how-to-build-your-own-arduino-based-robot/. [Accessed 2 April 2022].

[25] Pololu Robotics & Electronics, "A4988 Stepper

Motor Driver," Pololu Robotics & Electronics, [Online]. Available: https://www.pololu.com/product/1182. [Accessed 1 November 2022].

[26] B. D. R. L. M. A. K. T. Figliola RS, Theory and Design for Mechanical Measurements, New Jersey: John Wiley & Sons, 2011.

[27] R. Pinke, "Buying Guide: Comparing Field of view when buying a Conference room video camrera," video conference Gear, 22 January 2021. [Online]. Available: https://www.videoconferencegear.com/blog/updated-buying-guide-comparing-field-of-view-when-buying-a-conference-room-video-camera/. [Accessed 30 March 2022].

[28] T. Mortenson, "Limit Switch Explained | Working Principles," Realpars, 19 October 2020. [Online]. Available: https://realpars.com/limit-switch/. [Accessed 6 April 2022].

[29] A. Choudhary, " Controlling NEMA 17 Stepper Motor with Arduino and A4988 Stepper Driver Module," Circuit Digest, 10 September 2019. [Online]. Available: https://circuitdigest.com/microcontroller-projects/controlling-nema-17-stepper-motor-with-arduino-and-a4988-stepper-driver-module#:~:text=Maximum%20motor%20speed%20for%20NEMA,setSpeed(1000)%3B. [Accessed 24 June 2022].

[30] J. K. N. Richard G.Budynas, Shigley's Mechanical Engineering Design, New York: McGraw-Hill Education, 2015.

IX. Appendix A: Information for machine vision applications

A. Lighting

There are various very important aspects of lighting to consider when designing a machine vision system, these include selecting the correct light source, quantity, the placement relative to the camera and object and any possible reflections within work area. There is a very wide variety of equipment and there are multiple different lighting techniques, used to modify the perception of the object in regard to the camera [2].

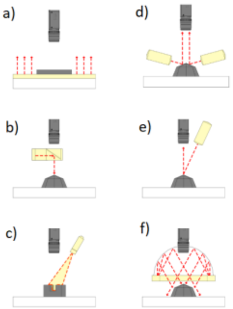


Figure 71: Lighting techniques

Back lighting highlights an object's outline and works great for when applications need only edge measurements. Back lighting detects shapes and makes dimensional measurements more reliable. Axial diffuse lighting couples light into the optical path from the side (coaxially). A semi-transparent mirror illuminated from the side, casts light downwards on the part. The part reflects the light to the camera through the use of a semi-transparent mirror resulting in an evenly illuminated and homogeneous image. Structured light is the projection of a light pattern at a predefined angle onto an object. It is useful for providing contrast-independent surface inspections, acquiring dimensional information, and calculating volume. Directional lighting can easily reveal surface defects and includes dark-field and bright-field illumination. Dark-field illumination generally is preferred for low-contrast applications. In dark-field illumination, specular light is reflected away and diffused light from surface texture and elevation changes are reflected into the camera. Bright-field illumination works great for high-contrast applications. Hot-spots and specular reflections on reflective surfaces require more diffused light sources to provide even illumination in the brightfield. Diffused dome lighting gives the most uniform illumination of features of interest and can mask irregularities that are not of interest. Strobe lighting is used in high-speed applications to freeze moving objects for examination [2].

a. Lenses

Lenses are used by cameras to capture the image and then deliver it to the image sensor. There are a lot of options when it comes to selecting lenses of different resolution or field of view (FOV) angle. This will vary based on the project requirements. There are two main types of lenses that most vision systems use: First we have interchangeable lenses, which are typically C-mounts or CS-mounts. The combination of lens and extension will produce the best possible image. Secondly, we have fixed lenses, which typically uses autofocus. Autofocus can be achieved either through the use of mechanically adjusted lenses or liquid lenses and usually has a fixed FOV at a given distance [2]. Figure 39 below outlines some of the specifications of different camara FOV's [27].

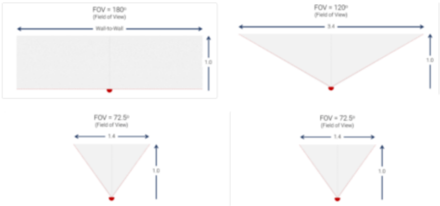


Figure 72: Different camera FOV dimensions

b. Image sensors

For a camera to capture an image it needs to have an image sensor. Image sensors mostly use charge couple devices (CCD) or complementary metal oxide semiconductors (CMOS) to generate electrical signals from incoming photons. Basically, the image sensor captures light and converts it into a digital image, while balancing the noise, sensitivity, and dynamic range. One should make sure that the camera has the right sensor resolution for the required application [2].

c. Vision Processing

Perhaps the most important part of Machine vision is the vision system algorithm. We will investigate multiple options in the rest of this literature review, that function in various different ways. The basic goal for most of these algorithms are to extract the required information from a digital image or video feed for a specific application [2]. There are two main methods for image processing namely: traditional image processing and machine learning methods. Both these methods will be further discussed in this literature review.

d. Communication

Communication needs to be established among the subsystems for the integrated machine vision system. This can be achieved by sending discrete input output (I/O) signals over a serial connection to a microcontroller. It is typical for serial protocols to use ethernet and RS-232, or new systems also use industrial protocols such as Ethernet/IP [2]. Communication between the operator and system is also very important, so it is standard practice to include a human machine interface (HMI). Typically, the HMI would display the process along with all the relevant information that the user would need to see, this will enable the user to interact with the system, or even control certain aspects of the system directly.

X. Appendix B: Information regarding robotic applications

A. Manipulators

Robotic arms are devices that are composed of a series of jointed segments that form an arm-like manipulator. The general construction of a robotic manipulator consists of rigid links that are connected by joints. The number of Degrees of Freedom (DOF) of a robotic manipulator dictates the movement capabilities of the end effector. The Axes of Motion (AOM) defines the maneuverability and complexity of the robotic design, this will refer to the types of motion of each joint (like prismatic or rotational) [21]. Generally, the first three joint are considered as the body, with the last three joints forming the wrist, but several variations of serial configurations exist. The structure of the robotic manipulator will determine the reach of its end-effector and its work envelope [19]. The main categories of configurations are cartesian, cylindrical, spherical, Selective Compliance Assembly Robot Arm (SCARA) and articulated. See Figure 73 below:

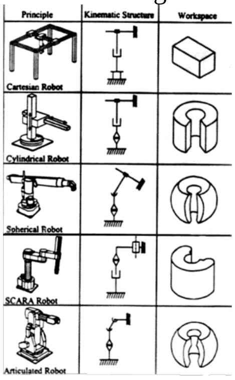


Figure 73: Different manipulators and their envelopes

These kinematic configurations are based on the combinations of prismatic (P) and revolute (R) joints of the manipulator. The joints for the different configurations are: cartesian (PPP), cylindrical (RPP), spherical (RRP), SCARA (RRP) and articulated (RRR).

B. Actuators

Overall motion of the robot is achieved through actuators, which are responsible for driving the joints. There are three main classes of actuators, which are electric, pneumatic and hydraulic. Robots mostly utilize electric actuators and generally consists of a combination of the following components: motor, encoder, castings, gearbox, timing belts and pulleys, drives and couplings. Actuators can either be used in direct-drive or could have an intermediate system to adjust the output to the required levels. This is often done with electric motors, through a gearbox or timing belts, to adjust torque output or to alter their levels of speed and precision. Different actuators, have different characteristics and it is important to select the correct actuators, based on the requirements of the robot [22].

For this project stepper motors will be the main source of actuation and thus the Table 4, as shown below, is considered.

C. Sensors

Sensors for robots are mostly aimed at collecting information about the joint associated variables, such as position, speed and acceleration. There are other sensors that could also be used to monitor the environment or current state of the robot before the robotic operation begins. Encoders are used to track joint movement and can either be absolute or incremental. There are several types of encoders: optical, magnetic, capacitive, and inductive. Encoder pulses per revolution, alongside the actuator and possible reduction system (gearbox or timing belts), are used to control the resolution of the robotic joints. Resolution is the smallest increment of motion, of a joint, that can be detected and controlled by the control system [22].

Sensors like mechanical limit switches or proximity sensors are also widely used in robotics to detect the presence or position of objects. They are often used to establish calibration sequences for robots or to define a custom safe operating area of the manipulator [28].

D. Controllers

Controllers are responsible for the motion control of the actuators. Controllers receive information from the processor and then coordinates the required motion, by driving the actuators. There are two main control methods namely: servo control (open-loop) and non-servo control (closed-loop). Servo control can send input commands to the joints as a function of the actuators, due to the built-in encoders of servo motors. Non-servo control methods make use of limit switches or other similar devices, to establish predetermined limits for all the different axes. For both these methods a control algorithm is required to execute on a computer to provide the input values that need to be sent to the joints [22].

Furthermore, there are two types of closed-loop control. The first is point-to-point control, which provides no control over the path taken. It simply moves from one point to another. The second is continuous path control, which provides control over the whole path taken between two points [22].

E. Processors

The processor is required to control the overall system of the robot. The processor will execute tasks like forward kinematics, inverse kinematics, velocity kinematics, trajectory planning and path planning, decoupled control, force control, dynamics and machine vision programs. To accomplish this, the processor operates like a computer. It requires an operating system, peripheral equipment and the applicable programs [22].

F. End Effectors

End effector refers to the component attached to the final manipulator axis and is the component that interacts with the environment to perform a certain task. The end effector is usually specially designed for the purpose of the robot and can either be controlled by the same controller as the robot or a separate I/O components controller [21]. End effectors can either grippers or tools. Tools can be any specific tools, like for example, welding rig or hot-ends for 3d-printing applications. There are also several different types of grippers, but can generally be defined as attachment that are used to manipulate objects by picking them up, holding and moving them [22].

XI. APPENDIX C: DESIGN STAGE CALCULATIONS

A. Conveyor

First, consider Figure 74:

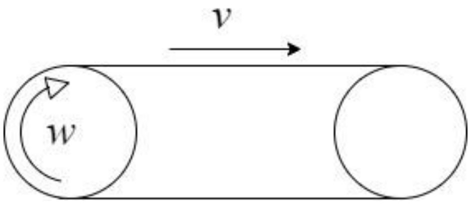


Figure 74: Conveyor representation diagram for linear and angular velocity.

The conveyor must be able to move the parts at a slow enough speed, so that the parts do not fall over when the conveyor starts or stops but should also not take too long to get the part to the required position. The student decided that it should take 3 seconds for a part placed on one end of the conveyor to move to the other end. With a conveyor length (L) equal to 535mm the required velocity will be:

$$v = \frac{L}{time} \qquad (9)$$

$v = 0.1783 m/s$

Now with a pulley diameter (D) of 30mm, we can calculate the angular velocity required:

$$v = \omega(\frac{D}{2}) \qquad (10)$$

$\omega = 11.89 rad/s$

Next, consider Figure 75:

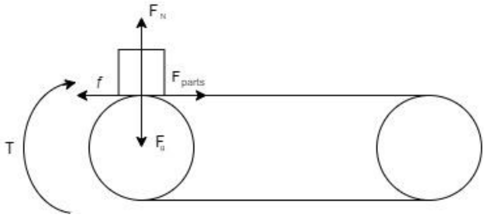


Figure 75: Conveyor free-body diagram.

The mass of the Vinyl belt i: $m_{belt} = 0.13kg$

The mass of the combined parts: $m_{parts} = 0.17kg$

For the friction coefficient value of the roller, the student chose to work with a largely exaggerated value: $f = 0.5$

This will ensure that even if there are slight imperfections in some of the components, the conveyor will still operate as required.

The force required to move the parts on the conveyor can then be calculated (with 'g' being the gravitational constant = 9.8):

$$F_{parts} = f\ g\ (m_{belt} + m_{parts}) \qquad (11)$$

F_{parts} = 1.47 N

The torque required from the motor to move the conveyor (with the parts) at the desired linear velocity is then calculated as:

$$T = \frac{F_{parts} \cdot v}{\omega} \qquad (12)$$

$$T = 0.022\ Nm$$

Initially the Nema 17 Pancake stepper motor (0.7A, 0.13Nm, 25mm) was selected, with a holding torque of 0.13Nm, but after reviewing the cost, the student discovered that the larger Nema 17 (1.7A, 4kg-cm) is less expensive and has a holding torque of 0.39 Nm. Thus, the Nema 17 (1.7A, 4kg-cm) will be used to drive the conveyor.

B. Base Pulley system

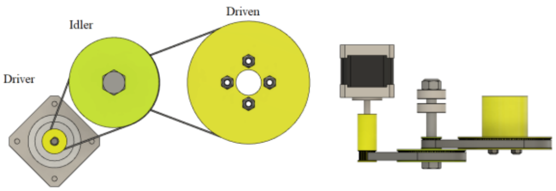


Figure 76: Robot base pulley system.

Driver:Idler = 1:4 and Idler:Driven = 1:5

Therefore, the ratio from Driver to Driven is 1:20. This means that it takes 20 full rotations from the driver to achieve one full rotation for the driven. The Nema17 stepper motor has a step angle of 1.8 degrees. This means that the theoretical resolution of the base rotation is 0.09 degrees thanks to the combination of the four pulleys. The maximum motor speed for a Nema 17 stepper motor driven by an A4988 driver, is 4688 RPM [29]. This means that the theoretical maximum RPM for SCARA-Robot base is:

$Theoretical\ Maximum\ RPM = \frac{Motor\ Maximum\ RPM}{20} =$ 234.4 RPM

C. Lifting Actuator

The mass which needs to be lifted is precisely 1.5kg (m =1.5 kg), if all the parts are 3d-printed with 100% infill. This is of course not the actual weight, because most of the parts are printed with between 20% and 40% infill, but 1.5kg is used as the theoretical weight for calculations. To calculate the required lifting toque the following equations from [30] is used.

Leadscrew specifications (Appendix F):

- Pitch (P) = 2mm
- Thread diameter (D) = 8mm
- Number of starts (n) = 4
- Friction Coefficient(u) = 0.21 (for brass nut [30])

$Lead\ (L) = P\ (n) = 8mm$

Displacement of Effort = x_E

Displacement of Load = x_L

$$Velocity\ Ratio\ (VR) = \frac{x_E}{x_L} = \frac{\pi(D)}{L} \tag{13}$$

$$VR = \frac{8\pi}{8} = \pi$$

$$\alpha = tan^{-1}\left(\frac{L}{\pi(D)}\right) \tag{14}$$

$$\alpha = 17.66^{\circ}$$
$$\beta = tan^{-1}(\mu)$$
$$\beta = 11.86^{\circ}$$

$$Leadscrew\ Efficiency\ (\eta) = \frac{\tan(\alpha)}{\tan(\alpha+\beta)} \tag{15}$$

$$\eta = 0.56$$
$$Mechanical\ advantage\ when\ lifting\ a\ load\ (M_A) = VR(\eta) = 1.759$$
$$Force\ generated\ by\ the\ mass\ (F_m) = m(g) = 14.7\ N$$
$$Required\ lifting\ force\ (F_{lift}) = \frac{F_m}{M_A} = 8.357\ N$$
$$Required\ lifting\ torque\ (T_{lift}) = F_{lift}\,.radius = 0.033\ Nm$$

The Nema 17 (1.7A, 4kg-cm) stepper motor (as mentioned above) has an output torque of 0.39 Nm, which exceeds the required lifting torque, thus it can be used to actuate the Z-axis of the SCARA-robot.

D. Arm pulley system

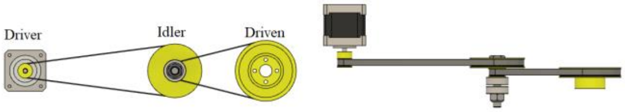


Figure 77: Robot arm pulley system.

Driver:Idler = 1:4 and Idler:Driven = 1:4

Therefore, the ratio from Driver to Driven is 1:16. Therefore, it takes 16 full rotations from the driver to achieve one full rotation for the driven. Similarly to the base, the theoretical resolution of the arm rotation is calculated to be 0.1125 degrees. This means that the theoretical maximum RPM for SCARA-Robot arm is:

$$Theoretical\ Maximum\ RPM = \frac{Motor\ Maximum\ RPM}{16}$$

= 293RPM

XII. Appendix D: Homogeneous Transform matrices

First the “A” matrices were derived:

$$H_1^{base} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} H_2^1 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H_{TCP}^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Next the above matrices are used to derive the following:

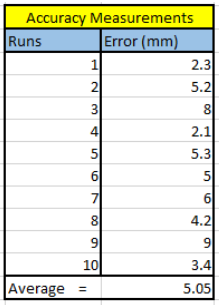
$$H_2^{base} = H_1^{base}H_2^1 = \begin{bmatrix} c_1 & -s_1 & 0 & c_1a_2 \\ s_1 & c_1 & 0 & s_1a_2 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_{TCP}^{base} = H_2^{base}H_{TCP}^2 = \begin{bmatrix} c_1c_3 - s_1s_3 & -c_1s_3 - s_1c_3 & 0 & c_1a_3c_3 - s_1a_3s_3 + c_1a_2 \\ s_1c_3 + c_1c_3 & -s_1s_3 + c_1c_3 & 0 & s_1a_3c_3 + c_1a_3s_3 + s_1a_2 \\ 0 & 0 & 1 & d_1 - d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

XIII. Appendix E: Data tables related to Performance Analysis section

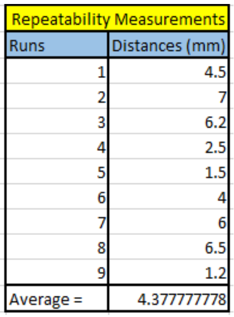
A. *Robot Test 2:*

Table 5: Test 2 Accuracy measurements



Accuracy Measurements	
Runs	Error (mm)
1	2.3
2	5.2
3	8
4	2.1
5	5.3
6	5
7	6
8	4.2
9	9
10	3.4
Average =	5.05

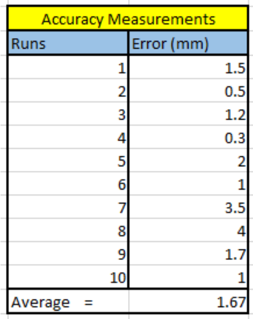
Table 6: Test 2 Repeatability measurements



Repeatability Measurements	
Runs	Distances (mm)
1	4.5
2	7
3	6.2
4	2.5
5	1.5
6	4
7	6
8	6.5
9	1.2
Average =	4.377777778

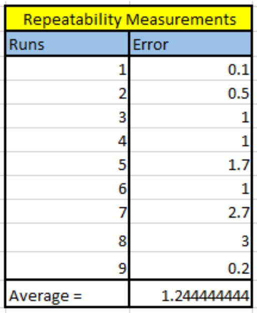
B. *Robot Test 3:*

Table 7: Test 3 Accuracy measurements



Accuracy Measurements	
Runs	Error (mm)
1	1.5
2	0.5
3	1.2
4	0.3
5	2
6	1
7	3.5
8	4
9	1.7
10	1
Average =	1.67

Table 8: Test 3 Repeatability measurements



Repeatability Measurements	
Runs	Error
1	0.1
2	0.5
3	1
4	1
5	1.7
6	1
7	2.7
8	3
9	0.2
Average =	1.244444444

C. *Outlier Test:*

Mean = $\bar{X} = \frac{1}{n}\sum_{k=1}^{N} X_k$

$\bar{X} = 1.24$

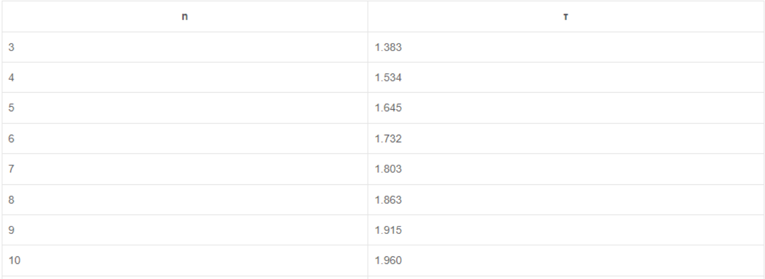
Standard deviation $s_x = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(X_k - \bar{X})^2}$

s_x= 1.033

$$T = \frac{|X_i - \bar{X}|}{S}$$

For run number 8 (suspected outlier):
T_8 = 1.7038

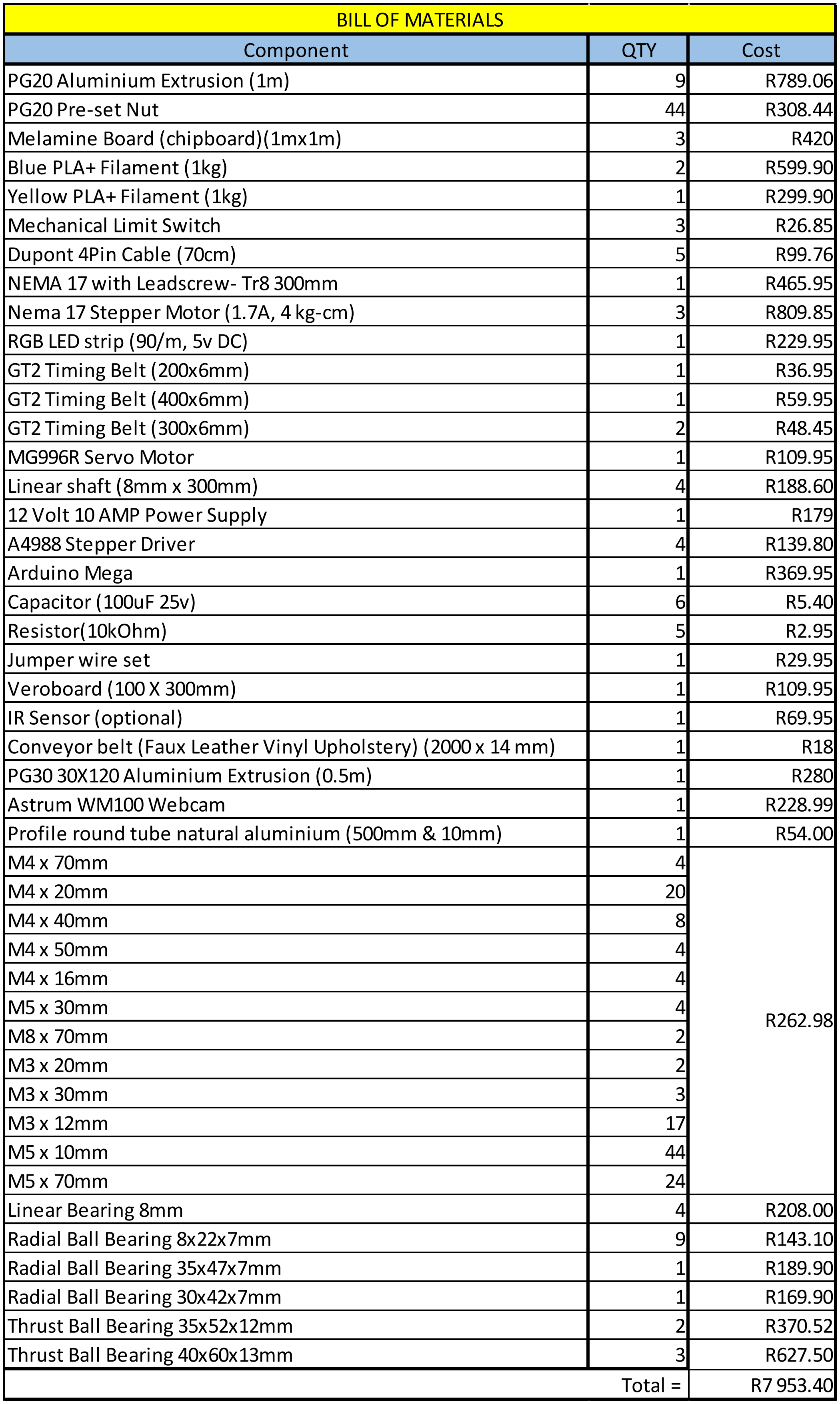
Table 9: Chauvenet's Criterion values



n	T
3	1.383
4	1.534
5	1.645
6	1.732
7	1.803
8	1.863
9	1.915
10	1.960

For n = 9:
$T_8 < T_C$
1.7038 < 1.915
Therefore run 8 cannot be considered as an outlier.

XIV. Appendix F: Bill of Materials and Component Specification Sheets



BILL OF MATERIALS		
Component	QTY	Cost
PG20 Aluminium Extrusion (1m)	9	R789.06
PG20 Pre-set Nut	44	R308.44
Melamine Board (chipboard)(1mx1m)	3	R420
Blue PLA+ Filament (1kg)	2	R599.90
Yellow PLA+ Filament (1kg)	1	R299.90
Mechanical Limit Switch	3	R26.85
Dupont 4Pin Cable (70cm)	5	R99.76
NEMA 17 with Leadscrew- Tr8 300mm	1	R465.95
Nema 17 Stepper Motor (1.7A, 4 kg-cm)	3	R809.85
RGB LED strip (90/m, 5v DC)	1	R229.95
GT2 Timing Belt (200x6mm)	1	R36.95
GT2 Timing Belt (400x6mm)	1	R59.95
GT2 Timing Belt (300x6mm)	2	R48.45
MG996R Servo Motor	1	R109.95
Linear shaft (8mm x 300mm)	4	R188.60
12 Volt 10 AMP Power Supply	1	R179
A4988 Stepper Driver	4	R139.80
Arduino Mega	1	R369.95
Capacitor (100uF 25v)	6	R5.40
Resistor(10kOhm)	5	R2.95
Jumper wire set	1	R29.95
Veroboard (100 X 300mm)	1	R109.95
IR Sensor (optional)	1	R69.95
Conveyor belt (Faux Leather Vinyl Upholstery) (2000 x 14 mm)	1	R18
PG30 30X120 Aluminium Extrusion (0.5m)	1	R280
Astrum WM100 Webcam	1	R228.99
Profile round tube natural aluminium (500mm & 10mm)	1	R54.00
M4 x 70mm	4	R262.98
M4 x 20mm	20	
M4 x 40mm	8	
M4 x 50mm	4	
M4 x 16mm	4	
M5 x 30mm	4	
M8 x 70mm	2	
M3 x 20mm	2	
M3 x 30mm	3	
M3 x 12mm	17	
M5 x 10mm	44	
M5 x 70mm	24	
Linear Bearing 8mm	4	R208.00
Radial Ball Bearing 8x22x7mm	9	R143.10
Radial Ball Bearing 35x47x7mm	1	R189.90
Radial Ball Bearing 30x42x7mm	1	R169.90
Thrust Ball Bearing 35x52x12mm	2	R370.52
Thrust Ball Bearing 40x60x13mm	3	R627.50
	Total =	R7 953.40

A. Component specification

Nema 17 (1.7A, 4kg-cm) stepper motor:
https://www.diyelectronics.co.za/store/stepper-motors/43-nema-17-stepper-motor-17a-4-kg-cm.html

MG996R servo motor:
https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf

A4988 Stepper motor driver:
https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf

Leadscrew:
https://www.diyelectronics.co.za/store/leadscrew-ballscrew/1497-8mm-metric-acme-lead-screw-only-300mm.html

Arduino Mega:
https://www.robotshop.com/media/files/PDF/ArduinoMega2560Datasheet.pdf

Power supply:
https://www.takealot.com/dc-12v-10a-120w-power-supply-mrul/PLID90071037?gclid=CjwKCAjwzY2bBhB6EiwAPpUpZmgr6eudPDjym2c5k1hGfkn9fv9_qdpZgINc70g-KE1ghVWhGFHE6hoC3loQAvD_BwE

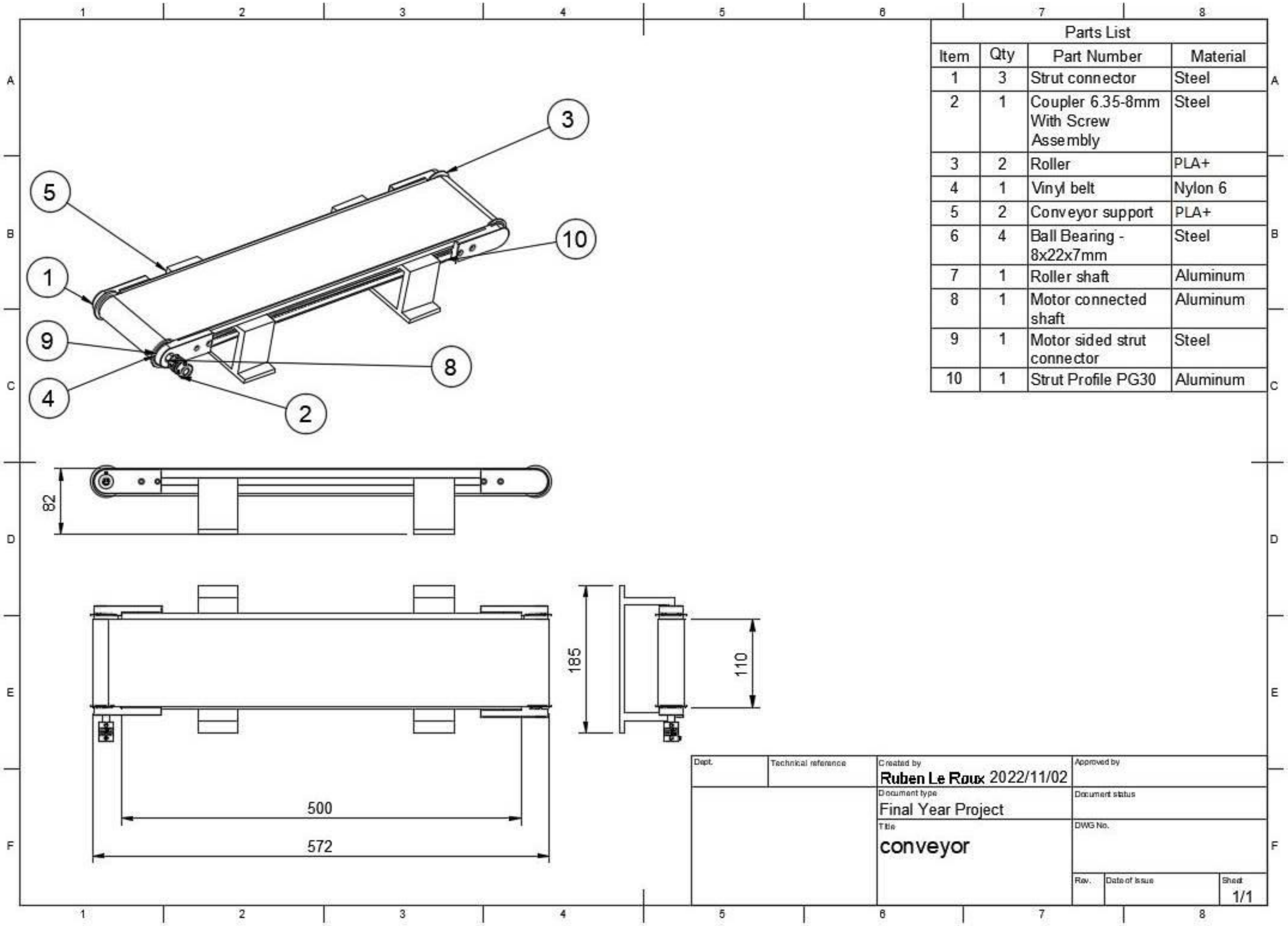
All the bearings:
https://www.bearings.co.za/catalogues/BI/index-h5.html?page=1#page=1

Webcam:
https://www.takealot.com/astrum-full-hd-usb-webcam-with-mic-wm100/PLID73052423

GT2 timing belts:
https://www.handsontec.com/dataspecs/gt2-belt-B.pdf

PG20 Aluminium profile:
https://moduasm.co.za/product/strut-profile-pg20-20x20-4-slots-1-11-20-020020-04-2/

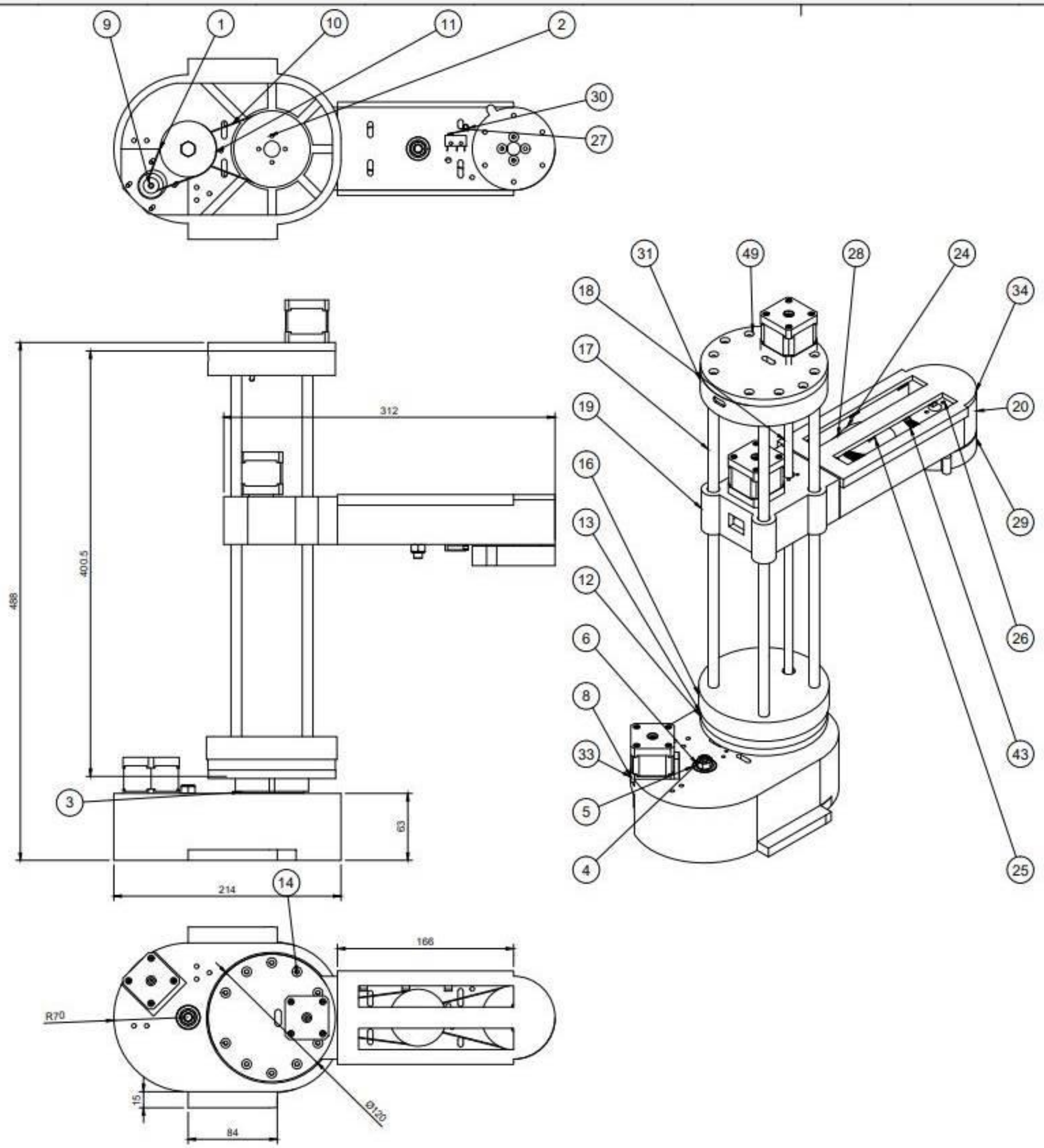
XV. Appendix G: CAD design drawings (Fusion 360)



Parts List

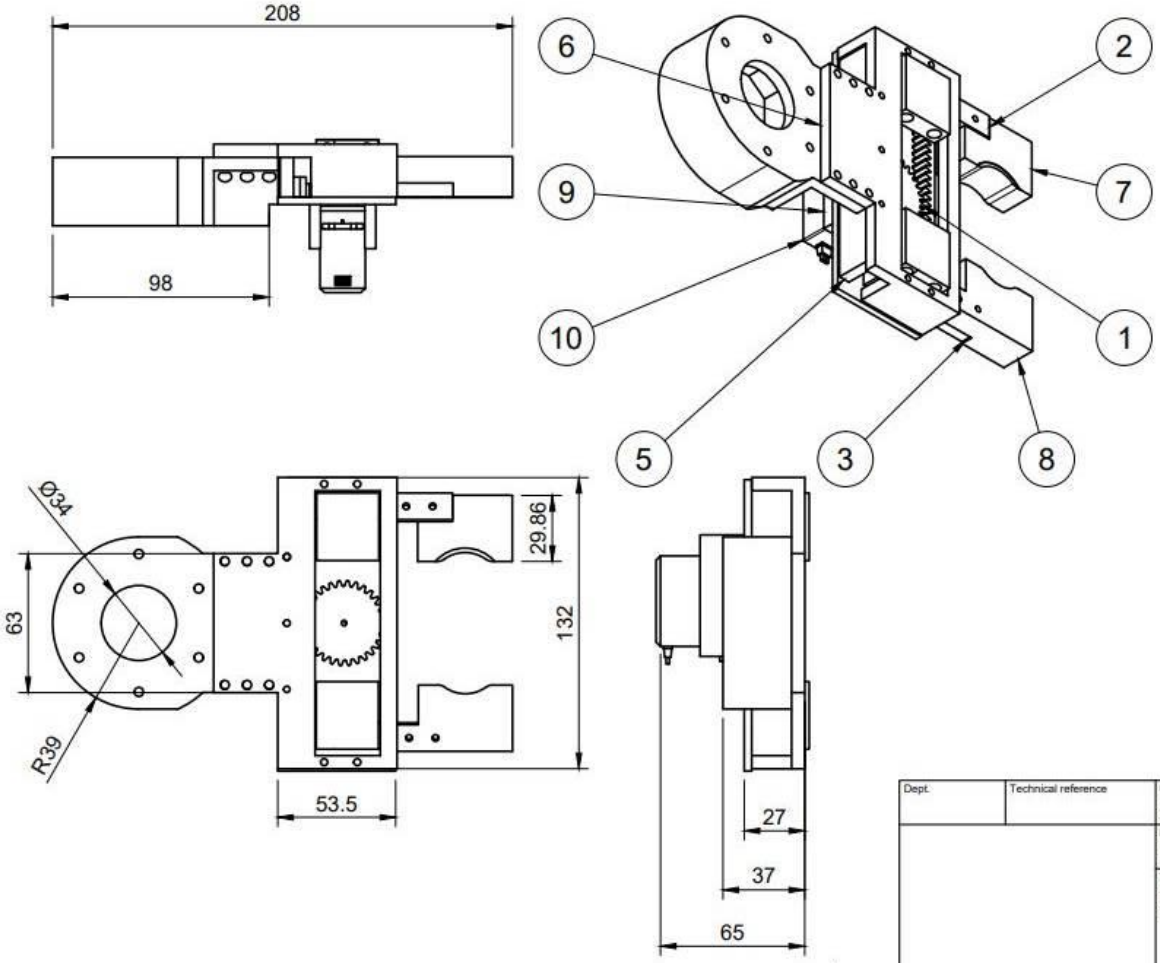
Item	Qty	Part Number	Material
1	3	Strut connector	Steel
2	1	Coupler 6.35-8mm With Screw Assembly	Steel
3	2	Roller	PLA+
4	1	Vinyl belt	Nylon 6
5	2	Conveyor support	PLA+
6	4	Ball Bearing - 8x22x7mm	Steel
7	1	Roller shaft	Aluminum
8	1	Motor connected shaft	Aluminum
9	1	Motor sided strut connector	Steel
10	1	Strut Profile PG30	Aluminum

Dept.	Technical reference	Created by Ruben Le Roux 2022/11/02	Approved by		
		Document type Final Year Project	Document status		
		Title conveyor	DWG No.		
			Rev.	Date of issue	Sheet 1/1



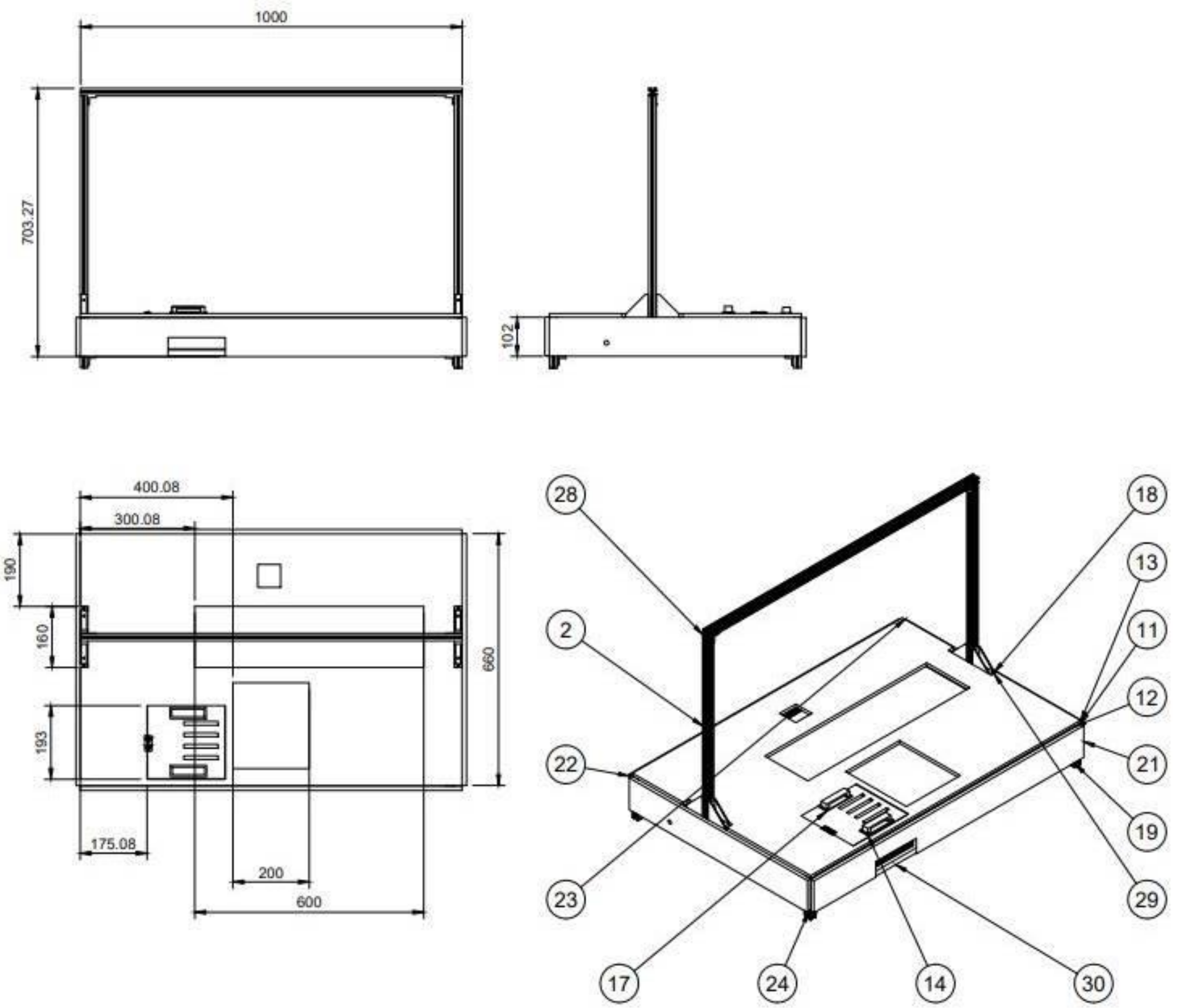
Parts List			
Item	Qty	Part Number	Material
1	1	22 - 80 teeth_GT2 Pulley	PLA+
2	1	110 teeth - GT2 Pulley	PLA+
3	3	Thrust Bearing - 40x60x13mm	Steel
4	1	Bolts_M8 70mm	Steel
5	4	Ball Bearing - 8x22x7mm	Steel
6	2	M8 Lock nut	Steel
7	1	Ball Bearing 30x42x7mm	Steel
8	3	NEMA 17 Stepper Motor	Steel
9	1	Base 20 teeth -GT2 Pulley	PLA+
10	1	GT2 Belt 300mm	Rubber, Black
11	1	GT2 Belt 200mm	Rubber, Black
12	1	Joint 1 coupler	PLA+
13	1	Z-axis Bottom Plate	PLA+
14	8	Smooth Rod Clamp	PLA+
15	1	Ball Bearing - 8x22x7mm	Steel
16	1	Rotation Base cover	PLA+
17	4	Smooth Rod 8mm	Steel
18	1	Lead Screw D8mm	Steel
19	1	Z-axis Mounting Platform	PLA+
20	1	SCARA Robot Arm 1	PLA+
21	1	Lead Screw Nut 8mm	Steel
22	4	Linear Bearing 8mm	Steel
23	1	20 teeth_GT2 Pulley	PLA+
24	1	23 - 80 teeth_GT2 Pulley	PLA+
25	1	Bolts_M8 45mm	Steel
26	1	92 teeth - GT2 Pulley	PLA+
27	1	GT2 Belt 300mm	Rubber, Black
28	1	GT2 Belt 400mm	Rubber, Black
29	1	Joint2 Coupler	PLA+
30	2	Limit switch	Steel
31	1	Top cover	PLA+
33	1	SCARA Robot Base	PLA+
34	1	Perspex top v5	PLA+
42	1	Ball Bearing 30x42x7mm_Ball Bearing 30x42x7mm	Steel
43	2	Thrust Bearing 35x52x12mm_Thrust Bearing 35x52x12mm	Steel
49	1	Z-axis Top Plate	PLA+

Dept.	Technical reference	Created by Ruben Le Roux 2022/11/03	Approved by
		Document type Final Year Project	Document status
		Title SCARA_Robot_Main design	DWG No.
			Rev. · Date of issue · Sheet 1/1



Parts List			
Item	Qty	Part Number	Material
1	1	Spur 2	PLA+
2	1	Rack1 v12	PLA+
3	1	Rack2	PLA+
5	1	Back	PLA+
6	1	Top	PLA+
7	1	Gripper end1	PLA+
8	1	Gripper end2	PLA+
9	1	Bottom plate	PLA+
10	1	Servo Motor MG996R 3D Model	

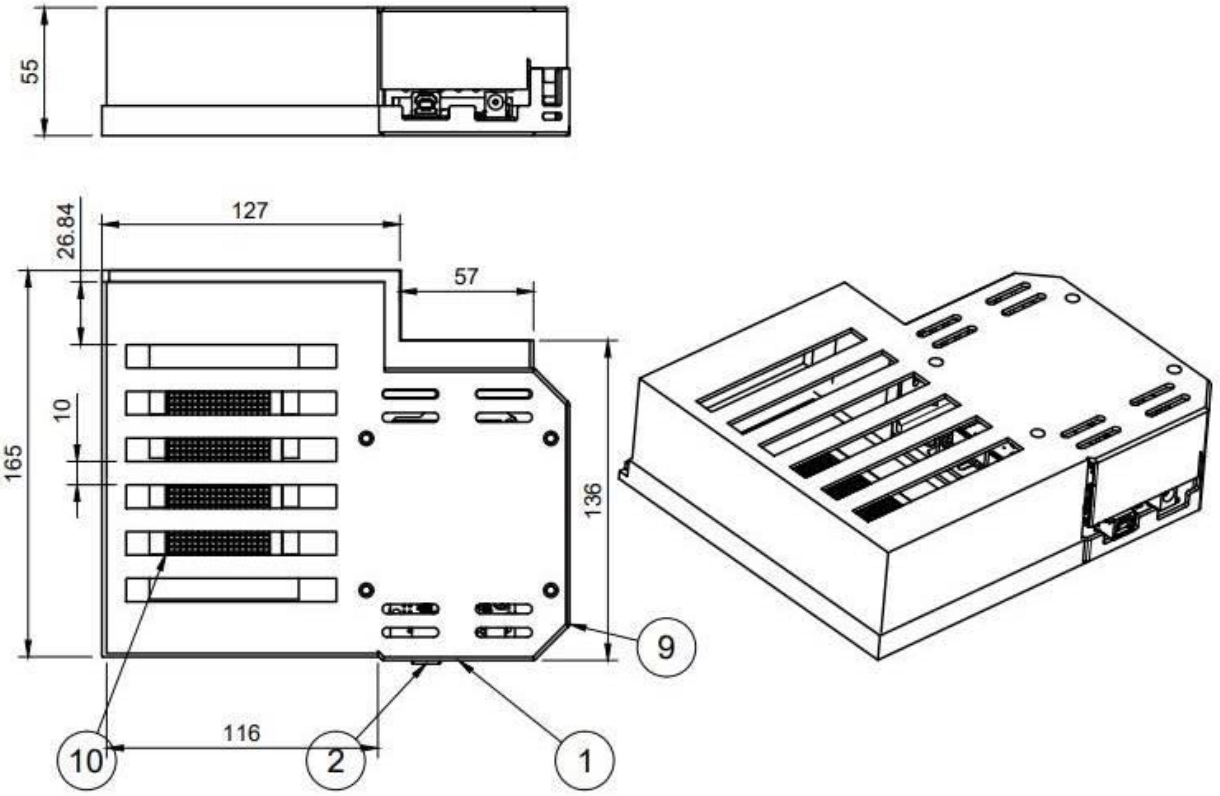
Dept.	Technical reference	Created by Ruben Le Roux 2022/11/03	Approved by
		Document type Final Year Project	Document status
		Title Second_arm_design	DWG No.
			Rev. Date of issue Sheet 1/1



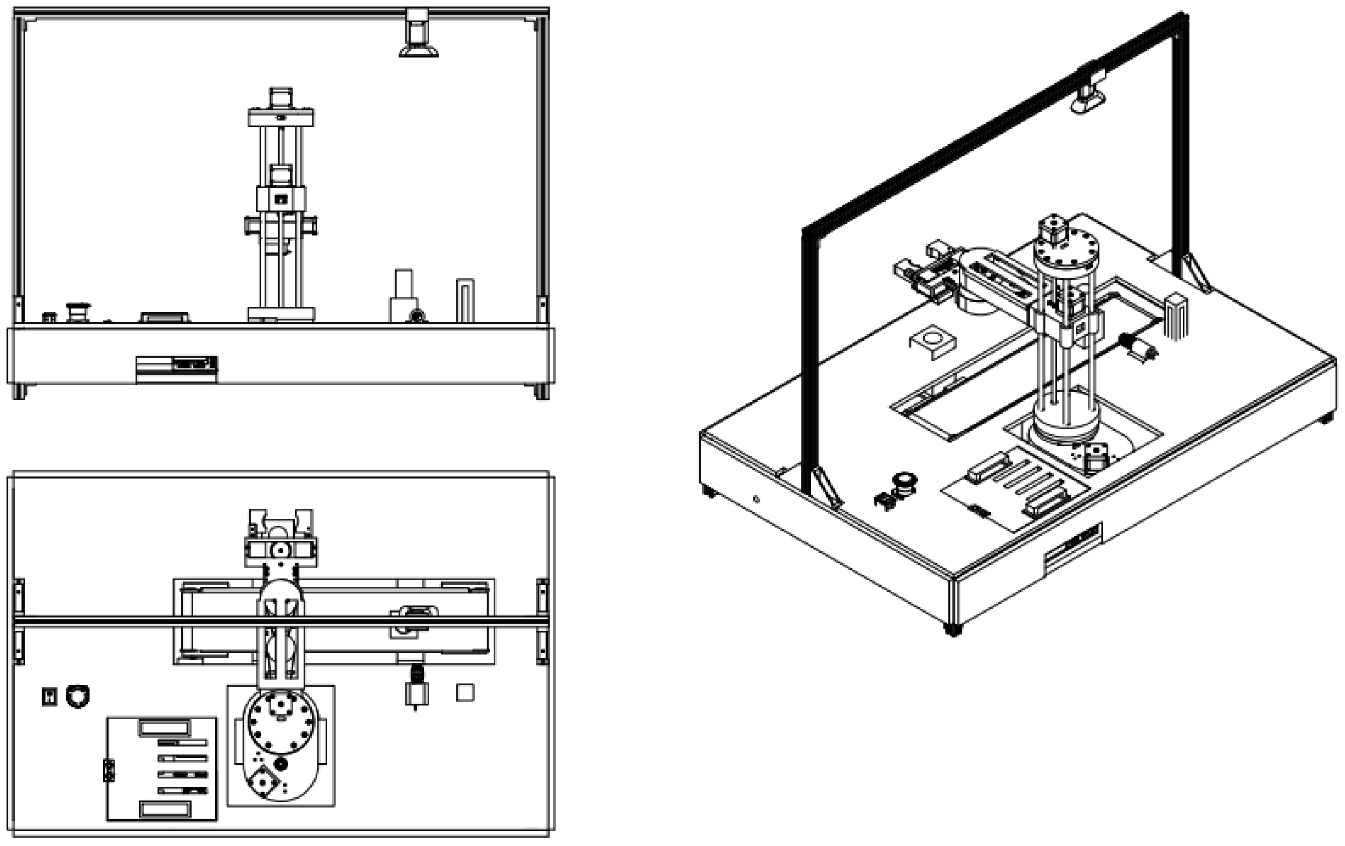
Parts List		
Item	Qty	Part Number
2	2	PG20 500mm
11	4	PG20 140mm
12	4	PG20 960mm
13	1	Chipboard top
14	1	Hatch door
17	1	Handle
18	4	90degree connector
19	18	L_connector
21	1	Front cover
22	1	Back cover
23	1	Right side cover
24	1	Left side cover
28	1	Camera mounting strut
29	4	PG20 720mm
30	1	Bottom board

Dept.	Technical reference	Created by Ruben Le Roux 2022/11/03	Approved by
		Document type Final Year Project	Document status
		Title Station Structure	DWG No.
			Rev. Date of issue Sheet 1/1

Parts List		
Item	Qty	Part Number
1	1	Case top
2	1	R1.6+AM2560
9	1	Case bottom
10	1	Veroboard 140x95mm



Dept.	Technical reference	Created by Ruben Le Roux 2022/11/03	Approved by
		Document type Final Year Project	Document status
		Title Electrical Case	DWG No.
			Rev. / Date of issue / Sheet 1/1



Dept.	Technical reference	Created by	Approved by
		Ruben Le Roux 2022/11/03	
		Document type	Document status
		Final Year Project	
		Title	DWG No.
		Fully Assembled Robotic System	
			Rev. / Date of issue / Sheet
			1/1

XVI. Appendix H: Self-Assessment

To indicate the parts of the report that present evidence of achieving the required outcomes, each requirement question will be stated, and an answer will be provided with supporting evidence from the report.

1. **Is it clear as to what will be delivered/accomplished as a result of this project?** The introduction section of this report provides a clear outline of the project requirements and what will be delivered as a result of those requirements. It introduces the project as a whole to the reader and indicates the project background, purpose and specific requirements.
2. **Is it clear that knowledge of mathematics, basic science and engineering sciences have been applied from first principles to solve engineering problems?** The background research and literature review section, along with Appendix A, B, C and D serves as evidence that a theoretical knowledge of engineering sciences and mathematics have been used to analyse and solve the various design related problems.
3. **Has a systematic approach been followed regarding the following tasks?** Clear evidence of a systematic approach is provided by the separate sections investigated in the literature review, the explanation of the system requirements, thorough problem analysis and the detailed design description for each main aspect of the project. Each section builds off the knowledge and design, of the previous sections.
   - 3.1. **Identify and formulate the problem.** The problem for this project is identified and formulated in the introduction and system requirements sections of the report. As the report progresses, the intricacy of the problem is further elaborated and addressed.
   - 3.2. **Plan the design process.** The design process is planned during the literature review and related background research sections. Then the various aspects of the design process are showcased in the problem analysis and design description section and in Appendix A, B and C. This is based on the formulated project problem statement and system requirements.
   - 3.3. **Acquire and evaluate the requisite knowledge, information, and resources.** Knowledge, information, and resources is acquired during the background research and literature review section, including Appendix A and B, of the report. As the problem analysis and design description part of project progresses, more knowledge, information and resources are identified and used where necessary.
   - 3.4. **Evaluate alternatives and preferred solution.** During the background research and literature review multiple different methods for solving related problems are investigated. In the problem analysis and design description section various alternative solutions are considered for the mechanical, electrical and software designs. The concepts of each design sub-section are then compared based on the various project criteria/requirements, before selecting the final design.
   - 3.5. **Assess impacts and benefits of the concept designs.** The impacts and benefits of the conceptual designs are discussed and assessed in the problem analysis and design description section. Different concepts are also analysed during background research and literature review section and in Appendix A and B.
4. **Is the theory correctly applied for solving a design/problem?** Theoretical sources are identified in the background research and literature review section and also in Appendix A & B. The theory is applied in the problem analysis and design description section, including Appendix C, D and E.
5. **Is it clear that appropriate engineering methods, skills and tools have been used in the project?** Various engineering skills, methods and tools are used throughout the whole report. A systematic design approach was followed in all the relevant sections of the report.
6. **Is the document written in a scientific and appropriate style?** The report is written in a scientific style and follows the IEEE format and layout.

Next the parts of the report that present evidence of achieving the required Exit Level Outcomes (ELO), as stated by the Engineering Council of South Africa (ECSA), is explained. Each ELO will be stated, and a brief description will be provided, on how the report meets the ELO’s requirements.

ELO 1 - Problem solving

The project problem is defined and analyzed in the report’s introduction and system requirements sections. The problems discovered throughout the project design process was solved by identifying the specifications of the problem and then using applicable engineering methods, knowledge, tools and skills.

ELO 2 - Application of scientific and engineering knowledge

Engineering knowledge obtained throughout the previous years of study, is utilized all over this report. Scientific, engineering, and mathematical methods are used and demonstrated when solving problems during the project execution.

ELO 3 - Engineering design

A systematic approach is followed throughout the relevant sections of the project report. A comprehensive literature review is performed, which covers various aspects of the project. Then the system requirements are formulated. Using the relevant information gathered in the literature review and the requirements of the system, the problem for this project is analyzed. Multiple conceptual designs, for the different aspects of the project are generated and then compared, based on the relevant specifications, before selecting the final design.

ELO 4 - Investigations, experiments, and data analysis

As part of the investigation into the project’s final design performance, data is experimentally collected and analyzed.

Graphical illustrations and relevant figures are then used to quantify the projects performance, and this is then compared to the system requirements of the project.

ELO 5 - Engineering methods, skills, and tools, including information technology

The application of relevant engineering methods, skills and tools is showcased throughout the project design and report. This includes the various forms of information technology used in all aspects of the design and for operating the project.

ELO 6 - Professional and technical communication

The report and all forms of presentations of this project, demonstrates technical and professional standards of communication.

ELO 7 - Impact of engineering activity

Operator safety is taken into consideration in the design of this project. The impact of the project design and components on the environment is minimized by using biodegradable materials. When the system is being manufactured/assembled proper disposing protocols is followed for component packaging and minimal materials will go to waste.

ELO 8 - Individual, team, and multi-disciplinary working

Throughout this project the student will demonstrate his competence to perform engineering tasks as an individual.

ELO 9 - Independent learning ability

Various categories of information, which used to fall outside the student’s competence/experience, was investigated and implemented throughout the entire project. These categories were researched and then illustrated in the background research and literature review section of the report. The student then demonstrated his learning capabilities by implementing the relevant researched information in the problem analysis and design description sections.

ELO 10 - Engineering professionalism

The student will conduct himself in a professional manner when presenting the project and will produce the project within the specified time period.

ELO 11 - Engineering management

Engineering management principles was applied by the student throughout the entire project for time schedule management and economic decision making.