Karlsen, Benjamin
Lersveen, Fridtjof Pedersen

Robotic assembly line for junction boxes

Bachelor's thesis in automation and robotics
Supervisor: Kleppe, Adam Leon
Co-supervisor: Mork, Ola Jon & Kleppe, Paul Steffen
May 2023

Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Karlsen, Benjamin
Lersveen, Fridtjof Pedersen

Robotic assembly line for junction boxes

Bachelor's thesis in automation and robotics
Supervisor: Kleppe, Adam Leon
Co-supervisor: Mork, Ola Jon & Kleppe, Paul Steffen
May 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Preface

This bachelor thesis was written by two students: Benjamin Karlsen & Fridtjof Pedersen Lersveen from NTNU Ålesund's Automatiseringsteknikk. It was originaly aquired by the ManuLAB at NTNU Ålesund when Pipelife came to them with a request to automate an accembly process for their junction boxes. Pipelife is Norway's biggest producer of plastic pipe solutions and have three factories located in Surnadal, Stathelle and Ringebu. As part of the project, the assembly line and its supporting components were designed, built, programmed, and tested.

Acknowledgement

We would firstly like to thank Pipelife for giving us the project.

Secondly, we want to express our gratitude to Adam Leon Kleppe, Ola Jon Mork, and Paul Steffen Kleppe, who served as our project's supervisors, for their guidance, encouragement, and support. Their insightful remarks helped us make better decisions and develop our ideas.

We also want to give a special appreciation to Ola Jon Mork and Irina-Emily Hansen, who repeatedly encouraged us by inviting nearby businesses into the lab where we worked, which resulted in the businesses providing valuable comments.

Finally, we would like to express our gratitude to the ManuLAB for funding the project. Without the financial assistance, this project would not have been as successful.

Summary

This report shows the development and results of an automated assembly line, which assembles junction boxes for Pipelife. The system includes various components such as robots, conveyors, linear drives, robot-tools, tool-docks, magazines, screw sorting and feeding system, jigs and fixtures. Components were manufactured and designed around 3D-printing and laser-cutting to utilize a cheap and efficient method of prototyping.

Individual testing of the components shows the overall success rate and optimal speed of the robots to ensure repeatability and minimum wear and tear while also reducing power consumption. The tests also revealed how the assembly line can be improved and optimized in future revisions to accommodate industrial standards and repeatability.

Contents

Terminology

HMI Human Machine Interface

CAD Computer Aided Design

CAM Computer Aided Manufacturing

CAE Computer Aided Engineering

PLC Programmable Logic Controller

I/O Input / Output

HVAC Heating, Ventilation and Air Conditioning

ACE Automation Control Environment

6-DOF Six degrees of freedom

AIV Autonomous Industrial Vehicle

RPM Rotations Per Minute

ROI Return On Investment

Notation

Kanban Material Flow technique in LEAN

Pull production Demand based production

Viper 1 & 2 The names of the two robotic arms used

List of Figures

List of Tables

Chapter 1

Introduction

This report will review a possible solution for assembling Pipelife's junction boxes. In total there are six variations of boxes, where this project focused on creating a base solution for one of the six variants. With this base solution, only small changes should be necessary to assemble all the different variations.

1.1 Background

Pipelife Norge AS is Norway's biggest manufacturer of plastic pipe systems for both infrastructure and the building industry. One of the products Pipelife offers are electrical junction boxes. This junction box can be mounted on poles from all four sides, offers a multitude of tube fittings at all ends, and can be mounted together to create bigger boxes. Figure 1.1 depicts a junction box and its parts.

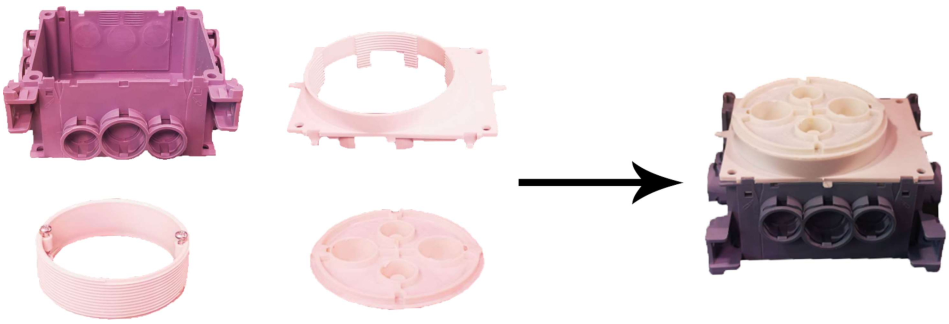


Figure 1.1: Pipelife's parts and assembled junction box

1.2 The process today

A chart of how the process is done today is displayed in figure 1.2

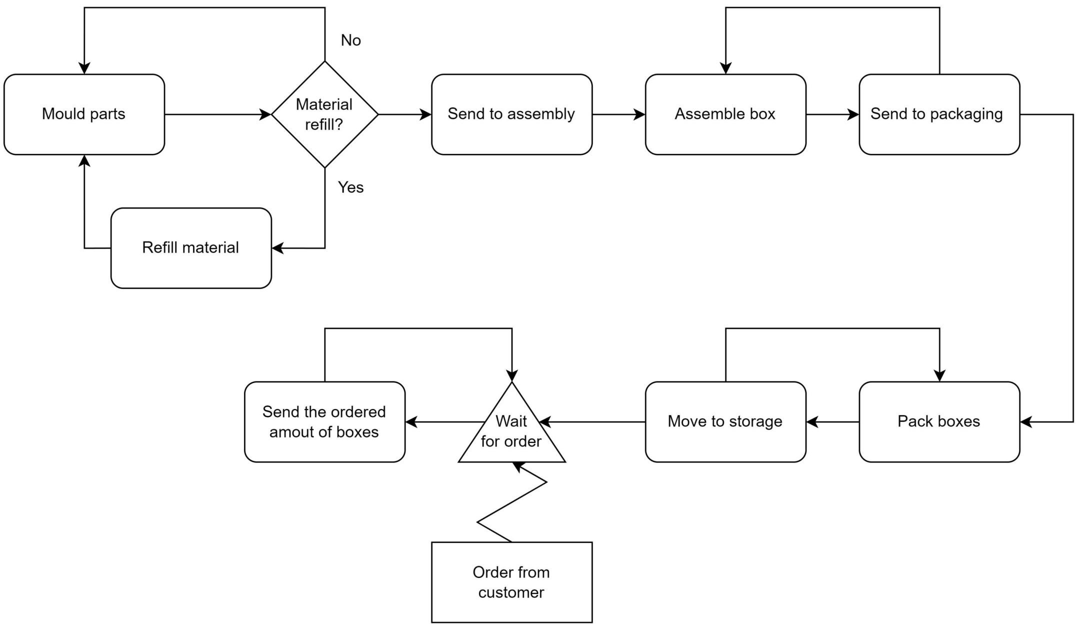


Figure 1.2: Chart of the process today

In this process the boxes are moulded at Pipelife, transported to a second company for assembly, transported to a third company for packaging, and lastly returned to Pipelife where it is stored until the boxes are ordered. This process has multiple cases of production wastes such as possible overproduction, transportation, inventory, extra-processing, and unused talents. In transportation alone, the cycle time of the production is increased by several days. Pipelife seeks to eliminate these wastes by changing the process from a manual assembly to an automated solution. This also reduces the production cost of the boxes, which can increase the global competitiveness of the product.

1.3 Problem Formulation

Before assembling a box, screws need to be inserted into the adjustment ring as shown in figure 1.3. These screws are delivered loosely, making it necessary to sort them. After sorting the screws a ring needs to be fed into the system where the location of its screw holes needs to be determined. When the holes locations are found, screws have to be accurately inserted into a given height. After insertion of the screws, the ring is ready for assembly.

Figure 1.3: Inserting screws into adjustment rings

Before starting the assembly the remaining parts needs to be fed into the system. The components are delivered loosely in a box, making it necessary to create a way to sort and feed them.

When assembling the box, the different components need to be placed with high accuracy due to their geometry. The first step in the assembly is inserting the top cover into the main box as shown in figure 1.4. The top cover has plastic clips that connects to both the inside and outside of the box, leaving next to no room for error.

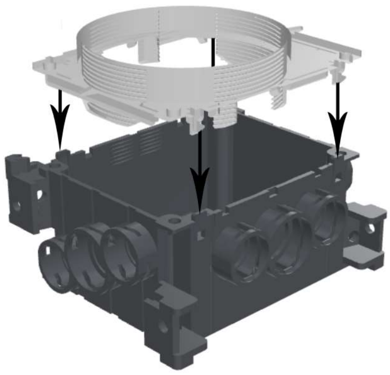


Figure 1.4: Inserting top cover into main box

The next step of the assembly is inserting the adjustment ring with screws into the top cover as shown in figure 1.5. The adjustment ring is connected with threads. Inserting the ring requires precise orientation, and knowledge of how far the ring has been rotated to make sure no damage to the threads occur.

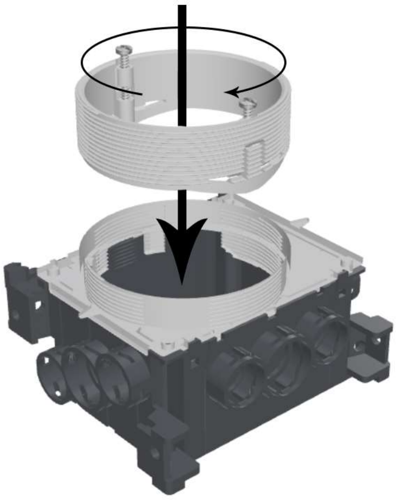


Figure 1.5: Inserting adjustment ring into top cover

The final part of the assembly is placing the lid on top of the adjustment ring as shown in figure 1.6. The lid connects to the screws in the adjustment ring using two slots inside the lid. The lid needs to be oriented correctly for the screws to fit into the slots. If the screws inserted into the adjustment ring are at the wrong height, the lid wont sit tightly.

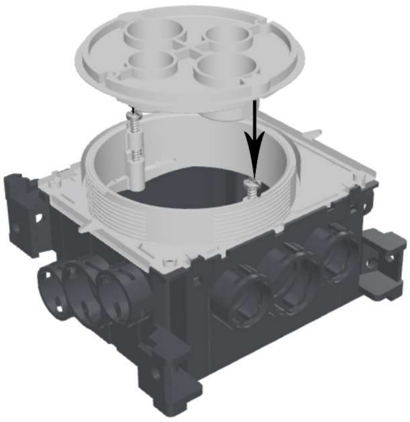


Figure 1.6: Placing lid on top of adjustment ring

After the assembly process, the box needs to be palletized to make the system ready for the next assembly.

When the system runs out of parts it needs to be restocked.

When the system has assembled multiple boxes, they need to be removed from the system.

1.4 Objectives

The system was developed with the following goals in mind:

1. Make the system flexible, so that other products can be included later.
2. Use AIVs to deliver cardboard boxes and remove cardboard boxes with products inside.
3. 20 second assembly time per box.
4. Reduce production costs.
5. Create an affordable solution.

   Some of these goals were also specified by Pipelife. All of their specifications can be seen in the appendix A.

1.5 Limitations

1.5.1 Robotic cell limitations

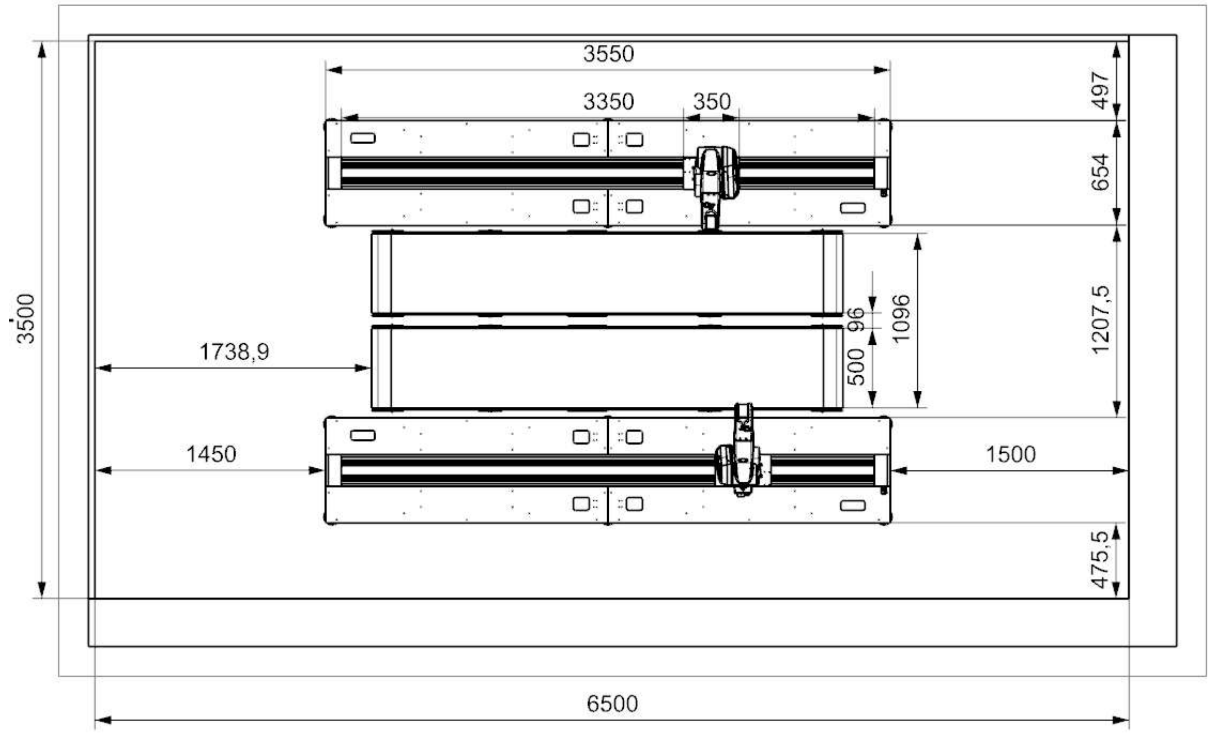


Figure 1.7: Standard Robotic Cell

The robotic cell used in this project was already present. This cell included two robots, two linear drives, two cameras, and two conveyors. Due to the restrictions on space in the cell, the assembly line's components were placed on top of the conveyors. The system might be further optimized and simplified by constructing the robotic cell in a different way. The cell and its measurements are shown in figure 1.7

1.5.2 Bin-picking limitations

This project encountered two major limitations with respect to the implementation of bin-picking. Firstly, the time required to bin-pick all components exceeded the specified limit of 20 seconds of assembly per box, with technical challenges such as object recognition, path planning, and part grasping taking too long. Secondly, the implementation of 3D-bin picking was down-prioritized due to the projects time restrains.

1.6 Approach

The project was separated into four stages.

The first stage dealt with planning and documentation of the project. Here, the group made a pre-project report (B), Gantt chart with the project deadlines (C), and a risk assessment matrix (D). The rest of the documentation was done in different stages, making this stage overlap with the rest throughout the entire project.

The second stage focused on designing and ordering all the components needed. Many different designs were tested to see which ones gave the best results.

The third stage used the components from stage two to build the system. In this part, the robots were programmed to first complete each step of the assembly separately before executing the entire assembly.

The fourth stage focused on optimizing the programs. In the previous stage, unnecessary movements were implemented to prevent the robot from crashing. These could now be removed. Other parameters, such as approach positions and movement speeds were also optimized.

1.7 Structure of the Report

The rest of the report is structured as follows.

Chapter 2 Gives an introduction to the theoretical background needed to understand and recreate the project

Chapter 3 Contains the software and hardware used

Chapter 4 Contains the methodology and materials that were considered throughout the project

Chapter 5 Contains a description of the finished system and its efficiency

Chapter 6 Contains a summary of goals, test results and system values

Chapter 7 Presents an overall conclusion

Chapter 2

Theoretical basis

2.1 Machine vision with Omron Ace

Machine vision is a technique used to display a model of the real world from images. Machine-vision algorithms record information about the image scene based on its two-dimensional projections. When using two-dimensional images to display three-dimensional models, information about the third dimension is lost and must be recovered. Knowledge of the objects in the scene and projection geometry are necessary to recover the information [12]. Figure 2.1 shows a model of a projected point in an image scene. Here the axes are represented as the vectors x, y, and z in a three-dimensional coordinate system.

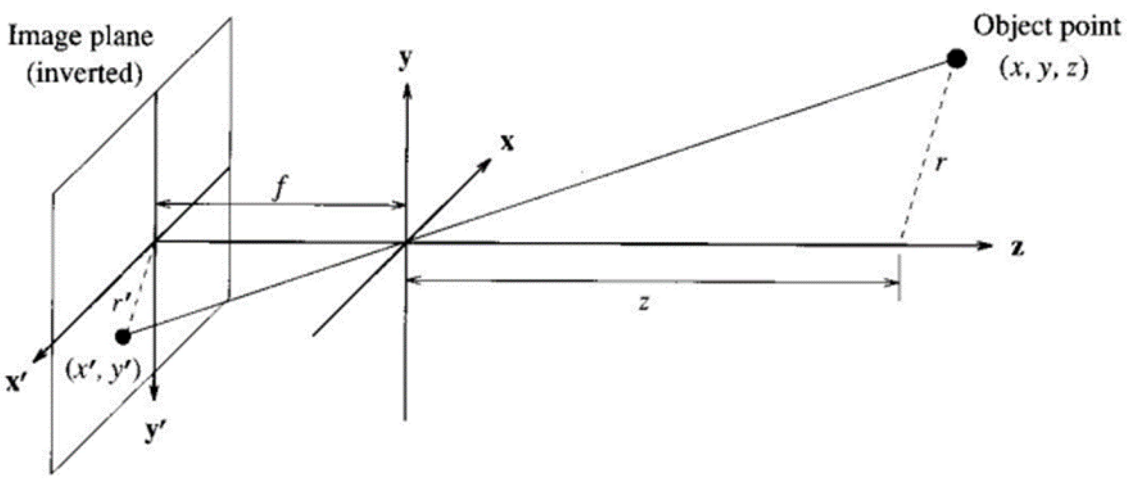


Figure 2.1: Model of a projected point in an image scene [12] ©McGraw-Hill Inc.

Since a traditional two-dimensional camera does not gain information about the vector z, different methods like stereo imaging or structured light can be used to regain this information [12]. In Omron Ace, the value of the z vector is regained when calibrating the robot's movements to the change in the image scene. In the initial step of the calibration, the robot's z value is manually inserted into the controller. Since the objects in the camera scene are always at the same height, this z value is fixed. From here, the robot will move the part to different locations and orientations within the camera scene. Before the calibration process, the camera needs to be calibrated to know the pixel-to-millimeter conversion rate within the image.

2.1.1 Camera Calibration

By assuming that the camera's central axis intersects the image plane in the center of the image array, the approximate transformation from pixel to real world distance can be used. In this method, the x-axis is defined as the direction of increasing column index, and the y-axis is defined as the direction of increasing row index in the opposite direction [12].

Equation 2.1 shows the transformation of the pixel coordinates [i, j] to image coordinates [x', y'], given that the spacing between rows and columns is the same [12]:

$$\begin{aligned} x' &= (j - \frac{m-1}{2}) \\ y' &= -(i - \frac{n-1}{2}) \end{aligned} \tag{2.1}$$

where n and m are the numbers of rows and columns in the image arrays, respectively.

In Omron Ace, cameras are calibrated using a calibration sheet such as the one seen in figure 2.2. This way, one can find the convection rate from pixels to millimeters by measuring the distance between each dot's center in both pixel distance and real-world distance. To find the conversion factor between pixel and millimeter, one must simply divide the distance in pixels by the distance in millimeters.

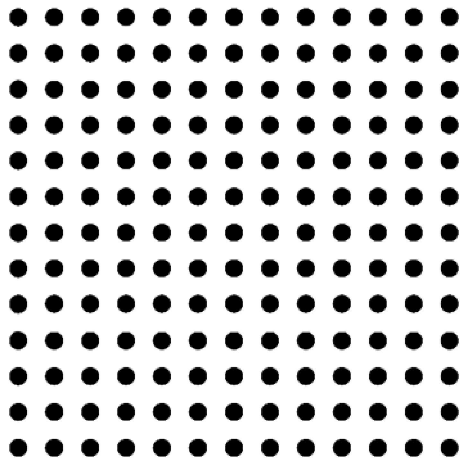


Figure 2.2: Calibration sheet used with Omron Ace

2.1.2 Edge Detection

After calibrating the camera, the system needs to be able to locate edges in the image scene to be able to find objects. The image edges are measured as significant local changes in the intensity of the image or as a change in the derivative of the intensity. These changes in intensity can be either:

- Step discontinuities. This is where the image intensity abruptly changes from one value on one side to a different value on the other side
- Line discontinuities. This is where the image intensity abruptly changes but returns to its starting value within a short distance.

In real images, abrupt changes rarely occur. Instead of these instant changes, real images change intensity over a finite distance [12].

One common way to measure local changes in an image's intensity is by defining the image arrays as a continuous function. From this, the gradient of the image can be used to measure local changes within the function. By using an approximation of the gradient, significant changes can be detected [12]. The gradient equation is defined as the vector 2.2:

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix} \tag{2.2}$$

For digital images, the derivatives in equation 2.2 are found from differences. The simplest approximation of said differences are:

$$\begin{aligned} G_x &\cong f[i, j+1] - f[i, j] \\ G_y &\cong f[i, j] - f[i+1, j] \end{aligned} \tag{2.3}$$

where j is the x direction and y is the negative y direction.

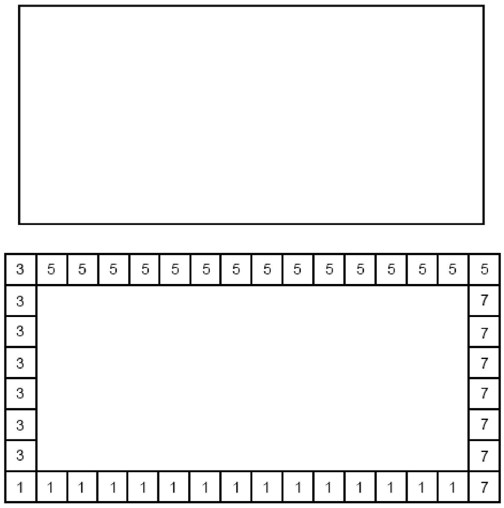
2.1.3 Contours

After finding edges in an image, they must be linked into a representation for a region boundary. This representation is called a contour. The system may use different algorithms for edge linking, such as the Hough transform and chain codes [12].

Chain codes specify the direction of a contour at each edge. These directions are represented as a value between 1 and 8, starting at the first edge and going clockwise around the contour [12]. Figure 2.31 shows how the chain codes are represented, while figure 2.32 shows an example of a contour and its chain code.

2	3	4
1		5
8	7	6

(1) Table of how the directions in a chain code are represented



(2) Example of a contour and its chain code

Figure 2.3: Chain code table and example

In Omron Ace, the user needs to define the boundary where the system will look for objects, as well as thresholds and other tuning parameters. The edge detection, liking, and contour creation all happen internally within the system.

2.2 3D Bin-picking

3D bin picking is a technique used to pick up objects scattered in a box. Here, the vision algorithms need to find each part's orientation and position, and then calculate a safe approach to the part to ensure no collisions in the robot's path. From here, the part is picked up and delivered to its assigned destination. A bin-picking operation can be split into 4 stages:

Scanning the bin.

In the first stage of the bin picking algorithm, a 3D sensor is used to measure the bin and create a point cloud of the objects in the bin. This point cloud can then be used to create a representation of the bin.

Object detection

The second stage used various techniques, such as CAD models of the objects or edge detection [3]. Using the CAD models of the objects in the bin makes it easy to convert the CAD file to a 3D point cloud. From here, the scanned point cloud and the object point cloud can be compared using different algorithms like the Random Sample Consensus (RANSAC) algorithm [6].

Path planning

Once the objects in the bin have been identified, a path planning algorithm is used to determine the optimal path for the robot to pick up each object. One way of determining the path is to create a 3D mesh of the gripper used. When an object is located, this gripper mesh can be positioned relative to the detected object's coordinate system at its pick pose. From this a collision measurement can be analyzed from the point cloud of the bin and the gripper mesh [3]

Object grasping

After all the algorithms are done, the detected object can be grasped. From here, the object is removed from the bin and placed in its desired position. It can be necessary to use collision avoidance when removing the part; in some applications, the same collision avoidance can be used as the one calculated in the path planning step.

2.3 Communication protocols

2.3.1 Ethernet/IP

Ethernet/IP is an application-layer communication protocol used in industrial automation applications for control and information exchange. It combines the Ethernet communication standard with the Control and Information Protocol (CIP), which is an upper-layer protocol used in industrial networks [13].

CIP

Every device in the CIP protocol is represented as a group of objects. Each object is a collection of associated data on a device. Three objects are required in the CIP protocol.

- The identity object. This object contains identifying information such as the vendor ID, serial numbers, and manufacturing date.
- Message router. The router routes requests among the objects on a device.
- Network object. This object holds communication data such as the IP address and other information about the interface [13].

The device can also contain application objects and vendor-specific objects.

Application objects

Application objects are the objects that define the data within a device. These items are specific to the type and function of the device [13]. An example can be an I/O module describing its signal types, signal variables, and resolutions. Application objects are usually predefined for most common devices, where all CIP devices need to contain the same series of application objects [13].

Vendor specific objects

These objects include additional information about the device's features. This information is determined by the vendor and may differ from one vendor to the next.

2.3.2 EtherCAT

EtherCAT is a protocol that is based on the Ethernet protocol. One of the main differences is that EtherCAT is designed for the I/O domain and focuses on:

- Broad applicability.
- Full conformity with the Ethernet standard.
- Smallest possible device granularity without using a sub-bus.
- Maximum efficiency.
- Short cycle times.
- Maximum deterministic properties.

With these focuses in mind, the EtherCAT protocol manages to process for instance 12 000 digital I/O signals in 350 microseconds, or 100 servo axes in 100 micro seconds [7]

EtherCAT is a master/slave architecture in which the EtherCAT master connects with several EtherCAT slaves via a single Ethernet cable. The EtherCAT master can be a controller or a computer, while the EtherCAT slaves can be sensors, actuators, drives, or other control devices.

The EtherCAT communication frame is made up of an Ethernet header followed by a series of data bytes that include control data, process data, and an error detection cyclic redundancy check (CRC). EtherCAT utilizes a unique "Telegram" concept in which the EtherCAT master sends a single data packet containing commands and data to all EtherCAT slaves, and each EtherCAT slave extracts the relevant data from the packet and adds its own data before forwarding it to the next slave, resulting in a linear topology, also known as a "daisy-chain" topology. An illustration of this topology can be seen in figure 2.4. This enables EtherCAT to achieve its high data transfer rate while reducing overall network load.

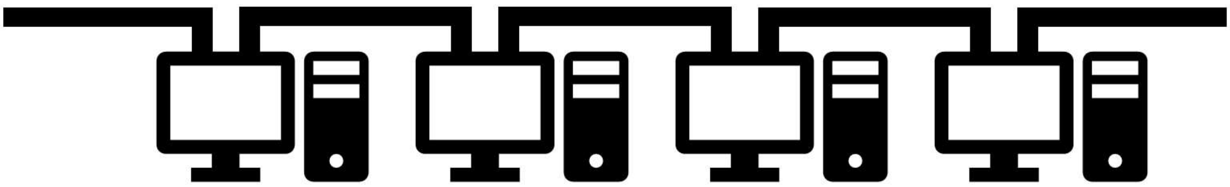


Figure 2.4: Four computes connected in a "daisy-chain" topology

2.4 Encoder

An encoder is a sensor used to measure the position, speed, or direction of motion of a mechanical component. The encoder generates electrical signals that determine the position or motion of the object being measured. Typically, these signals are created by rotating a disc and measuring the rotation. Some common methods of measuring the rotation are:

- Optical Encoder. This encoder typically utilizes LED lights and a light sensor to measure light pulses through the encoder disk.
- Magnetic Encoder. This encoder uses a magnet and a magnetic sensor to detect changes in the magnetic field as the disk rotates.
- Capacitive Encoder. This encoder changes the disk's capacitance and measures it to detect motion.

An incremental optical encoder's rotation disc and signals are depicted in figure 2.5

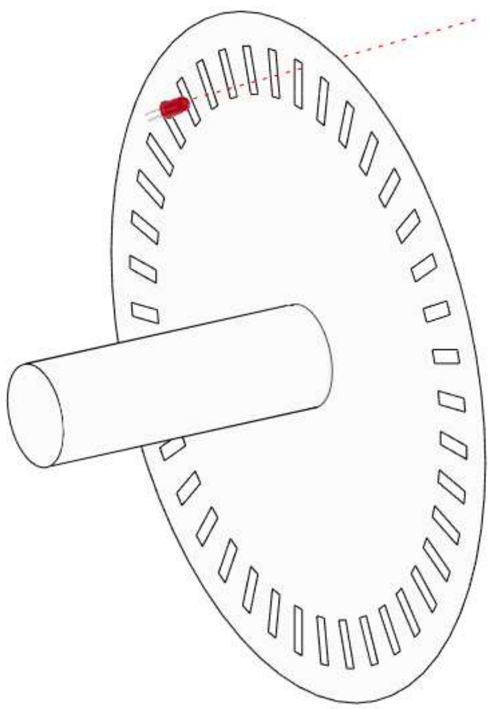


Figure 2.5: Incremental encoder disc with LED

To have a reference, the position of an incremental encoder must be reset between power cycles. The rotation components in this project did not have the capability of having such a reset point; hence, all encoders were absolute encoders. An absolute encoder uses a specific pattern, code, or sector on the encoder disc to represent the absolute position information. The pattern is designed in such a way that each unique position corresponds to a unique pattern, which can be read by sensors or detectors to determine the absolute position value. The disc of an absolute encoder is displayed in figure 2.6

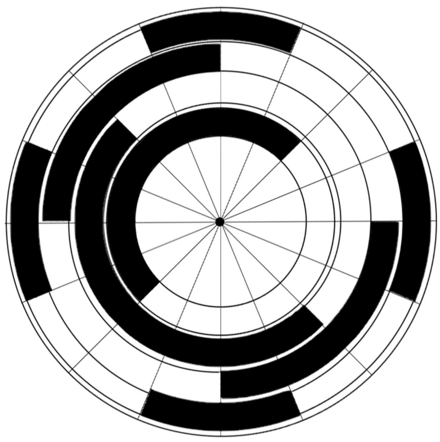


Figure 2.6: An absolute encoders disc

2.5 Jigs and fixtures in manufacturing

Jigs and fixtures are tools used to manufacture similar or identical components. They are tool-guiding or work-holding devices, which help with manufacturing a large quantity of parts by eliminating the need for a set-up procedure for every workpiece. The main difference between a jig and a fixture is that jigs can guide the tool to its precise position. The jig can also assist in the location and support of the workpiece [10].

Some advantages of using jigs and fixtures in a manufacturing process are:

- Improved accuracy. Jigs and fixtures are designed to ensure accurate positioning of work-pieces
- Increased productivity. By holding workpieces, they reduce the need for manual holding

and positioning, which reduces time when producing batches.

- Improved safety. By holding the workpieces, they reduce the risk of unstable components and also minimize the need for uneconomic work conditions.
- Cost-effective. Jogs and fixtures are cost effective by ensuring accurate and repeatable results

2.6 Automated screw fastening

Automated screw fastening is the method of automatically driving screws into workpieces with specialized equipment such as screwdrivers or screwdriving machines. The goal of automating this process is to improve efficiency, lower labor costs, and improve product quality. Automated screw festering requires precise torque control, screw depth, and alignment for reliable results [8]. Maintaining such high precision in automated screw fastening can be especially challenging due to variations in tolerances, material properties, and wear on equipment over time. With these factors, it is challenging to maintain good repeatability in the system. This could require frequent maintenance and calibration.
The screw fastening process can be divided into four main steps [8].

2.6.1 Feeding

Since screws usually come in boxes or bags, they need to be oriented and sorted before being fed to the screwdriver. Some methods used to feed screws are vibratory bowl feeders, tape feeders, pneumatic feeding, and magazines. Figure 2.7 shows different feeding methods.

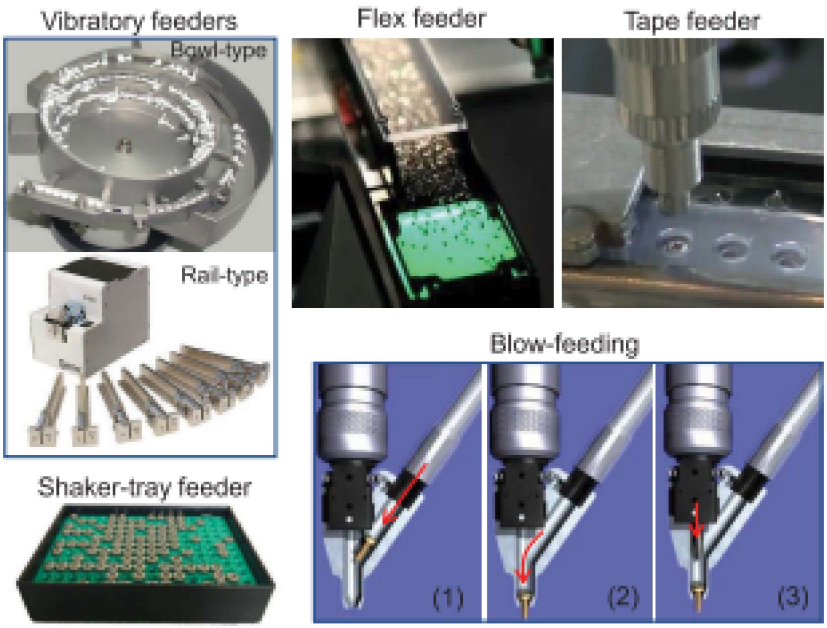


Figure 2.7: Different screw feeding methods [8]

In this project, the screws were oriented by a screw tumbler and fed to the screwdriver by a pneumatic feeder. In this method, a pneumatic signal is used to direct a screw from the bowl, through a tube, and into the screwdriver. With this method, the feeding system can be easily automated with a pneumatic valve and sensors.

2.6.2 Alignment

After acquiring the screw, the screw must be aligned with the part or parts it will join. This step is essential for the screw to be fastened successfully. Some common methods of aligning are jigs, force sensors, and machine vision. Jigs are commonly used when screws are inserted into identical components multiple times. A jig can place the component in the same orientation every time, so that the screwdriver goes to the same position every time. Force sensors can be used to provide real-time feedback so that the system can make adjustments to align the screw with the part. Machine vision can be used when the screw position varies to some extent between components. By using image processing algorithms, the system can compensate for these variations in position by finding the points on the part where the screw should be aligned.

2.6.3 Fastening

In this step, the screwdriver approaches the hole and starts rotating. The fastening process can be divided into four stages [14]. The different stages are:

1. Initial mating. Depending on whether the hole is pre-threaded or not, here is where the screw discovers the threads or begins threading it.

2. Run-down. In this stage, the screw is inserted into the hole until the head collides with the part.

3. Snug. This stage is the tightening between the screw and the workpiece.

4. Elastic clamping. In this stage, the slope of the torque-angle curve is constant.

5. Post yield. This stage is where plastic deformations occur. Some safety-critical applications need the screw to be tightened to this stage.

Figure 2.8 shows an example of a typical torque-angle curve.

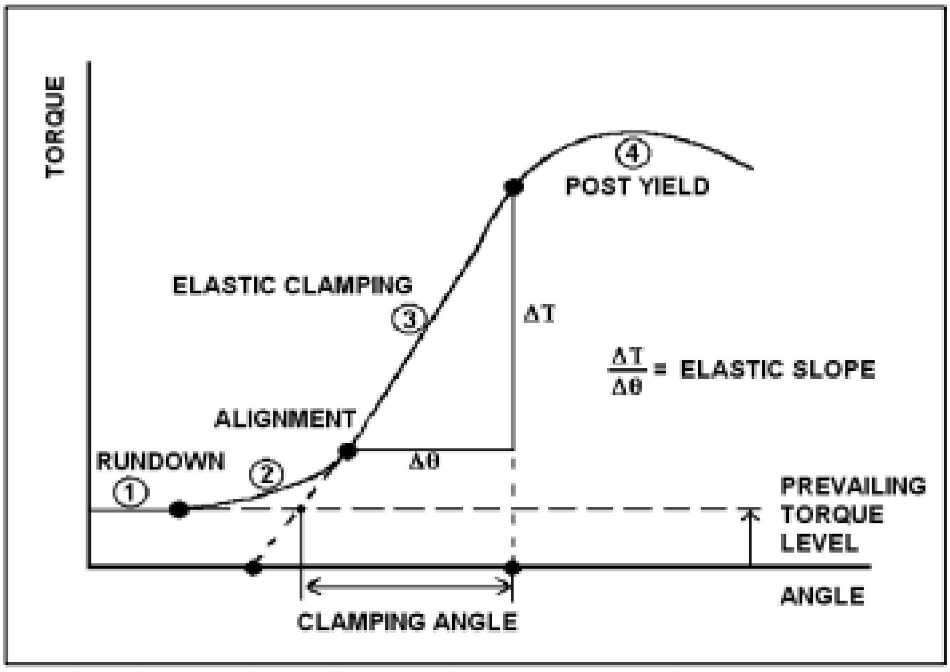


Figure 2.8: A typical torque-angle curve [14]

Monitoring the torque, and thus what stage of the fastening process the screw is currently in, is important to ensure no failures have occurred and the repeatability of the process. Multiple methods exist, but this project utilized the torque-only method. This approach controls and monitors only the driving torque and assumes that the related surfaces have known torsional stiffness [8].

2.6.4 Post fastening

This step moves the screwdriver back to its initial position, and from here the steps are repeated for the next screw.

2.7 Thread pitch compensation

Ensuring successful insertion of a threaded component in a machine or robotic system requires proper compensation for the thread pitch, which refers to the distance between each tread on the component. The tread pitch determines how far the component travels per rotation and is therefore a key parameter in achieving accurate and efficient insertion. Figure 2.9 shows a close-up of a bolt's thread pitch.

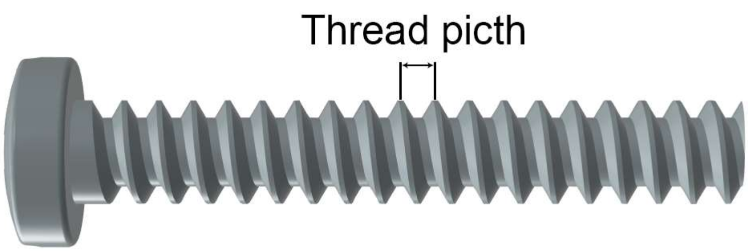


Figure 2.9: A bolts thread pitch, and how its measured

By compensating for the thread pitch, the machine or robot can accurately account for the component's movement along the thread axis during insertion. Failure to compensate for the pitch can result in the system moving at undesirable speeds. If the system moves too slowly, it can lead to delays and potential damage to the components. On the other hand, if the system moves too quickly, it can apply too much pressure to the components, causing damage.

Proper compensation for thread pitch offers several benefits, including increased accuracy, efficiency, robustness, and reliability. Accuracy is improved as the machine or robot can align the component precisely with its threaded counterpart. Efficiency is enhanced as the system can rapidly insert the component without delays or rework. Robustness is increased as the compensation accounts for variations in thread pitch due to manufacturing tolerances or wear. Reliability is improved as consistent and proper engagement of the component reduces the risk of loosening or vibrations after insertion.

One way of compensating for the components travel during insertion is by using the equations 2.4 for metric threads and 2.5 for imperial threads:

$$\frac{\text{Distance}}{\text{Rotation}} = \text{pitch} \tag{2.4}$$

$$\frac{\text{Distance}}{\text{Rotation}} = \frac{1}{\text{threads/inch}} \tag{2.5}$$

By adding the rotation speed of the component, the system knows how fast it needs to move to compensate for the insertion.

$$\frac{\text{Distance}}{\text{Seconds}} = \frac{\text{Rotation}}{\text{Seconds}} \cdot \text{pitch} \tag{2.6}$$

$$\frac{\text{Distance}}{\text{Seconds}} = \frac{\text{Rotation}}{\text{Seconds}} \cdot \frac{1}{\text{threads/inch}} \tag{2.7}$$

2.8 Vacuum ejector

A vacuum ejector is a component that creates a vacuum when receiving a pneumatic signal by utilizing the Venturi principle [4], which is the reduction of fluid pressure that occurs when a fluid flows through a constricted section of a pipe. The vacuum ejector consists of an inlet nozzle, a Venturi tube, and a muffler/diffuser. The compressed air or steam is passed through the nozzle at a high velocity, which creates a low-pressure area in the Venturi tube. This low-pressure area causes the process medium, usually air or gas, to be drawn in through an inlet port and mixed with the high-speed jet of compressed air or steam. The combined medium then travels through the Venturi tube and into the diffuser, where the velocity decreases and the pressure increases. As a result, a vacuum is created at the inlet port, and the fluid is sucked into the system [5]. Figure 2.10 shows an example of a vacuum ejector's principle.

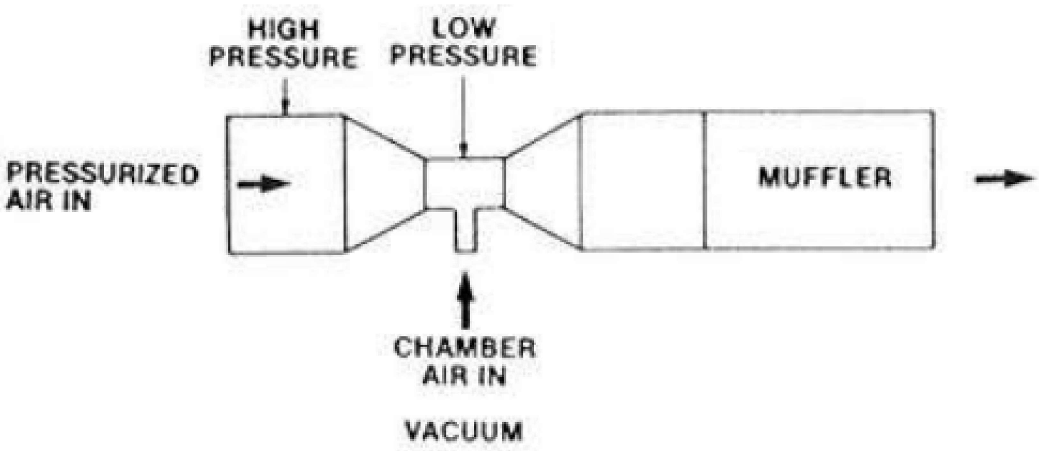


Figure 2.10: Vacuum ejector using the Venturi principle [5]

2.9 LEAN manufacturing

LEAN manufacturing is a methodology aimed at optimizing the flow of production by eliminating waste and maximizing value for the customer. The different type of wastes in an production are categorized as overproduction, waiting, transportation, non-value-added-processing, excess inventory, defects, excess motion, and underutilized people [9].

The LEAN manufacturing thought process utilizes different methods to reduce or eliminate these wastes. Some examples of these methods are pull production and Kanban. Pull production is a method where production is based on customer demand, where as Kanban is a technique used to organize a stations material flow [9].

Chapter 3

Software & Hardware

3.1 Software

3.1.1 Fusion 360

Autodesk Inc.'s Fusion 360 is a cloud-based 3D CAD/CAM program. It is used in product design, engineering, and manufacturing to build complex 3D models, renderings, and animations. Fusion 360 also includes simulation, documentation, and collaboration functions, making it an all-in-one solution for the whole design and engineering process. Fusion 360's ability to merge mechanical, electrical, and electronic design into a single, unified environment is one of its distinguishing advantages. As a result, it's an excellent tool for designing products with both mechanical and electronic components, such as consumer electronics, robots, and industrial machinery [2].

3.1.2 Siemens NX

Siemens NX is a CAD/CAM/CAE software developed by Siemens PLM Software. It is used by engineers, designers, and manufacturers across a wide range of industries, including aerospace, automotive, and industrial machinery. NX is known for its advanced capabilities in design, simulation, and manufacturing, including 3D modeling, drafting, assembly modeling, and parametric modeling.

One of the advantages of NX is its ability to handle large and complex designs, allowing users to work with very large assemblies and perform complex simulations. NX also includes a range of advanced tools for product development, such as digital twin technology, which allows users to create a virtual representation of a product and simulate its behavior in real-world conditions [15].

3.1.3 Omron ACE 4.0

Omron ACE is a software package used for programming and configuring Omron PLCs, robot controllers, and vision controllers [11]. Users can use ACE to create and manage controller programs, configure devices, and execute a variety of testing and troubleshooting tasks. The software contains capabilities such as automatic code generation, online debugging, and remote monitoring that serve to streamline the automation process. Both robots used in this project are programmed in ACE 4.0.

3.1.4 Sysmac Studio

Sysmac Studio is another software developed bu Omron, and is also used for programming and configuration. Sysmac is designed for Omrons PLCs, industrial PCs, and other automation controllers. It supports various programming languages used in PLCs, including structured text, function blocks, and ladder logic. Sysmac allows for different applications such as motion, logic sequencing, safety, drives, vision and HMIs to be configured in the same platform. In this project Sysmac Studio is used to program the main PLC, I/O-modules, frequency inverters, servo drives, and most of the other electrical components. All communication between components is also programmed in Sysmac.

3.2 Hardware

3.2.1 PLC

The majority of the project's components interact with one another via a PLC, which is an Omron NX1 Modular CPU. The PLC can control 4 servos, 2 Ethernet connections, and a total of 64 EtherCAT nodes. It is programmed using Sysmac Studio.

3.2.2 EtherCAT I/O Module

The robotic cell is placed far away from the PLC. Therefore, I/O ports are moved into the cell using EtherCAT modules, simplifying and organizing wiring. Omrons NX-ECC203 I/O modules are the type of EtherCAT modules used.

3.2.3 Robotic Arms

In this project, Omron Adept Viper 850 robotic arms were used. The robots are 6-DOF arms that are capable of handling all project-related movement, insertion, and assembly activities. They feature an 855 mm reach and a 5 kilogram payload capacity. For the boxes to be consistently assembled properly, their 0.03 mm repeatability is essential.

3.2.4 Robot Controller

Omron's eMotrionBlox 60R, which utilizes the eV+ operating system, is the robot controller in use. The movements of the robot are programmed and controlled by the controller using Omron ACE.

3.2.5 Servo Motor

For operations requiring the insertion of screws or threads, two Omron 1S servo motors were employed. The motors have a 3000 rpm rated speed and a 0.637 Nm rated torque at 200 W of power. A built-in absolute encoder that measures the position of the motor shaft is also present in the motor.

3.2.6 Servo Drive

Two Omron 1S servo drivers are utilized to control the servo motors. They both have a rated output of 200W.

3.2.7 Vision Controller

The vision controller used is a Omron SmartVision MX. The machine vision algorithms in the system are programmed and processed using this controller

3.2.8 Frequency Inverter

There are two conveyors in the robotic cell. Omron MX2 frequency inverters are used to control the motors in each conveyor.

3.2.9 Sensors

To determine whether parts are present, some system operations require sensors. There are three sensors used in this project. The first is an inductive sensor made by Omron (E2A-S), the second is an ultrasonic sensor made by Takex (US-S25AN), and the third is a photoelectric sensor made by Omron (E3C).

3.2.10 Pneumatic Linear Actuator

To assemble parts, the robots require grippers to grasp each part individually. Two SMC MHF2 parallel linear pneumatic actuators were utilized to achieve this. The actuators have a repeatability of ± 0.05 mm and a maximum gripping force of 141 N at 0.5 MPa.

3.2.11 Pneumatic Piston

There are two Festo DSNU-20-150-P-A double-acting pneumatic pistons used in the project. The pistons can function between 1 and 10 bar of pressure and have a 150 mm long stroke. They were utilized throughout the project to push and/or hold various elements in position.

3.2.12 Pneumatic 5/2 Valve

The system's various pneumatic components were controlled by 5/2 valves. The two valves were Festo solenoid valves, model numbers VUVG-L10-M52-MT-M5-1R8L.

Chapter 4

Method

4.1 Project Organisation

4.1.1 Previous experience

The group's two members have both finished apprenticeships to become automation technicians, giving some background knowledge of automated systems. The group members are familiar with Sysmac Studio and Omron ACE and have previously used the robotic cell in the Industry 4.0 course. Additionally, one of the group's members have taken a CAD design course.

4.1.2 Distributed work and responsibilities

The project group's workload and responsibilities were divided as follows:

Benjamin

- Project leader
- Programming robots

Fridtjof

- Secretary
- Programming PLC

The remaining workload was shared between both members.

4.1.3 Meetings with supervisors

The project team's supervisors and members were supposed to meet once every two weeks initially, but the meeting schedule had to be somewhat modified as some weeks required little to no follow-up or questioning while other weeks required more. The meetings mostly focused on what had been accomplished, what to concentrate on, and how those objectives could be achieved.

4.2 Process overview

The system was made to be able to merge the assembly and packaging process into one. This systems process chart can be seen in figure 4.1

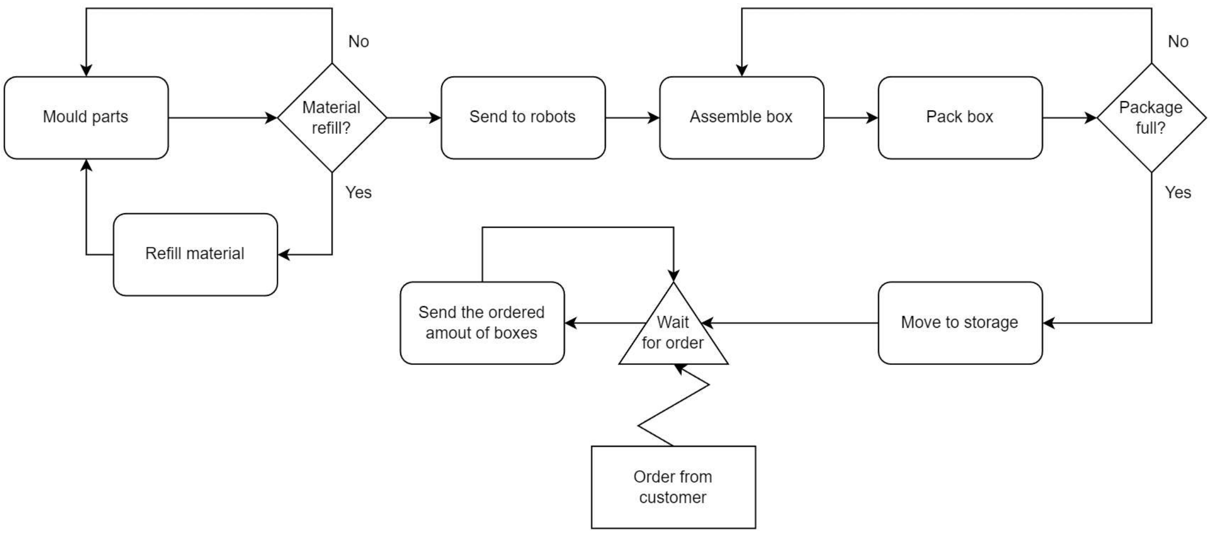


Figure 4.1: Chart of a process using this projects system

In this process chart the assembly process is represented as a single action. In reality the task involves multiple components and steps.

4.3 Design of components

Each part and component was designed using numerous design processes to explore various approaches to problem solutions. When designing parts for a semi-modular robotic cell, all of the components had to be developed around the cell and depend on the equipment provided by the cell. Since 3D printing and laser cutting were the only available manufacturing methods, the designs in this project had to be built around those methods.

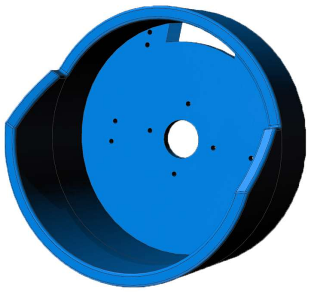
4.3.1 Design process

During the design process, all components were created with their specific application in mind. Initially, magazines were created to feed all components into the system. After the magazines were implemented, the grippers were designed, and docks were built to switch between grippers. When the system was capable of moving the components, jigs and fixtures were created to hold the components during the assembly procedures. Both Fusion 360 and Siemens NX were used in the design process as they were CAD software that the group were familiar with. In these software, elements could be simulated/animated to see if the design was sufficient. The following subsections go into detail about how each component of the system was designed.

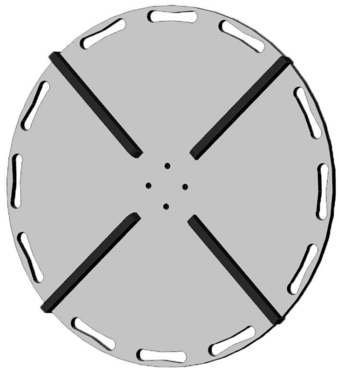
4.3.2 Tumbler

The Tumbler consists of four main parts as shown in figure 4.3. The assembly is shown in figure 4.2.

Figure 4.2: Tumbler assembly

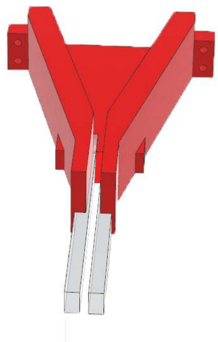


(1) Drum



(2) Rotary Plate

(3) Motor/Motor-mount



(4) ramp

Figure 4.3: Tumbler main parts

The rotary plate is housed inside the drum and is attached to the motor through the drum's center. The holes in the plate picks up the screws and carry them up to the top, where they fall out onto the ramp. For the feeding system to function properly, the ramp ensures that the screws hang from the head with their threads facing downward.

The plate design was made with rapid feeding in mind so that the tumbler would not "fall behind" and be the weakest link in the feeding process. It is equipped with 12 slots that carries the screws to the top where they fall into the ramp. In addition to the slots in the plate four "fingers" were added for continuous movement of all the screws.

The ramp orients the screws in the right direction before the feeder. When the screws drop out of the drum, they fall down into a chute where the head is guided by two rails while the threads fall downwards as shown in figure 4.4.

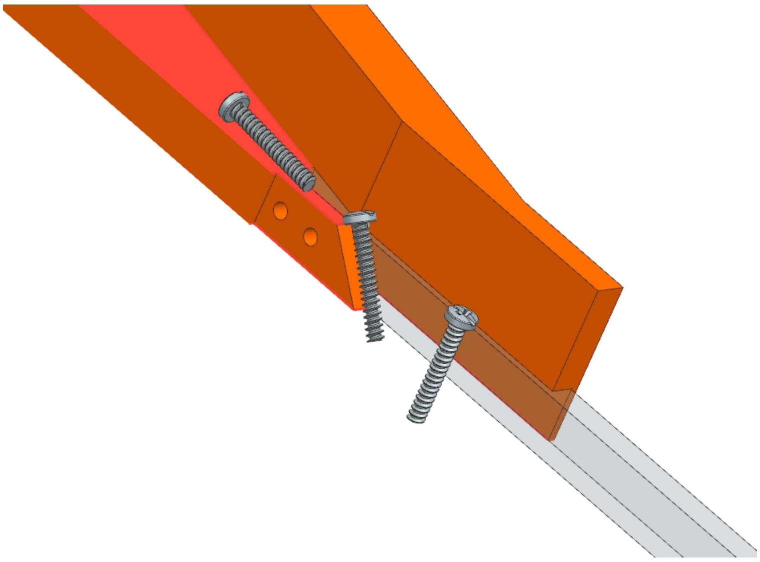


Figure 4.4: Ramp-Slide

4.3.3 Feeder

After the process of sorting and orienting the screws, they are fed into the screw driving unit when requested. To achieve this a feeder or hopper was placed between the tumbler and screw driving unit. The feeder is a simple cylinder with 2 holes on each side to pick up one screw at a time from the ramp and drop them in the pneumatic system that transports the screw with pressurized air. The feeder is shown in figure 4.5

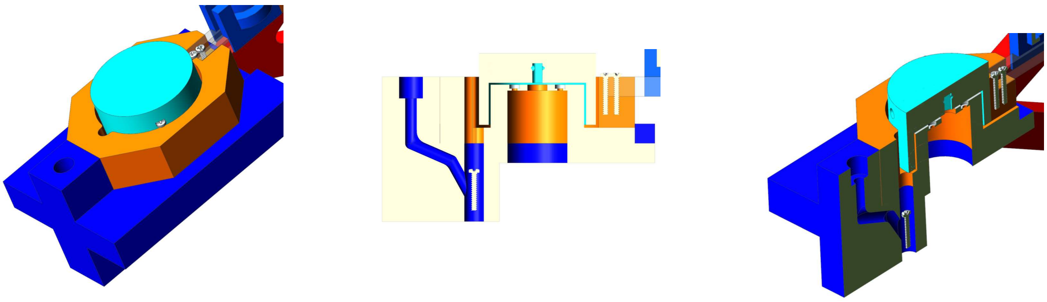


Figure 4.5: Feeder-Assembly

4.3.4 Screw driving unit

The screw driving unit is a key component to insert screws into the adjustment ring. It is based on designs used in the industry today, altered with a few iterations for a tailored task. The first prototype was made from a sketch on a whiteboard as shown in figure 4.6.

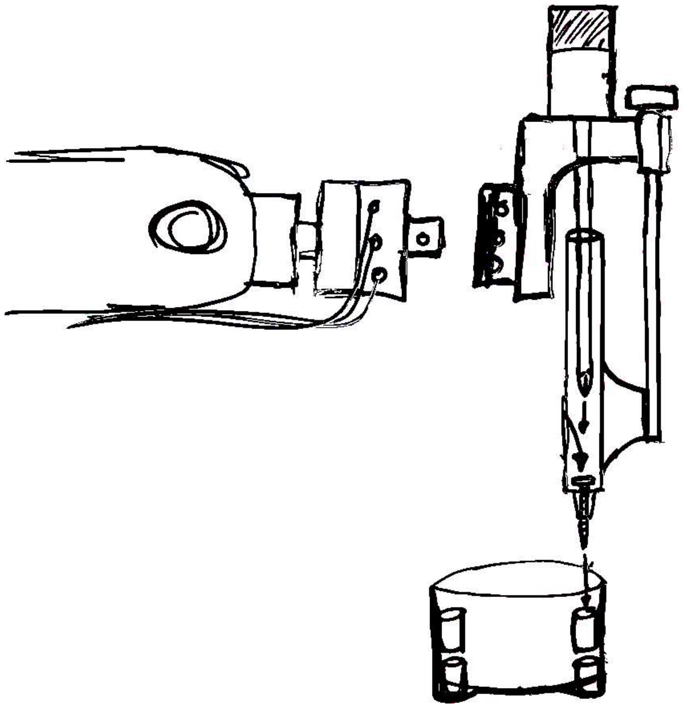


Figure 4.6: Whiteboard sketch of the screw driving unit

The main function of seating a screw into a "holder" with pressurized air and descending the screwdriver downwards until it engages the screw is more or less unchanged since the first prototype, with the only addition of a spring to make the slider return to the top. The screw collector/seat on the other hand has gone through many iterations due to changes in the setup of the magazines and how the robot engages the adjustment ring. The screw driving unit works by using a Omron motor to rotate the screwdriver with a precise amount of given rpm and torque. When the robot moves down to the adjustment ring, it presses the bottom of the screw collector onto it which proceeds to slide the Igus linear glide upwards. This makes the screwdriver move downwards, while the screw collector and the screw sits at a fixed position. Figure 4.7 shows the current screw driving unit.

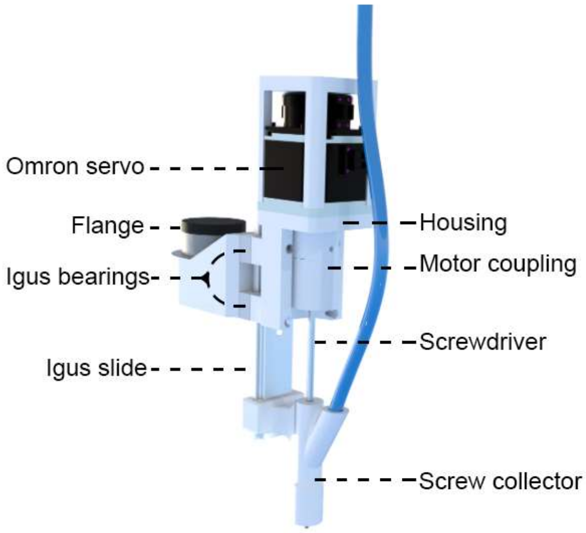
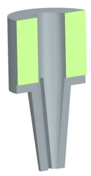


Figure 4.7: Screw driving unit

The screw driving unit consists of 7 parts. The first and most important part is the Omron servo that is the "heart" of the entire assembly. It is mounted to the housing which holds the entire unit together. On the side of the housing, four Igus sliding bearings are mounted so that the linear slide can slide straight up and down with little friction. The robot flange is mounted on the outside of the Igus bearings to add stiffness and a central position of the robot tip. The screw collector was designed in stages starting with the tips shown in figure 4.8

(1) Tip 1

(2) Tip 2



(3) Tip 3

Figure 4.8: Tip prototypes

Tip 1 shown in figure 4.81 was the first prototype made to hold the head of the screw. In this version the "lip" inside could sometimes jam with the screw threads, preventing the screw from exiting the tip correctly.This lead to the second design in figure 4.82 which was made to guide the screw more accurately "through" the fingers where they would catch the head of the screw. The downside with this design proved to be the stiffness of the fingers which lead to excessive resistance when pushing the screw out. The fingers would latch on to the screw too hard and snap. To eliminate this problem the last prototype was designed to have flexible fingers while still holding on to the screw up until the point of expected release by using less material for all fingers as shown in figure 4.83.

After designing the tip, the screw collector was introduced. In total there were 24 iterations of the screw collector, where twenty of the them had minor adjustments to the tip distancing as well as minor alterations to the tube fitting. The design changed more significantly in the remaining four iterations, which are detailed below.

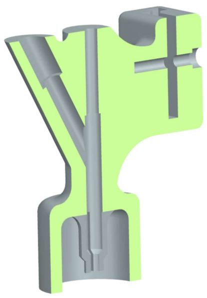
- Version 1

(1) Version 1 split (2) Version 1

Figure 4.9: Screw Collector v1

The first version shown in figure 4.9 started with a main objective of catching the screw and holding it until the screwdriver threaded the screw into the adjustment ring. This worked as a concept but did not work consistently and could only insert 2-3 screws in a row before one did not enter the hole correctly. This version worked by centering the four finger tip over the hole and pressing down until the screw was down to the correct height.

- Version 2



(1) Version 2 split

(2) Version 2

Figure 4.10: Screw Collector v2

The second version included some smaller changes and the introduction of "the cup" shown in figure 4.10. The four fingers on the tip were changed to better encounter the screw head with a "lip" to catch it and still let the threads effortlessly slide through. All this while holding the screw securely in place until the screwdriver "drives" it out with little resistance. With a shorter height of the entire part the screwdriver was able to press the screw further past the end of the tip. The largest change was the cup that surrounds the tip. This was introduced to make the insertion of the screw more accurate and put even pressure on top of the ring to prevent it from shifting or flip.

- Version 3

(1) Version 3 split (2) Version 3

Figure 4.11: Screw Collector v3

Version 3, shown in figure 4.11, was made to clamp the ring inside the cup and hold it in place until the screw was in place. This was a further improvement of the "cup" since the last version did not have enough friction on the ring to hold it from rotating when the torque from the screwdriver increased proportionally with the depth of the screw. The cup could hold the ring from either sides and did not depend on the rotation of the ring.

- Version 4

(1) Version 4 split (2) Version 4

Figure 4.12: Screw Collector v4

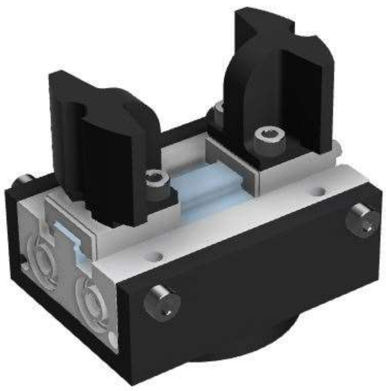
Version 4, shown in figure 4.12, was made to work with a jig that the adjustment ring would be placed in before inserting the screws. The cup was made smaller and the four finger tip is able to be swapped out in case any of the fingers where to snap off, caused by feeding multiple screws at once. The four finger tip's threads are left-handed to counter the rotation from the screw driver.

4.3.5 Linear-Grippers

The two linear grippers were designed to relocate and assemble parts during the assembly process. Viper 1's linear gripper moves the adjustment rings from a conveyor to a jig, where screws are inserted, and finally into a Kanban. Viper 2's linear gripper picks all different components and either places them in a jig or assembles them with the previous part.

Viper 1's linear gripper

This gripper is built around a pneumatic linear actuator and contains a mount between the robot flange and the actuator, as well as two fingers for picking up the adjustment ring. The mount is attached to the actuator by four screws on the side, and the fingers are attached with two screws on each finger. Figure 4.131 shows the gripper, while figure 4.132 shows the gripper grasping a ring.



(1) Viper 1's linear gripper

(2) Linear gripper fetching a ring

Figure 4.13: Viper 1's linear gripper

Viper 2's linear gripper

Just as Viper1's gripper this gripper was built around the pneumatic linear actuator. The goal for this gripper was to assemble the four main parts of the box which consists of the main box, top cover, adjustment ring and lid. To achieve this the gripper was designed to pick up all of the parts and place them with the necessary support to press the parts in place. Figure 4.14 shows the gripper.

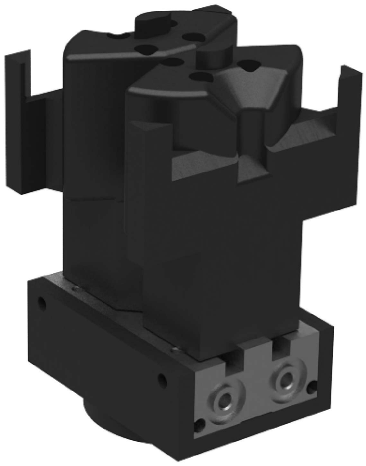


Figure 4.14: Viper2's Linear Gripper

4.3.6 Rotary Gripper

The adjustment ring must be put into the top cover during the assembling process by rotating the ring approximately nine times. If the robot rotates the ring with its linear gripper, the constraints of its sixth joint would require the robot to release the ring, rotate back, and continue the insertion. This procedure was deemed too time consuming, so a revolving gripper was developed to make it more efficient. This gripper achieves rotational movement with a 200W, 3600 rpm servo motor with an internal absolute encoder. The gripper is made up of a frame that connects the robot flange to the motor, and a head that is mounted on the motor shaft. The head was designed to fit inside the lid, allowing the insertion of the adjustment ring to be the final step in the assembling process. This reduced the number of gripper changes required by the robot. Figure 4.15 shows the gripper.

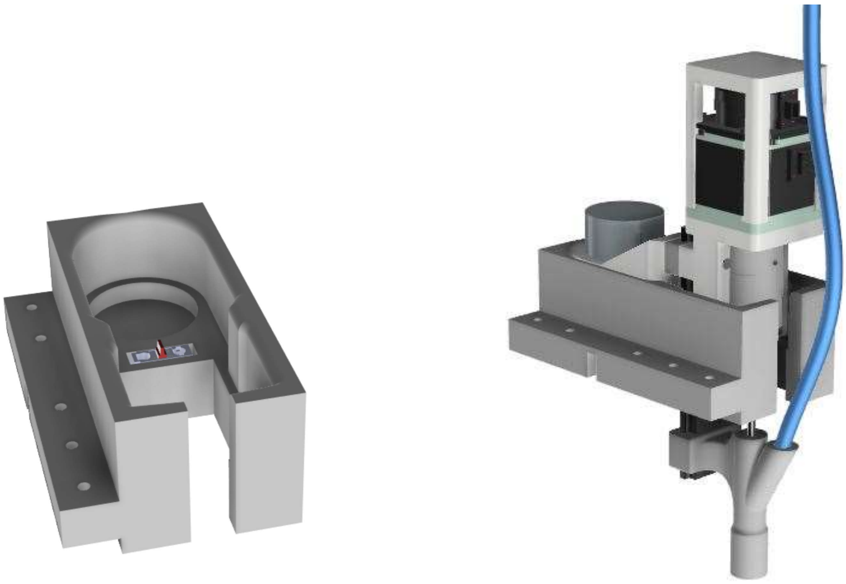
Figure 4.15: Rotating gripper

4.3.7 Tool Docks

Docking stations were implemented to store tools that were not in use, allowing robots to switch between tools. Both robots have their own docks, with Viper 1 having docks for its linear gripper and screw driving unit and Viper 2 having docks for its linear gripper and rotary gripper. The docks were positioned near relevant jigs to allow for a quick tool change.

Screw driving unit dock

The screw driving unit dock was designed to allow the tip and tube to hang freely while the dock supports the units body. As a result, the unit was free to rotate the screwdriver while docked. A switch was added to the dock to guarantee that the screw driving unit is correctly placed in the dock. When the unit is successfully docked it presses the switch, delivering a signal to the PLC. Figure 4.161 shows the screw driving unit dock with the end switch inserted, and figure 4.162 shows the dock with the unit inside.

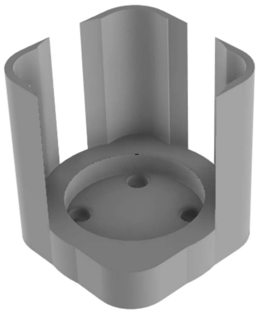


(1) Screw driving unit dock (2) Dock with screw driving unit

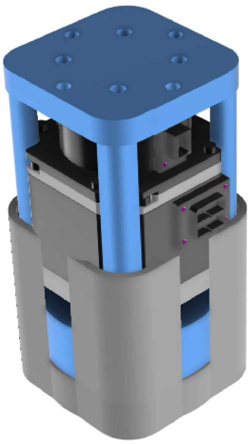
Figure 4.16: Screw driving unit dock

Rotary gripper dock

The dock for the rotary gripper was designed so that the gripper could freely revolve when docked. This allowed the gripper to return back to its home position before being put on the robot. The dock was designed to hold the gripper's sides while also providing space for the motor and encoder cables on each side. The dock is depicted in figure 4.171. In this figure, one side is made invisible to show the dock's internal geometries. The dock is shown in figure 4.172 with the rotating gripper inserted.



(1) Rotary dock

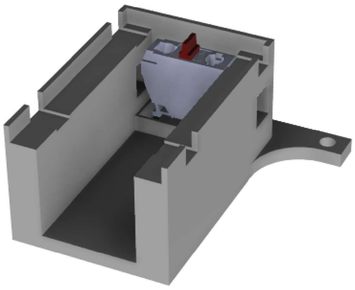


(2) Rotary dock with gripper

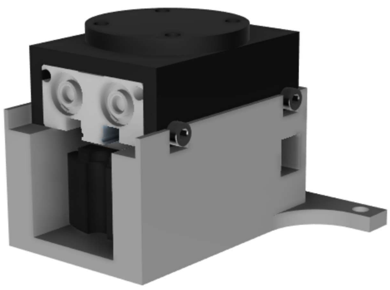
Figure 4.17: Rotary gripper dock

Linear gripper docks

The dock for Viper 1's gripper, like the screw driving unit dock, uses a switch to ensure a successful docking. To ensure that the gripper sits tightly in the dock, the dock has slots that fit the screws on the sides of the gripper. Viper 1's linear gripper dock is shown in figure 4.181, and in figure 4.182 where the gripper is inserted into the dock.



(1) Linear dock for Viper 1



(2) Linear dock for Viper 1 with gripper

Figure 4.18: Viper 1's linear gripper dock

The dock for Viper2's gripper is made to house the gripper tightly to ensure that the robot can pick from the same position each time. The dock is shown in figure 4.19.

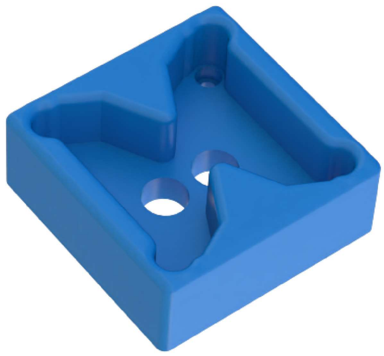


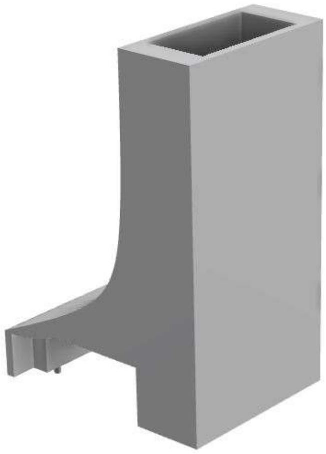
Figure 4.19: Viper2-Gripper Dock

4.3.8 Magazines

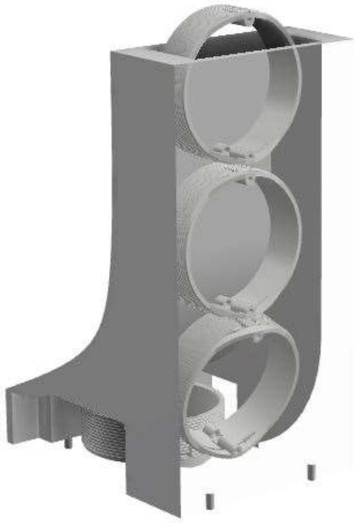
Magazines were used to allow the system to be fed several components without the need for manual input. Three magazines were designed in total.

Adjustment ring magazine

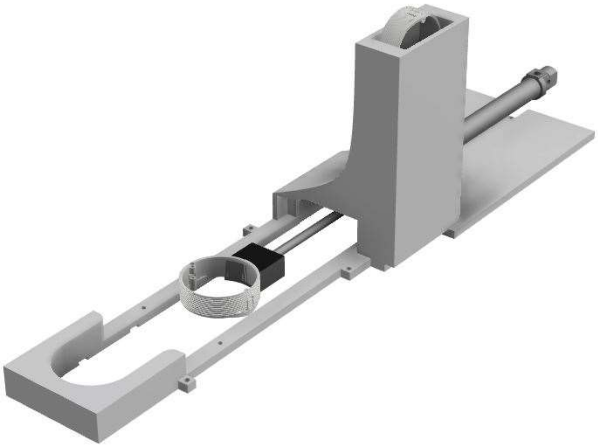
The first magazine stores the adjustment rings before they are fed to Viper 1. Multiple magazines were designed and two of the designs were tested . In the first design depicted in figure 4.201 the rings are stacked on top of each other vertically. In the bottom of the magazine the rings slides on a ramp that orients them horizontally. From here the bottom ring is pushed into the robots reach by a pneumatic piston. Figure 4.202 shows how the rings are stacked inside the magazine, and figure 4.203 shows a ring being pushed out of the magazine.



(1) First design of adjustment ring magazine



(2) First design with rings inside



(3) First magazine assembly

Figure 4.20: First design of adjustment ring magazine

This design had both advantages and disadvantages. Its advantage was the ability of its guides along the piston stroke to precisely guide the adjustment rings into the robot's reach. This meant that the ring could be located within a smaller search area. The magazine's drawbacks were that the guides frequently became stuck to the ring when the robot picked it up, that the magazine had a maximum capacity of four rings, and that the slope inside the magazine was too steep resulting in rings regularly becoming stuck. As a result of the drawbacks, the design was dismissed and version two was created.

Version two is designed in the shape of a ramp, allowing the rings to slide on a guide down onto a conveyor. The rings are moved by this conveyor to a latch sensor where they are picked up by the robot. Since this magazine used the angle of the ramp to move the rings onto the conveyor, the pneumatic piston was no longer required, allowing the magazine to be larger and fit a total of 10 rings. Figure 4.21 shows the final design of the magazine.

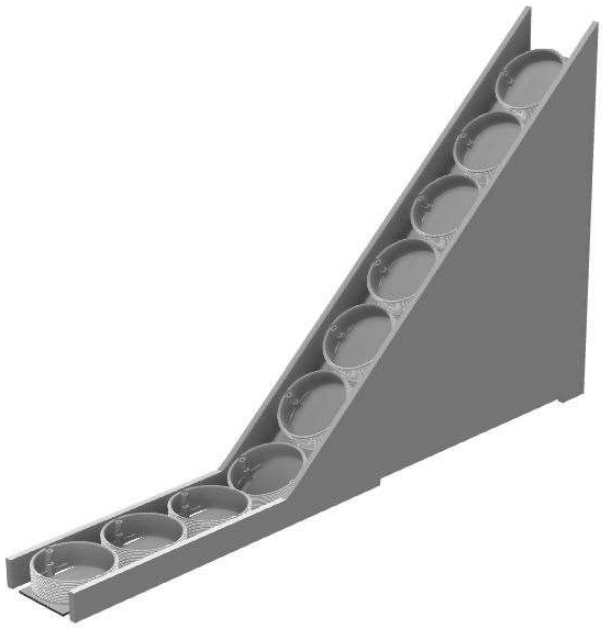


Figure 4.21: Final design of adjustment ring magazine

Adjustment ring Kanban

After the screws have been fitted into the adjustment ring, it is placed in a Kanban. This Kanban is based on the final design of the adjustment ring magazine. On the side of the Kanban is an ultrasonic sensor that signals when the Kanban is full. Viper 1 only inserts screws into adjustment rings when the Kanban requires extra parts. This was done in order to achieve a pull production. The bottom plate of the Kanban was made of see-through acrylic so that the ring could be detected more easily using machine vision. Figure 4.22 depicts the Kanban.

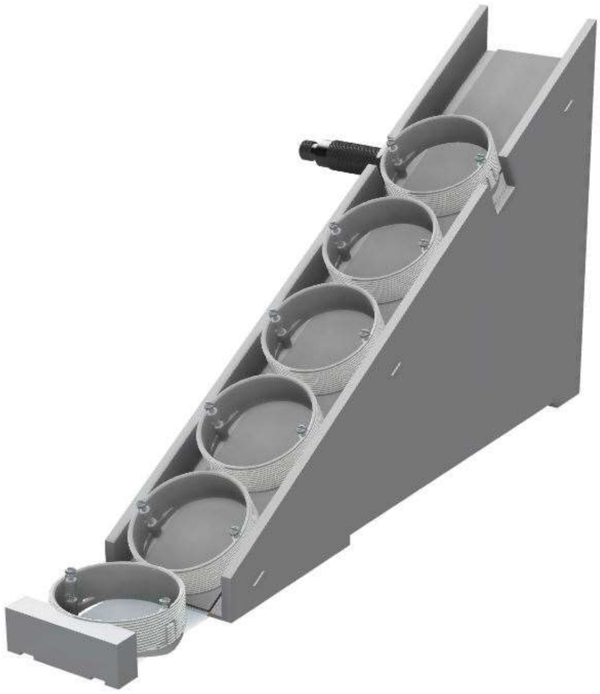
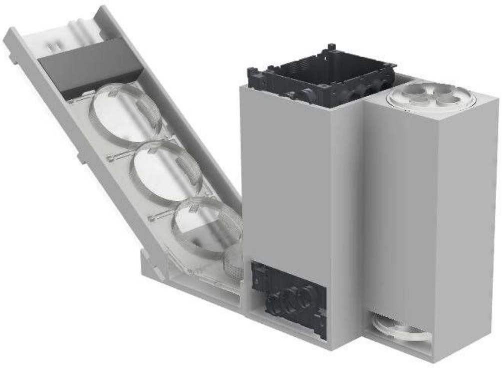


Figure 4.22: Kanban with rings and sensor

Triple magazine

To present the main box, top cover, and lid in a simple and efficient way, a magazine was created to push all parts into the machine vision window at the same time using a single pneumatic piston. This magazine is essentially an assembly of three magazines and a common pushing mechanism. In both the magazines for the main box and the lid the components are stacked on top of each other, while in the magazine for the top cover the parts are placed in a ramp. Initially the top covers were also stacked on top of each other, but due to their geometry they got stuck in each other. When refilling the magazine, a weight is placed behind the last top cover to ensure they are represented in a consistent manner. Figure 4.231 displays the triple magazine, while figure 4.232 displays the magazine with the piston and pushing mechanism.



(1) Triple magazine front

(2) Triple magazine back, with pushing mechanism

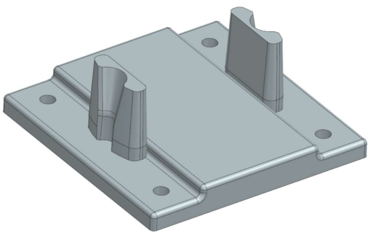
Figure 4.23: Triple magazine

4.3.9 Jigs/fixtures

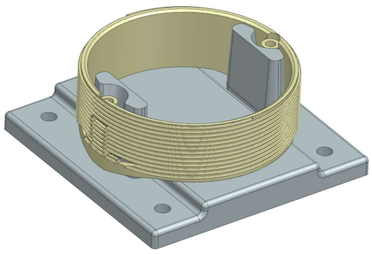
There are 3 types of jigs in this project:

1. Adjustment ring jig

   This jig is designed to hold the ring to make the screw task both easier and faster for the robot. It will correct for some rotation on the ring when the robot inserts it and then holds the ring in a fixed orientation. The jig is depicted in figure 4.241, and in figure 4.242 where a ring is inserted.



(1) Adjustment ring jig

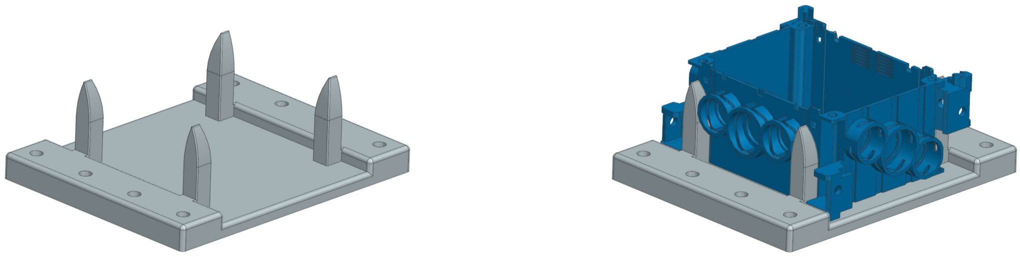


(2) Adjustment ring jig with ring

Figure 4.24: Adjustment ring jig

2. Assembly jig

The jig is designed to hold the main box until all the other parts are mounted to it. When the robot is inserting the box, the jig will correct for some misalignment in the x- and y-plane and hold the box in a fixed position until the process is done. The jig is depicted in figure 4.251, and in figure 4.252 where a main box is inserted.



(1) Assembly jig (2) Assembly jig with box

Figure 4.25: Assembly jig

A plexi plate with mounting-holes was fixed to the top of both conveyor belts in the cell to position the jigs close to the robot while not interfering with the belts on the conveyors. The plate has a template of holes in it to make it easy to fasten other fixtures or jigs in the future if needed. The plate with the assembly jig mounted is shown in figure 4.26.

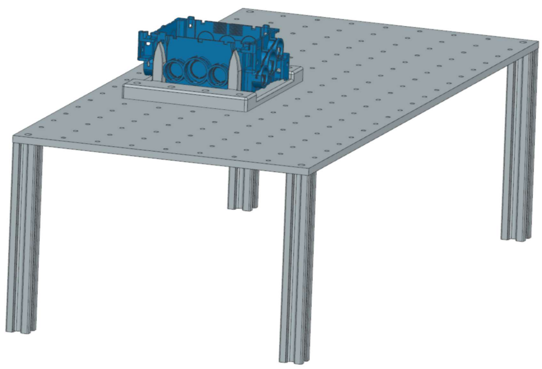
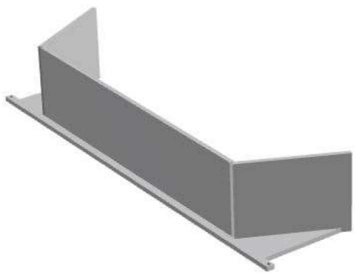


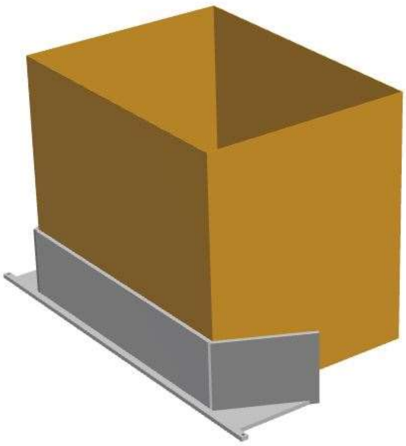
Figure 4.26: Mounting Plate

3. Cardboard box jig

   The final jig was designed to guide the cardboard box that holds the finished products into the same position every time. This jig is placed on top of Viper 2's conveyor and uses the movement of the conveyor to guide the cardboard box. Figure 4.271 and 4.272 shows the jig and the jig with a cardboard box inserted.



(1) Cardboard box jig



(2) Jig with cardboard box inserted

Figure 4.27: Cardboard box jig

4.4 Programs

In total there are three programs distributed across both robots and the PLC. The robots were programmed with Omron ACE, whereas Sysmac Studio was used to program the PLC. The methods used to accomplish the assembly goals are described in the following subsections, together with the pseudo-codes for each program. Since the pseudo codes are simplified versions of the real programs, they do not contain all the specifics in every step.

4.4.1 Viper 1

Algorithm 1 Viper 1 pseudo code

```
Move to safe position
if Tool_attached then
    Dock attached tool
end if
if Both_tools_docked = False then
    STOP
end if
Attach linear gripper
while (run = True) do
    Drive conveyor
    while Latchsensor = False do
        Wait
    end while
    Feed first screw to screw driving unit
    Pick adjustment ring
    Place adjustment ring
    Dock linear gripper
    Attach screw driving unit
    Insert first screw
    Feed second screw to screw driving unit
    Insert second screw
    Dock screw driving unit
    Attach linear gripper
    Pick adjustment ring
    Deliver adjustment ring
end while
```

Communication

In steps 2, 5, 8, 9, 11, 14, 17, 18, 20, 22, 23 and 25 in Viper 1's pseudo code the robot sends signals to or receives signals from the main PLC via an Ethernet/IP connection.

In the docking and attaching steps the signals are sent from the master PLC and are used to check if the tools are inside their docks. If one of the signals from the docks is in an unexpected state, for instance if the gripper dock switch is false after the robot has docked the gripper, then the program stops to assure that no crashes occur.

In step 9 the robot gets its start signal from the PLC. This signal was implemented to assure that the communication is running when the robot starts the main loop. If the communication should fall, then the program will exit the main loop causing the robot to stop.

In step 11 the PLC sends a signal based on a latch sensor located on the conveyor. This way the robot program will only proceed when a ring is located inside the machine vision area.

In step 14 and 20 a signal is sent from the robot to the PLC. This signal tells the PLC to feed one screw to the screw driving unit. The first signal is placed right after a ring is detected by the latch sensor. This way the first screw is ready when the screw driving unit is attached, removing the need to wait for the screw. The second screw can only be fed after the first has been inserted, therefore a short wait function is hidden inside step 21.

Machine vision

For the robot to be able to pick an adjustment ring from the conveyor a camera is used to take an image of the ring. By training an object location model in Omron ACE, the program can locate the position and orientation of the ring. This information is then fed to the robot to pick the ring. Figure 4.28 shows the locator model.

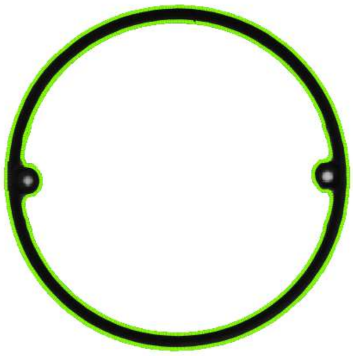


Figure 4.28: Locator model adjustment ring

Screw insertion

For the robot to be able to insert a screw successfully, the rotational speed of the screw driving unit and the speed of the robot needs to match how far the screw is inserted each rotation. By using equation 2.4 the distance the robot needs to travel per rotation can be calculated by inserting the screws thread pitch of 1mm.

$$\text{Distance} = \text{Rotation} * 1\text{mm} \tag{4.1}$$

Equation 4.1 shows that the robot needs to move 1mm down to compensate for one rotation of the screwdriver. Using equation 2.6 the travel speed can be calculated from the motors rpm.

4.4.2 Viper 2

Algorithm 2 Viper 2 pseudo code

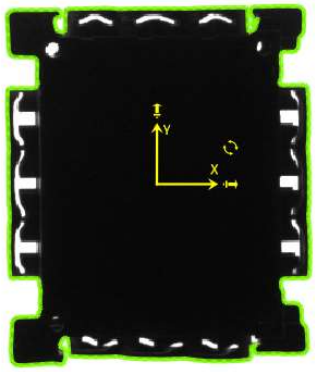
```
Move to safe position
Attach linear gripper
while (run = True) do
    Pick main box
    Place main box
    Pick top cover
    Place top cover
    Pick adjustment ring
    Place adjustment ring
    Pick lid
    Place lid
    Dock linear gripper
    Attach rotary gripper
    Insert adjustment ring
    Dock rotary gripper
    Attach linear gripper
end while
```

Communication

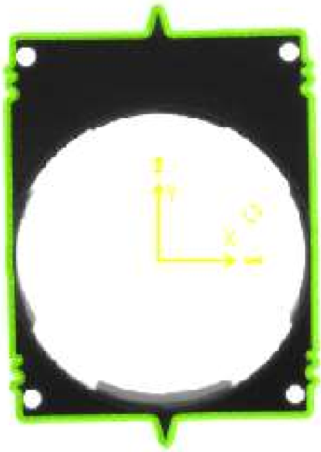
In the insertion step of the program, the robot sends a signal to the master PLC when the rotary gripper needs to start rotating. When the ring is inserted, the robot moves the gripper back towards its dock. While moving, the robot sends a signal to return the motor back to its initial position to make it ready for the next ring insertion.

Machine vision

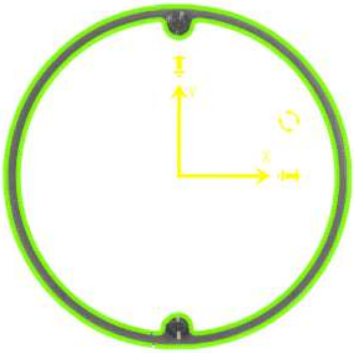
Similar to Viper 1's program, a camera captures the location of the desired component during the steps related to picking each component. Four object detection models were developed since Viper 2 picks four separate parts. In case one of the parts has moved since the last photo, a new picture is taken in each step. The detection model for the main box, top cover, adjustment ring with screws, and lid are shown in figures 4.291,4.292,4.293, and 4.294 respectively.



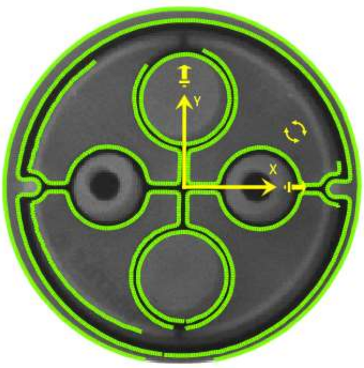
(1) Main box model



(2) Top cover model



(3) Adjustment ring with screws model



(4) Lid model

Figure 4.29: Viper 2 machine vision models

Adjustment ring insertion

Similarly to the screw insertion in Viper 1's program, Viper 2 needs to compensate for the thread pitch of the adjustment ring and the rotation speed on the rotary gripper when inserting the adjustment ring. By using equation 2.4 and the rings thread pitch of 1.5 mm the distance can be found.

$$\text{Distance} = \text{Rotation} * 1.5\ \text{mm} \tag{4.2}$$

Equation 4.2 shows that the robot needs to move 1.5 mm per rotation of the rotary gripper. Using equation 2.6 the travel speed of the robot can be found from the rotation speed of the gripper.

4.4.3 Master PLC

The master PLC is in charge of:

- **Screw driving unit motor**

  Controlling and monitoring the speed and torque to make it possible to calculate the necessary speed to match the speed of the robot.

- **Rotating gripper motor**

  Controlling and monitoring the speed and torque to make it possible to calculate the necessary speed to match the speed of the robot.

- **Conveyor belts**

  Controlling the speed and acceleration of the conveyors.

- **Safety**

  Monitoring the safety around the robot-cell and emergency stops.

- **Lights**

  Turning on and off the light for the machine vision work area.

- **Sensors**

  Monitoring the sensors related to the system.

- **Relays**

  Actuating the relays used in the system.

- **Feeder**

  Monitoring and controlling the feeder station for when the robot request a screw.

- **Magazine actuator**

  Actuating the piston for the magazine when the robot requests more parts.

- **Dock switches**

  Monitoring the docks for when the tools are inserted.

Chapter 5

Result

5.1 Finished system

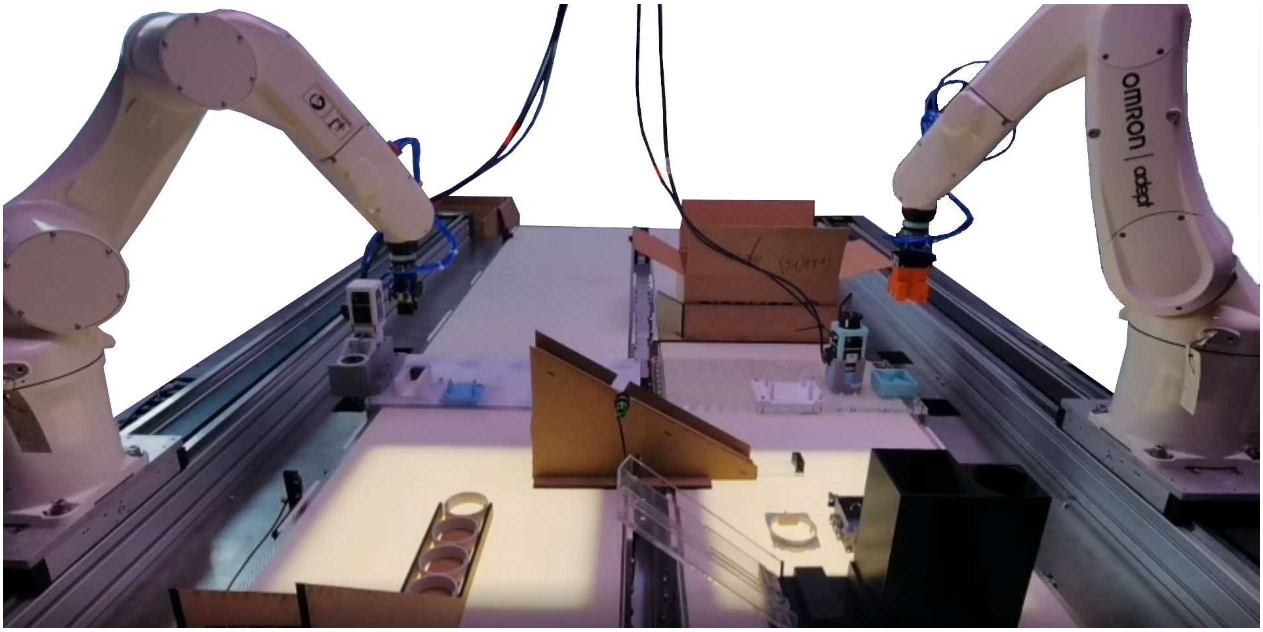


Figure 5.1: Final version of the system

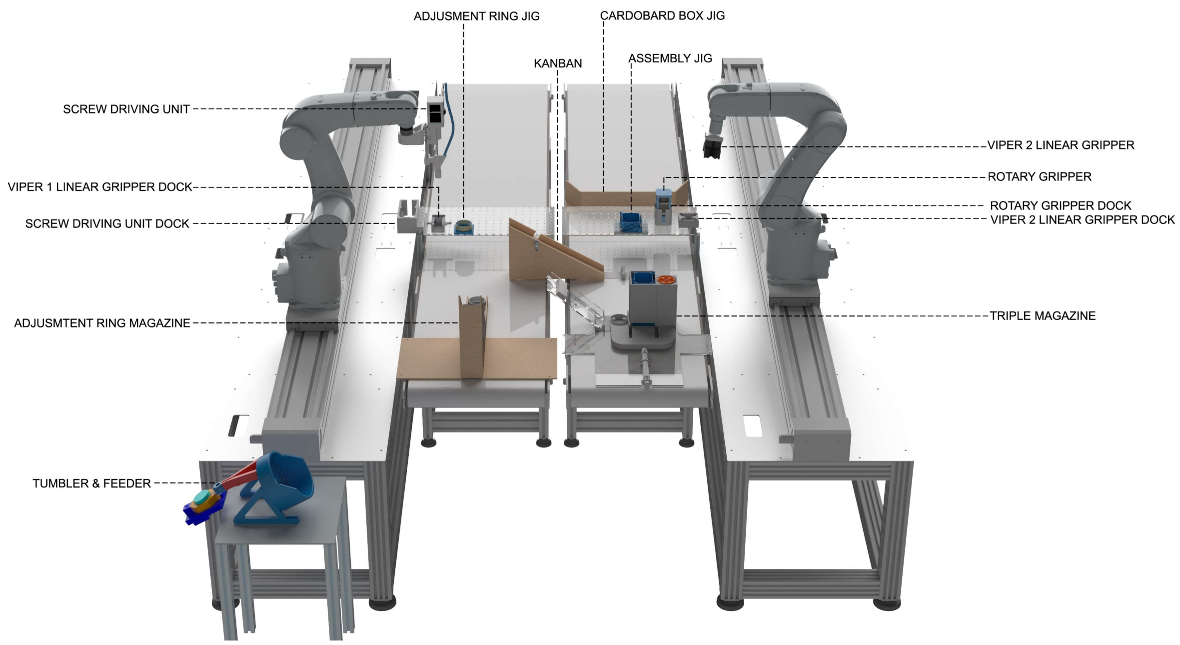


Figure 5.2: Component placement

The system is shown in its final form in 5.1 and 5.2. Figure 5.2 is a render of the cell with pointers to where each component is placed.

5.2 System efficiency

One of Pipelife's specifications was to assemble a junction box every 20 seconds. The "assemble box" process from figure 1.2 has been deconstructed in figure 5.3 to show all steps in the process and the time spent in each step.

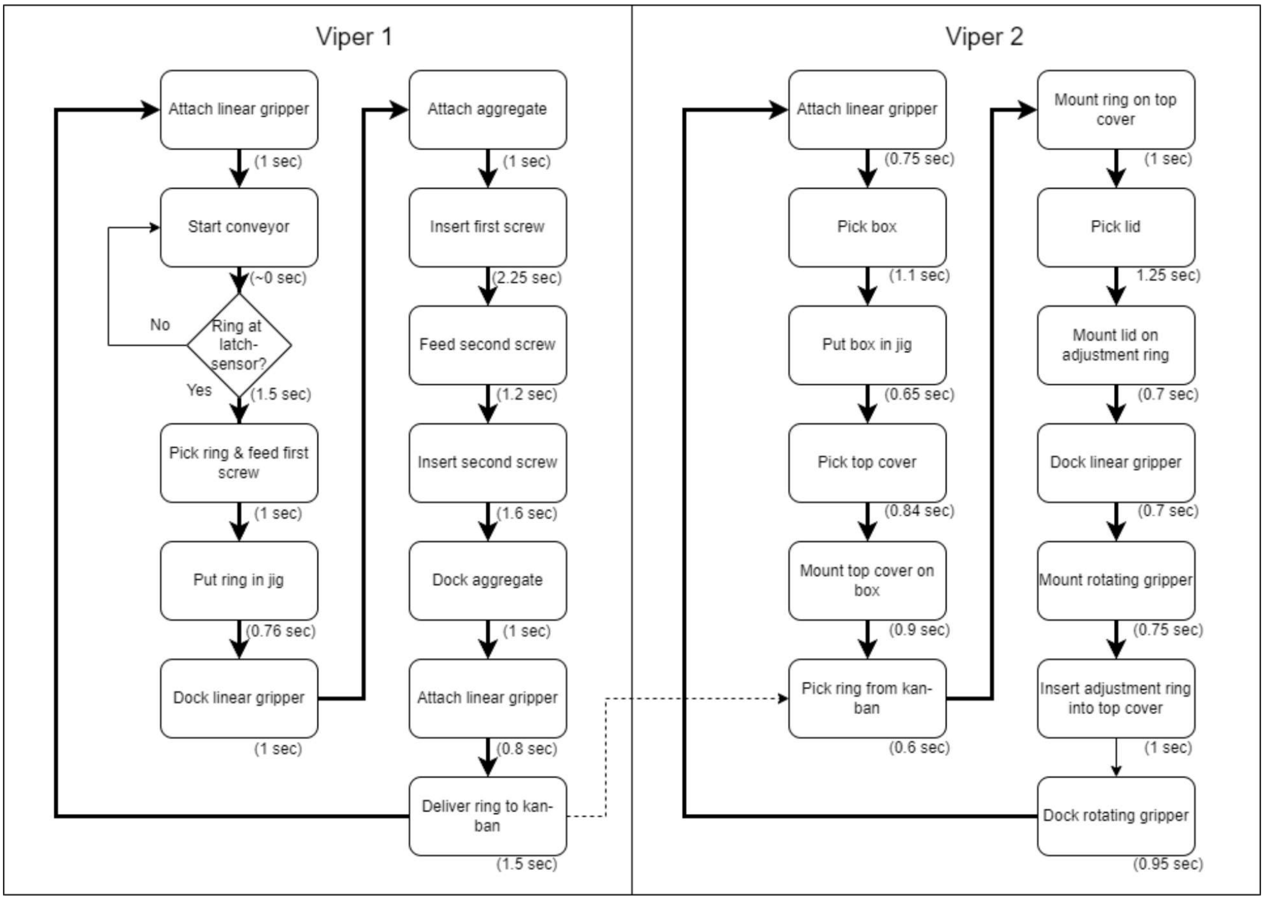


Figure 5.3: Assembly processes in both robots with time stamps

The total assembly time of a junction box is found by summing the timestamps in each step in Viper 2's assembly process, resulting in an assembly time of 11.19 seconds. Viper 1 inserts screws and delivers a ring to the Kanban in 14.61 seconds, making it the slowest part of the process. This results in Viper 2 sometimes needing to wait for an adjustment ring to enter the Kanban, making the box assembly slightly slower.

5.2.1 Component success rate

All component were individually tested to establish their success rate. The linear grippers and rotating grippers were excluded from the tests since their success rate is equal to the success rate of the servos and linear actuators they are built on. The tested components and their success rate are shown in table 5.1

Components	Number of tests	Success rate
Screw feeder	100	95 %
Screw driving unit	100	98 %
Adjustment ring magazine	100	97 %
Kanban	100	97 %
Triple magazine	100	98 %

Table 5.1: Component success rate

The overall success rate can be calculated by multiplying the success rate of each component. Equation 5.1 shows this calculation.

$$\text{Total success rate} = (0.95 \cdot 0.98 \cdot 0.97 \cdot 0.97 \cdot 0.98) \cdot 100\% = 85.85\% \tag{5.1}$$

Since the rest of the system's success rate is not taken into account, the number found in equation 5.1 is regarded as an approximation of the system's total success rate.

5.3 Technical objectives

The technical objectives in the project were:

1. Making the system flexible, so that other products can be included later.

2. Use AIVs to deliver cardboard boxes and remove cardboard boxes with products inside.

3. 20 second assembly time per box, including packaging.

The system achieves flexibility by using two robotic arms and removable fixtures and jigs. To introduce other products, the system needs to include a new program to assemble the products, as well as new jigs that fit the product.

The goal to use an AIV to deliver and remove cardboard boxes was not achieved.

The assembly time of each box is 11.19 seconds. This beats the goal of 20 seconds with 8.81 seconds, making the system capable of running at 56% speed and still achieve the goal.

Chapter 6

Discussion

6.1 System shortages

Some objectives and components were not implemented in the finished version of the system. These include the AIV delivering and fetching cardboard boxes, the triple magazine and mounting of metal brackets.

6.1.1 AIV

The reason for not including the AIV was that all AIV's were occupied in other projects. Still, an AIV was programmed to fetch a cardboard box filled with junction boxes. The AIV needs to be programmed to deliver a new cardboard box. Both the delivery and retrieval of the cardboard box needs to be included in the robot and PLC program. Implementing the AIV was not prioritized because the group only had one lid, making the system only able to assemble one box. This lead to the system never being able to deliver a full cardboard box to an AIV.

6.1.2 Triple magazine

The solenoid on the 5/2 valve controlling the pneumatic piston broke during testing. Without the solenoid, the movement of the piston could no longer be controlled from the PLC, rendering the magazine useless. The magazine was removed from the robot program, but is still wired to the PLC I/O. Fixing the magazine was not prioritized due to having one lid, making the magazine only able to feed one set of parts to the system before needing a refill.

6.1.3 Metal brackets

The mounting of metal brackets was deemed too time consuming. Pipelife agreed and wanted the group to focus solely on the box assembly. Therefore the group excluded the metal brackets from the project objectives.

6.2 Test results

6.2.1 Screw feeder

The screw feeder is working well, but there are some minor faults that should be fixed in newer revisions. The screws sometimes jams when they are supposed to enter the pneumatic tube, and the slots on the feeder wheel does not have enough clearance which can get a screw jammed.

6.2.2 Screw driving unit

While feeding screws to the screw driving unit, errors caused by the feeder were excluded from the test results. The errors with the unit occurred when a screw jammed inside the four finger tip and when it failed to drive the screw into the adjustment ring. The errors were eliminated by swapping the four finger tip with a new one.

6.2.3 Adjustment rings magazine

Since the magazine works by sliding adjustment rings down a ramp, the rings are free to rotate slightly while sliding on the guide. This resulted in the ramp working 97/100 times. The three fails occurred when one adjustment ring rotated in the guide and got jammed by colliding with another ring. This problem can be solved by either rounding the edges of the guide, or by making the ramp less steep.

6.2.4 Kanban

The Kanban is constructed the same way as the adjustment ring magazine with a guide inside a ramp. In this magazine, the adjustment rings also jams 3/100 times. These jams were also caused by rotations of the ring while sliding down the ramp. Here the ring is not jammed by colliding with another ring, but rather due to its speed when getting dropped of by the robot. To fix this issue, the guide can be rounded in the edges, or the ramp can be made less steep.

6.2.5 Triple magazine

The triple magazine works 98/100 times. Both times where the magazine failed were due to the main box getting caught in the one stored above it. This was found to happen when the boxes were not oriented the same way, as they have one side with connectors in the top of the corners. When the boxes were oriented the same way, the magazine stopped jamming. Figure 6.1 shows a box with the connectors annotated and examples of right and wrong orientations.

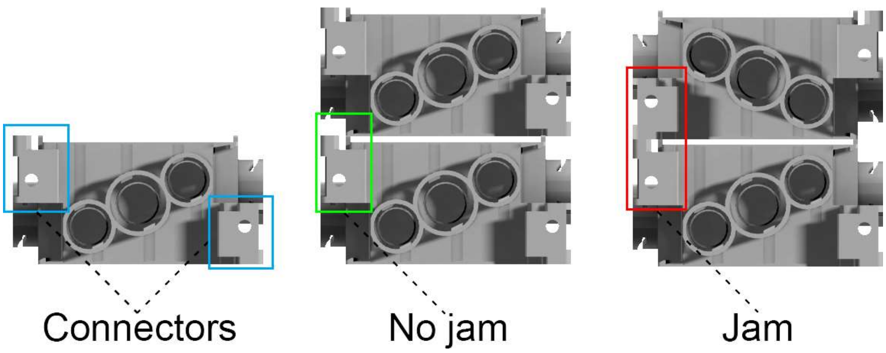


Figure 6.1: Box with connectors annotated and example of right and wrong orientations.

6.3 System efficiency

Since inserting screws into the adjustment rings is the slowest part of the system, it can be seen as a possible bottle neck. In a worst-case scenario, Viper 2 has to wait for 14.61 seconds for Viper 1 to deliver an adjustment ring. In this instance the assembly time is increased to 25.8 seconds. Since Viper 2 has yet to be programmed to palletize the boxes, this scenario is considered unrepresentative of a final system. Palletizing in this project would require Viper 2's linear drive to move the robot over to the cardboard box, likely making Viper 2's process longer than Viper 1's screw insertion. The screw insertion would then no longer be a bottle neck.

When assembling a box in 11.19 seconds the goal of 20 seconds is achieved when running at approximately 56% speed. This means that Viper 2 can use 54% less energy when assembling a box, assuming that the energy consumption is linear. Not only the power consumption is reduced by running at lower speeds, but also the wear-and-tear, and noise. In addition the robots repeatability can increase.

The systems success rate of 85.85% reduces the systems total efficiency since the system fails 14.15% of the time. To improve the success rate, and therefore improve the systems efficiency, the components need to be redesigned and manufactured in a more industrial standard. This can include machining the components in metal and/or other robust materials, in addition to implementing the changes previously mentioned. This would require the design of the parts to be changed corresponding to the chosen manufacturing method.

6.4 System value

Two of the goals for this project were reduction of production cost, and to make an affordable system.

Reduction of production cost

To produce a box in Pipelifes process today, the box needs to be transported to assembly, assembled, transported to packaging, packed, transported back to Pipelife, and lastly sent to the customer. In this process there are employees in each of the mentioned steps requiring six workers to assemble one box. The system made in this project requires little transportation and human involvement. This in combination with an assembly time of 11.19 seconds creates a possibility for reduced production cost.

Making the system affordable

The robotic cell is approximated to cost around 1.5 million NOK. The components created in the project is hard to put a price tag on since they need to be re-manufactured to a more industrial standard. Therefore they have been given an approximated value at 500 000 NOK. The price Pipelife sells their boxes to their wholesalers are also unknown. This price is estimated to be 30% lower than wholesale, which sells the box for 45 NOK [1], making the estimated price 31.5 NOK.

The number of boxes the system needs to produce to pay for itself can be found by dividing the systems total estimated cost by the estimated price Pipelife sells their boxes for.

$$\text{Boxes needed} = \frac{2000000}{31.5} \approx 63493 \tag{6.1}$$

The time used to return the investment can be found by multiplying the number of boxes, found in equation 6.1, with the assembly time of one box.

$$\text{Time to return investment} = 63493 \cdot 11.19 \approx 710487\text{seconds} \approx 11842\text{minutes} \approx 198\text{hours} \quad (6.2)$$

Equation 6.2 shows that the system could be able to return the investment in 198 hours of continuous, fault free production. It is worth to note that these calculations does not take into account any material costs in the production process. By estimating the material cost of the box to be 50% of the boxes price, the number of boxes needed is doubled. This also doubles the total time to return the investment to 396 hours. It is important to note that this number is a rough estimate, and therefore does not represent the ROI of the system in a real world scenario.

Chapter 7

Conclusions

In this project an automated assembly line was made to assemble a junction box for the company Pipelife. The system included two 6-axis robots, two conveyors and two linear drives. Additional components were designed, manufactured and built during the project. These components were tools, tool docks, magazines, sorting mechanisms, feeding systems, and various jigs/fixtures. The components were designed for 3D-printing and laser-cutting as these were the two methods available in the manufacturing process. This made the design process focus on methods for efficient and simple manufacturing.

After implementing the components and programming the system, the success rate of each component was tested, resulting in an overall success rate of 85.85%. The total assembly time was 11.19 seconds, meaning the robot speed can be reduced to 56% to the specified 20 second cycle time.

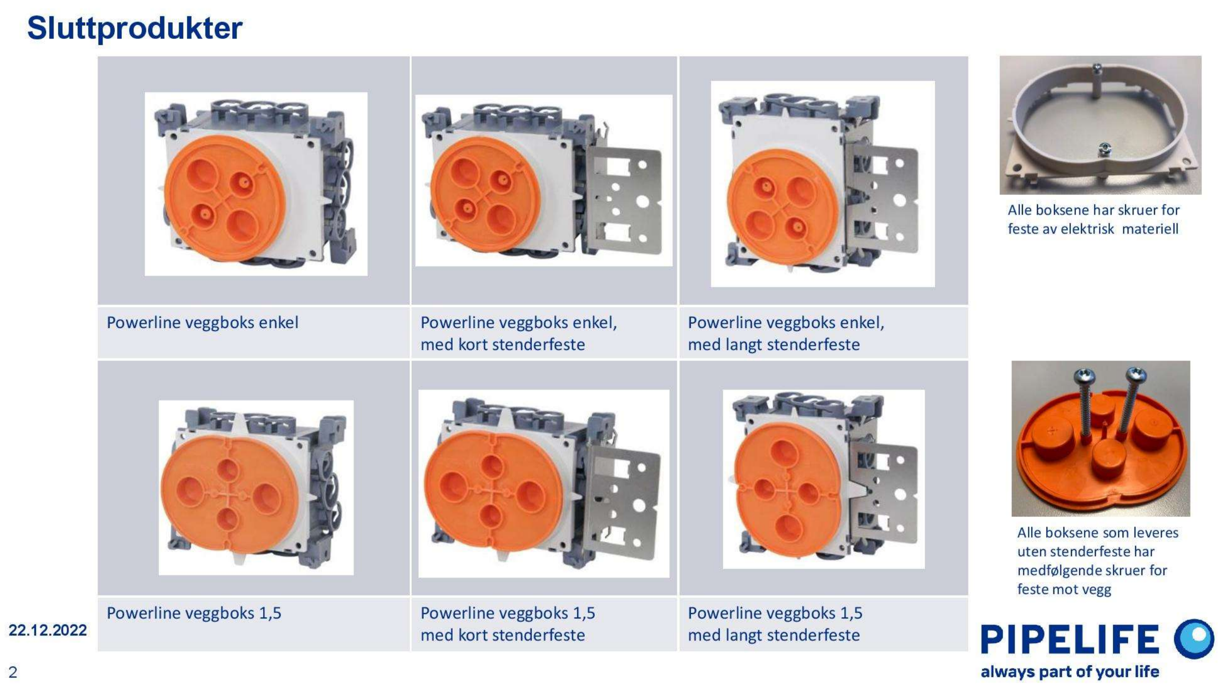
7.1 Further work

Several component upgrades and expansions to the cell is required to fully automate the system.

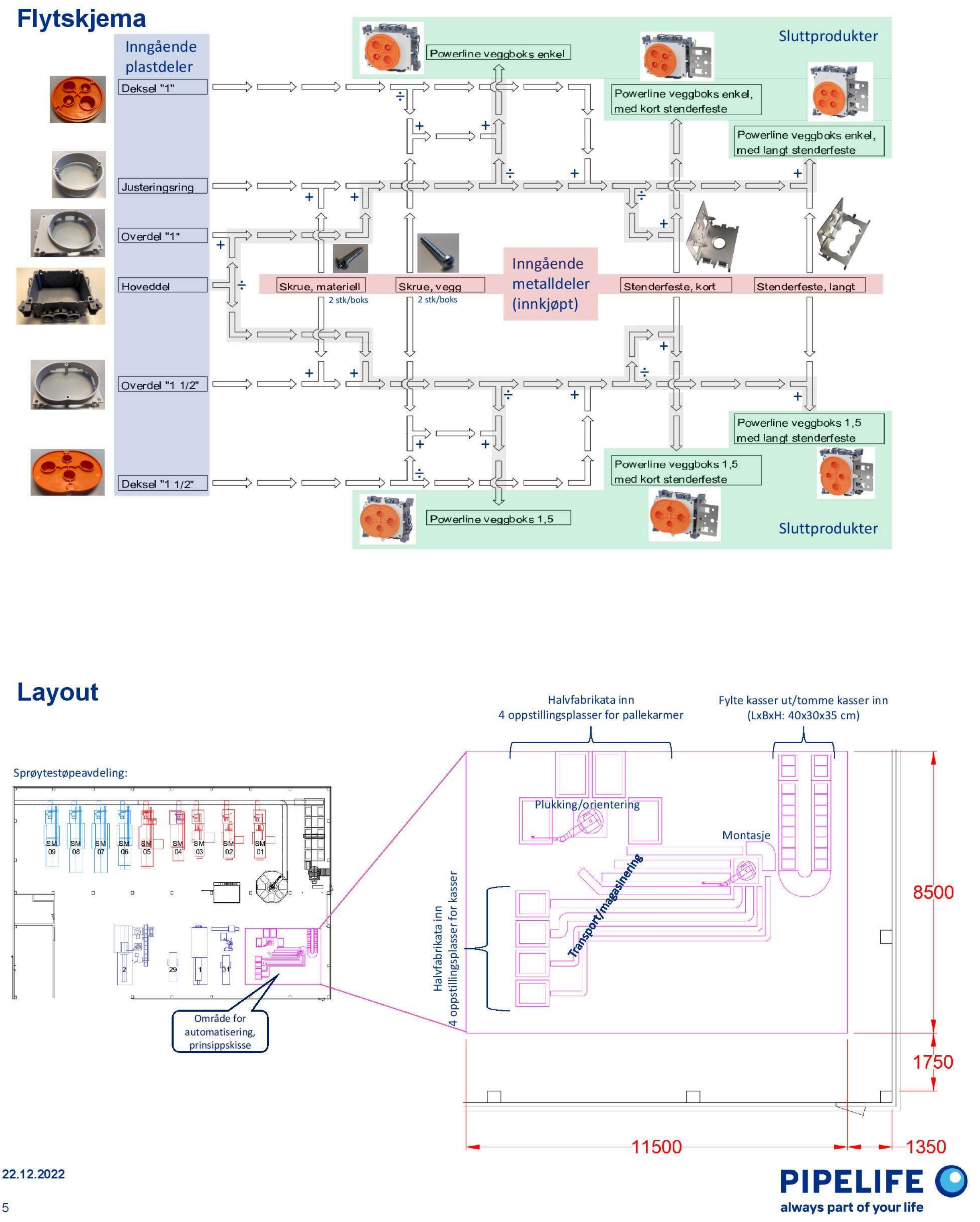
- Implement an AIV to deliver and retrieve a cardboard box.
- Fix the triple magazine, or implement bin-picking to replace magazines.
- Redesign components to improve their success rate.
- Expand the system to assemble different variations of the box.

Appendices

A Project specification


Sluttprodukter
Alle boksene har skruer for feste av elektrisk materiell
Powerline veggboks enkel
Powerline veggboks enkel, med kort stenderfeste
Powerline veggboks enkel, med langt stenderfeste
Alle boksene som leveres uten stenderfeste har medfølgende skruer for feste mot vegg
Powerline veggboks 1,5
Powerline veggboks 1,5 med kort stenderfeste
Powerline veggboks 1,5 med langt stenderfeste
22.12.2022
2
PIPELIFE
always part of your life


Inngående halvfabrikat
Hoveddel
Overdel «1 ½»
Deksel «1 ½»
Skrue, materiell
Stenderfeste, kort
Justeringsring
Overdel «1»
Deksel «1»
Skrue, vegg
Stenderfeste, langt
22.12.2022
3
PIPELIFE
always part of your life


Flytskjema
Inngående plastdeler
Deksel "1"
Justeringsring
Overdel "1"
Hoveddel
Overdel "1 1/2"
Deksel "1 1/2"
Sluttprodukter
Powerline veggboks enkel
Powerline veggboks enkel, med kort stenderfeste
Powerline veggboks enkel, med langt stenderfeste
Skrue, materiell
2 stk/boks
Skrue, vegg
2 stk/boks
Inngående metalldeler (innkjøpt)
Stenderfeste, kort
Stenderfeste, langt
Powerline veggboks 1,5 med langt stenderfeste
Powerline veggboks 1,5 med kort stenderfeste
Powerline veggboks 1,5
Sluttprodukter
Layout
Sprøytestøpeavdeling:
Område for automatisering, prinsippskisse
Halvfabrikata inn
4 oppstillingsplasser for pallekarmer
Fylte kasser ut/tomme kasser inn
(LxBxH: 40x30x35 cm)
Plukking/orientering
Montasje
Transport/magasinering
Halvfabrikata inn
4 oppstillingsplasser for kasser
8500
1750
11500
1350
22.12.2022
5
PIPELIFE
always part of your life

Øvrige spesifikasjoner

- Veggboksene skal monteres helautomatisk, med de inngående delene som vist tidligere i presentasjonen, og deretter pakkes i kasser.
- Halvfabrikater av plast leveres inn til anlegget i pallekarmer.
- Halvfabrikater av andre materialer leveres i mindre kasser.
- Pappkasser leveres inn til anlegget fra AGV. Kassene er ferdig oppreist med lokkene på toppen stående opp.
- Ferdig fylte pappkasser skal leveres til AGV-transport for videre palletering. Kassene lukkes i eksisterende palleteringsanlegg.
- Kassene for veggboksene har målene 40x30x35 cm (LxBxH), men det må tas høyde for andre kassestørrelser til evt. andre produkter.
- Det er 35 veggbokser i hver kasse.
- Anlegget må ha kapasitet for å ferdigmontere og håndtere 1 veggboks per 20 sekund.
- Det er viktig at anlegget lages for fleksibilitet, og at andre produkter kan være mulig å håndtere.

22.12.2022

6

PIPELIFE
always part of your life

B Pre-project report

FORPROSJEKT - RAPPORT
FOR BACHELOROPPGAVE

NTNU
Kunnskap for en bedre verden

TITTEL:
Manulab: Automatisk Montering av Elektro Bokser

KANDIDATNUMMER(E):			
BENJAMIN KARLSEN & FRIDTJOF PEDERSEN LERSVEEN			
DATO: **10.01.2023**	EMNEKODE: **IELEA2920**	EMNE: **Bacheloroppgave automatisering**	DOKUMENT TILGANG: - Åpen
STUDIUM: **AUTOMATISERING OG ROBOTIKK**		ANT SIDER/VEDLEGG: 13 / 3	BIBL. NR: • Ikke i bruk -

OPPDRAGSGIVER(E)/VEILEDER(E):
Ola Jon Mork, Paul Steffen Kleppe, Adam Leon Kleppe

OPPGAVE/SAMMENDRAG:

Oppgaven handler om å lage en helautomatisert løsning for montering av elektrobokser. For å gjøre dette på en mest mulig effektiv måte, vil det bli tatt i bruk to roboter, hvor en plukker delene og den andre monterer de sammen. Her vil de bli prøvd å bruke et 3D-kamera for «bin picking» av komponenter. Det vil også bli nødvendig å lage flere forskjellige jigger for orientering av komponenter under monteringsprosessen.

Denne oppgaven er en eksamensbesvarelse utført av student(er) ved NTNU i Ålesund.

INNHOLD

1 INNLEDNING

Bachelorgruppens valg av oppgave kom fra et tidligere fag, hvor kandidatene startet på å løse en forenklet versjon av oppgaven. Etter at oppdragsgiveren Pipelife så på den tidligere løsningen, ble det aktuelt å videreutvikle systemet. Problemstillingen i bacheloroppgaven handler om å automatisere en produksjon av «elektrobokser». Per dags dato blir disse produsert av oppdragsgiver, hvor de så fraktes til montering. Alt av montering skjer for hand og blir i etterkant av montering lagret før de sendes til en kunde. Denne oppgaven har som formål å erstatte de manuelle oppgavene i dagens prosess. Samtidig er et viktig formål å minimere frakt mellom forskjellige ledd. Dermed skal den ferdige løsningen ha som mål å plukke, montere og pakke deler i samme celle.

2 BEGREPER

- PLS (Programmerbar Logisk Syring)
- Bin picking (Plukking av tilfeldig orienterte deler)
- Viper (Industriell robot brukt i denne oppgaven)
- Vision (Maskinsyn)

3 PROSJEKTORGANISASJON

3.1 Prosjektgruppe

Studentnummer(e)
Benjamin Karlsen - 536151 Fridtjof Pedersen Lersveen - 536141

Tabell: Studentnummer(e) for alle i gruppen som leverer oppgaven for bedømmelse i faget ID 302906

3.1.1 Oppgaver for prosjektgruppen - organisering

- 3D tegning
- Programering Viper
- Programering PLS/Omron
- Maskinering
- 3D printing
- Feilsøking
- Dokumentering
- Vision programering
- “research”

3.1.2 Oppgaver for prosjektleder

Prosjektlederen har ansvar for:

- Arbeidet som blir utført følger planen. Dersom det er nødvendig å avvike fra planen skal planen oppdateres.
- Å informere veilederne om framgang, avvik og andre bemerkninger. Dette kan bli gjort under de planlagte møtene.
- Kontakte bedriften ved eventuelle spørsmål eller oppdateringer.
- Sørge for at arbeidsplass(er) er trygge og at arbeidet er forsvarlig.

3.1.3 Oppgaver for sekretær

Sekretæren sitt ansvar er:

- Møterefferat.
- Passe på at dokumentasjon blir gjort undervegs og stemmer med fysiske anlegg.
- Kalle inn til møter.
- Formidle dokumentasjon og filer internt og til andre parter.

3.2 Styringsgruppe (veileder og kontaktperson oppdragsgiver)

Pipelife kontaktperson:

- Navn: **Per Holten**
- Mail: per.holten@pipelife.com
- Tlf: +47 907 51 212

Veiledere:

- Navn:
  - Adam Leon Kleppe,
  - Ola Jon Mork,
  - Paul Steffen Kleppe.
- Mail:
  - adam.l.kleppe@ntnu.no
  - ola.j.mork@ntnu.no
  - paul.s.kleppe@ntnu.no
- Tlf:
  - 91896882
  - 70161490
  - 41526987

4 AVTALER

4.1 Avtale med oppdragsgiver

Avtalen mellom bachelorgruppen og oppdragsgiver er at gruppen skal utvikle en industrialisering av dagens produksjon. Gruppen skal lage robuste løsninger som fungerer industrielt, og forbedrer tider, kvalitet og kapasitet.

4.2 Arbeidssted og ressurser

Tilgang til arbeidsplass:

- Alle medlemmer av gruppen har døgntilgang til arbeidsstedhele uken. Arbeidssted er L101 Manulab ved NTNU Ålesund.

Tilgang til ressurser:

- Ett av medlemmene har tilgang til tunglabb i vanlige arbeidstider. Denne labben vil bli brukt til saging, boring og laserkutting.
- Den mekaniske labben samt plastlabben trenger søknad om tilgang. Labben vil bli brukt til større mekaniske oppgaver som for eksempel maskinering av metalldeler
- Alle medlemmene har tilgang til elektrolabben. Denne labben blir brukt til lodding, samt henting av elektriske komponenter.

Tilgang til personer:

- Per Holten (Kontaktperson for bedriften)
- Adam Leon Kleppe (Veileder)
- Ola Jon Mork (Veileder)
- Paul Steffen Kleppe

4.3 Gruppenormer – samarbeidsregler – holdninger

Medlemmene i oppgaven må kunne samarbeide på en profesjonell og effektiv måte, noe som medfører at de må:

- Forholde seg til avtalte tidspunkt.
- Møte i tide og ikke komme for sent gjentatte ganger.
- Kunne diskutere og samarbeide selv om forslag og ideer blir nedstemt.
- Gjøre jobben og oppgavene som man har fått tildelt.

5 PROSJEKTBESKRIVELSE

5.1 Problemstilling - målsetting - hensikt

Formuleringer av den grunnleggende problemstillingen og hva en skal komme fram til i løpet av prosjektet – hovedmål og evt. delmål. Gjerne med en inndeling eller beskrivelse som skjelner mellom effektmål (verdimål), resultatmål og prosessmål.

Prosjektets mål er å ved hjelp av to industrielle roboter automatisere en monteringsprosess for en type elektro boks. Denne boksen består av totalt 10 komponenter som kan kombineres for å skape 6 forskjellige bokser.

Hovedmålet i dette prosjektet er lage en monteringscelle som klarer å montere en av de 6 boksene, som da kan skaleres etter dette. De forskjellige oppgavene til cellen blir delmål som må oppnås for å klare hovedmålet. Disse delmålene er:

- Lage et GUI/HMI hvor man bestemmer hvilken boks som skal produseres
- Sette opp en form for plukking av inngående deler. 3D bin picking vil bli prioritert.
- Lage jigger for montering, palletering og mellomlagring.
- Lage stasjon for pakking.
- Programmere roboter til å samsvare med GUI/HMI og kameraer.
- Videre utvikle cellen til å montere forskjellige boks typer

5.2 Spesifikasjoner

Spesifikasjonene er tatt fra Pipelife sin prosjektforespørsel (Vedlegg 1).

- Veggboksene skal monteres helautomatisk, med de inngående delene som vist tidligere i presentasjonen, og deretter pakkes i kasser.
- Halvfabrikater av plast leveres inn til anlegget i pallekarmer.
- Halvfabrikater av andre materialer leveres i mindre kasser.
- Pappkasser leveres inn til anlegget fra AGV. Kassene er ferdig oppreist med lokkene på toppen stående opp.
- Ferdig fylte pappkasser skal leveres til AGV-transport for videre palletering. Kassene lukkes i eksisterende palleteringsanlegg.
- Kassene for veggboksene har målene 40x30x35 cm (LxBxH), men det må tas høyde for andre kassestørrelser til evt. andre produkter.
- Det er 35 veggbokser i hver kasse.
- Anlegget må ha kapasitet for å ferdigmontere og håndtere 1 veggboks per 20 sekund.
- Det er viktig at anlegget lages for fleksibilitet, og at andre produkter kan være mulig å håndtere

5.3 Planlagt framgangsmåte(r) for utviklingsarbeidet – metode(r)

Fremgangsmåten for gruppen på de forskjellige oppgavene kan forklares kort og presist for de fleste oppgavene.

- Diskutere mulige løsninger på problemstilling/oppgaven
- Lage prototype av deler/program og teste dette ut
- Feilsøking av eventuelle feil og eller nye problem
- Ta feilsøking/feil videre og vidareutvikle prototype
- Ny runde med testing
- Optimalisering og ferdigstilling av prototyper

5.4 Informasjonsinnsamling – utført og planlagt

Samtlige av medlemmene i gruppen har fagbrev fra relevante fagområder, og har gjennom dette vært med på å montere lignende systemer.

I tillegg til tidligere erfaring har oppdragsgiver vist hvordan noen av komponentene blir produsert i eksisterende anvendelser, samt gitt sine ønsker om prosessflyt og layout.

For videre informasjon vil det primært bli kommunisert med Pipelife. For mer teknisk informasjon vil datablader bli brukt. Om nødvendig informasjon ikke skulle kunne finnes i datablader, kan Omron kontaktes direkte.

5.5 Vurdering – analyse av risiko

Med den gitte tidsrammen ser vi ingen mulighet i å utvikle hele cellen som blir etterspurt av oppgavegiveren. Derfor har oppgavens mål blitt å produsere en av de seks etterspurte variantene, for å så skalere cellen videre om mulig.

For å kunne lykkes med dette er det viktig å jobbe systematisk mot satte mål, samt å følge kjente sikkerhetsprosedyrer. Det er også utført en risikovurdering i form av en risikovurderingsmatrise. Denne er vedlagt som vedlegg 2.

5.6 Hovedaktiviteter i videre arbeid

(F = Fridtjof, B = Benjamin)

Nr	Hovedaktivitet	Ansvar	Kostnad	Tid/omfang
A1	Designe deler	F/B	0	30 dager
A11	Konstruksjon av monteringsjigg	F	0	
A12	Designe og lage griper	F	0	
A13	Lage mellomjigg for deler	F	>100kr	
A14	Lage jig for metall braketter	F/B	>100kr	
A15	Lage stasjon for palletering	F/B	>100kr	
A16	Lage sortering for skruer	F/B	>100kr	
A17	Lage stasjon for griperskift	B	>100kr	
A2	Oppbygging av celle	F/B	100000kr	18 dager
A21	Montere 3D kamera	B	0	
A22	Sette opp rotasjonsbord	F/B	0	
A23	Montere div. jigger	F/B	0	
A24	Montere div komponenter	F/B	0	
A25	Sette opp stasjon for griperskift	F/B	0	
A3	Programmering i ACE	B	0	30 dager
A31	Sette opp kommunikasjon	F	0	
A32	Programmere vision	B	0	
A33	Programmere roboter	B	0	
A34	Optimalisering	B	0	
A4	Programmering i Sysmac	F	0	30 dager
A41	Sette opp kommunikasjon	F/B	0	
A42	Få roboter til å samkjøre	F/B	0	
A43	Programmere transportband	F	0	
A44	Programmere linjeføringer	F	0	
A45	Programmere GUI/HMI	F/B	0	
A46	Programmere rotasjonsbord	F	0	

5.7 Framdriftsplan – styring av prosjektet

5.7.1 Hovedplan

Milepæler fortløpende gjennom semmesteret/prosjektet:

- Automatisk sortering av skruer
  Ferdigmontering og testkjøring av sorteringsautomat for skruer.
  - Programmering (Arduino eller Sysmac Studio) — Benjamin/Fridtjof
  - Montering — Benjamin/Fridtjof
  - Design — Benjamin/Fridtjof

- Skrue aggregat plukker opp og skrur inn skruer
  Griper/aggregat på robot plukker opp skruer fra automat og skrur inn skruer i justeringsring.
  - Programmering (ACE) — Benjamin
  - Montering — Benjamin/Fridtjof
  - Design/Tegning — Fridtjof
  - Testing/Feilsøking — Benjamin/Fridtjof

- Jigg for montering av justeringsring og boks
  Monteringsjigg laget for å minimere tid brukt på å montere justeringsring i boks er tatt i bruk.
  - Programmering (ACE og Sysmac Studios) Benjamin/Fridtjof
  - Montering Benjamin
  - Design Benjamin
  - Testing/Feilsøking Benjamin/Fridtjof

- Montering av første komplette boks u/skruer og veggfeste
  Montering av boks med jigger laget spesifikt for plastdeler er programmert og virker med høy repeterbarhet.
  - Programering (ACE og Sysmac Studio) Benjamin/Fridtjof

- Jigg for montering av skruer
  Egen jigg for å holde justeringsring for montering av skruer er tatt i bruk med høy repeterbarhet.
  - Programering (ACE) Benjamin
  - Montering Fridtjof
  - Design Fridtjof

- Bin-picking av deler
  Bin-picking begynner å virke på enkelte deler og kan videreutvikles for prosjektet.
  - Programmering (ACE/Vision Controller) Benjamin

- Jigg for veggfeste
  Jigg for montering av veggfester på boks er implementert.
  - Programmering (ACE) Benjamin
  - Design Fridtjof

- Montering av første boks med skruer og eller veggfeste
  Automatisk montering av boks med skruer og eller veggfeste tar plass uten bin-picking og er optimalisert med høy repeterbarhet.
  - Programmering (ACE og Sysmac Studio) Benjamin/Fridtjof

- Bin picking av flere typer deler med rotasjonsbord.
  Bin-picking er videreutviklet og har muligheten til å plukke nødvendige deler for montering av en komplett boks.
  - Programmering (ACE og Sysmac Studio) Benjamin/Fridtjof

- Samarbeid og montering med begge Viper roboter
  Begge Viper robotene er programert og har mulighet til å sammarbeide for montering av halv-ferdige eller komplette bokser.
  - Programmering (ACE og Sysmac Studio) Benjamin/Fridtjof

- Start/Stop av operasjoner fra HMI/GUI
  HMI/GUI er implementert og det er mulig å styre hele prosessen fra dette.
  - Programmering (ACE og Sysmac Studio) Benjamin/Fridtjof

- *Kanskje* Implementering av forskjellige typer bokser med resept-basert system
  Implementering av resepter kommer etter alt. Dette er et punkt vi ikke ser på som nødvendig for et fullstendig og ferdig prosjekt da gruppen fokuserer på montering av en type boks ved utvikling av cellen. Dette utelukker selvfølgelig ikke at cellen skal utvikles for å kunne montere alle 6 boksene.
  - Programmering (ACE og Sysmac Studio) Benjamin/Fridtjof
  - Montering Benjamin/Fridtjof
  - Design Benjamin/Fridtjof

5.7.2 Styringshjelpemidler

I prosjektet blir det brukt primært Gantt diagram for ha en illustrasjon på fremdriften og hvordan progresjonen er i de forskjellige arbeidsoppgavene som er satt opp. Samt vil framdriften bli dokumentert fortløpende i form av framdriftsrapporter. Disse rapportene vil bli gjennomgått med veiledere for å få innspill til videre arbeid.

5.7.3 Utviklingshjelpemidler

- Programvarer for:
  - Roboter
  - Kamera
  - Master PLS
  - 3D tegning
  - 3D pritning
  - Maskinering
  - Tegneprogram for koblingsskjema

5.7.4 Intern kontroll – evaluering

Internkontroll vil bli utført ved oppdatering av gantdiagrammet. I dette diagrammet vil framdriften bli tydelig vist opp imot planlagt tidsbruk. For at et mål/delmål skal være oppnådd må målets innhold være oppfylt samt testet. Kun når målet er feilfritt vil det bli satt som oppnådd.

5.8 Beslutninger – beslutningsprosess

Beslutningen om at prosjektoppgaven var for stor til å ha som mål med å bli helt ferdig ble tatt i lag med veileder Ola Jon Mork. Her ble det enighet om at gruppen skulle fokusere på å skape en av boksene, for å så skalere cellen videre om det var mulig tidsmessig.

For beslutninger videre i prosjektet, vil de bli tatt felles i gruppen. For større beslutninger vil det bli tatt opp på møter med veiledere, og også kommunisert med oppdragsgiver om nødvendig.

6 DOKUMENTASJON

6.1 Rapporter og tekniske dokumenter

Dokumentasjonen i prosjektet kan komme i form av:

- Koblingsskjemaer
- Fremdriftsdiagram (gantdiagram)
- Arrangement tegninger

Under dokumentering vil dokumentene bli tatt opp i møter med veiledere, hvor de vil bli sett nærmere på for godkjennelse.

7 PLANLAGTE MØTER OG RAPPORTER

7.1 Møter

7.1.1 Prosjektmøter

Møter mellom gruppemedlemmer og veiledere skal skje en gang annen vær uke, samtidig kan det kalles inn til møter utenom denne planen om det skulle bli nødvendig.

I møtene skal veiledere bli oppdatert på prosjektets framgang og hvordan denne framgangen ligger an i forhold til planen. Nye dokumenter vil også bli lagt frem, slik gruppen får en rask tilbakemelding på forbedringspotensialer. Det vil også bli diskutert hvordan gruppen skal jobbe frem til neste møte.

Alle møtene skal oppsummeres i et møtereferat.

7.2 Periodiske rapporter

7.2.1 Framdriftsrapporter (inkl. milepæl)

Framdriftsrapporten blir i form av et gantdiagram som kontinuerlig oppdateres. For å forsikre at ingen informasjon blir utelukket fra diagrammer, skal innholdet gås over i slutten av hver uke.

Det skal også bli lagt en rapport for hvert møtes innhold i form av et referat.

Imellom hvert møte skal det skrives en framdriftsrapport som beskriver mål for perioden, planer for å oppnå målene, samt en beskrivelse av arbeidet som faktisk ble utført.

8 PLANLAGT AVVIKSBEHANDLING

Ved avvik i prosjektets plan vil det bli dokumentert og vist i gantdiagrammet.

Ved små avvik som for vil ikke det være nødvendig å endre noe på prosjektplanen. Arbeidet kan gå videre som planlagt uten at det har noen stor konsekvens.

Ved drastiske avvik må gruppen kontakte veiledere for å bli enige om alternative løsninger eller nye mål for gruppen. I slike tilfeller må planen oppdateres, samt må det begrunnes nøye hvorfor et slikt avvik oppstod.

Om det skulle oppstå et avvik som gjør at prosjektet ikke kan utføres som planlagt skal ansvaret ligge på gruppen som en helhet siden gruppen kun består av to medlemmer.

9 UTSTYRSBEHOV/FORUTSETNINGER FOR GJENNOMFØRING

I prosjektet vil det være nødvendig med følgende nytt utstyr:

- Maskin for skru sortering. (Denne vil først bli forsøkt å lage av gruppen).
- Skrumaskin til industriell robotarm. (Denne vil først bli forsøkt å lage av gruppen).
- Elektrisk aktuert lineær griper til robot.

Noen av punktene vil bli forsøkt lagt av gruppen, men om det skulle bli en forhindring vil det bli sett på muligheter for å kjøpe et tilsvarende produkt.

C Gantt chart

The Gantt chart was made in the website Notion.so. The chart can be seen by clicking the following hyperlink: Gantt chart. The Gantt charts major activities and their planned time can be summarized as:

Aa Name	Date	Date Created
Documentation	@January 17, 2023 → May 22, 2023	@January 17, 2023 11:33 PM
Programming in Sysmac	@March 1, 2023 → April 30, 2023	@January 17, 2023 11:32 PM
Programming in ACE	@March 1, 2023 → April 30, 2023	@January 17, 2023 11:29 PM
Building the cell	@February 13, 2023 → March 5, 2023	@January 17, 2023 11:26 PM
Designing and ordering parts	@January 19, 2023 → February 19, 2023	@January 17, 2023 11:22 PM

This summary does not include the sub-activities within each major activity.

D Risk assessment matrix

		Consequence				
		A Harmless	B Some	C Dangerous	D Critical	E Very critical
Probability	5 Extremely likely					
	4 More than likely					
	3 Likely					
	2 Unlikely					
	1 Very unlikely					

Risk acceptance criteria	
Level	**Measures**
HIGH	Red is not acceptable and measures must be taken immediately
MEDIUM	Yellow indicates the need for implementation of risk-reducing measures
LOW	Green indicates acceptable risk, introduction of measures is considered where it obviously brings good effects

Possible risks				
Project scenario	**Probability**	**Consequence**	**Risk class**	**Measures**
Short circuit of equipment	1	B		Installation done by capable workers. Work done on powerless system
Crash between robots and humans	1	E		Follow all safety procedures when the robots are running.
Mechanical crash with robots	3	B		Keep out of the cell when robots are moving.
Wrong usage of components	2	A		Read documentation before useage
Pinching	2	B		No measures necessary
Cuts	2	B		No measures necessary

Bibliography

[1] Pipelife-junction-box-wholesale. URL https://www.ledonline.no/powerline-veggboks-enkel.html?srsltid=AR57-fDQmRZgHvPOUW_mNb_YYfAEA3JMjNuWDiFgrPOK7uelu76RP1SWtco.

[2] Autodesk. What is fusion 360? URL https://www.autodesk.no/products/fusion-360/overview?term=1-YEAR&tab=subscription.

[3] Dirk Buchholz. Bin-picking. *Studies in Systems, Decision and Control*, 44:3–12, 2016.

[4] Aurelio Agliolo Gallitto, Roberto Zingales, Onofrio Rosario Battaglia, and Claudio Fazio. An approach to the venturi effect by historical instruments. *Physics Education*, 56(2):025007, 2021.

[5] Marsbed H Hablanian. *High-vacuum technology: a practical guide*. Routledge, 2017.

[6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[7] Dirk Jansen and Holger Buttner. Real-time ethernet: the ethercat solution. *Computing and Control Engineering*, 15(1):16–21, 2004.

[8] Zhenzhong Jia, Ankit Bhatia, Reuben M Aronson, David Bourne, and Matthew T Mason. A survey of automated threaded fastening. *IEEE Transactions on Automation Science and Engineering*, 16(1):298–310, 2018.

[9] Jerry Kilpatrick. Lean principles. *Utah Manufacturing Extension Partnership*, 68(1):1–5, 2003.

[10] Charles Chikwendu Okpala and Ezeanyim Okechukwu. The design and need for jigs and fixtures in manufacturing. *Science Research*, 3(4):213–219, 2015.

[11] Omron. Ace. URL https://industrial.omron.no/no/products/ace.

[12] Brian G. Schunck Ramesh Jain, Rangachar Kasturi. *Machine Vision*. McGraw-Hill Inc., 1995. ISBN 0-07-032018-7.

[13] John Rinaldi. An overview of ethernet/ip. *Real Time Automation< www. rtaautomation. com*, 2003.

[14] Ralph S Shoberg. Engineering fundamentals of threaded fastener design and analysis. i. *Fastening*, 6(2):26–29, 2000.

[15] Siemens. Nx cad for product engineering. URL https://plm.sw.siemens.com/en-US/nx/cad-online/.

NTNU

Norwegian University of
Science and Technology