TECHBOOST

Cobot assembly

-technology demo report

Inspired by Dynaset Oy

Elinkeino-, liikenne- ja ympäristökeskus
Työ- ja elinkeinoministeriö
Arbets- och näringsministeriet
Tampereen yliopisto
HAMK Hämeen ammatti-korkeakoulu
Metropolia
Aalto-yliopisto
Aalto-universitetet
Aalto University
OAMK
Oulun ammattikorkeakoulu
Tampereen ammattikorkeakoulu

Rasmus Hyypiä

DYNASET HYDRAULIC GENERATOR ASSEMBLY

Implementing machine vision and nutrunner-based fastening for enhanced efficiency

Tampere University
December 2024
TECHBOOST Project Report

CONTENTS

1. Introduction

This report presents the findings from the TECHBOOST initiative's Dynaset case study. Techboost is a collaborative program that supports the growth of small and medium-sized enterprises (SMEs) by partnering with Finnish higher education institutions. By leveraging academic expertise, TECHBOOST helps SMEs adopt, develop, and implement new technologies, particularly in the fields of robotics and artificial intelligence (AI). In total, TECHBOOST oversees 30 business projects across Finland. Among these is the Dynaset project—carried out by Tampere University—aimed at enhancing Dynaset's manufacturing processes using collaborative robotics and machine vision.

Dynaset is a company based in Ylöjärvi, Finland, that specializes in developing solutions to increase the productivity of mobile work machines by converting hydraulic energy into various forms of power, including electricity, compressed air, or mechanical power.

This project focuses on automating a specific step in Dynaset's hydraulic generator assembly process. This step involves four key components: the generator body, the flange, the hydraulic motor, and the fastening bolts. The generator body serves as the main structure to which the other parts are attached. Although Dynaset produces about ten variations of hydraulic generators, the scope of this project centered on three distinct variations, with the final demonstration narrowing down to two. In the actual production environment, these parts are delivered on a pallet that travels along a roller conveyor from a preceding workstation and continues down the line after assembly. For this project, however, a physical mockup of the robot cell was created in a university laboratory. An identical height-adjustable table provided by Dynaset was utilized to simulate the real factory environment.

The primary goal of this project was to gather practical information and best practices, documenting the encountered challenges and the solutions developed. This documentation aims to prevent the recurrence of similar issues for Dynaset personnel or other interested readers. A key target, established during the kickoff meeting, was to achieve a five-minute cycle for the assembly as a feasibility benchmark. More broadly, the project sought to build and test a system flexible to handle varying conditions and reliable enough for potential use in a production environment.

To test the system's robustness, the mockup cell was designed as a "worst-case scenario", with minimal constraints to increase complexity. One challenge was the height-adjustability of the workstation table, which workers can alter for ergonomic reasons. Dynaset wants the station to be used by both robots and humans interchangeably, requiring the robot to adapt to these height changes. Details about both the university test cell and the real factory environment are discussed in Chapter 2.

Another challenge lay in the pallet design. Although parts rest on holders to facilitate pickup, their positions are not fixed in known locations. Instead, they can freely shift and rotate within defined boundaries. This variability necessitates machine vision for accurate detection and localization. Furthermore, the robot cell was deliberately exposed to non-standardized lighting conditions to mirror real factory conditions, where fluctuating lighting can affect vision performance. To effectively handle the dynamically positioned parts on the pallet, approaches to designing custom gripper fingers and component holders are described in Chapter 3, while Chapter 4 covers the vision-related methods and their integration into the system.

In addition, the project evaluated how a Cleco Cellcore cordless battery-powered nutrunner—originally designed for handheld use—could function as a part of the robotized assembly process. Unlike traditional nutrunners, the Cleco Cellcore features adjustable torque values, and programmable modes, and requires a dedicated controller. This integration, including I/O connections, physical mounting to the robot, and strategies for consistent fastening operations, are elaborated in Chapter 5: Integration and Mounting Design of the Cellcore Nutrunner.

2. System Overview

This chapter provides an overview of the robot cell developed to assemble hydraulic generators. Section 2.1 covers the university's robot cell, presenting its critical elements. Section 2.2 outlines Dynaset's real production environment. Section 2.3 reviews the three different hydraulic generator variations and their respective parts, while Section 2.4 details the program's operational flow.

2.1 University robot Cell

The university robot cell serves as the experimental setup where all the project-related processes are developed, tested, and refined. This mockup replicates Dynaset's production conditions, enabling realistic testing that can be extended to the real manufacturing setting. Notably, the mockup adheres to the "worst-case scenario" setup, meaning in cases where test conditions differ from Dynaset's real conditions, it is by adding complexity. This approach aims to ensure that if the system operates reliably under the test conditions, it should also perform reliably within the actual production environment.

The layout of the university robot cell is illustrated in Figure 1 and Figure 2. Figure 1 provides a general overview of the entire system, while Figure 2 offers a view of the interior of the robot cell cabinet.

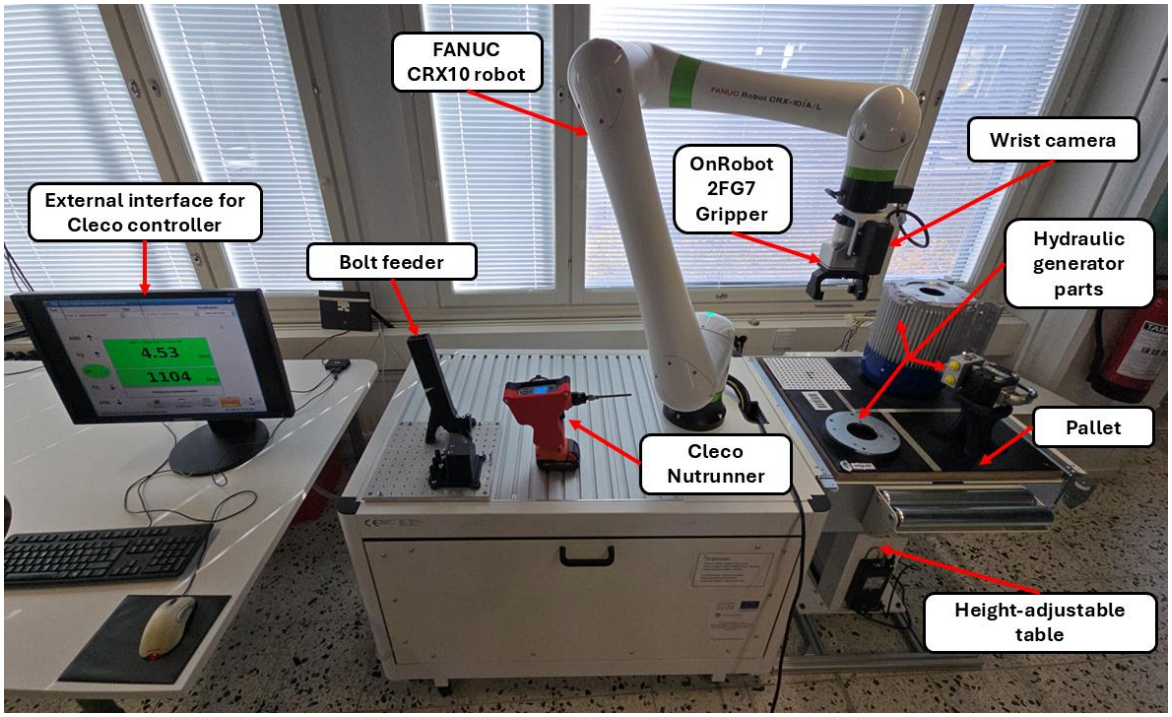

Figure 1: General overview of the university robot cell

Figure 2: Interior view of the robot cell cabinet

The assembly robot cell is based on the FANUC collaborative educational package, which includes the collaborative robot, robot controller, and a trolley table where the robot is mounted. Key components within this assembly cell include:

- **FANUC CRX-10iA/L**: Collaborative robot from Fanuc with a 10kg payload capacity and 1418mm reach.
- **OnRobot 2FG7:** An electric parallel gripper with an 11 kg maximum payload and an adjustable gripping force (20-140 N). This gripper is utilized for all grasping operations by switching between different gripper fingers.
- **Wrist Camera:** Fanuc iRVision 2D camera mounted on the robot's wrist for part detection and localization.
- **Cleco Cellcore Nutrunner**: A cordless battery-powered nutrunner housed within its mounting kit. Includes OnRobot quick changer that fast manual switching between the nutrunner and the gripper.
- **Bolt feeder**: A 3D-printed device that consistently supplies M8 bolts for fastening operations.
- **Workstation table**: A height-adjustable table, identical to those in Dynaset's factory. It includes several pneumatic controls: one to lift the pallet off the rollers, another to stop pallet movement, and a third to lock the table's rotational movement.
- **Pallet**: A thick plywood base with a rubber surface for holding parts during assembly.
- **Screw fastening test station**: A 3D-printed test station with brass inserts, used to test the nutrunners capabilities.
- **Robot controller**: The Fanuc R-30iB Mini Plus controller.
- **Cleco controller**: The Cleco mPro200 series controller handles the programming of the nutrunner and manages communications between the robot and the nutrunner.

2.2 Estimated costs

The used demonstration cell is not indented for industrial use but most of the components are of industrial grade. Price estimates below are tentative and don't include the needed engineering work of safety design nor the needed pedestal for the robot arm.

- **FANUC CRX-10iA/L:** 45000 €
  - Includes gripper&camera
  - Robot alone 33500 €
- **Cleco nutrunner:** 8500 €
  - Includes controller, battery…

2.3 Dynaset's production environment

This chapter provides an overview of Dynaset's actual manufacturing setting where hydraulic generators are currently assembled. It features a larger assembly line where pallets flow from one manufacturing process to the next via a roller conveyor. Figure 3 illustrates the assembly line, highlighting the workstation tables positioned on both sides of the conveyor. These workstations are currently operated manually, with workers performing identical assembly steps on each side.

Figure 3: Layout of Dynaset's production environment

Unlike the university's mockup cell, Dynaset plans to mount the robot on the ceiling. This ceiling-mounted configuration serves two primary purposes. By elevating the robots, valuable floor space is freed up, allowing more efficient use of the production area. Secondly, it enhances

interchangeability between human workers and the robot by ensuring that both human operators and the robot can access the workstation tables without interference.

2.4 Components in assembly

This section provides an overview of all the products and components involved in the assembly process. Dynaset supplied three distinct hydraulic generator products, designated as **variation 1, variation 2, variation 3**, which are depicted in Figures 4, 5, and 6 respectively. The naming scheme aligns with how the generators are referenced throughout this report and in the robot programming code.

Figure 4: Fully assembled hydraulic generator – variation 1

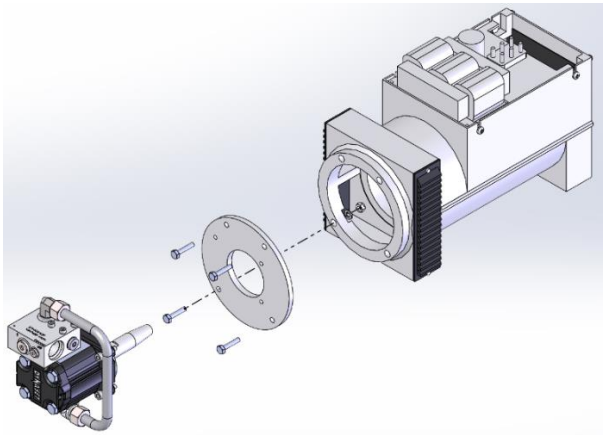

Figure 5: Exploded view of hydraulic generator – variation 2

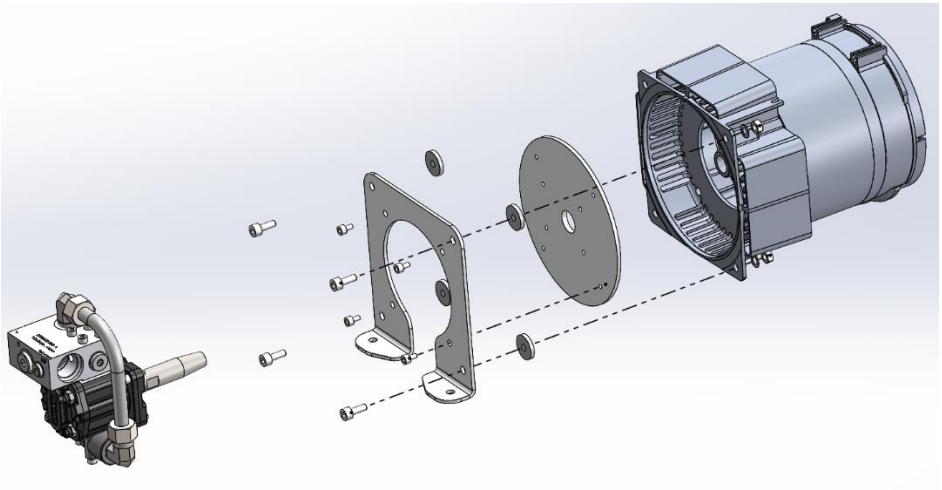

Figure 6: Exploded view of hydraulic generator – variation 3

Each product variation is assembled from three primary components: the generator body, the flange, and the hydraulic motor. In addition to these, the assemblies use different bolts, washers and nuts for fastening. To maintain clarity and consistency, each component type is labeled with a numerical identifier corresponding to its variation. This is summarized in the table 1 below:

Table 1: Marker scale measurements at different table heights

Component	Assembly 1	Assembly 2	Assembly 3
Hydraulic generator body	Body 1	Body 2	Body 3
Flange	Flange 1	Flange 1	Flange 2
Hydraulic motor	Motor 1	Motor 1	Motor 2

Although three variations were provided, the demonstration focused only on variation 1 and variation 2. Both assemblies share the same flange and hydraulic motor, differing only by the generator body. To streamline the fastening process, a design modification was implemented to the type 1 and 2 generator bodies. Instead of using traditional bolt, nut, and washer setup to tighten the flange to the generator body, thread inserts were installed in the existing holes. This allows the flange to be directly installed to the body with only bolts. While this modification may increase the manufacturing cost of the flange, it greatly simplifies the robotization process.

2.5 Program flow

This section provides an overview of the hydraulic generator assembly program's flow, outlining the key steps involved in assembling the hydraulic generator. Figure 7 illustrates the main program flow, serving as a reference to understand the sequence of operations and how various processes interconnect throughout the assembly.

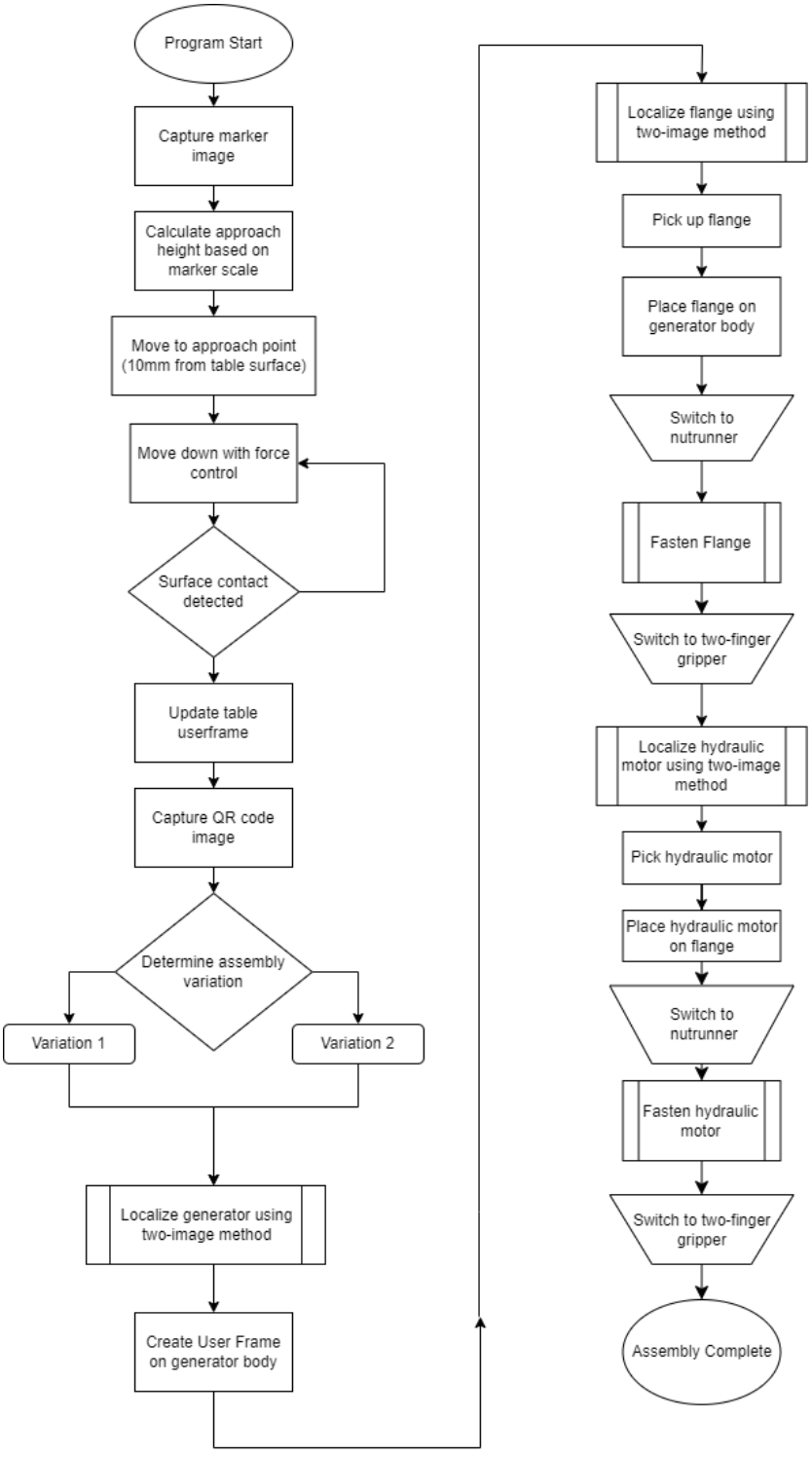

Figure 7: Flowchart of the program flow

The program starts by capturing an image of a reference marker placed on the adjustable workstation table. By analyzing the marker's scale in the image, the robot calculates a safe approach height, positioning its end-effector approximately 10 mm above the table's surface. After moving to this point, the robot employs force control to gently contact the table surface. Upon detecting surface contact, it updates the table's user frame to reflect the current table height, establishing a reference for subsequent operations.

Next, the robot captures an image of a QR code to determine which assembly variation to build. This choice affects specific adjustments in the image-capturing positions. The assembly process begins with the localization of the generator body using the two-image method detailed in Chapter 4.4. This method accurately identifies parts even when their positions and orientations vary. The two-image method repeated for all component pick-ups. Once the generator body's positions and orientation are determined, a user frame is established on the generator body, which is used in subsequent assembly operations.

With the user frame set, the robot localizes and retrieves the flange using the same two-image technique, which is then placed onto the generator body using force control to ensure correct seating. A manual tool change is then performed to switch from the two-finger gripper to the nutrunner. The fastening operation

After setting the user frame, the robot proceeds to localize and pick up the flange using the same two-image technique. With the flange grasped, the robot places it onto the generator body (using force control). A manual tool change is then performed to switch from the two-finger gripper to the nutrunner. The fastening operation (outlined in Appendix B) is initiated to secure the flange. This process involves a loop which repeats four times until all the bolts are fastened.

After fastening the flange, the tool is switched back to the two-finger gripper, and the gripper fingers are changed to handle the hydraulic motor. Again, utilizing the two-image method, the robot localizes, picks up, and places the hydraulic motor on top of the flange.

The tool is then switched back to the nutrunner to fasten the hydraulic motor. Since the hydraulic motor is also mounted with four bolts, the same fastening sequence is repeated. Once all the components are assembled, the tool is switched back to the original two-finger gripper, and the robot is ready to begin the next assembly cycle.

3. Design of gripper fingers and component holders

In this chapter, we discuss the design and selection of gripper fingers and component holders used in the hydraulic generator assembly process. These elements arere critical for ensuring precise handling and placement of component. We also discuss alternative designs that were considered but ultimately not adopted, providing insights into the design decisions and challenges encountered.

3.1 Overview of the OnRobot 2FG7 Gripper

The OnRobot 2FG7 gripper was selected for this project due to its compatibility with collaborative robotic environments, ease of integration, and compact design, which allows for operations in confined spaces and minimizes the robot's payload burden. It features a payload capacity of up to 11 kg in form-fit applications and 7 kg in force-fit scenarios, with a total stroke length of 38 mm. The adjustable gripping force ranges from 20 N to 140 N, enabling the handling of various components required for the assembly.

The gripper operates electrically, eliminating the need for compressed air and simplifying installation. Features such as lost-grip detection further enhance operational accuracy, ensuring stable performance during automated tasks. Detailed component drawings included in the gripper’s datasheet facilitated the design of custom fingers, discussed in the following sections.

The gripper is paired with an OnRobot quick changes, allowing for fast, manual tool changes. While this manual step may affect assembly efficiency in high-volume production, automated tool changers—pneumatic or electric—could be considered in more demanding industrial environments to reduce downtime and improve throughput.

3.2 Gripper fingers Design

3.2.1 Custom gripper fingers

The final robot program employed custom gripper fingers, 3D-printed from PETG filament, to handle different components—specifically, the flange and the hydraulic motor. Figure 8 shows CAD models of these custom fingers: the left set is designed for internal gripping of the flange via its center hole, while the right set provides external gripping for the hydraulic motor. Both sets are form-fitting, ensuring the full 11 kg payload capacity is available for secure handling of all required parts

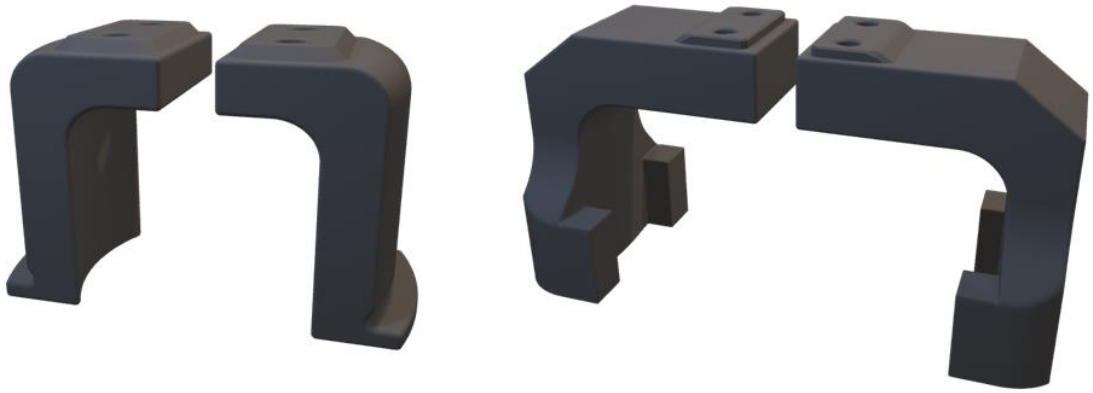

Figure 8: CAD models of the gripper fingers (left=flange, right=hydraulic motor)

A key design goal was to minimize the length of the gripper fingers. The gripper achieves its maximum force of 140 N only when the contact point is near the finger platform, specifically at approximately 36 mm. Therefore, keeping the fingers as short as possible is essential to maintain adequate gripping force. Another important design consideration was the approach clearance, which defines the gripping range of the custom fingers. The design needed to accommodate a total stroke length of 38 mm, ensuring that the fingers could extend towards the part without any physical obstruction while also closing sufficiently to securely grasp the part.

Determining the optimal grasping point also improves stability and self-centering capabilities. Unlike a three-finger gripper, which naturally centers the part, a two-finger gripper requires specific design features to achieve a similar effect. For instance, the flange gripper fingers are rounded on the outer picking side to match the flange's inner diameter, guiding the part into proper alignment even if the initial pick-up position is slightly off-center.

3.2.2 Dual gripper design

A dual gripper design was explored as a potential solution to reduce tool switching during the assembly process. The concept involved a single gripper equipped with two sets of fingers: one set pointing forwards for handling the hydraulic motor, and another set oriented horizontally along the face of the tool flange for grasping the flange. This design aimed to enable the robot to handle both components without the need for manual tool changes, thereby improving efficiency. The dual gripper design is presented in the Figure 9.

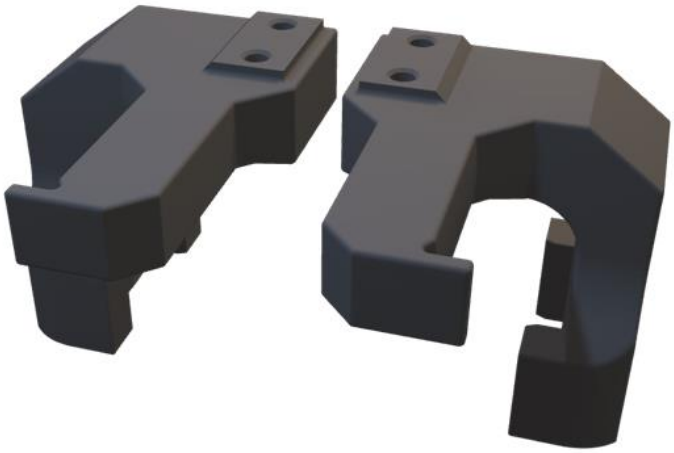

Figure 9: Dual gripper design

Despite its potential advantages, the dual gripper design faced several significant challenges, leading to its exclusion from the final implementation. A primary issue was misalignment caused by the altered orientation of the gripper fingers, which shifted the Tool Center Point (TCP) away from the robot's sixth axis. Typically, rotations are performed around the sixth axis due to its extensive range of motion. However, with the TCP displaced, rotations required the use of multiple axes, increasing the complexity of movements and the risk of encountering non-reachable positions.

These limitations made the dual gripper less suitable for our application, where part orientations and movements are highly variable. The dual gripper design might be more feasible in scenarios with constrained part orientations and limited movement ranges. To capitalize on the efficiency benefits of reduced tool changes, workers would need to ensure that parts are consistently oriented when placed on the pallet. While standardized procedures can ensure consistent part orientation, this limits flexibility, requiring a careful analysis of whether efficiency gains outweigh the potential reduction in adaptability.

3.2.3 Adaptations for assembly variation 3

Assembly Variation 3 introduced a smaller hydraulic motor and a different flange design. The new flange’s very small center hole made it unsuitable for the two-finger gripper approach used in Variations 1 and 2. Without a suitable gripping feature, the 2FG7 could not reliably hold this flange. A suction-based solution was considered as an alternative. The OnRobot VGC10 electric vacuum gripper, which was available for testing, offered adjustable configurations by using different suction cups and adapter plates. Although the VGC10 had sufficient lifting capacity, none of the tested cup configurations could securely grip the flange due to the presence of multiple screw holes that disrupted suction.

In the future, a custom adapter plate could be designed to position suction cups around these holes, potentially enabling a stable vacuum grip. However, due to time constraints and not being the project's primary focus, this approach was not pursued further.

3.3 Component holders design

Component holders are essential in ensuring proper part positioning for robotic manipulation. Without holders, components lying flat on the pallet surface can be difficult to grasp securely, and some parts may not have a stable foundation for accurate positioning. This section discusses the design considerations for hydraulic motor holders and bolt feeders.

3.3.1 Design of the red hydraulic motor holder

The hydraulic motor holder was designed with stability and manufacturability in mind. Its primary purpose is to support the hydraulic motor during both robot pick-up operations and transportation along the conveyor line. Because the holder is not fixed to the pallet, it must remain stable during conveyor movement and sudden stops. To achieve this, the holder features a wide base that lowers its center of gravity and ensures even weight distribution, as illustrated in Figure 10.

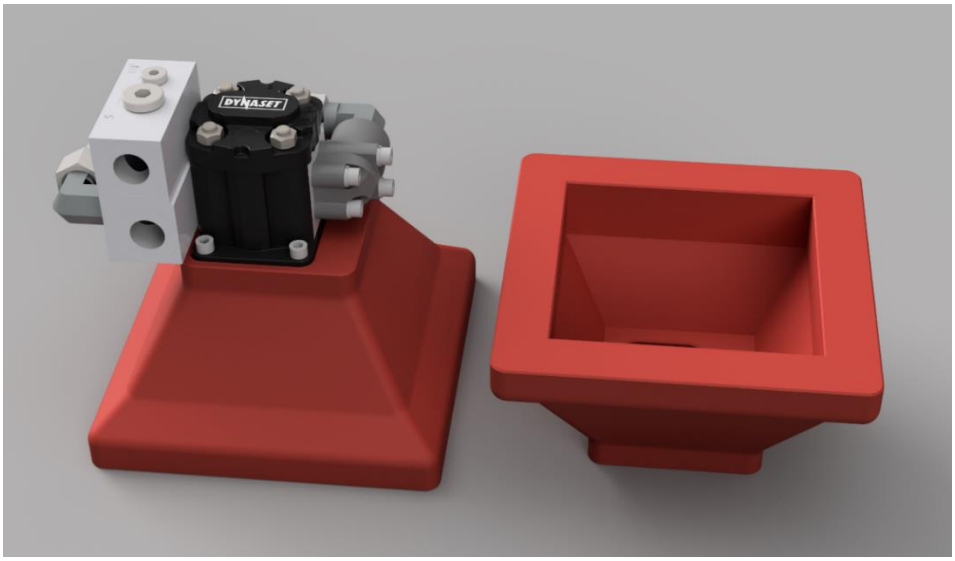

Figure 10: Design of the hydraulic motor holder – Variation 2

Like the other holders, this one was produced using 3D printing, which required additional considerations for manufacturing feasibility. In real production environments where holders are printed, weight and printability are essential considerations. The holder should be designed to use as little material as possible without compromising its structural integrity, thereby reducing both printing time and material costs. This can be achieved by using efficient infill structures with lower infill densities and by reducing wall thickness. Another strategy would be to eliminate unnecessary features, such as removing excess material from non-load-bearing areas. However, this approach

conflicted with another important consideration, printability. The goal was to design the printable model in such a way that it does not require any support material. This minimizes post-processing, allowing the holders to be ready for use directly after printing.

Another advantage of this holder's design is its stackability. Stackable holders conserve storage space, reduce clutter, and improve logistics by enabling faster organization and deployment of multiple units. While this feature offers limited benefit in a test setup producing only a few holders, it becomes highly valuable in a production environment where hundreds of pallets—and thus many holders—must be managed efficiently.

3.3.2 Bolt feeder design

The bolt feeder plays a critical role in the assembly process by supplying bolts for fastening operations. Its function is to align bolts so that the robot can pick them up easily, ensuring a smooth and efficient assembly workflow. Figure 11 shows the bolt feeder developed for this test setup.

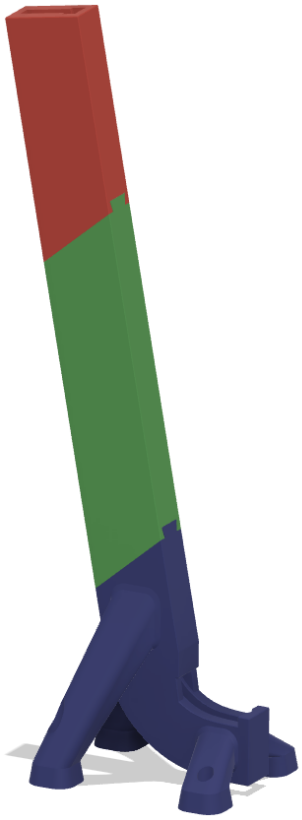

Figure 11: Design of the bolt feeder

The bolt feeder operates as a simple gravity-based system: bolts slide down a chute, naturally orienting themselves for robotic pick-up. This design was chosen for its simplicity and cost-effectiveness, avoiding the complexity, maintenance, and higher costs associated with more advanced mechanical or vibratory systems. A key feature of this bolt feeder is its modular and extendable design. To overcome the constraints of the 3D printer's build volume, the feeder was

printed in multiple sections that could be joined together. This approach allows for easy customization of the feeder's capacity and enables rapid prototyping, testing, and modification of individual sections without reprinting the entire assembly.

In real production settings, gravity-based feeders are often larger, holding thousands of bolts to minimize refilling frequency. Such feeders, although occupying more space, are acceptable in industrial facilities designed for large-scale equipment. However, high-volume applications frequently rely on automatic screw feeders based on vibratory bowl technology. These systems handle larger volumes of fasteners more reliably and require less human intervention.

4. Camera and machine vision integration

4.1 Pallet overview and visual markers

This section introduces the layout of the pallet used in the hydraulic generator assembly, highlighting special organization and the key components used in the machine vision processes. As shown in Figure 12, the primary assembly components—the generator body, the hydraulic motor, and the flange—are arranged on the pallet. Tape lines on the pallet's surface define allowed movement limits for each component, allowing them to rotate or shift without compromising the camera's ability to locate them accurately

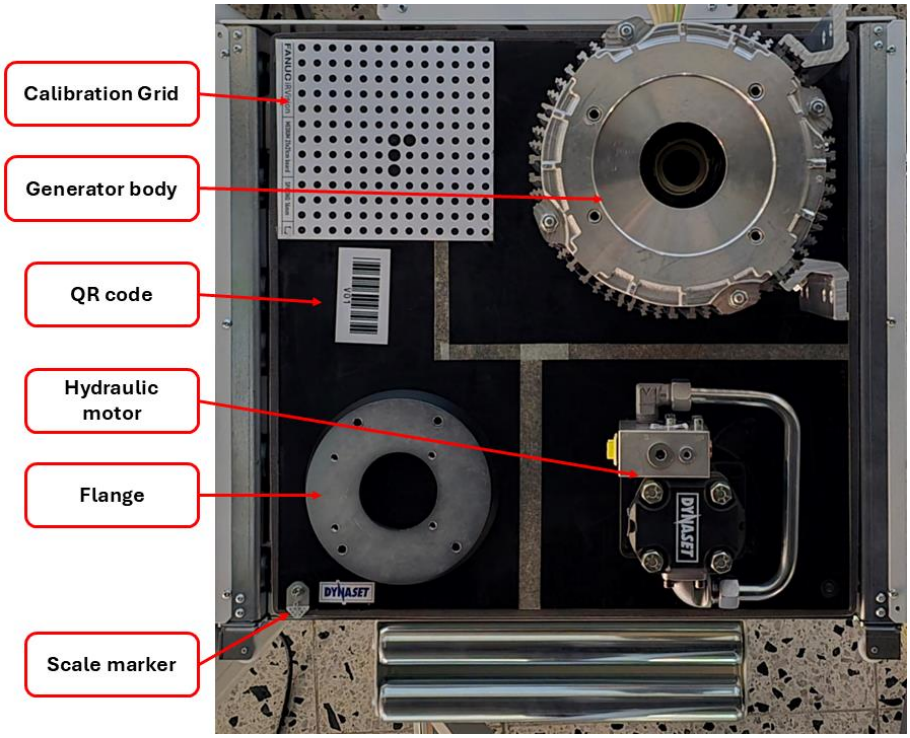

Figure 12: Top-down view of the pallet and key component on it

In addition to the assembly components, the pallet incorporates several crucial visual markers, including a scale marker, a calibration grid, and the QR code. The scale marker is a small and

distinct object integrated into the pallet's design, providing a fixed reference for assessing table height. Its stable placement ensures that height measurements remain unaffected by the parts' positions or orientations.

The calibration grid, positioned in the upper-left corner of the pallet, serves two essential functions. First, it is used for camera calibration which determines intrinsic parameters, including focal length and the optical center. These parameters are crucial for correcting distortions, particularly radial distortion where straight lines may curve near the edges of the image. This effect is especially pronounced with the wide-angle lenses used in this application. Second, the calibration grid defines a default user frame for the table. This user frame can be adjusted later if the table height changes, ensuring consistent spatial references throughout the assembly process.

Although the calibration grid could be used to assess table height, much like the scale marker, it is not employed for this purpose due to spatial and practical considerations. The calibration grid occupies a relatively large area—feasible in a test setup but not ideal in real production environments, where many pallets would need such grids. Using a small, dedicated scale marker instead is more cost-effective and easier to implement in large-scale operations.

4.2 Adjustable table height handling using machine vision and force control

This section outlines the methods developed to enable the robotic system to adjust its approach based on the current table height. By combining machine vision and force control, the system can dynamically adapt to changes in table height. The subsections below detail the initial approach, the encountered challenges, and the improved solution.

4.2.1 Initial solution using linear approximation

The first approach used FANUC iRVision to measure a specific circular feature—the flange's inner diameter—in pixels at different table heights. Two reference values were recorded:

- **Reference Height ($\mathrm{H_{ref}}$):** The baseline table height where initial measurements are taken.
- **Maximum Height ($\mathrm{H_{max}}$):** The maximum tested table height (150 mm above the reference surface)
- **Measured Diameters:**
    - D_{ref}: Diameter at reference height
    - D_{max}: Diameter at maximum height.

The initial solution was based on the assumption of linearity, supposing that the relationship between the measured pixel diameter and the table height would remain linear within the 150 mm range. Under this assumption a pixel-to-height ratio (R) is calculated using the formula:

$$\mathrm{R} = \frac{H_{max} - H_{ref}}{D_{max} - D_{ref}}$$

By capturing an image of the circular feature and measuring the current diameter (D_{curr}), the height offset $(\Delta\mathrm{H})$ can be calculated as follows:

$$\Delta\mathrm{H} = R \cdot (\mathrm{D}_{\mathrm{curr}} - D_{ref})$$

In this setup, the default user frame for the table was defined at its lowest position, and the approach point was set at a safe distance of approximately 10 mm along the z-axis relative to the table frame. The calculated height offset $(\Delta\mathrm{H})$ is used as an offset to this approach position ensuring that the robot consistently maintained the safe distance from the table surface.

For the final 10 mm of movement toward the table, the system employed the collaborative robot's **Push Function**. This function is designed to move the robot towards the table and upon reaching a predefined **target force**, it begins to push with a specified **pushing force** for a set **duration**. The push function is typically used to exert force upon contact, but here it was repurposed to make gentle contact with the surface.

One potential issue arises if the robot moves too far without reaching the target force threshold. In this implementation, the approach point is positioned close enough to the table surface to prevent such problems. However, in an earlier height-check implementation that relied solely on force control, the starting position would need to be set higher than the maximum table height to avoid collision risks. If, for instance, the table height was close to the default position, the push distance would exceed 10 cm, which would trigger an error in the push function due to the excessive distance.

The linear approximation also presented significant challenges, ultimately making it unsafe to use within the program flow. The assumption of a linear relationship between pixel measurements and table height proved inaccurate in practice, with errors especially pronounced at intermediate table heights. At these heights, the approach point was sometimes too close to the table surface, and in some instances, it even positioned below the table surface, causing collisions. Another factor that contributed to the measurement errors was the use of the flange's inner diameter in calculations. Since the flange was not a fixed, its movement in x and y direction impacted diameter measurements, even at the same table height.

4.2.2 Improved solution using quadratic model

To overcome the challenges with the initial method, particularly the inaccuracies at intermediate table heights, a more robust approach was developed. Instead of measuring the flange's diameter, a fixed scale marker on the pallet was used, providing a stable reference independent of the parts' positions or variations.

To determine the actual relationship between the scale percentage measured by the camera and the physical table height, measurements were taken at 10 mm intervals within the 0 to 150 mm range. The data collected is summarized in Table 2 below.

Table 2: Marker scale measurements at different table heights

Difference (mm)	Scale %	Height (mm)
	100	0
2,3	102,3	10
2,5	104,8	20
2,3	107,1	30
2,6	109,7	40
2,7	112,4	50
2,6	115	60
3,1	118,1	70
3,3	121,4	80
3,2	124,6	90
3,4	128	100
3,7	131,7	110
4,1	135,8	120
4,2	140	130
4,5	144,5	140
4,8	149,3	150

Procedure for data collection:

- The table was raised by 10 mm at each step using table controls.
- At each height, a caliber was used to verify that the table reached the exact desired height.
- An image of the fixed marker was captured using the vision system, and the scale percentage was recorded for each increment.

To validate the improved approach, the collected data was analyzed by plotting the scale percentage against the table height, revealing a non-linear relationship that the initial linear model could not accurately represent. A quadratic function was then fitted to the data, achieving an excellent fit with $\boldsymbol{R^2}$ **value of 0.9999**.

The relationship between the scale percentage S and the height H is described by the following quadratic equation:

$$H = a \cdot S^2 + b \cdot S + c$$

Where a, b and c are coefficients determined through regression analysis of the measured data.

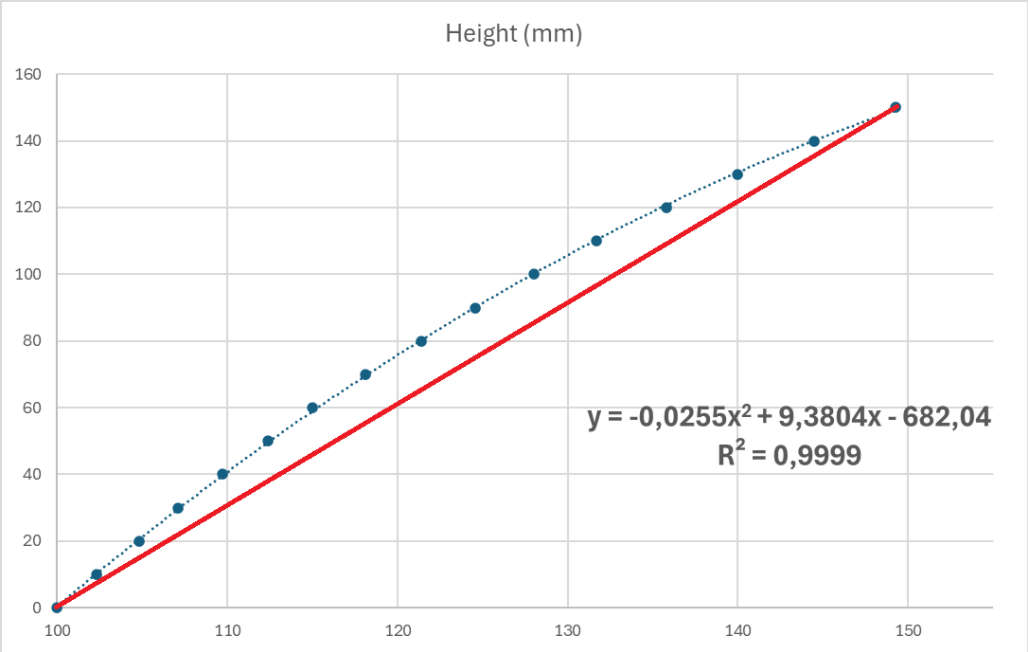

Figure 13: Comparison of measured data and fitted models

As shown in Figure 13, the blue data points represent the actual measured values, while the blue dotted line illustrates the fitted quadratic function. For comparison, the red line indicates the previous linear assumption. The linear approximation clearly fails to capture the non-linear nature of the data, particularly between the maximum and minimum values, which aligns with the previous experiences where the robot occasionally moved too close to the surface.

On the robot's side, the height calculation is implemented by storing each of the calculated coefficients in dedicated register values. By inputting the captured scale percentage into this same quadratic function, the system calculates the required height offset, enabling safe adjustments for varying table heights.

4.2.3 Linear XY-compensation for table height misalignment

The method described in section 4.2.2 provided sufficient accuracy to consistently reach the approach point. However, another source of error was identified during subsequent pick and place operations especially at table height differing from the default bottom position. The core of the issue stems from how the user frame is updated.

In the previous approach, only the z-value of the user frame was adjusted based on the table height. Since user frames are defined relative to the robot's world frame, whose z-axis does not perfectly align with the vertical axis of the table's movement, this adjustment inadvertently

introduced slight shifts in the X and Y directions. These shifts became more pronounced at higher table heights, resulting in minor but noticeable misalignments during pick-and-place tasks.

To observe this misalignment, we used the FANUC IRVision automatic grid frame setting. First, a vision frame was established on the table surface at the lowest height, and the process was repeated at the maximum height. Since this function generates user frames relative to the global user frame, we were able to observe and compare the X and Y directional shifts between the two heights.

To correct for the observed offsets, a linear XY-compensation method was implemented. This method assumes that the offsets increase linearly with the table height within the range of 10 to 150 mm. Using data points measured at the minimum and maximum table heights, we calculated the linear compensations for ΔX and ΔY.

Given measurements:

$$H_{min} = 0mm: \mathrm{X}_{\min} = -227.8\mathrm{mm}, \mathrm{Y}_{min} = 445.7\mathrm{mm}$$

$$H_{max} = 150mm: \mathrm{X}_{max} = -228.9\mathrm{mm}, \mathrm{Y}_{max} = 444.7\mathrm{mm}$$

Calculating the differences:

$$\Delta\mathrm{X} = X_{max} - X_{min} = -228.9mm - (-227.8mm) = -1.1mm$$

$$\Delta\mathrm{Y} = Y_{max} - Y_{min} = 444.7mm - 445.7mm = -1.0mm$$

Determining the slopes:

$$\mathrm{k_x} = \frac{\Delta\mathrm{X}}{H_{max} - H_{min}} == -\frac{1.1}{150} \approx -0.00733\ mm/mm$$

$$\mathrm{k_y} = \frac{\Delta\mathrm{Y}}{\Delta H_{max} - \Delta H_{min}} = -\frac{1.0}{150} \approx -0.00666\ mm/mm$$

Linear correction equations:

$$\Delta\mathrm{X} = k_x \cdot H = -0.00733 \cdot H$$

$$\Delta\mathrm{Y} = k_y \cdot H = -0.00666 \cdot H$$

Here, H represents the current table height obtained from height probing. By applying these correction values, the robot compensates for any misalignment in the X and Y directions as the table height changes.

As seen here, the compensation coefficient is quite small, amounting to a bit over one millimeter at maximum table height. While this finding does not negate the reality of a slight misalignment in the

axis, it suggests that the error observed in subsequent operations were not primarily caused by this misalignment.

4.3 Assembly Variation determination

Automating the assembly of hydraulic generators requires a system capable of distinguishing between different product variations. To ensure that the robot executes the correct assembly sequence for each variation, we utilized Fanuc's barcode reading tools.

Fanuc IRVision provides a versatile barcode reading tool that supports several types of 1D barcodes, including ITF, Code 39, NW7, EAN, and UPC, as well as 2D barcodes such as QR codes and Data Matrix codes. For our application, we conducted tests using Code 39 format barcodes and QR codes. These barcodes contained labels "Var1", "Var2", and "Var3", representing the three different hydraulic generator variations. To facilitate easy identification, a small image of the corresponding assembly was displayed on the backside of each barcode. Since the barcodes were printed on regular paper, we applied lamination to provide extra protection against physical damage. This setup is illustrated in Figure 14 below.

Figure 14: QR code image

During operation, the robot employs the barcode reader program to scan the presented barcode. The detected string is then utilized within the robot's code to determine which assembly variation to execute.

2D barcodes, such as QR codes, are more robust against damage, distortion, or partial obstruction because the data stored in the codes is redundant. This redundancy allows for accurate decoding even if part of the code is damaged or obscured. In our tests under different lighting conditions, QR

codes performed better than 1D barcodes, particularly in darker environments where the lines of the 1D barcodes started to blend into each other, which lead to detection errors.

4.4 Two-image localization method

To ensure flexible and precise localization of various components, the two-image method was implemented. This approach was adopted after it was observed that the accuracy of the vision system varies depending on whether the detected object is positioned at the center of the camera's field of view (FOV) or near its edges. Given that the parts have limited freedom of movement within the pallet, a single image was insufficient to guarantee favorable image-taking conditions.

The two-image method operates independently of specific type of vision process used (template matching, circle finder, blob tool) and follows a straightforward workflow outlined in Figure 11. This method is designed to maintain high accuracy while allowing the flexibility needed in a test setup with variable part positioning.

4.4.1 Two-step imaging process

The two-image method start by capturing an image from a height that allows the entire defined area to be visible within the cameras field of view. This wide FOV ensures that the vision process can locate the object, regardless of its position within the designated limits. Using template matching (or another vision process), the system then performs a preliminary localization based on this initial image. This rough estimate provides enough accuracy to move the robot closer to the object.

With the robot positioned closer, the camera captures a second, more focused image. This adjustment centers the object within the FOV, reducing errors caused by edge distortions. A second round of localization is performed on the close-up image, resulting in a more precise measurement. This refined localization can be used for tasks such as picking up objects or positioning them accurately on the generator body.

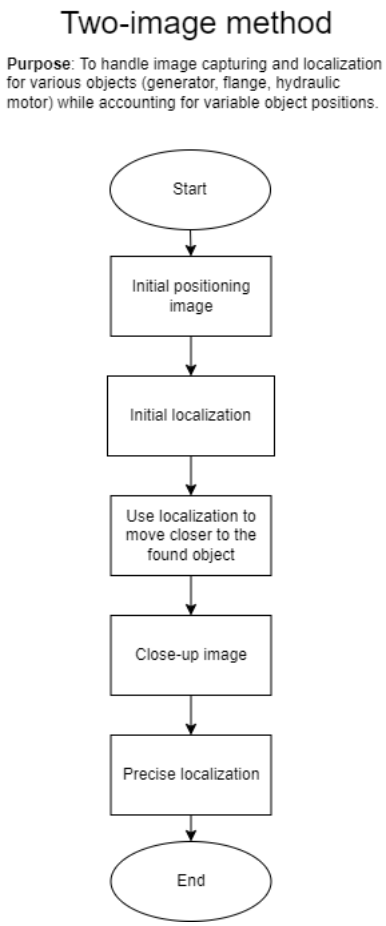

Figure 11: Flowchart for the Two-step imaging process

4.4.2 Efficiency considerations for the two-image method

One potential consideration with the two-image method is efficiency, as taking an additional image introduces extra processing time and robot movements. However, this added step does not present a significant issue in practice due to various settings available in IRVision.

In particular, the parameters for allowable limits in **scale**, **orientation**, and **aspect ratio** play a critical role in optimizing processing times. For the first image, these parameters need relatively loose allowances: orientation can vary by ±180 degrees, and scale can fluctuate by several percentage points, since the detected object's scale can change depending on its position within the camera's FOV. Naturally, these broader settings lead to a longer processing time.

For the second, closer image, however, these limits can be tightened significantly. The scale range, for instance, can be restricted to 0.98–1.02 rather than the 0.85–1.15 used for the initial image. Similarly, orientation can be limited to ±10 degrees, as the close-up image is always taken from a consistent angle. These narrower parameters allow for much faster processing times. What previously required between 2–6 seconds can now be processed in around 100 milliseconds. While the robot requires some time to move between the two image-taking positions, this delay is minor compared to the advantages of the two-image method in terms of accuracy and reliability.

4.5 Comparison of detection methods

In developing the machine vision processes for this setup, three different object detection methods were explored and adapter based on the specific needs of various assembly tasks. This section compares a typical IRVision object detection pipeline with two custom methods implemented to address additional requirements, such as higher accuracy and ease of use.

4.5.1 Standard IRVision detection with relative movements

The typical IRVision object detection pipeline relies on **relative movements**. In this approach, the vision process is taught a reference image along with a corresponding reference pick position that matches the results of this reference image. During runtime, when a new image is captured and the object is detected, the system calculates any positional changes relative to the reference image.

For example, if the object has shifted by 20 mm in the x-direction, 10 mm in the y-direction, and rotated by 30 degrees, these offsets are applied to the originally taught reference pick position, adjusting the robot's movements accordingly. This method proved effective for handling translation, maintaining accuracy even when the object was moved from the reference. However, accuracy declined as the detected object's orientation deviated from the reference orientation. In the current implementation this method is used for picking up the hydraulic motor.

Attempts to address this rotational inaccuracy led to the development of the alternative methods discussed in the following sections.

4.5.2 Absolute position-based detection

The second method, **absolute position-based detection**, is an alternative method that is not dependent on relative offsets. Here, the vision process output provides coordinates (x, y, z) and orientation (pitch, yaw, roll), which the robot directly uses in its program. Rather than applying relative adjustments to a predefined reference position, the robot simply moves to the exact coordinates provided by the vision system.

This method relies on a manually set origin point within the image detection tool, which becomes the reference point for detected parts. As vision system is equipped with a 2D-camera, the output naturally provides only x, y, and roll, to represent the object's position and orientation on a 2D plane. **z-height**, which is the height from the table surface to the detectable feature is known as this information is required in the vision process. The remaining pitch and yaw are assumed to be zero since the detectable features are parallel to the table surface.

In the current implementation, absolute position-based detection is used for picking up the flange. Here, the origin point is set to a clearly identifiable feature—the flange's center hole. This works

well as the center hole is also used as the picking point. However, absolute positioning becomes less practical in placement operations on the generator body, where multiple parts must be positioned at different locations, such as threaded holes for bolts. The thread insert features are also not easily detected by the vision process, making the placement of the origin point less reliable.

4.5.3 User frame-based method

The third method builds upon absolute positioning by creating a user frame directly on the detected object, which becomes the reference for all subsequent operations. Rather than using the vision process output solely as a pick point, this approach defines a custom user frame based on the object's detected position and orientation.

By anchoring the user frame to a specific feature on the object, such as the center hole on the generator body, it becomes an intuitive reference for placement tasks. This approach is particularly advantageous for the hydraulic generator body, where multiple components need to be placed accurately in different positions. Figure 12 illustrates this concept with a CAD drawing of the flange. Since the flange is positioned at the center of the generator body, the placement coordinates are simply set to x=0 and y=0, making it easy to position the component. Fastening points for the flange can also be determined directly from the CAD dimensions, simplifying the setup further.

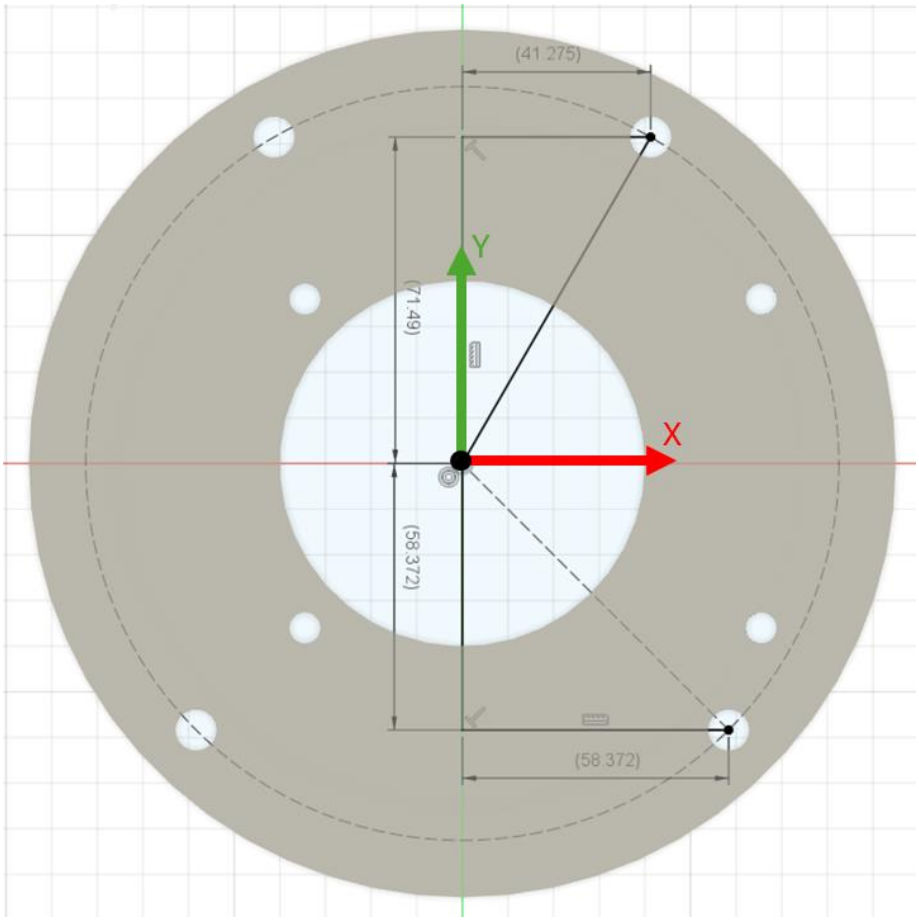

Figure 12: Illustration of the generated user frame on the flange

4.5.4 Ease of use

One key challenge in multi-step assembly processes is managing situations where multiple assembly actions rely on a single vision process result. While the advantages of the different methods were discussed in previous sections, here we compare the methods from the perspective of **ease of use** and **re-teaching**.

Consider a scenario where all assembly operations on the generator body are performed using a single vision process with the standard IRVision method. In this setup, each placing and fastening move is made relative to one reference image of the generator body. For this approach to work, the generator must remain completely static throughout while teaching all placement and fastening operations. Even a slight shift in the body's position during teaching will introduce inconsistencies, requiring the programmer to re-teach all subsequent positions. This limitation is especially challenging in the test setup where parts are allowed some movement on the pallet. Testing robustness of this system requires trying assembly operations with the generator body in different positions. However, moving the generator body on the pallet surface means that the exact reference position is lost, as it difficult to return the generator body to the precise starting point.

To mitigate this issue with standard method, multiple "identical" vision processes were created to capture the generator body's position. While this takes a bit more time as the camera captures and processes three separate images, it enables the assembly process to be divided into manageable parts. In an earlier setup, this division was organized as follows:

- Hydraulic motor and flange placements
- Flange bolt fastening operation
- Hydraulic motor bolt fastening operation

This arrangement allowed each task to be adjusted individually. For instance, once the hydraulic motor and flange placements were reliably taught, the vision process for the flange could be refined without affecting the others.

The user frame-based method, however, eliminates this complexity entirely. When all positions reference a user frame that can be reliably redefined to the generator body, adjusting positions becomes straightforward. In this approach, each position can be modified independently without affecting other steps in the process, making testing and re-teaching much less tedious.

4.5.5 Combining detection tools for accuracy

The choice of detection tools is crucial in ensuring accurate object detection during assembly. Two primary tools—GPM Locator (a template matching tool) and Circle Finder—were explored and refined throughout the development process to improve accuracy.

The GPM Locator is a template-matching tool that detects an object's position and orientation by comparing a taught template to the captured image. This tool is effective for complex shapes with distinct features, making it a versatile solution for various detection problems. In contrast, the Circle Finder tool specializes in identifying circular features with a predefined diameter. It provides a simple, fast, and highly accurate solution for objects with prominent circular elements that can reliably serve as origin points. However, as a rotationally symmetrical tool, Circle Finder detects only the (x, y) coordinates and does not capture the object's orientation.

Positioning using the Circle finder tool can be more accurate than template matching due to fundamental differences in how each method processes visual information. The circle finder often achieves sub-pixel accuracy because it calculates the center based on the best fit of multiple edge points around the circle's perimeter. Additionally, it is robust to changes in lighting, scale, or partial occlusions, if enough of the circular edge is visible to calculate the center accurately. In contrast, template matching is typically limited to pixel-level accuracy and can be sensitive to changes in appearance and lighting. In this test setup where parts can move under the camera's field of view, the parts might appear deformed from different perspectives, meaning the template does not match the detected object well, leading to inaccuracies.

Initially, the GPM Locator was chosen for its ability to define both the position and orientation of objects. However, as the program evolved, we explored additional methods to enhance accuracy. The final solution combined both tools in a hybrid approach to achieve a fully defined, higher-accuracy result. In this process, template matching with the GPM Locator is first used to locate the initial origin point and angle of the object. Then, the Circle Finder refines this detection by precisely pinpointing the (x, y) origin, enhancing the overall accuracy of the vision process.

4.6 General good practices

Implementing an effective machine vision system involves several foundational practices that ensure reliability and accuracy, especially in varied environmental conditions. This section covers general good practices that, while not extensive enough to warrant individual sections, play significant role in system performance and consistency.

4.6.1 Lighting Considerations

Lighting stability and quality are essential for machine vision accuracy, as they directly influence the camera's ability to capture clear, high-contrast images. In our laboratory setup, we intentionally tested our vision system under suboptimal lighting conditions, relying solely on ceiling lights and ambient sources. This setup was designed as a worst-case scenario to evaluate system performance in challenging environments. The limitations of this lighting approach required adjustments through camera settings and robot movement to maintain image quality. In an industrial context, however, establishing a stable and suitable lighting environment is recommended for reliable, consistent image capture over the long term.

Key factors in lighting design include the type, direction, and consistency of the light source, as well as the reflective properties of the materials being imaged. For instance, direct lighting can lead to glare on metallic surfaces, whereas diffuse lighting often minimizes reflections, which is particularly beneficial when imaging reflective materials.

In our setup, the ceiling light source was positioned directly above the pallet, aligned with the robot's wrist camera. This direct overhead lighting caused glare issues, especially on metallic objects, and the laminated surfaces of the barcodes. Unable to adjust the light source itself, we instead moved the robot arm. By shifting from a direct top-down view to a 30-degree angle, we effectively eliminated glare, which had occasionally led to failed QR code detections.

4.6.2 Image Filtering Techniques

Image filtering, or preprocessing, is an essential part of machine vision, as it enhances key image features and reduces noise, thereby making object detection more reliable. Most machine vision systems offer a range of filtering options, such as sharpening filters to accentuate edges and blurring filters, like Gaussian or median blurs, to minimize background noise.

In our tests, the impact of filtering was particularly noticeable in Section 4.3, where we experimented with reading QR codes and 1D barcodes under variable lighting. In low-light conditions, salt-and-pepper noise—random black and white speckles in the image—became a recurring issue, interfering with detection accuracy. To address this, we applied a median blur filter with two iterations, which removed the noise, followed by Gaussian sharpening to enhance the QR code's details. Figure 15 illustrates a comparison of a QR code image before and after applying these filters.

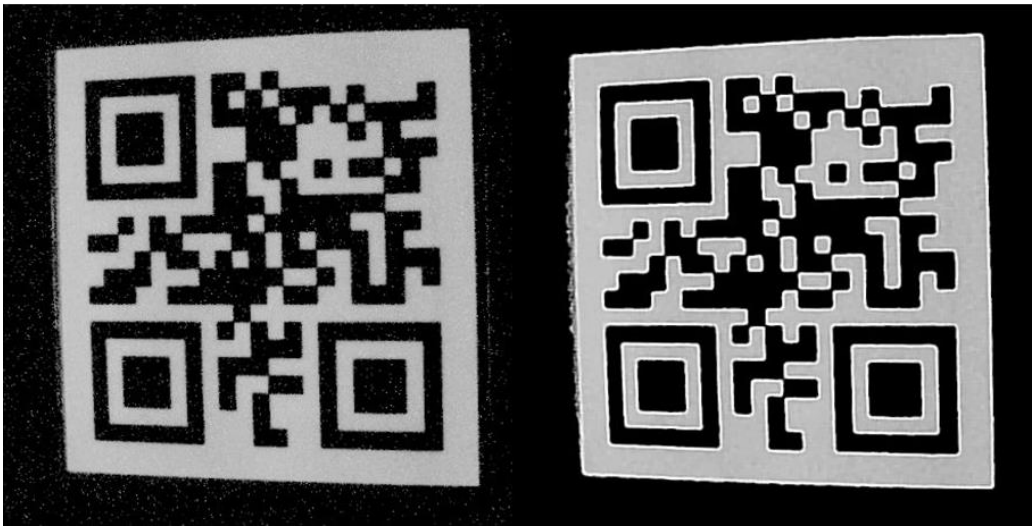

Figure 15: QR code before and after applying filters

5. Integration and mounting of the Cleco Cellcore nutrunner

The integration of the Cleco Cellcore nutrunner into the automated assembly process is a central component of this project. Unlike typical robotic nutrunners, the Cleco Cellcore is a pistol-style tool originally designed for manual use, which presents both challenges and opportunities when adapting it for robotic automation. Robotic nutrunners are often designed with a specialized arm to counteract the fastening torque, redirecting it away from the sixth joint of the robot, which is the weakest in terms of torque capacity. This additional support helps maintain stability and accuracy during high-torque operations. One of the key objectives of this project is to determine how effectively the Cleco nutrunner can operate in an automated environment, despite its manual-use design.

In the following sections, we provide a comprehensive overview of both the physical mounting and communication setups between the Cleco Cellcore nutrunner and the robot controller. This includes the mounting solution used to attach the nutrunner to the robot and the IO connections that enable data exchange. Additionally, the nutrunner's programming capabilities and their implementation in our assembly process are discussed.

5.1 Physical mounting of the nutrunner

The physical integration of the Cleco Cellcore nutrunner began by mounting the tool onto the robot arm using a custom 3D-printed mounting case. As illustrated in Figure 16, the mounting case consists of two parts that secure the body of the nutrunner with bolts and nuts, providing a stable attachment. The mounting case also includes a dedicated slot for an OnRobot quick changer, which facilitates easy manual swapping of the nutrunner, similar to the process used for grippers.

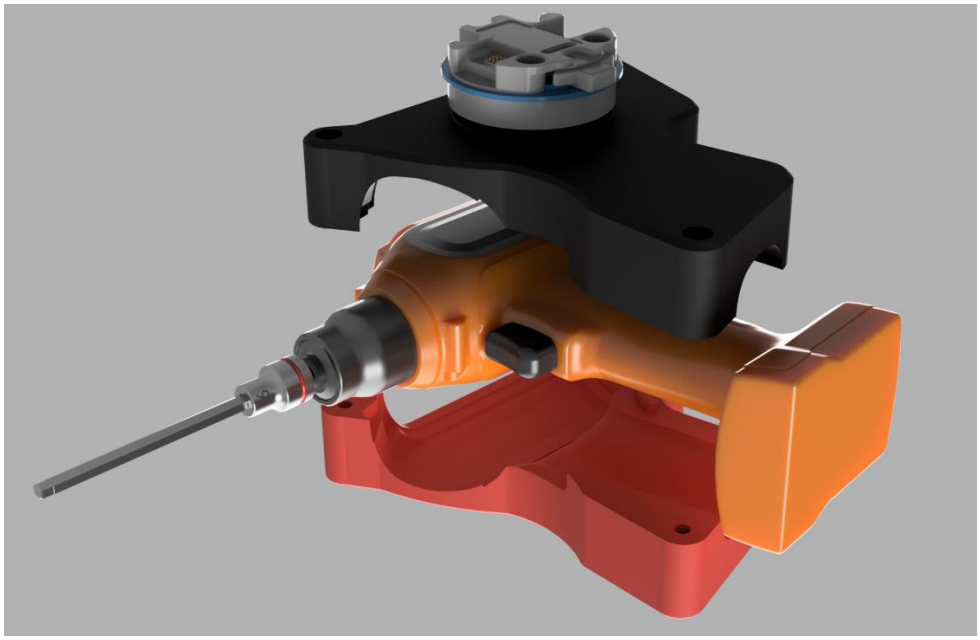

Figure 16: Rendered image of the nutrunner and its mounting setup

The main design consideration for the mounting case was to position the nutrunner perpendicular to the robot's sixth axis. If the nutrunner were aligned along the sixth axis, all the torque generated during the fastening operation would be applied directly to this joint. Since the sixth axis is the weakest in terms of torque-handling capability, this would risk overloading it. By mounting the nutrunner perpendicular to the sixth axis, the torque forces are distributed more effectively, minimizing the strain on this joint and ensuring reliable operation during high-torque tasks.

The nutrunner uses the **Wera 8740 B Hf Zyklop bit socket** for fastening. This is a longer profile version, which allows the tool to access all fastening positions without colliding with other components. It also features a holding function that relies on a ball detent mechanism, pressing against the base of the bolt to ensure the bolt remains securely held during handling.

5.2 Connection setup

The integration of the Cleco Cellcore nutrunner with the FANUC robot controller requires a robust setup, which is divided into two primary components:

1. **Connection between the Nutrunner and the Cleco Controller**
2. **Connection between the Cleco Controller and the FANUC Robot Controller**

These connections enable the direct control of the nutrunner and facilitating seamless command exchange and feedback between the robot and the nutrunner within the automated assembly process.

The Cleco controller is equipped with two Ethernet cards, allowing it to create distinct networks for different communication purposes. One network is used for communication between the nutrunner

and the Cleco controller. The nutrunner supports wireless connectivity via both Wi-Fi and Bluetooth. In our setup, we opted for a Wi-Fi connection to establish communication between the nutrunner and its controller.

The second connection is between the Cleco controller and the FANUC robot controller.
This connection allows the robot to transmit commands to the nutrunner and receive status feedback, enabling synchronized operations during the assembly process. While the Cleco controller supports various connectors and communication protocols, including Ethernet/IP and Fieldbus systems, we selected simple digital input/output (I/O) connections using Phoenix connectors for ease of integration, and compatibility with the existing hardware.

A total of four outputs and four inputs were configured on the FANUC robot controller to establish two-way communication with the Cleco controller. This setup enables the robot to initiate fastening commands, select applications, and receive status feedback from the nutrunner. The outputs from the robot controller, serving as inputs to the Cleco controller, are assigned as follows:

Table 3: Robot Outputs to Cleco Controller

Robot Output (to Cleco Controller)	Function
Output 1	Remote Tool Start
Output 2	Application Selection Bit 0
Output 3	Application Selection Bit 1
Output 4	Application Selection Bit 2

Remote tool start initiates the fastening operation when activated. The application selection bits are used collectively as a 3-bit binary system to select one of up to **eight fastening applications** programmed into the Cleco controller.

The inputs to the robot controller, receiving signals from the Cleco controller, are configured as follows:

Table 4: Robot Inputs from Cleco Controller

Robot Input (from Cleco Controller)	Function
Input 1	TOOL OK
Input 2	TOOL NOK
Input 3	Tool Online
Input 4	Battery Low

- TOOL OK (input 1): Indicates a successful fastening operation.
- TOOL NOK (input 2): Indicates a failed fastening operation.

- Tool Online (input 3): Confirms the nutrunner is connected and ready to use.
- Battery Low (input 4): Indication turns on when the battery level falls below a set threshold.

5.3 Fastening operation backing challenges and solutions

The integration of the Cleco Cellcore nutrunner introduced specific challenges during the retraction phase of the fastening operation. These challenges are primarily due to the design of the hexagon bit used with the nutrunner and the robot's sensitivity settings.

The hexagon bit utilized is an extended version with a slight clearance that allows the tip to move slightly. This design is beneficial during bolt insertion, as it helps guide bolts into their respective holes by accommodating minor misalignments. Additionally, there is extra clearance when the bolt is attached to the bit socket, resulting in the bolt not being perfectly perpendicular to the axis of rotation. While this additional clearance can be advantageous for guiding bolts, it also introduces uncertainty regarding the bolt's exact position.

After fastening, the robot initiates a retraction maneuver to pull the nutrunner tip away from the screw. Due to possible misalignments and slight positional adjustments made during insertion, the robot's end position may differ from its initial position. Attempting to return the tool to the exact starting point can result in unintended lateral forces being applied to the nutrunner tip. These forces can exceed the robot's safety thresholds, potentially triggering a safety stop and causing the assembly process to fail.

5.3.1 Initial solution: Fixed point backing

The most straightforward approach was to program the robot to retract to the exact point where it began the fastening operation. This method assumes perfect alignment between the starting and ending positions. However, due to the hexagon bit's clearance and the minor adjustments made during screw insertion, this assumption proved invalid. The robot occasionally encountered resistance during retraction, leading to unintended forces on the nutrunner tip and triggering safety stops.

5.3.2 Improved solution: Utilizing local positioning with low sensitivity

To address these issues, a combined solution employing the 'Switch to Low Sensitivity' setting and the FANUC Local Positioning (LPOS) feature was implemented. The 'Switch to Low Sensitivity' setting temporarily reduces the robot's force limit sensitivity, allowing the robot to tolerate minor force deviations without triggering a safety stop. The LPOS function then captures the robot's exact position at the end of the fastening operation, saving this position to a position register so that the robot can initiate retraction from the location where the screw was secured. This ensures that the retraction movement aligns with the actual tool position, accounting for any positional adjustments

made during insertion. This solution significantly reduced the occurrence of safety stops, improving the system's overall reliability.

5.4 Torque handling test

This project aimed to evaluate the performance of the Cleco Cellcore nutrunner in an automated setup despite being designed for manual use. A big part of this test was figuring out how much torque the nutrunner could handle without causing safety issues or putting too much strain on the robot's joints, especially the sixth axis, which is the weakest joint when it comes to handling torque.

To evaluate the torque-handling capabilities of the Cleco nutrunner, a series of tests were conducted where an M8 bolt was fastened into the generator body at different torque values, ranging from 1 Nm to 21 Nm, the operational limits of the nutrunner. The test setup ensured perfect alignment, thereby eliminating positional errors as a variable. This provided an ideal baseline to understand the relationship between torque values and the operational limits of the collaborative robot setup. The test aimed to establish a threshold at which torque would start causing either safety triggers or operational instability. The results were categorized into three distinct levels of behavior:

- **Torque Levels 0–10 Nm**: The fastening operations were performed smoothly without any noticeable strain on the robot joints. The Cleco nutrunner did not trigger safety stops or errors.
- **Torque Levels 10-18 Nm**: In this torque range, a different type of failure was observed. While the robot did not outright trigger immediate safety stops, there was a noticeable recoil in the robot arm after fastening. This recoil introduced a twisting motion in the robot arm, which sometimes resulted in contact force exceeding errors during the retraction phase.
- **Torque Levels 18-21 Nm**: At these torque levels, the system experienced outright failures at the end of fastening operation. An alarm would be raised at the end of the fastening process due to excessive torque, which would trigger the robot's safety stops.

The tests highlighted that the Fanuc robot with the current mounting setup could operate effectively at torque levels up to 10 Nm without compromising system stability. Once torque levels exceeded 10 Nm, the recoil effect introduced a potential risk of failure during the retraction phase, and above 18 Nm, the system's reliability was compromised, with frequent safety stops.

In the future, it might be helpful to repeat this study with other robot brands that have similar payload capacities to see if they can handle the tightening operation better. Future work could also explore different mounting setups for connecting the nutrunner to the robot's tool flange, such as using torque reaction arms or adjusting the fastening angles, to see if higher torque levels could be managed more effectively.

6. Conclusion

The project presented here demonstrates how a collaborative robotic system can be implemented for assembling hydraulic generators, going through mechanical design decisions and machine vision techniques. By focusing on a worst-case scenario method that introduced complexities beyond those found in Dynaset's actual production environment, the work aimed to ensure that the developed solutions are robust and transferable to real-world manufacturing conditions. Key challenges included handling adjustable workstation table heights, compensating for non-standardized lighting, and ensuring accurate part localization despite variable component positioning on the pallet.

Adapting to variations in table height was a central focus area. Initial attempts using a linear approximation to relate camera measurements and table height were inadequate, often placing the robot's end-effector too close to the table surface. Introducing a quadratic model and a fixed scale marker as a stable reference allowed the robot to calculate height offsets reliably, even at intermediate table positions. This approach, along with minor XY-compensations, ensured accurate robot operations regardless of the table height.

Integrating machine vision methods to handle variability in part location and orientation was another key aspect of the project. A two-image localization technique – capturing an initial wide field-of-view image followed by a refined close-up – was essential for achieving high accuracy. Narrowing scale and orientation limits during the second image capture significantly reduced processing times without compromising precision. Analysis of detection methods emphasized the benefits of absolute positioning and user frame-based strategies, which simplified re-teaching positions. Anchoring a custom user frame to a known part feature further streamlined assembly tasks and improved reliability under changing conditions.

In parallel with these machine vision improvements, the project tackled mechanical design related considerations. Custom 3D-printed gripper fingers were developed to handle varying shapes and sizes of components, ensuring stable and self-centering grips. Component holders and bolt feeders were designed with printability and stackability in mind. Incorporating threaded inserts into the generator bodies eliminated the need for nuts and washers, simplifying the fastening process and making it more suitable for robotic handling.

The integration of the Cleco Cellcore nutrunner demonstrated the feasibility of of incorporating a handheld-style tool in an automated process, though it also highlighted limitation when working across its full torque range. Operations were stable up to moderate torque levels, but higher torques introduces recoil effects alignment issues. Utilizing FANUC's local positioning and reduced sensitivity setting helped mitigate some of these problems. Although the nutrunner's design posed constraints, the lessons learned provide a foundation for future optimization, whether by exploring alternative mounting setups, or different robot brands better suited to high-torque operations.

Ultimately, the project's outcomes suggest that achieving at least a five-minute cycle time is a realistic goal. By examining each step in the assembly process—detection, picking, placing, and fastening—and documenting encountered challenges and their solutions, this work provides a reference that can help other people avoid similar issues. While these approaches were tailored to a specific product line, robot model, and set of conditions, the underlying techniques—such as machine vision techniques, gripper design ideas—can be adapted to other robotic assembly tasks.

APPENDIX A: Flow Chart Subprogram