UNIVERSITÀ
DI PAVIA

FACULTY OF ENGINEERING DEPARTMENT OF ELECTRICAL,
COMPUTER AND BIOMEDICAL ENGINEERING

MASTER'S DEGREE IN INDUSTRIAL AUTOMATION ENGINEERING

Thesis Title

Development of a flexible robotic system with a custom end-effector for helmet mechanism endurance testing

Supervisor:
Prof. Hermes Giberti

Co-supervisor:
Davide Zhou

Candidate:
Sanaz Isar 531552
Farshad Roshani Kohal 531555

A.Y. 2025/2026

Abstract

In modern product development, testing systems are increasingly expected to combine repeatability with the ability to adapt to different product geometries. Dedicated mechanical rigs can provide reliable operation for a specific component, but their fixtures and actuation mechanisms often require substantial redesign when the product or test sequence changes.

This thesis presents the design and preliminary experimental validation of a robotic platform for repetitive functional testing of motorcycle helmet mechanisms. The system combines a Dobot CR10 collaborative robot, a custom 3D-printed end-effector, a stepper-motor actuation mechanism, an Arduino-based low-level controller, and computer-based supervisory control. The robot provides spatial positioning, while the end-effector performs the local opening and closing action required by the selected helmet component.

The platform was evaluated on the main visor, internal sun visor, upper air vent, and chin mechanism of one helmet, together with the main visor of a second helmet having a different geometry. Each configuration was operated for twenty consecutive cycles. The tests verified successful automated actuation, collision-free execution, and cycle-to-cycle consistency of the controller-reported trajectories and timing. The quantitative analysis is therefore interpreted as a preliminary assessment of system consistency rather than as an independent measurement of absolute robot accuracy or long-term component durability.

The results support the feasibility of using a reprogrammable robotic platform for laboratory-scale functional testing while reducing repetitive manual intervention

Keywords:

Industrial robotics, functional testing, trajectory repeatability, robotic end-effector, helmet mechanism

Contents

INTRODUCTION

Protective helmets are safety-critical products whose movable mechanisms, including visors and ventilation controls, must remain functional during repeated use. In addition to impact and retention requirements, manufacturers therefore need practical methods for checking whether these mechanisms can be operated repeatedly under controlled and reproducible conditions [1-3].

Standardized helmet testing primarily addresses safety performance such as impact attenuation, penetration, and retention. The present work does not replace those conformity tests. Instead, it addresses a complementary engineering problem: the automated repetition of functional opening and closing motions during prototype evaluation [1,2].

Dedicated cyclic rigs commonly use servo, pneumatic, or stepper-driven linkages to reproduce a prescribed motion. Such systems can be effective for a fixed product, but the fixture and linkage are normally tailored to one geometry. Product changes may therefore require mechanical modification, new calibration, and additional setup time [4,5].

A programmable robot offers a different approach. The spatial path can be modified in software, while a purpose-built end-effector provides the local mechanical interaction. This separation between robot positioning and end-effector actuation can improve reconfigurability when the location or orientation of a helmet mechanism changes [6-8].

Industrial robots are particularly suitable for repetitive operations because they can execute predefined sequences with consistent timing and without operator fatigue. Their use in a test cell can also reduce operator-dependent variability and make the experimental procedure easier to document and repeat [6,7,9].

Robot performance terminology must nevertheless be used carefully. ISO 9283 distinguishes pose accuracy, pose repeatability, path accuracy, and path repeatability. The measurements analysed in this thesis are obtained from the internal controller feedback of the Dobot robot. They are therefore used to evaluate controller-reported trajectory consistency and are not presented as an independent ISO 9283 accuracy test [10].

The Dobot CR10 provides six degrees of freedom, a nominal 10 kg payload, a 1300 mm working radius, and a manufacturer-specified repeatability of ±0.03 mm. These characteristics, together with network-based programming, made it a suitable platform for the proof-of-concept developed in this work [11].

The objective of this research is to design, integrate, and preliminarily validate an automated platform for repetitive functional testing of motorcycle helmet mechanisms. The study evaluates whether one robotic cell and one end-effector can execute predefined actuation routines on several components and can be reconfigured for a second helmet geometry mainly by changing robot target positions.

The following chapters review related robotic testing approaches, describe the mechanical and control architecture, document the experimental protocol, and analyses the resulting controller-reported motion data. The scope is limited to functional feasibility and short-run consistency; force, wear, and certified durability are outside the present validation.

The thesis is structured as follows:

- **Chapter 1**: **State of the Art and Research Gap** reviews conventional and robotic testing approaches and identifies the research gap.
- **Chapter 2**: **Aim, Objectives and Scope** define the objectives and the scope of the work.
- **Chapter 3**: **Robotic Platform Design and Experimental Methodology** presents the system design, integration, and experimental procedures,
- **Chapter 4**: **Experimental results and Analysis** present and analyses the experimental results.
- **Chapter 5**: **Conclusions and future work** summarize the main findings, limitations and future developments.

Chapter 1: STATE OF THE ART

The increasing demand for reliable and rapidly reconfigurable test systems has encouraged the use of industrial robots beyond conventional manufacturing. Repetitive functional testing is a relevant example because it requires controlled motion, repeatable sequencing, and adaptation to product variants [12,4].

Traditional test rigs remain appropriate when a single component must be exercised for very large numbers of cycles. Their principal limitation is product specificity: the actuator, linkage, and fixture are normally designed around one geometry and one motion profile.

Robotic platforms provide a complementary solution when flexibility is important. Their multi-axis motion can be reprogrammed, and different interaction tools can be mounted on the same robot [6,8]. This chapter reviews conventional rigs, robot performance terminology, and representative robotic testing applications before defining the research gap addressed by this thesis.

Table 1.1 summarizes the main conceptual differences between dedicated mechanical test rigs and robot-based testing platforms. Dedicated systems are typically optimized for a specific component and motion sequence, whereas robotic platforms shift part of the reconfiguration effort from mechanical redesign to programmable motion and end-effector adaptation.

Table 1.1. Comparison of traditional vs robotic testing

Traditional	Robotic
fixed setup	programmable
single product	multi-product
low flexibility	high flexibility

The comparison does not imply that robotic systems are universally superior; rather, the preferred solution depends on the required cycle count, product variability, motion complexity, and test objective.

1.1 Conventional Testing Systems for Mechanical Durability

Dedicated mechanical test rigs typically combine an actuator, a linkage, a rigid specimen fixture, and a control unit [13]. Electric servo motors provide accurate position and speed control, pneumatic cylinders offer simple and robust actuation, and stepper motors are useful when discrete commanded motion is sufficient [5,14].

The strengths of a dedicated rig are simplicity, high cycle capacity, and a mechanical design optimized for one task [13,15]. Its weakness is limited reconfigurability: a change in helmet shell, visor tab, or mechanism location may require a new fixture or linkage.

Lu et al. [16] developed a miniature durability test benchfor electric sliding-door components. The authors developed a compact test bench specifically for the electronic lock system and power-output mechanism of the door. The bench reproduces the required trajectory and resistance using a four-bar mechanism, spring elements, and a dedicated powertrain. Although the system provides an economical and effective solution for high-cycle durability testing, its mechanical architecture is closely tailored to the specific sliding-door application [16].

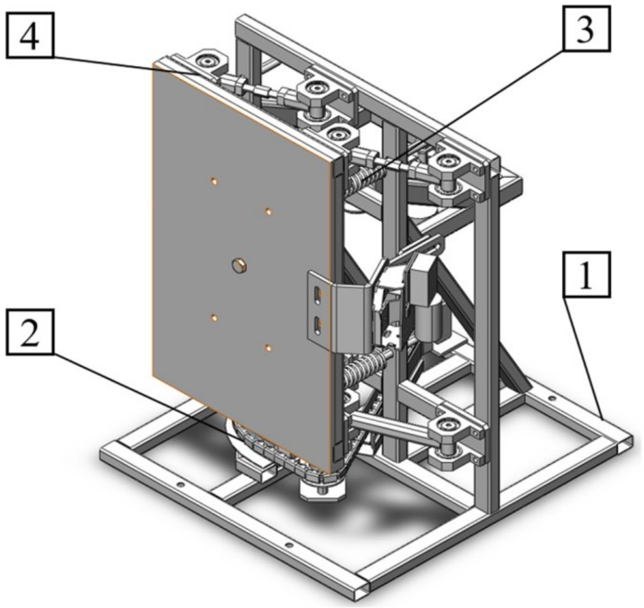

Figure 1.1. Miniaturized durability test bench for electric sliding-door components

1. bench system
2. powertrain system
3. spring resistance system
4. door system

As shown in Figure 1.1, the testing architecture relies on several dedicated mechanical subsystems specifically arranged around the sliding-door mechanism

For helmet mechanisms, the repeated operation of a visor or vent can be reproduced with a purpose-built actuator. However, such a rig normally provides one or two controlled axes and is not intended to approach several mechanisms located at different positions on the same product.

The architecture of a conventional setup can therefore be summarized as actuator, transmission, specimen fixture, and optional feedback. This application-specific structure motivates the use of a programmable robot when several product variants or interaction directions must be accommodated.

A robotic platform does not remove the need for a secure fixture or a suitable contact tool, but it transfers much of the reconfiguration effort from dedicated mechanics to trajectory programming and end-effector design [6,8].

The appropriate choice depends on the test objective. Dedicated systems remain preferable for standardized high-cycle qualification, whereas robotic systems are attractive for flexible laboratory testing, early design evaluation, and mixed product families.

For this thesis, the relevant advantage is the ability to reach multiple helmet mechanisms with one six-axis platform while keeping the local actuation mechanism unchanged.

The following sections therefore focus on robot performance terminology and prior examples of robots used as automated testing platforms.

1.2 Industrial Robot Performance Standards and Repeatability

ISO 9283 defines performance criteria and test methods for manipulating industrial robots [10], including pose accuracy, pose repeatability, path accuracy, and path repeatability. Accuracy describes agreement with a commanded or reference value, whereas repeatability describes the dispersion obtained when the same condition is reproduced [10].

The standard requires controlled test conditions and an appropriate external measurement procedure. Consequently, internal controller coordinates alone are not sufficient to claim an ISO 9283 accuracy measurement. In this thesis, controller feedback is used only to compare repeated executions of the programmed trajectories.

Repeatability can be influenced by payload, robot configuration, temperature, mechanical compliance, speed, and the stiffness of the tool and fixture [10,17]. For a robotic test cell, the complete interaction chain must therefore be considered rather than the nominal robot specification alone.

Each investigated test routine was executed for twenty cycles. This number is sufficient for a preliminary check of short-run functional consistency, but it is not a long-term fatigue campaign and cannot establish wear resistance or service-life durability.

1.3 Robots in Testing Applications

Industrial robots have been used as testing platforms when a specimen must be operated repeatedly, approached from several directions, or instrumented during motion. Their value lies in programmable trajectories and the possibility of integrating force, vision, or non-destructive-testing sensors.

Sotgiu et al. developed an automated robotic test bench for a helicopter sliding door [18]. The system used industrial robots and force measurement to automate a long-duration campaign reported at 500,000 operating cycles, demonstrating the feasibility of robot-based repetitive mechanical testing (Figure 1.2).

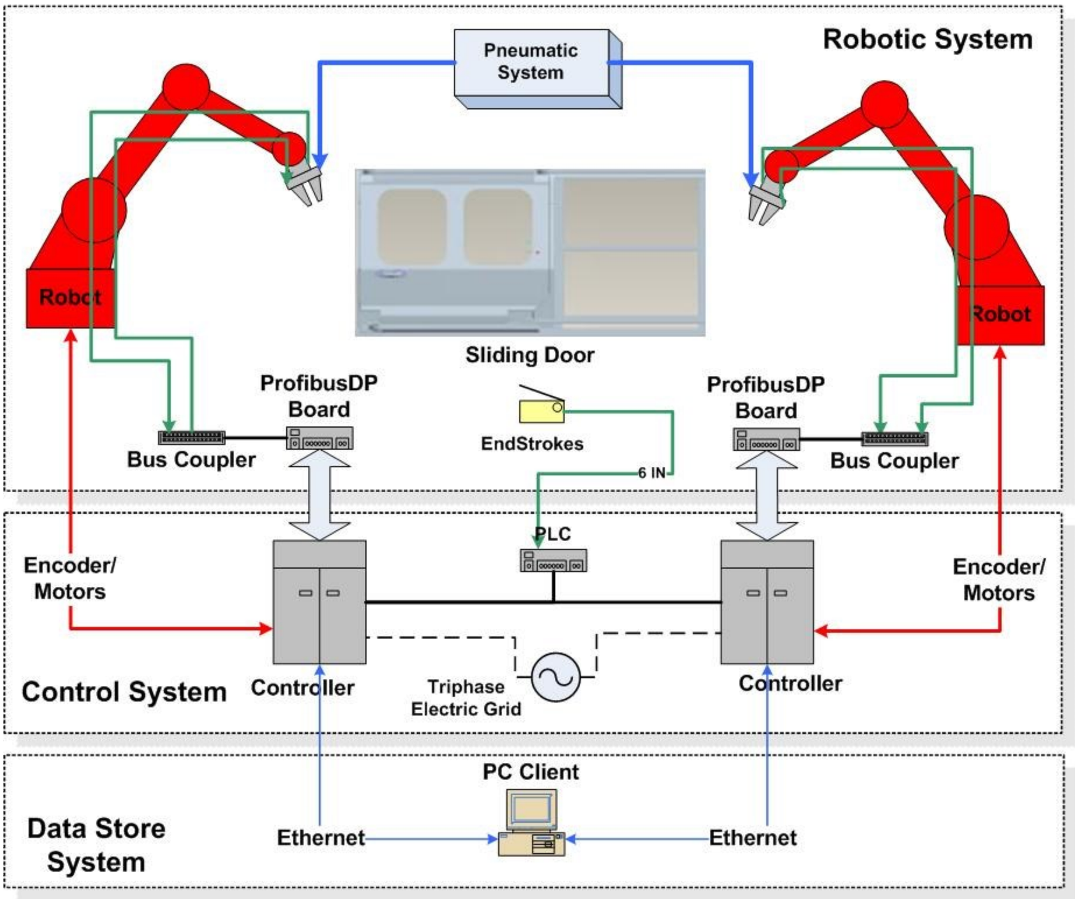

Figure 1.2. Robotic work-cell architecture for sliding-door fatigue testing

Klankers et al. investigated robotics-based, human-like testing of an upper-body exoskeleton using standard robotic manipulators as motion generators [19]. The authors evaluated three different approaches for reproducing human-like interaction with the exoskeleton, including direct use of the robotic arm, a customized robot–exoskeleton attachment, and a human-arm dummy mounted on the robot end-effector. Their study demonstrates the potential of robotic platforms for objective and repeatable exoskeleton testing while also highlighting the need for specialized hardware and algorithms to realistically reproduce human interaction conditions. (Figure 1.3)

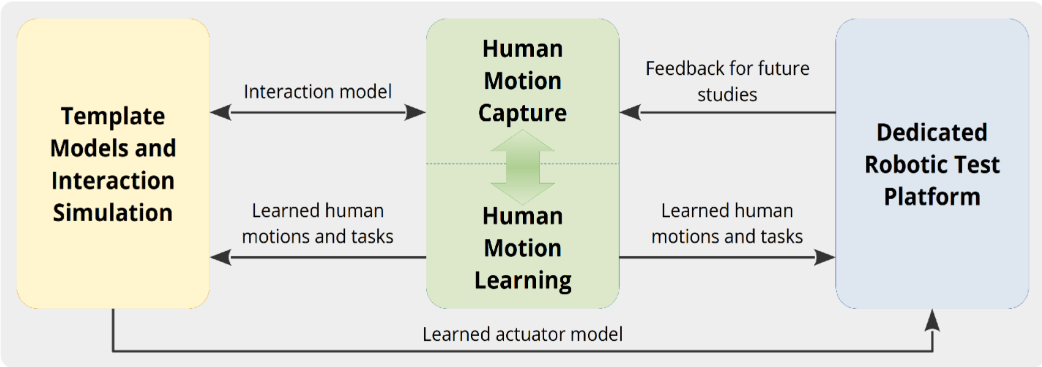

Figure 1.3. Research domains involved in robotics-based testing of upper-body exoskeletons

Hanke et al. proposed a software-defined robotic testing facility for flexible mechanical component testing. (Figure 1.4) The system uses six-axis industrial robots to apply loads to components from different directions and to accommodate specimens with different sizes and geometries. In contrast with dedicated test machines, the proposed approach transfers part of the test reconfiguration from fixed mechanical hardware to programmable robot motions and sensor-based control [20].

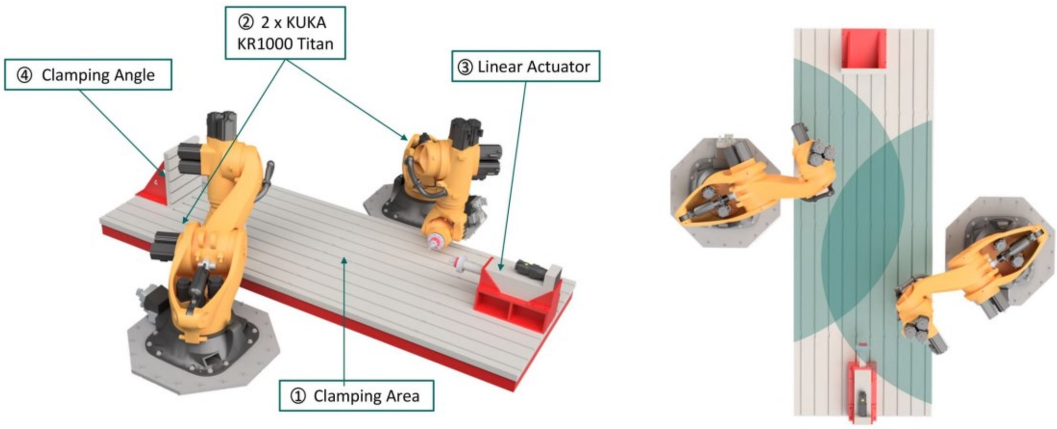

Figure 1.4. Flexible robot-based facility for mechanical component testing

Figure 1.4 illustrates how the use of multi-axis robots allows the loading direction and component position to be changed without redesigning the complete mechanical test structure.

A further example was presented by Valera et al., who developed a robot-based platform for automated durability testing of automotive seat fabrics. Human ingress and egress motions were measured and subsequently reproduced by an ABB industrial robot equipped with a force sensor and a dedicated dummy. The approach demonstrates how a robot can reproduce repeated human-like motion and contact loading for durability evaluation [21].

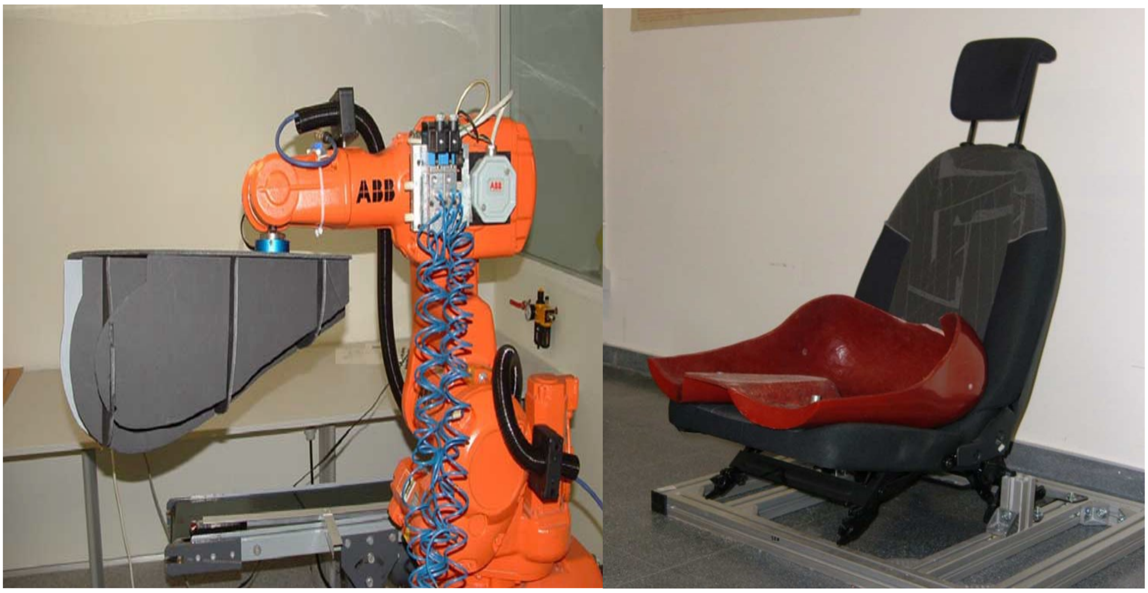

Figure 1.5. Industrial robot and test dummy used for automated anthropomorphic seat testing

This example shows that robotic testing can reproduce not only simple linear loading but also more complex interaction patterns derived from human motion. (Figure 1.5)

Collaborative robots have also been introduced into industrial testing and inspection tasks. Müller et al. developed a collaborative testing system for automotive final assembly using a UR10 robot and a thermographic camera. The system was designed to automate water-leak inspection while maintaining the flexibility required for integration into an existing production environment [22].

Figure 1.6. Collaborative robotic system for automated automotive water-leak inspection

As shown in Figure 1.6, the collaborative robot provides a reprogrammable platform for positioning the inspection tool directly around the product being evaluated.

Industrial robots are also widely used in non-destructive testing [23,24]. Robot-guided ultrasonic systems can maintain a sensor path over complex geometries while improving coverage and reconfigurability compared with fixed inspection equipment [23].

Recent developments in robotic testing have further expanded the role of robots from simple motion reproduction platforms toward intelligent inspection and evaluation systems. Robotic non-destructive testing approaches combine industrial manipulators with sensors, path planning algorithms, and data processing methods to improve accessibility, repeatability, and automation in complex inspection tasks [25,26].

The integration of collaborative robots with monitoring and sensing capabilities has also increased interest in flexible laboratory work cells, where robotic motion, tool interaction, and collected data can be coordinated within a single experimental platform [27]. Moreover, robotic grasping and interaction research highlights the importance of adaptable end-effectors when dealing with components having different geometries and uncertain contact conditions [28].

These examples demonstrate the flexibility of robotic testing, but they also show that credible results depend on an application-specific tool, a clearly defined metric, and suitable sensing. A robot is therefore an enabling platform rather than a complete test method by itself.

The present thesis applies this principle to repetitive functional actuation of motorcycle helmet mechanisms. The contribution is the integration of one robot, one custom end-effector, and several software-defined approach trajectories. The study is intentionally positioned as a proof-of-concept rather than as a standardized fatigue or certification test.

Comparison criteria	Conventional testing system	Robotic testing systems
System configuration	Dedicated fixtures	Reprogrammable platform
Flexibility	Low	High
Motion capability	Limited motion (1–2 DOF)	Multi-axis motion (up to 6 DOF)
Reconfiguration effort	Mechanical modification required	Mainly software and trajectory modification
Product adaptation	Product-specific setup	Adaptable to different geometries
End-effector/ Tooling	Dedicated actuation hardware	Modular or adaptable end-effector
Motion profile	Simplified predefined motion	Programmable complex trajectories
Scalability to product variants	Limited	Higher

Figure 1.7. Qualitative comparison of conventional and robotic testing systems across key system characteristics

Figure 1.7 compares conventional and robotic testing systems in terms of system configuration, flexibility, motion capability, reconfiguration effort, product adaptation, tooling, motion profile, and scalability to product variants. Conventional systems are typically designed for specific components and fixed motion sequences, whereas robotic systems offer greater flexibility through programmable multi-axis motion and adaptable end-effectors. The comparison is qualitative and does not imply that robotic systems are superior for every testing application.

Taken together, these studies show that robotic platforms can support a wide range of testing tasks, from mechanical loading and repeated human-like interaction to sensor-based inspection. Their main advantage is the ability to modify trajectories, interaction directions, and end-effector functions without constructing a completely new test machine. However, the reported applications are generally designed around one specific component, inspection method, or testing objective.

1.4 Research Gap

Prior work confirms that industrial robots can automate repetitive operation, human-like interaction, and sensor-guided inspection [18,19,23,24]. However, published examples are generally developed for one specialized object or one sensing task.

Recent reviews confirm that the use of collaborative robots in quality control remains relatively limited compared with their use in handling and assembly. Puttero et al. found that most reported cobot-based quality-control applications are concentrated on visual inspection, while other forms of testing and inspection remain less represented [29].

A similar trend was reported by Verna et al., whose review of academic and industrial applications showed that visual inspection represented the dominant inspection category, whereas conformity or functionality testing accounted for only a small portion of the identified applications [30].

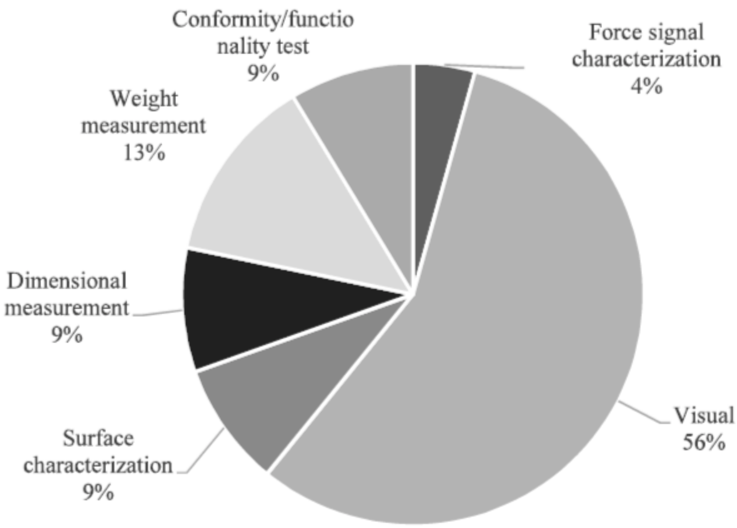

Figure 1.8. Distribution of inspection types reported in collaborative-robot quality-control studies and industrial case studies

Figure 1.8 indicates that cobot-based quality control has been dominated by visual inspection, while conformity and functionality testing have received substantially less attention.

A practical gap remains in the design of compact laboratory platforms that can use one end-effector to operate several small mechanisms on products with different geometries, while documenting cycle-to-cycle motion consistency.

The research question addressed here is therefore whether a six-axis collaborative robot and a single stepper-driven gripper can be reconfigured mainly through target-point and trajectory changes to operate multiple helmet mechanisms.

Chapter 2: AIM, OBJECTIVES AND SCOPE

The work aims to develop a flexible laboratory platform for repetitive functional testing of motorcycle helmet mechanisms. The motivation is not to replace standardized helmet certification equipment, but to investigate whether a reprogrammable robotic cell can reduce the amount of product-specific mechanical redesign required during early-stage testing [31,32].

The proposed architecture separates two functions. The Dobot CR10 provides six-axis spatial positioning, while the custom end-effector supplies the short local stroke required to open, close, push, or release the selected mechanism [6,8]. This division allows the robot path to be changed without redesigning the entire actuation unit [7].

The specific objectives are: (i) design and manufacture a compact rack-and-pinion end-effector; (ii) integrate robot, stepper motor, Arduino controller, and supervisory software; (iii) define safe and repeatable approach and withdrawal trajectories; (iv) execute twenty-cycle functional trials on several helmet mechanisms; and (v) analyze controller-reported motion data to compare cycle timing, path length, joint motion, TCP speed, and trajectory dispersion.

The expected contribution is a proof-of-concept showing that one robotic platform can operate multiple mechanisms and can be adapted to a second helmet geometry primarily through software-defined target changes. Absolute robot accuracy, interaction force, long-term wear, and conformity with a helmet durability standard are outside the present experimental scope.

The workflow begins with mechanical design and system integration, continues with programmed functional cycles and controller-data logging, and ends with a preliminary evaluation of feasibility and consistency.

The overall workflow of the proposed study is summarized in

Figure 2.1, from mechanical design and system integration to programmed functional testing and controller-data analysis.

Figure 2.1. Overview of the proposed robotic testing system: objectives and workflow

The workflow reflects the proof-of-concept nature of the study, with emphasis on system integration, repetitive execution, and preliminary consistency analysis.

Chapter 3: MATERIALS AND METHODS

This chapter presents the materials, equipment, and experimental methods used to develop the robotic system proposed in this research. The aim of the work was to automate the functional testing of motorcycle helmet mechanisms by combining a collaborative robot with a custom-designed end-effector and an electronic control system [33]. The complete experimental platform was developed to perform repetitive tests automatically while maintaining consistent operating conditions throughout the experiments.

The robotic platform consists of a Dobot CR10 collaborative robot equipped with a custom end-effector designed specifically for this study. A stepper motor was integrated inside the end-effector to generate the mechanical movement required to operate the different helmet mechanisms. The motor was controlled by an Arduino Uno through a stepper motor driver, while the overall testing sequence and the robot motion were managed using a Python program running on a personal computer.

Two different motorcycle helmets were selected for the experimental investigation. The first helmet included four mechanisms to be tested: the visor, the internal sun visor, the upper air vent, and the chin mechanism. Since the second helmet had a different mechanical design, only the visor mechanism was tested. For each test, the selected mechanism was actuated twenty consecutive times in order to evaluate the repeatability of the proposed robotic system.

The development of the experimental platform involved several stages. The first stage focused on designing the custom end-effector according to the geometry of the helmet mechanisms. After the design was completed, the components were manufactured using 3D printing, assembled, and mounted on the collaborative robot. The electronic hardware and the control software were then integrated to allow communication between the robot and the end-effector, enabling the automatic execution of the testing sequence.

This chapter first describes the experimental setup and the equipment used in the study. It then presents the design and fabrication of the custom end-effector, followed by the implementation of the electronic control system and the software developed for robot control. Finally, the robot programming, calibration procedure, and experimental methodology adopted during the testing campaign are described in detail.

3.1 Experimental Setup

A dedicated experimental setup was developed to evaluate the feasibility of using a collaborative robot for the automated functional testing of motorcycle helmet mechanisms [33]. The objective of the setup was to reproduce the manual operations normally performed during helmet inspection while ensuring that each test could be executed under consistent and repeatable conditions [4].

The experimental platform was designed by integrating a collaborative robot, a custom-designed end-effector, an electronic control system, and two motorcycle helmets used as test specimens. Each component was selected or developed to perform a specific task within the overall testing process. The collaborative robot was responsible for positioning the end-effector accurately, while the custom end-effector generated the mechanical interaction required to operate the different helmet mechanisms. The electronic control system synchronized the movement of the end-effector with the robot motion, allowing the complete testing procedure to be performed automatically.

Two commercially available motorcycle helmets with different geometrical characteristics were used during the experiments. The first helmet was employed to evaluate four different mechanisms, including the visor, the internal sun visor, the upper air vent, and the chin mechanism. The second helmet had a different design, and therefore only the visor mechanism was selected for testing.

The following sections describe the main elements of the experimental setup in detail, including the collaborative robot, the helmet specimens, the custom-designed end-effector, and the electronic control system used throughout the experimental work.

3.1.1 Dobot CR10 Collaborative Robot

The robotic platform used in this study is based on the Dobot CR10 collaborative robot. This robot was selected because it combines a relatively large working range with high positioning accuracy while maintaining the safety features required for laboratory environments. Unlike conventional industrial robots, collaborative robots are designed to operate safely in shared workspaces, making them suitable for research activities and experimental testing where frequent interaction between the operator and the robotic system may be required [34,35].

In this work, the Dobot CR10 was used exclusively to position the custom-designed end-effector during the testing procedure. The robot was programmed to move between predefined positions corresponding to the different helmet mechanisms. Once the desired position was reached, the end-effector performed the required mechanical action to operate the selected component.

Since all movements were generated automatically through the control software, the robot followed the same trajectory during every testing cycle. This ensured that each mechanism was actuated under identical operating conditions, improving the repeatability of the experiments and reducing the variability associated with manual testing [10].

The robot was mounted on a rigid laboratory table to provide a stable working environment throughout the experiments. During the development of the testing procedure, the robot trajectories were carefully adjusted to ensure smooth movements while avoiding collisions with the helmet fixture or the surrounding equipment. Before the experimental campaign, all programmed positions were verified through preliminary trials to confirm that the robot could safely reach each testing location.

The main technical specifications of the Dobot CR10 collaborative robot are summarized in Table 3.1

As shown in Figure 3.1, the robot was installed on a rigid laboratory support and provided

Figure 3.1. Dobot CR10 collaborative robot used in the experimental setup

the six-axis positioning required to approach the different helmet mechanisms.

, while the robot used in the experimental setup is shown in Error! Reference source not found. [36].

As shown in Figure 3.1, the robot was installed on a rigid laboratory support and provided the six-axis positioning required to approach the different helmet mechanisms.

Table 3.1. Main specifications of the Dobot CR10 collaborative robot

Parameter	Value
Robot type	Collaborative robot
Degrees of freedom	6
Maximum payload	10 kg
Reach	1300 mm
Repeatability	±0.03 mm

The 1300 mm reach provided sufficient workspace to access the different helmet mechanisms, while the 10 kg payload was well above the mass of the custom end-effector. The manufacturer-specified repeatability of ±0.03 mm is reported here as a robot specification and should not be interpreted as an independently measured value in the present experiments. (Table 3.1)

Robot motion was defined with reference to the coordinate systems provided by the Dobot controller. Figure 3.2 illustrates the robot joint coordinates, while Figure 3.3 and Figure 3.4 show the base and tool coordinate systems used to describe the position and orientation of the end-effector. These reference frames were important during trajectory definition and calibration because all programmed target poses were expressed relative to the selected robot and tool frames.

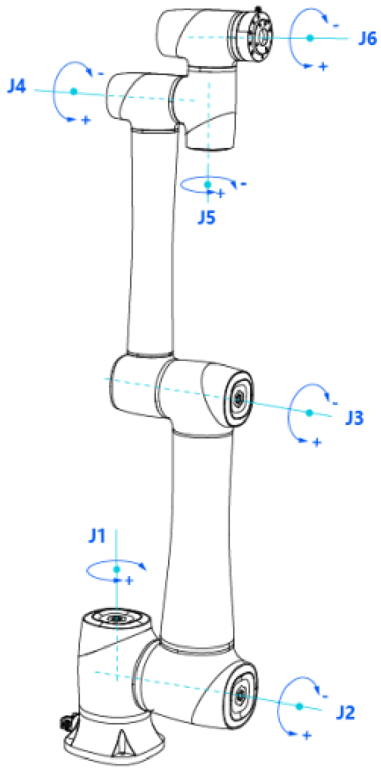

Figure 3.2. Base coordinate system of CR10 robot

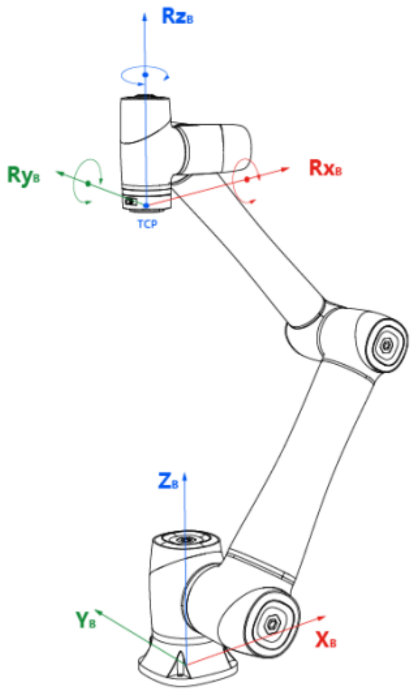

Figure 3.3. Joint coordinate system of the Dobot CR10

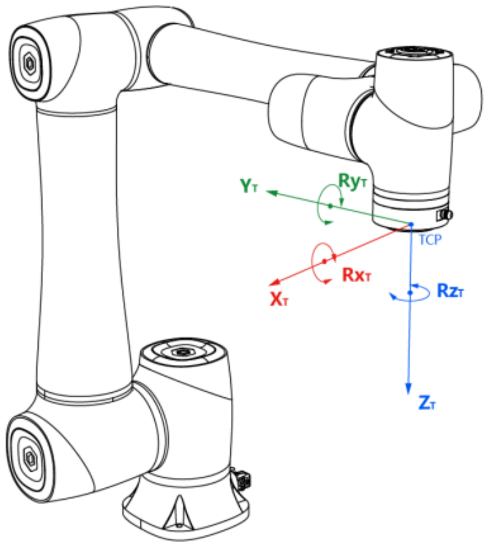

Figure 3.4. The default Tool coordinate system of CR10 robot

These coordinate definitions were subsequently used when defining the robot targets and the orientation of the custom end-effector around the helmet.

3.1.2 Helmet Specimens

Two different motorcycle helmet models were selected as test specimens for this study. The use of two helmets with different geometrical characteristics made it possible to evaluate the proposed robotic system on more than one helmet configuration while using the same experimental setup.

Before each experiment, the helmets were securely fixed to the experimental workbench using dedicated mechanical fixtures. These fixtures prevented unwanted movement during robot operation and ensured that each helmet remained in the same position throughout the testing procedure. Maintaining a fixed position was important to guarantee that the robot approached the helmet mechanisms under the same conditions during every testing cycle [37].

The first helmet, hereafter referred to as Helmet A, is shown in Figure 3.5 and was used for the main experimental investigation.

Four functional mechanisms were selected for automated testing:

- Main visor
- Internal sun visor
- Upper air vent
- Chin mechanism

Figure 3.5. Helmet A used for testing the main visor, internal sun visor, upper air vent, and chin mechanism

As shown in Figure 3.5, Helmet A includes several independently operated mechanisms located at different positions and orientations, making it suitable for evaluating the ability of the robot to approach multiple interaction points.

The second helmet, hereafter referred to as Helmet B, is shown in Figure 3.6. It presented a different external geometry and visor configuration; therefore, only the main visor was selected for the reconfiguration trial.

Figure 3.6. Helmet B used for the main-visor reconfiguration trial

The geometrical differences between Helmets A and B required changes in robot target positions and trajectories, while the same principal robotic platform and custom end-effector were retained.

Each mechanism was tested through 20 consecutive operating cycles following the same testing procedure.

The second helmet **Error! Reference source not found.** had a different external geometry and a different visor mechanism. Because of these design differences, only the main visor was included in the experimental tests. The visor was also actuated for 20 consecutive cycles using the same robotic procedure adopted for the first helmet.

The use of two helmet models provided a preliminary demonstration of reconfiguration across different geometries while retaining the same principal robotic platform and end-effector structure.

3.1.3 Custom End-Effector

To enable the collaborative robot to interact with the helmet mechanisms, a custom end-effector was developed as part of this research. The end-effector acts as the interface between the robot and the helmet, allowing the programmed movements of the robot to be translated into the mechanical actions required to operate the different helmet components [7,38].

The end-effector was designed to perform the investigated functional operations using the same principal mechanical configuration. Instead of using different tools for each helmet mechanism, one end-effector was developed to interact with the main visor, the internal sun visor, the upper air vent, and the chin mechanism. This approach simplified the experimental setup and reduced the time required to switch between different testing procedures.

A stepper motor was integrated inside the end-effector to generate the mechanical movement required during the tests. The motion produced by the motor was transmitted to the contact element of the end-effector, allowing the selected helmet mechanism to be actuated in a controlled and repeatable manner. Depending on the component being tested, the robot positioned the end-effector at the appropriate location, while the stepper motor provided the actuation needed to complete the operation.

The end-effector was mounted directly on the flange of the Dobot CR10 collaborative robot and operated together with the robot throughout the experimental campaign. Its compact structure allowed the robot to access different areas of the helmet while maintaining stable and repeatable contact with the selected mechanisms.

The design methodology, fabrication process, and mechanical characteristics of the end-effector are presented in detail in Sections **3.2** and **3.3**.

3.1.4 Electronic Control System

The electronic control system coordinates end-effector actuation with the programmed robot sequence. The Arduino generates STEP and DIR signals for the motor driver, while the computer issues high-level commands and waits for completion acknowledgements before advancing the robot routine.

The system consists of an Arduino Uno microcontroller, a stepper motor driver, a stepper motor, an external power supply, and a personal computer running the control software [39]. The Arduino Uno was used to control the stepper motor, while the motor driver supplied the electrical power required for its operation. The personal computer managed the overall testing process and communicated with the Arduino through a USB serial connection.

During the experiments, the collaborative robot positioned the end-effector at the selected helmet mechanism, while the electronic control system provided the actuation required to perform the corresponding mechanical operation. By combining the robot motion with the controlled movement of the stepper motor, the complete testing sequence could be executed automatically and repeatedly with minimal operator intervention.

The electronic control system was integrated with the robotic platform throughout the experimental campaign and operated continuously during all testing procedures. A detailed description of the hardware configuration, electrical connections, and communication strategy is presented in **Section 3.4**.

3.2 Design of the Custom End-Effector

The end-effector is one of the key components of the proposed robotic testing system, as it provides the physical interaction between the collaborative robot and the motorcycle helmet [33]. Since the robot is only responsible for positioning, the end-effector must generate the mechanical action required to operate the different helmet mechanisms. For this reason, particular attention was given to its design in order to ensure reliable operation during all experimental tests.

The design process started by identifying the functional requirements of the end-effector. Several concepts were considered before the final configuration was selected. The objective was to develop a single end-effector capable of performing all the required tests while remaining compact, lightweight, and easy to mount on the robot [37]. The complete design was created in Autodesk Inventor and was gradually refined as the experimental setup was developed.

The following sections describe the design process, beginning with the design requirements, followed by the measurements taken from the helmet mechanisms, the

CAD modelling procedure, the motion transmission mechanism, and the virtual assembly of the complete end-effector.

3.2.1 Design Requirements

Before starting the CAD design, the functional requirements of the end-effector were carefully defined based on the characteristics of the helmet mechanisms and the objectives of the experimental tests. Since the end-effector would be used repeatedly throughout the testing campaign, it needed to provide stable and reliable operation while remaining simple to manufacture and assemble.

One of the main design objectives was to develop a single end-effector that could be used for different helmet mechanisms without replacing or modifying the tool during the experiments. The end-effector therefore had to interact with the main visor, the internal sun visor, the upper air vent, and the chin mechanism using the same mechanical structure.

Another important requirement was compatibility with the Dobot CR10 collaborative robot. The end-effector had to be mounted securely on the robot flange while maintaining a compact size that would not restrict the robot motion or interfere with the surrounding environment. The overall dimensions and weight were therefore considered throughout the design process.

To generate the required mechanical motion, a stepper motor was integrated inside the end-effector. The internal structure was designed to accommodate the motor while providing sufficient support for the moving components. At the same time, the transmission mechanism had to convert the motor rotation into a controlled movement that could actuate the different helmet mechanisms without causing damage to the helmet surface.

Ease of manufacturing was also considered during the design stage. Since the end-effector was produced using 3D printing, the geometry of each part was designed to simplify fabrication and reduce the need for post-processing. In addition, the assembly procedure was kept as straightforward as possible so that the components could be easily installed, replaced, or adjusted if it is necessary during the experimental work.

Finally, repeatability was considered one of the most important design requirements [10]. Because every helmet mechanism was tested over multiple consecutive cycles, the end-effector had to produce the same mechanical movement repeatedly while maintaining stable contact with the helmet. Achieving consistent operation throughout the testing process was essential for obtaining reliable experimental results.

3.2.2 Measurements of Helmet Mechanisms

Before starting the CAD modelling of the end-effector, the selected helmet mechanisms were carefully examined to understand their geometry and operating characteristics. This preliminary step was essential to ensure that the end-effector would interact correctly with each component during the experimental tests.

The measurements were carried out directly on the two helmet models used in this study. A digital caliper was used to inspect the geometry of the mechanisms and to obtain the information required for the design process. Particular attention was given to the visor, the internal sun visor, the upper air vent, and the chin mechanism, since these components would be actuated by the robotic system during the experiments.

In addition to the dimensions of the individual mechanisms, their position on the helmet and the available space around them were also evaluated. This information was important for determining how the end-effector should approach each component without interfering with adjacent parts of the helmet or with the supporting fixture.

The measurements also highlighted the geometrical differences between the two helmet models. Although both helmets served the same purpose, the shape and location of some mechanisms were not identical. These differences were considered throughout the design process to ensure that the end-effector could operate correctly on the selected test specimens while maintaining the same overall mechanical configuration.

Rather than designing the end-effector based on estimated dimensions, the CAD model was developed using the measurements obtained from the actual helmets. This approach reduced the need for repeated design modifications during fabrication and helped achieve proper alignment between the end-effector and the helmet mechanisms during the experimental tests.

3.2.3 CAD Design in Autodesk Inventor

After completing the measurement of the helmet mechanisms, the end-effector was designed using Autodesk Inventor. The CAD model was developed step by step, allowing the geometry to be refined throughout the design process until the final configuration was obtained [40].

The design started with the main body of the end-effector, which serves as the supporting structure for all the mechanical components. During this stage, the overall dimensions were selected to provide sufficient rigidity while keeping the structure as compact as possible. A compact design was preferred to allow the robot to access the different helmet mechanisms without unnecessary interference with the helmet or the surrounding workspace.

Once the main body had been defined, the remaining parts were designed according to their specific functions. Particular attention was given to the internal space required to accommodate the stepper motor, as the motor was integrated inside the end-effector. The internal geometry was arranged so that the motor could be securely positioned while leaving enough space for the moving components responsible for transmitting the motion to the contact element.

The contact element of the end-effector was designed to interact with different helmet mechanisms using the same mechanical structure. Since the visor, sun visor, air vent, and chin mechanism have different shapes and operating directions, several design iterations were carried out to obtain a configuration capable of performing all the required tests. During this process, the geometry of some components was modified to improve accessibility and ensure smoother interaction with the helmet.

Another important aspect considered during the CAD modelling process was the assembly of the different parts. The position of fastening holes, clearances between moving components, and interfaces for mounting the end-effector on the robot flange were incorporated into the design from the beginning. This reduced the need for modifications after fabrication and simplified the final assembly of the system.

The main components developed during the CAD design stage are shown in

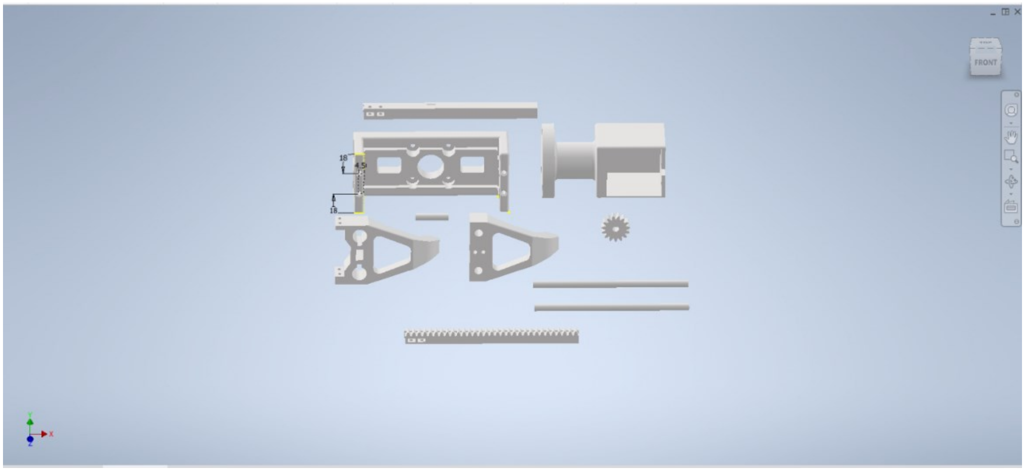

Figure 3.7.

Figure 3.7. Main CAD-designed components of the custom end-effector developed in Autodesk Inventor

The model includes the main structural body, moving jaw, fixed jaw, rack-and-pinion elements, guide rods, and the robot mounting interface. The CAD environment was used to define the geometry and relative arrangement of these components before fabrication and assembly.

The end-effector was designed with a circular mounting flange having a diameter of 63 mm. The axial distance from the mounting-flange reference plane to the distal jaw tip is approximately 199.3 mm, while the overall span between the external jaw surfaces is 130 mm. The gripping mechanism consists of one fixed jaw, rigidly attached to the main body, and one movable jaw with an operational stroke of 76 mm. Consequently, the functional distance between the jaw tips ranges from 0 mm in the fully closed configuration to 76 mm in the fully open configuration. The movable jaw is guided by two solid steel rods, each 130 mm long and 6 mm in diameter, together with two LM6UU linear bearings.

Throughout the design process, the complete assembly was continuously evaluated in the CAD environment to verify the fit of all components and to identify potential mechanical interferences before manufacturing. This virtual verification made it possible to improve the design progressively and reduced the number of adjustments required after the parts had been fabricated [41].

The final CAD model served as the basis for the fabrication of the end-effector and for its subsequent integration with the collaborative robot. The complete virtual assembly and the individual components are presented in the following sections.

3.2.4 Rack-and-Pinion Motion Transmission

The motion transmission mechanism was designed to convert the rotational motion of the stepper motor into the linear movement required for the operation of the end-effector. Since the robot was responsible only for positioning the end-effector, all mechanical interaction with the helmet mechanisms was generated by the actuation system integrated inside the end-effector.

Motion is transmitted directly from the NEMA 17 motor shaft to the movable jaw through a rack-and-pinion mechanism. The pinion is mounted directly on the motor shaft and engages with the rack connected to the movable jaw. The spur pinion has 15 teeth, a module of 1.375 mm, a pressure angle of 20°, a pitch diameter of 20.625 mm, and a face width of 8 mm. It engages with a 145.547 mm-long rack containing 31 teeth. The moving jaw slides along two parallel guide rods through two LM6UU linear bearings, while the opposite jaw remains rigidly fixed to the main body.

Since the pinion is mounted directly on the motor shaft, no intermediate rotational reduction is used, corresponding to a motor-to-pinion transmission ratio of 1:1. For the rotary-to-linear conversion, one complete pinion revolution produces a theoretical rack displacement of approximately 64.8 mm, calculated from the pinion pitch circumference. Therefore, the full 76 mm stroke of the movable jaw corresponds to approximately 1.17 pinion revolutions.

The rack-and-pinion transmission mechanism developed for the end-effector is illustrated in

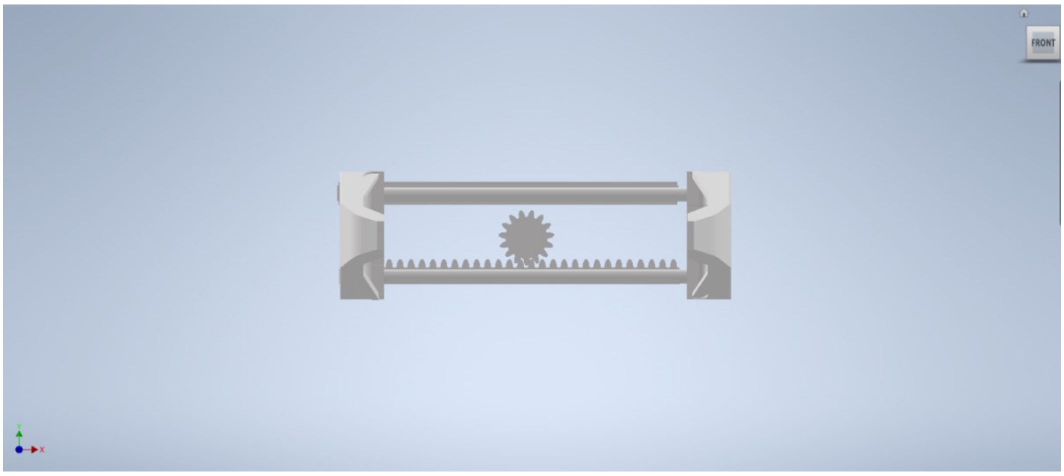

Figure 3.8.

Figure 3.8. Rack-and-pinion transmission mechanism of the custom end-effector

As shown in

Figure 3.8, the rotation of the pinion gear produces the linear displacement of the rack, while the guide rods help maintain alignment and support the moving parts during operation. This configuration allows the end-effector to generate the controlled linear motion required to actuate the selected helmet mechanisms.

The use of a rack-and-pinion mechanism provided several advantages for this application. It offered a simple and compact solution for transforming rotary motion into linear motion while maintaining good positional repeatability throughout the testing cycles [42]. In addition, the mechanism could be integrated inside the end-effector without significantly increasing its overall dimensions, making it suitable for operation within the limited space around the helmet.

During the design stage, particular attention was given to the alignment between the pinion gear, the rack, and the guide rods. Proper alignment was essential to ensure smooth movement of the jaws and to reduce unnecessary friction during repeated operation. The internal arrangement of the components also contributed to the rigidity of the end-effector and improved the stability of the gripping mechanism during the experiments.

The final transmission mechanism enabled the end-effector to generate the controlled linear motion required for the functional testing of the selected helmet mechanisms. By combining the accurate positioning of the Dobot CR10 with the repeatable actuation provided by the rack-and-pinion system, the robotic platform was able to perform the programmed testing cycles under consistent operating conditions.

3.2.5 Virtual Assembly

Once the individual components of the end-effector had been designed, they were assembled in the Autodesk Inventor environment to verify the complete mechanical structure before fabrication. Creating a virtual assembly made it possible to evaluate how the different parts interacted with one another and to identify potential design issues at an early stage, reducing the need for modifications after manufacturing [41].

During the assembly process, all components were positioned according to their intended operating configuration. The main body of the end-effector served as the supporting structure, while the stepper motor, rack-and-pinion transmission mechanism, guide rods, moving jaws, and fastening elements were assembled to form the complete system. This virtual model provided a clear representation of the final end-effector and allowed the integration of all components to be evaluated before production.

Particular attention was given to the alignment of the moving components. The guide rods were checked to ensure smooth linear motion of the jaws, while the engagement between the pinion gear and the rack was verified to provide proper transmission of motion. The available internal space was also examined to confirm that the stepper motor and transmission mechanism could be accommodated without interference between adjacent components.

The virtual assembly also made it possible to verify the interfaces between the end-effector and the Dobot CR10 collaborative robot. The mounting holes and connection surfaces were checked to ensure that the end-effector could be securely attached to the robot flange while maintaining the correct orientation during operation.

In addition to verifying the mechanical assembly, the CAD model was used to inspect the overall dimensions of the end-effector and to evaluate its accessibility to the different helmet mechanisms. This helped confirm that the proposed design could operate within the available workspace without interfering with the helmet fixture or limiting the robot motion.

Performing the complete assembly in the CAD environment before fabrication significantly simplified the manufacturing stage. Minor adjustments to the geometry and the position of individual components could be made quickly in the virtual model, avoiding unnecessary reprinting of parts and reducing the overall development time.

The completed virtual assembly of the custom-designed end-effector is shown in Figure 3.9.

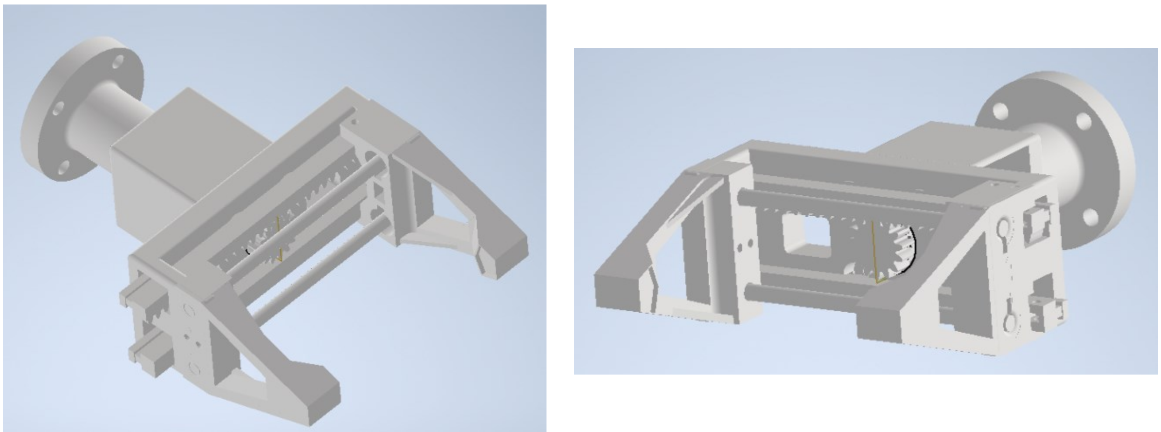

Figure 3.9. Virtual assembly of the custom end-effector in Autodesk Inventor

The virtual assembly was used to verify the relative position of the main body, moving jaw, fixed jaw, guide rods, rack-and-pinion mechanism, stepper motor, and robot mounting interface. It also allowed potential interference between components to be identified before fabrication.

3.3 Fabrication and Integration

After completing the design and verifying the virtual assembly, the end-effector components were manufactured and assembled to produce the final prototype used throughout the experimental work. The fabrication stage transformed the CAD model into a functional device that could be mounted on the collaborative robot and integrated with the electronic control system.

The manufacturing process included 3D printing of the designed parts, installation of the mechanical components, integration of the stepper motor, and final assembly of the complete end-effector. Once assembled, the end-effector was mounted on the Dobot CR10 collaborative robot and verified before starting the experimental tests.

The following sections describe the fabrication procedure and the integration of the end-effector with the robotic system.

3.3.1 3D Printing Process

The structural components of the end-effector were manufactured using 3D printing. This manufacturing method was selected because it allowed the designed parts to be produced directly from the CAD models with good dimensional accuracy while significantly reducing fabrication time [43,44]. In addition, 3D printing made it possible to modify or reproduce individual components whenever small design improvements were required during the development of the prototype.

After the CAD design was completed in Autodesk Inventor, each component was exported in STL format and prepared for printing using slicing software. Before printing, the orientation of each part was selected to obtain better surface quality and to minimize the need for support structures. Printing parameters were adjusted according to the geometry of each component to achieve an appropriate balance between printing time, dimensional accuracy, and mechanical strength.

The end-effector was manufactured using PLA filament, which was considered suitable for this prototype because of its ease of printing, good dimensional stability, and sufficient rigidity for the loads encountered during the experimental tests. Since the end-effector was intended for laboratory evaluation rather than heavy industrial operation, PLA provided an appropriate solution while keeping the manufacturing process simple and cost-effective [44].

Once the printing process was completed, the parts were visually inspected to identify any printing defects or dimensional inaccuracies. Support material was removed where necessary, and the contact surfaces were cleaned before the assembly stage. This preparation ensured that the printed components could be assembled correctly and that the moving parts operated without unnecessary interference.

The fabricated components were then ready for the installation of the mechanical elements and the assembly of the complete end-effector, as described in the following sections.

3.3.2 Mechanical Integration

The printed structural components were integrated with the guide rods, moving jaw, fixed jaw, rack-and-pinion transmission, stepper motor, and mechanical fasteners to form the complete end-effector. The assembly was designed so that individual printed components could be replaced or modified without manufacturing the entire tool again.

The main mechanical requirements during integration were the parallel alignment of the guide rods, correct engagement between the rack and pinion, and alignment of the pinion with the stepper-motor shaft. These conditions were necessary to obtain smooth linear motion and prevent excessive friction, lateral displacement, or mechanical interference. The moving elements were therefore checked manually over their complete operating range before the electronic system was connected.

The stepper motor was secured to the central housing using four mounting screws. Its position maintained direct alignment between the motor shaft and the pinion, while the motor cables were routed away from the moving components. After integration, the jaws were actuated repeatedly at low speed to verify smooth travel, stable rack-and-pinion engagement, and adequate mechanical rigidity.

The motor was fixed to the main body of the end-effector using **four mounting screws**, providing a rigid and stable connection throughout the experimental tests. This mounting arrangement prevented unwanted movement of the motor during repeated actuation cycles and ensured consistent engagement between the pinion gear and the rack. Care was taken to position the motor accurately within the housing so that the transmission mechanism could operate smoothly without excessive friction or mechanical interference.

After the motor had been installed, the transmission mechanism was manually inspected by rotating the motor shaft to verify that the rack moved freely along the guide rods and that the moving jaws travelled smoothly throughout their operating range. Any minor misalignment identified during this inspection was corrected before the electronic control system was connected.

Particular attention was also given to the routing of the motor cables. The cables were arranged to avoid contact with the moving parts of the end-effector and to prevent interference with the robot during operation. This simple precaution improved the reliability of the experimental setup and reduced the possibility of cable damage during repeated testing cycles.

3.3.3 Mounting on the Robot

The mounting interface of the end-effector was designed to connect directly to the flange of the Dobot CR10 without requiring an additional adapter. The interface provided a rigid mechanical connection and maintained the orientation of the tool relative to the robot tool coordinate system.

The final mechanical integration of the custom end-effector with the Dobot CR10 is shown in Figure 3.10.

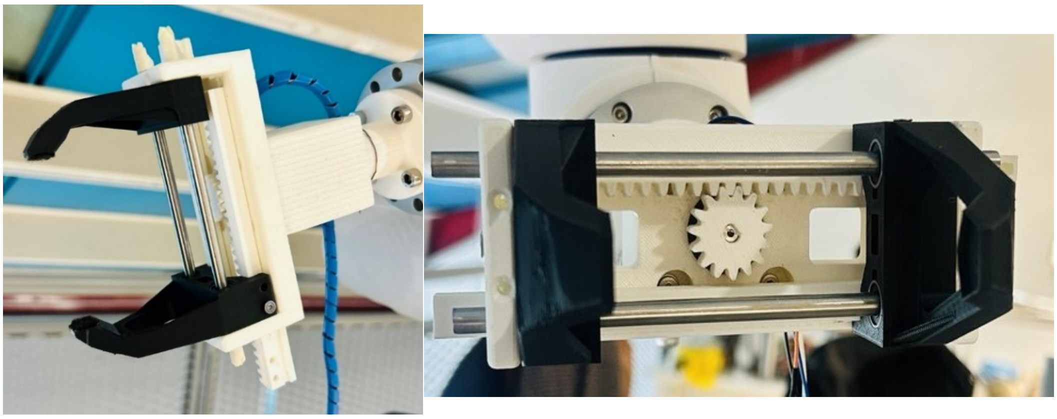

Figure 3.10. Custom-designed end-effector mounted on the Dobot CR10

As shown in Figure 3.10, the end-effector was mounted directly on the robot flange, creating the mechanical interface used to interact with the helmet mechanisms during the experiments.

After installation, the tool orientation, mechanical rigidity, cable clearance, and accessibility of the robot workspace were checked through low-speed trial motions. No visible displacement of the mounting interface or interference between the end-effector, helmet fixture, and surrounding equipment was observed during these preliminary movements.

3.3.4 Functional Verification

Before starting the experimental tests, the assembled robotic system underwent a functional verification to confirm that all mechanical and electromechanical components operated as expected. This preliminary stage was carried out to identify any potential issues before beginning the repetitive testing procedure.

The verification focused on the overall operation of the end-effector rather than on the performance of the helmet mechanisms themselves. The movement of the jaws was checked to ensure that they opened and closed smoothly throughout their operating range. The motion generated by the stepper motor was observed to confirm that it was transmitted correctly through the rack-and-pinion mechanism without irregular movement or mechanical interference.

The integration between the collaborative robot and the end-effector was also verified. Trial robot motions were executed to ensure that the end-effector maintained its orientation during movement and that no unexpected contact occurred with the helmet fixture or the surrounding workspace. These preliminary movements also confirmed that the cables connected to the stepper motor remained clear of the moving parts throughout the robot trajectory.

Following the mechanical verification, several trial actuations were performed on the helmet mechanisms at a reduced operating speed. These preliminary operations were intended only to confirm that the end-effector could establish proper contact with the selected mechanisms and complete the required motion successfully. Any minor adjustments to the robot approach positions were carried out during this stage before the experimental protocol was finalized.

Once all components operated correctly and the interaction between the robot and the helmet mechanisms had been verified, the system was considered ready for the experimental testing described in the following sections.

3.4 Electronic Control System

The electronic control system was developed to provide reliable actuation of the custom-designed end-effector and to ensure coordinated operation between the mechanical and electrical components during the experimental tests. While the collaborative robot was responsible for positioning the end-effector, the electronic system controlled the movement generated by the stepper motor, allowing the helmet mechanisms to be actuated in a controlled and repeatable manner.

The system consists of several hardware components working together, including an Arduino Uno microcontroller, a stepper motor driver, a stepper motor integrated inside the end-effector, an external power supply, and a personal computer used for system control. Each component performs a specific function, and together they form the complete control architecture adopted in this research.

The Arduino Uno was selected as the main control unit because it provides a simple and reliable platform for generating the electrical signals required by the stepper motor driver [39]. Rather than driving the motor directly, the Arduino sends pulse and direction signals to the driver, which supplies the current needed to rotate the motor accurately.

The stepper motor driver acts as the interface between the Arduino and the motor [45]. It receives the control commands generated by the Arduino and converts them into the appropriate electrical excitation sequence for the motor windings. Using a dedicated driver also protects the microcontroller from the higher current required by the motor while improving the stability of the actuation system during repeated operation.

The stepper motor was installed inside the custom-designed end-effector and served as the actuator responsible for producing the mechanical motion. Through the rack-and-pinion transmission mechanism described in the previous section, the rotational motion of the motor was converted into the linear movement of the jaw. This arrangement enabled the end-effector to interact with the different helmet mechanisms in a controlled manner.

An external DC power supply was used to provide electrical power to the motor driver and the stepper motor. Separating the motor power supply from the USB connection ensured stable motor operation throughout the experimental tests and prevented excessive electrical loading of the control electronics.

Communication between the computer and the Arduino Uno was established through USB serial communication. The architecture is sequential: the robot first reaches and settles at the commanded pose, the stepper actuation is then executed, and the robot moves only after the Arduino returns a completion message.

The overall hardware architecture adopted in this study is illustrated in Figure 3.11.

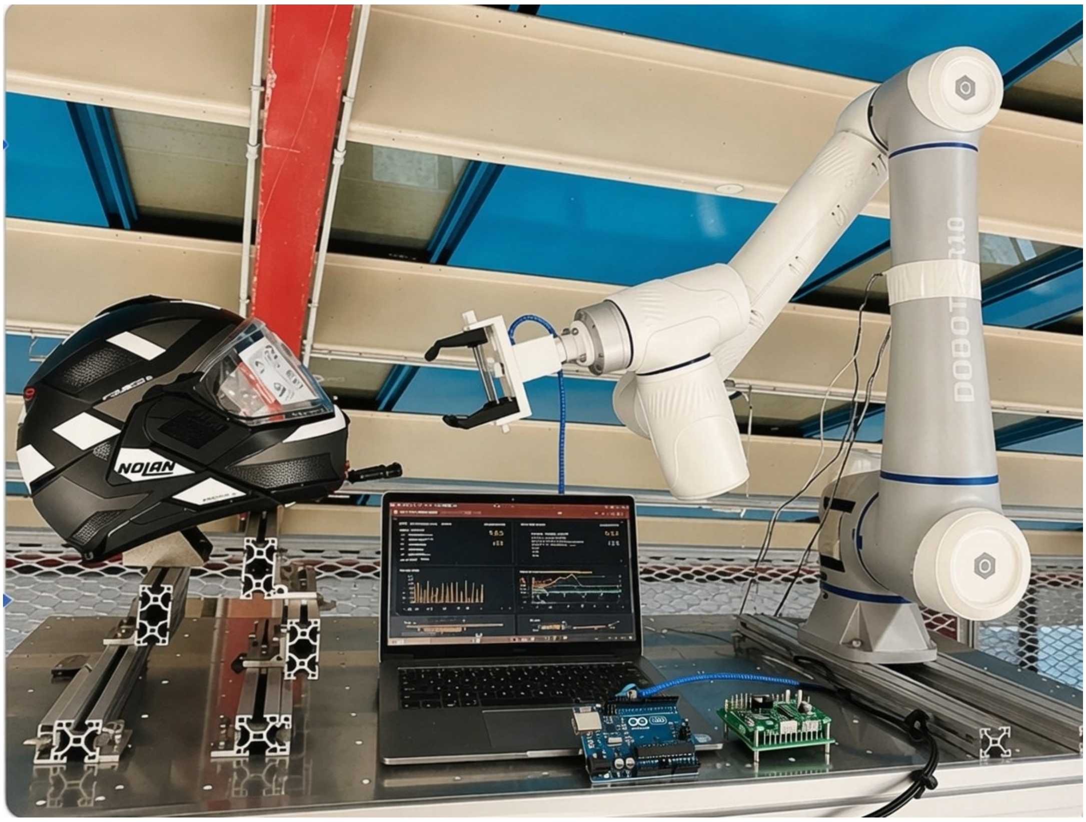

Figure 3.11. Overview of the integrated robotic test cell

The supervisory computer coordinates the robot and the Arduino-based actuation system, while the motor driver provides the electrical interface between the Arduino and the stepper motor. This architecture enables robot positioning and end-effector actuation to be executed sequentially during each test routine.

3.5 Software Architecture

The software architecture was developed to coordinate the operation of the collaborative robot and the custom-designed end-effector during the experimental tests. Rather than controlling the mechanical components independently, the software was designed to execute the entire testing sequence in a predefined order, ensuring that each operation was performed only after the previous step had been completed successfully.

Python was used as the main programming environment because it provides a simple and flexible platform for integrating robot motion with external devices [46]. The software acted as the central controller of the experimental setup, managing the communication with both the Dobot CR10 robot and the Arduino Uno throughout the testing process.

The sequence implemented by the supervisory software is summarized in Figure 3.12.

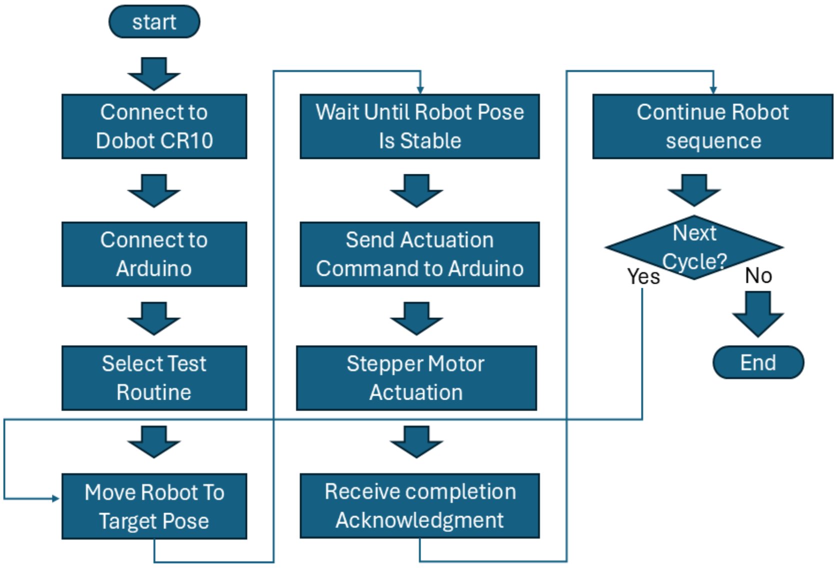

Figure 3.12. Flowchart of the supervisory software used to coordinate robot motion and end-effector actuation

As shown in Figure 3.12, the software follows a sequential control logic. The robot is first moved to the required target pose, after which the Arduino-based actuation command is issued. Once the end-effector motion is completed and the confirmation signal is received, the robot continues with the next programmed step or the next test cycle.

At the beginning of each test, the Python program initialized the required communication channels and established the connection with the robot and the Arduino. Once the system was ready, the robot was instructed to move to the predefined position corresponding to the selected helmet mechanism. During this stage, no actuation of the end-effector was performed, allowing the robot to reach the target position safely.

After the robot reached the desired position, the Python program transmitted a command to the Arduino through the serial communication interface. Upon receiving the command, the Arduino executed the required motion of the stepper motor, which in turn actuated the end-effector. Once the movement was completed, the Arduino returned a confirmation signal indicating that the operation had finished successfully.

This communication strategy ensured that the robot and the end-effector operated sequentially rather than simultaneously. As a result, the mechanical interaction with the helmet always occurred while the robot was stationary, reducing unnecessary dynamic loads and improving the consistency of the testing process.

The software was organized in a modular structure, where each stage of the testing sequence was handled independently [47]. This approach simplified the development process and allowed modifications to be made without affecting the overall operation of the system. For example, changes to the actuation sequence of the end-effector could be implemented without altering the robot trajectory, while modifications to the robot motion could be introduced without changing the Arduino program.

The adopted software architecture also facilitated the execution of repetitive testing cycles. Once the testing parameters had been defined, the complete sequence could be repeated automatically without requiring operator intervention, ensuring that all experiments were performed under the same operating conditions.

The complete Python and Arduino source codes developed for the experimental platform are presented in **Appendix A** and **Appendix B**, respectively.

3.6 Robot Programming

The Dobot CR10 collaborative robot was programmed to perform a sequence of predefined movements required for the automated testing of the helmet mechanisms. The objective of the robot program was to position the custom-designed end-effector accurately at each interaction point while ensuring smooth, safe, and repeatable motion throughout the experiments [6,7].

Before creating the motion sequence, the position of each helmet mechanism was identified within the robot workspace. The corresponding target positions were then defined so that the end-effector could approach each mechanism from an appropriate direction while maintaining a stable orientation. This procedure allowed the robot to interact with the helmet components without causing unwanted contact with adjacent parts or the supporting fixture.

To improve the consistency of the experiments, the robot was programmed to follow the same motion sequence during every testing cycle. The travelling speed, acceleration, and transition between consecutive positions were adjusted to achieve smooth movements and to minimize unnecessary vibration during operation [7]. At each relevant target pose, the software waited until the controller-reported robot position had stabilized. The corresponding end-effector command was then executed, and the robot sequence continued only after the Arduino returned a completion acknowledgement.

Each helmet mechanism required a different interaction position because of its location and operating direction. Although the motion paths varied depending on the component being tested, all trajectories were generated using the same programming strategy. This simplified the implementation of the experimental procedure while ensuring that every test followed a consistent and repeatable sequence.

The robot program was executed directly from the Python control software, which coordinated the movement commands with the operation of the end-effector. After the robot reached the target position, the corresponding actuation command was issued to the Arduino controller. Once the mechanical operation had been completed, the robot continued to the next programmed position or returned to its initial position, depending on the selected testing sequence.

Before starting the experimental campaign, the complete robot program was verified through several trial runs without applying any load to the helmet mechanisms. These preliminary tests were carried out to confirm that the programmed trajectories were correct, that all target positions could be reached safely, and that the end-effector maintained the required orientation throughout the robot motion.

3.7 Experimental Procedure

After completing the design, fabrication, and system integration, the experimental tests were carried out using the robotic platform developed in this study. Before each testing session, the experimental setup was inspected to ensure that all mechanical, electrical, and software components were operating correctly. Once the system had been verified, the testing procedure was performed following the same sequence for every experiment to ensure consistent operating conditions.

The first step consisted of securely fixing the selected helmet onto the experimental workbench using the dedicated support fixture. Care was taken to position the helmet correctly so that the target mechanism was accessible to the robot throughout the entire testing process. Maintaining the same mounting position for all experiments was important to ensure that the programmed robot positions remained valid during every testing cycle.

After the helmet had been installed, the Dobot CR10 collaborative robot was initialized and moved to its predefined home position. Starting from the same reference position before every experiment helped maintain a consistent sequence of robot movements and reduced the possibility of positioning errors.

Once the robot had reached the home position, it moved towards the predefined approach position corresponding to the selected helmet mechanism. This intermediate position allowed the robot to approach the helmet safely before moving to the exact interaction point where the end-effector would perform the required mechanical operation.

When the end-effector reached the interaction position, the actuation sequence was initiated. The end-effector operated the selected helmet mechanism using the motion generated by the internal stepper motor. After completing the required movement, the end-effector returned to its initial configuration while the robot remained stationary.

Following the completion of the actuation, the robot moved away from the helmet and returned either to the home position or to the starting position for the next testing cycle. This sequence was repeated automatically according to the predefined testing parameters until the required number of cycles had been completed.

Throughout the experiments, no manual intervention was required once the testing sequence had started. The robotic platform executed all programmed movements automatically, ensuring that each cycle followed the same procedure and operating conditions. This automated approach minimized the variability associated with manual testing and improved the repeatability of the experimental process [4].

The overall sequence adopted during the experiments is summarized below:

1. Helmet fixation on the experimental fixture.
2. Robot initialization and movement to the home position.
3. Robot movement to the predefined approach position.
4. Positioning of the end-effector at the interaction point.
5. Actuation of the selected helmet mechanism.
6. Return of the end-effector to its initial position.
7. Robot withdrawal from the helmet.
8. Preparation for the next testing cycle.
9. Automatic repetition of the sequence until the required number of cycles was completed.

3.8 Testing Procedure

The testing procedure was designed to evaluate the capability of the proposed robotic system to perform repetitive functional operations on different motorcycle helmet mechanisms. Two helmet models were considered in this study, and the testing protocol was adapted according to the characteristics of each helmet.

The first helmet was selected as the primary test specimen because it included several mechanisms that are commonly operated by the user during normal use. Four functional components were investigated: the main visor, the internal sun visor, the upper air vent, and the chin mechanism. Each mechanism was tested independently to ensure that the robot interacted with only one component at a time.

For every mechanism, the same operating sequence was repeated for **20 consecutive cycles**. This number of repetitions was selected to evaluate the consistency of the robotic system during repetitive operation while maintaining identical testing conditions throughout the experiment. Between consecutive cycles, no changes were made to the robot program, the end-effector, or the position of the helmet, allowing each mechanism to be tested under the same operating conditions.

A second motorcycle helmet was also included in the experimental campaign. Although it was designed for the same application, its external geometry and visor mechanism differed from those of the first helmet. Because of these differences, only the main visor was selected for testing on this model. The remaining mechanisms were not evaluated, as their geometry was not compatible with the testing configuration developed for the first helmet.

The visor of the second helmet was subjected to the same testing protocol used for the first helmet, consisting of **20 consecutive operating cycles**. Applying the same number of cycles to both helmets ensured that the experimental conditions remained consistent and allowed the robotic system to be evaluated on two different helmet geometries.

The complete experimental campaign is summarized in Table 3.2.

Table 3.2. summarizes the experimental tests performed in this study

Helmet	Tested Component	Number of Cycles
Helmet A	Main visor	20
Helmet A	Internal sun visor	20
Helmet A	Upper air vent	20
Helmet A	Chin mechanism	20
Helmet B	Main visor	20

As shown in Table 3.2, four configurations were evaluated on Helmet A, corresponding to 80 operating cycles, while an additional 20-cycle visor test was performed on Helmet B. The complete experimental campaign therefore comprised 100 operating cycles. The Helmet B trial was used as a separate functional reconfiguration demonstration and was not included in the four-routine quantitative trajectory comparison presented in Chapter 4.

3.9 Calibration Procedure

Before testing, a geometric setup and alignment procedure was performed to reproduce the relative position of the helmet fixture, robot, and end-effector. This procedure was based on fixture references, low-speed jogging, and visual verification. It should not be interpreted as an independent metrological calibration of absolute robot accuracy [10].

The first step consisted of verifying the mounting position of the helmet on the supporting fixture. The helmet was carefully placed in the same orientation used during the development of the robot program, ensuring that the relative position between the helmet and the robot remained unchanged throughout the experiments.

After the helmet had been fixed, the robot was moved manually to the predefined interaction points corresponding to each helmet mechanism. At every position, the alignment between the end-effector and the target mechanism was visually inspected. If necessary, small adjustments were made to the robot coordinates to ensure that the contact element reached the intended location without touching adjacent parts of the helmet.

Particular attention was given to the orientation of the end-effector during the calibration process. In addition to reaching the correct position, the end-effector had to maintain an appropriate approach angle to interact properly with the visor, the sun visor, and the ventilation mechanisms. Verifying both the position and orientation helped ensure stable and repeatable contact during the experimental tests.

The operation of the end-effector was also checked after the robot reached each target position. A preliminary actuation was performed to confirm that the moving jaws completed the required motion without interference and that the selected helmet mechanism could be operated correctly. These verification steps were carried out before starting the automatic testing sequence.

Once the interaction points had been verified, the corresponding robot target poses were retained for the entire test session. The procedure reduced setup variation, but the achieved physical contact accuracy was not measured with an external instrument.

3.10 Data Acquisition and Analysis

Two synchronized data logs were generated during the automated experiments. The first log contained the controller-reported robot trajectory data, while the second recorded the commanded and controller-reported target positions at selected motion endpoints.
The recorded trajectory dataset included:

- Elapsed time
- Test routine name
- Cycle number
- Motion label
- Cartesian TCP pose: x, y, z, r_x, r_y, r_z
- Controller-reported joint angles J_1–J_6

The target-position log contained the commanded Cartesian coordinates and the corresponding controller-reported coordinates at the end of selected robot motions.
The recorded data were processed to obtain the following motion and consistency indicators:

- **TCP speed:** estimated from the Euclidean displacement between consecutive Cartesian samples divided by the corresponding time interval [48].

- **Joint motion range:** calculated as the difference between the maximum and minimum controller-reported angle of each robot joint.

- **Cycle duration:** calculated from the time difference between the first and last logged sample associated with each cycle number.

- **Cartesian path length:** calculated as the cumulative Euclidean distance between consecutive TCP positions within each cycle.

- **Coefficient of variation (CV):** calculated as $CV = 100s/\bar{x}$for cycle duration and Cartesian path length in order to compare the relative cycle-to-cycle variation between routines [49].

- **Trajectory consistency:** evaluated for steady cycles 2–19. Each cycle was normalized according to its progress along the Cartesian path and interpolated onto 200 equally spaced path-progress points. A mean trajectory was then calculated for each routine, and the root-mean-square Cartesian deviation of the individual cycles from this mean trajectory was used as the trajectory-consistency metric.

Cycles 2–19 were used for the steady-cycle analyses in order to reduce the influence of transition effects associated with the beginning and end of the programmed routines. The first and final cycles were retained in the general cycle-duration and path-length plots so that these boundary effects could be identified explicitly (Figure 3.13).

All Cartesian positions and joint angles used in the quantitative analysis were obtained from the internal Dobot controller. Therefore, the resulting values describe controller-reported trajectory consistency and should not be interpreted as independent measurements of absolute TCP position or ISO 9283 positioning accuracy [10].

The quantitative analyses presented in Chapter 4 primarily refer to the four Helmet A routines identified in the dataset as **MAIN_VISOR**, **DARK_VISOR** (internal sun visor), **HELMET_VENT**, and **CHIN**. The Helmet B main-visor test was analysed separately as a reconfiguration case.

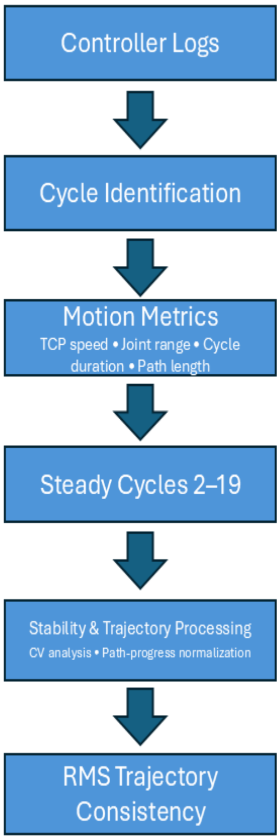

Figure 3.13. Data-processing workflow used for controller-based motion and cycle-consistency analysis

Chapter 4: RESULTS

This chapter presents the functional outcomes and controller-data analysis obtained from the developed robotic testing platform. The evaluation focuses on whether the programmed test routines were completed successfully and whether their timing and controller-reported trajectories remained consistent over repeated cycles. The chapter first presents the functional results obtained for the investigated helmet mechanisms. It then examines TCP speed, joint motion, cycle duration, Cartesian path length, and cycle-to-cycle trajectory consistency. The results are interpreted as a short-run evaluation of functional feasibility and controller-reported motion consistency; no claim of independently measured absolute positioning accuracy or long-term helmet durability is made.

4.1 Experimental Results

The experimental campaign evaluated the ability of the developed robotic platform to perform repetitive functional operations on two motorcycle helmets. Four mechanisms of Helmet A—the main visor, internal sun visor, upper air vent, and chin mechanism—were each operated for twenty consecutive cycles. A separate twenty-cycle main-visor trial was also performed on Helmet B to evaluate reconfiguration for a different helmet geometry.

The same robotic cell, principal end-effector, and control architecture were retained throughout the campaign. Reconfiguration between mechanisms was achieved primarily by modifying the programmed robot target poses and motion sequences according to the location and operating direction of the selected component.

Across the reported trials, all programmed functional routines were completed without recorded collision stops or missed actuation events. The end-effector remained mechanically intact, and no visible damage to the tested mechanisms was observed during the short experimental campaign.

These observations establish short-run functional feasibility. They do not demonstrate fatigue life, structural strength, or wear resistance because each configuration was operated for only twenty cycles and no force or deformation sensor was used.

The following subsections describe the functional outcome of each tested mechanism. The quantitative controller-data analyses presented later in this chapter refer to the four logged routines of Helmet A, whereas the Helmet B visor trial is treated separately as a functional reconfiguration demonstration.

4.1.1 Main Visor Test

The main visor was tested on both Helmet A and Helmet B for twenty consecutive cycles per configuration. In both cases, the programmed opening and closing routine was completed successfully using the same principal end-effector and control architecture. For Helmet B, only the robot target poses and motion sequence were adapted to the different visor geometry, providing a preliminary demonstration of system reconfigurability.
The two main-visor test configurations are shown in Figure 4.1.

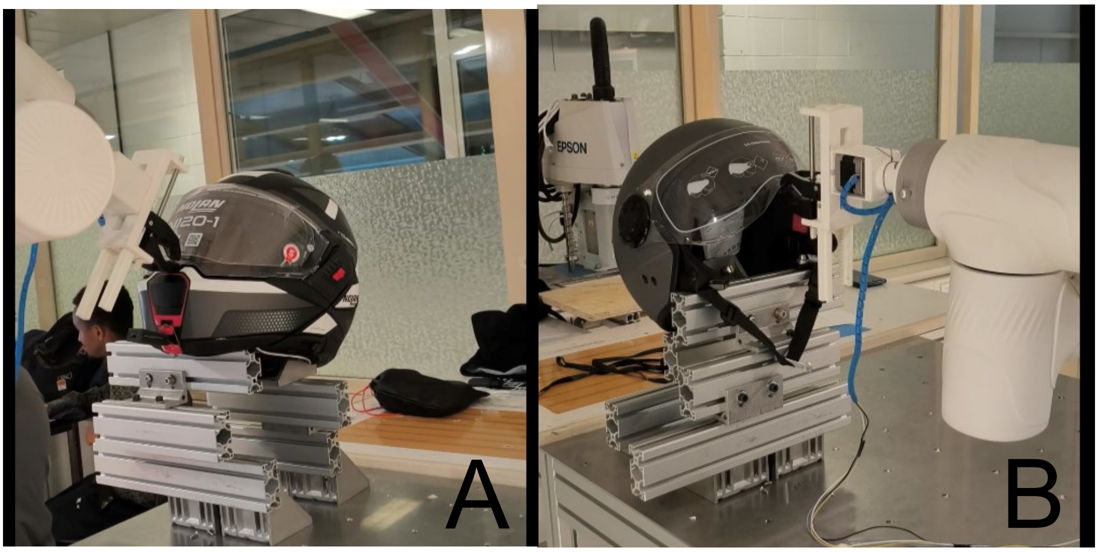

Figure 4.1. Main-visor functional test configurations: (a) Helmet A and (b) Helmet B.

Figure 4.1 shows that the same robotic platform and principal end-effector were retained, while the robot configuration was adapted to the different visor geometries.

4.1.2 Internal Sun Visor Test

The internal sun visor of Helmet A was operated for twenty consecutive cycles using the same robotic platform and end-effector. Compared with the main visor, this mechanism required a different interaction point and a shorter actuation stroke. All programmed cycles were completed successfully during the short experimental trial.

The robotic configuration used for the internal sun-visor test is shown in Figure 4.2.

Figure 4.2. Robotic functional test configuration for the internal sun visor of Helmet A

As shown in Figure 4.2, the robot was repositioned to access the sun-visor control while the same principal end-effector was retained.

4.1.3 Air Vent Test

The upper air vent of Helmet A was operated for twenty consecutive cycles using the same robotic platform and principal end-effector. Compared with the visor mechanisms, the air vent required a shorter actuation displacement and a more localized interaction point. All programmed cycles were completed successfully during the short experimental trial.

The robotic configuration used for the upper air-vent test is shown in Figure 4.3.

Figure 4.3. Robotic functional test configuration for the upper air vent of Helmet A

As shown in Figure 4.3, the robot was repositioned to access the upper air vent while the same principal end-effector and control architecture were retained.

4.1.4 Chin mechanism Test

The chin mechanism of Helmet A was operated for twenty consecutive cycles using the same robotic platform and principal end-effector. Because the mechanism is located in a different region of the helmet, a dedicated robot approach path was required to avoid interference with the surrounding geometry. All programmed cycles were completed successfully during the short experimental trial.

The robotic configuration used for the chin-mechanism test is shown in

Figure 4.4.

Figure 4.4. Robotic functional test configuration for the chin mechanism of Helmet A

As shown in

Figure 4.4, the robot was repositioned to approach the chin mechanism from a different direction while the same principal end-effector was retained.

4.2 TCP Speed Distribution

The distribution of the controller-derived TCP speed for the four logged Helmet A routines is shown in Figure 4.5. In the figure, the internal sun visor corresponds to the routine identified as DARK_VISOR in the recorded dataset. The horizontal line inside each box represents the median speed, while the triangular marker represents the arithmetic mean. The four routines exhibit clearly different speed distributions. The chin-mechanism routine has the highest mean TCP speed, approximately 81.6 mm/s, while the upper air-vent routine has the lowest mean value, approximately 15.0 mm/s. The internal sun visor has a mean speed of approximately 32.5 mm/s and a median of 25.1 mm/s. For the main visor, the mean is approximately 35.8 mm/s, whereas the median is only approximately 9.8 mm/s. The difference between the mean and median is particularly pronounced for the main-visor and chin-mechanism routines. This right-skewed distribution results from the combination of relatively long low-speed or stationary phases with shorter high-speed robot motions. Therefore, the speed distribution primarily characterizes the different programmed motion profiles, while cycle-to-cycle consistency is evaluated separately in the following sections.

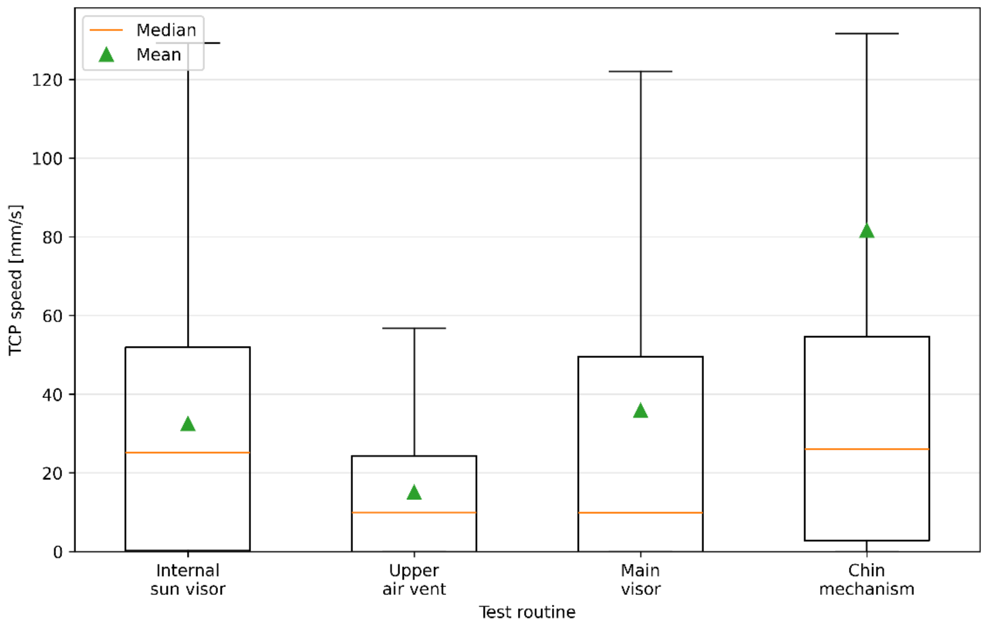

Figure 4.5. TCP speed distribution for the four logged test routines

The horizontal line inside each box represents the median speed, while the triangular marker represents the arithmetic mean. The box limits correspond to the first and third quartiles, and the whiskers are defined using the 1.5×IQR criterion. Statistical fliers are omitted from the Figure 4.5 for visual clarity but remain included in the calculations.

4.3 Joint Motion Range

Joint-angle range and the complete TCP speed trace were examined to verify that the programmed workspace was used without abrupt unplanned discontinuities in the controller data [6].

The overall angular range recorded for each robot joint during the four logged routines is shown in Figure 4.6. The angular range was calculated as the difference between the maximum and minimum controller-reported joint angle over the complete dataset.

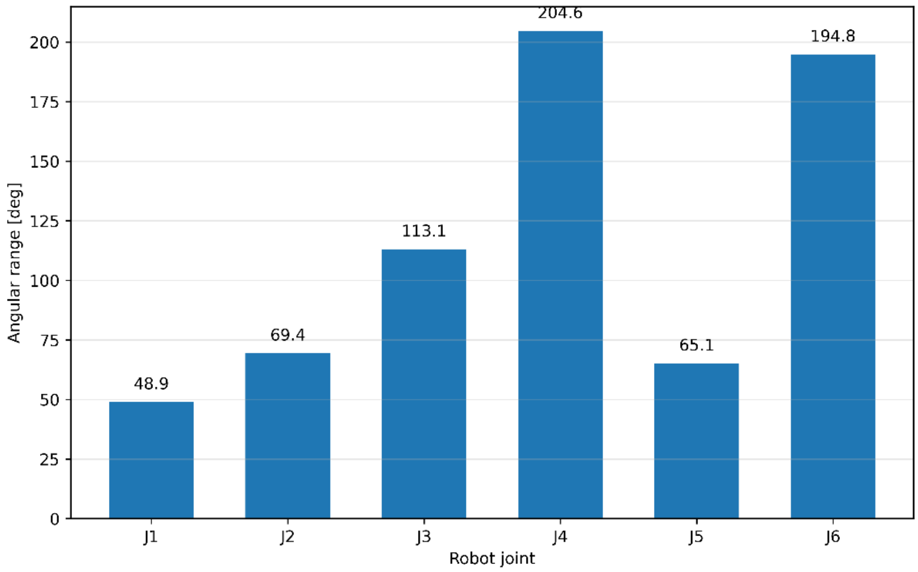

Figure 4.6. Overall joint-angle range recorded during the experimental campaign

As shown in Figure 4.6, J4 and J6 exhibit the largest angular ranges, approximately 204.6° and 194.8°, respectively. J3 also covers a relatively large range of approximately 113.1°, whereas J2 and J5 span approximately 69.4° and 65.1°, respectively. J1 shows the smallest range, approximately 48.9°. The large angular ranges of J4 and J6 indicate that substantial tool-orientation changes were required to access the different helmet mechanisms. These values describe joint utilization during the experimental campaign and do not indicate proximity to the manufacturer-defined joint limits.

4.4 Routine- wise TCP Speed Profiles

The controller-derived TCP speed profiles of the four logged Helmet A routines are shown in Figure 4.7. To improve readability, the routines are presented separately rather than as a single full-run graph.

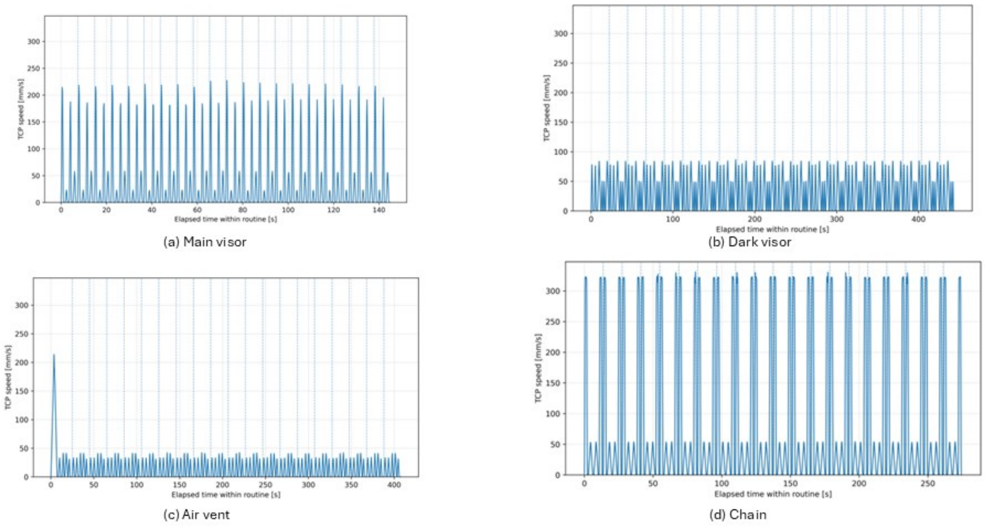

Figure 4.7. Controller-derived TCP speed profiles of the four logged Helmet A routines: (a) internal sun visor, (b) upper air vent, (c) main visor, and (d) chin mechanism

The separated profiles reveal distinct and highly repetitive motion patterns for the four mechanisms. The main-visor routine contains repeated high-speed peaks of approximately 180–230 mm/s together with lower-speed motion segments. The internal sun-visor routine operates predominantly below approximately 90 mm/s, while the chin-mechanism routine exhibits the highest repeated peaks, reaching approximately 320 mm/s. The upper air-vent routine is concentrated mainly below approximately 45 mm/s after the initial transition. The larger peak at the beginning of the air-vent profile is associated with the first-cycle transition and is consistent with the longer duration recorded for that boundary cycle.

The recurrence of similar speed patterns between successive cycle boundaries indicates that the programmed motion structure remained consistent during the repeated executions, while the substantial differences between routines reflect their different trajectories and motion commands.

4.5 Cycle-to-Cycle Timing and Path Consistency

Cycle duration and Cartesian path length were evaluated across the twenty repetitions of each logged routine. Each repetition was identified from the **cycle number** assigned by the supervisory Python program. The start of a repetition was defined as the first logged sample associated with that cycle number, and the end was defined as the last sample carrying the same cycle number. Cycle duration was calculated from the difference between these timestamps, while Cartesian path length was obtained by summing the Euclidean distances between consecutive controller-reported TCP positions within the same cycle.

The cycle duration recorded for each repetition is shown in Figure 4.8.

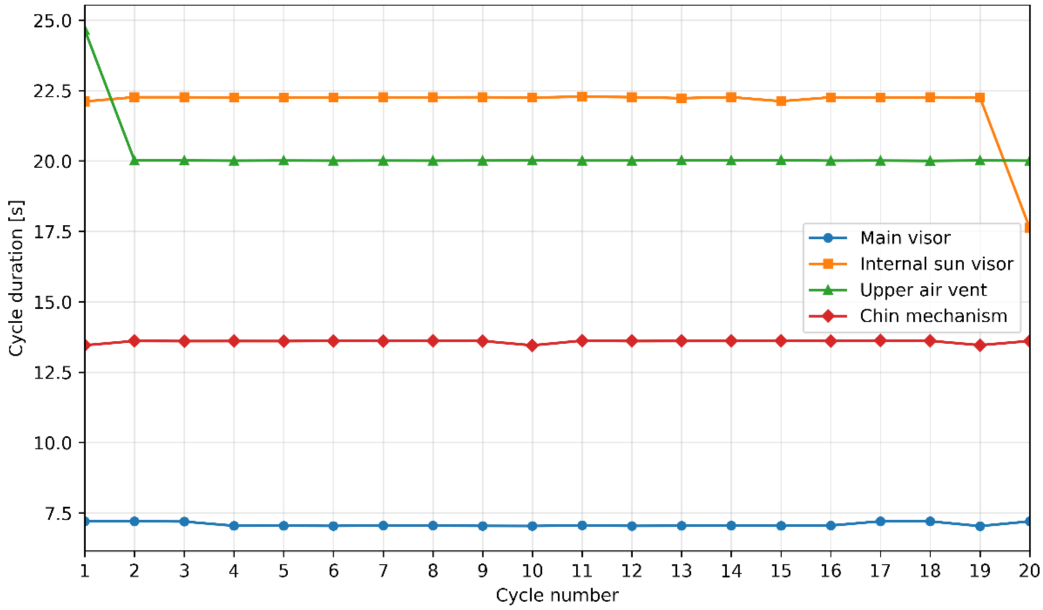

Figure 4.8. Cycle duration for the twenty repetitions of the four logged Helmet A routines

The steady repetitions show relatively stable cycle durations within each routine. For cycles 2–19, the mean duration was approximately 7.08 s for the main visor, 22.25 s for the internal sun visor, 20.01 s for the upper air vent, and 13.59 s for the chin mechanism. The main boundary effects are visible in the first air-vent cycle, which lasts approximately 24.65 s, and in the final internal-sun-visor cycle, which lasts approximately 17.62 s. These deviations originate from transition motions associated with the beginning or end of the programmed routine rather than from random cycle-to-cycle timing variation.

The corresponding controller-derived Cartesian path length for each repetition is shown in Figure 4.9.

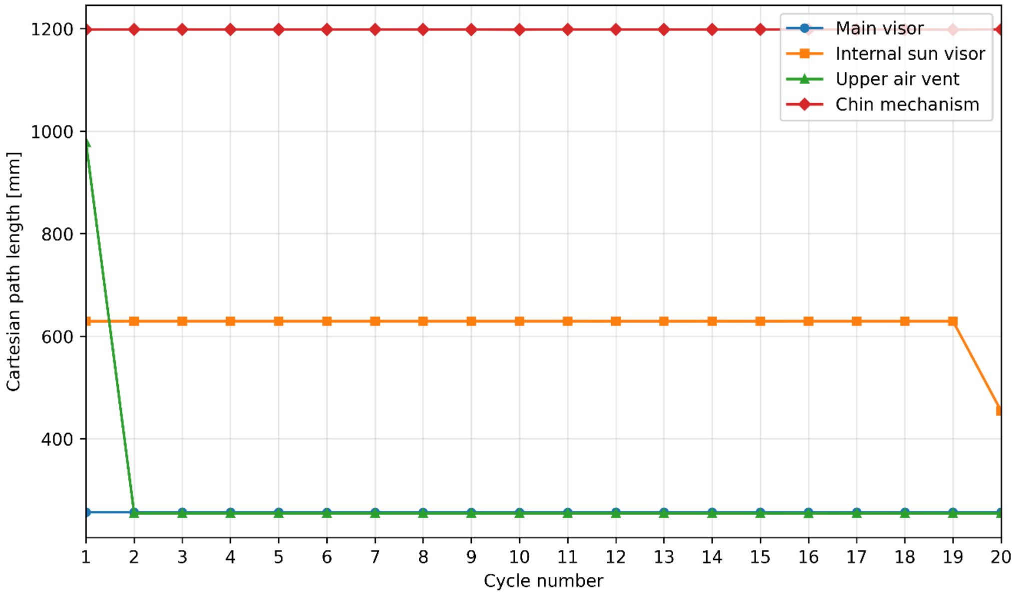

Figure 4.9. Cartesian path length for the twenty repetitions of the four logged Helmet A routines

The steady-cycle path lengths are nearly constant for all four routines. For cycles 2–19, the mean path length was approximately 256.5 mm for the main visor, 629.4 mm for the internal sun visor, 254.5 mm for the upper air vent, and 1198.4 mm for the chin mechanism. The first upper-air-vent cycle has a substantially larger path length of approximately 978.4 mm because it includes the initial transition into the repeated sequence. Conversely, the final internal-sun-visor cycle decreases to approximately 454.1 mm because the inter-cycle return motion is not executed after the final repetition.

Excluding these programmed boundary effects, the nearly horizontal traces indicate that the controller-reported geometric path length remained highly consistent between repeated executions. These results describe the consistency of the programmed controller trajectories and should not be interpreted as measurements of absolute physical TCP accuracy.

4.6 Comparative Stability Indicators

To compare the cycle-to-cycle stability of routines having different absolute durations and path lengths, the coefficient of variation (CV) was calculated for the steady cycles 2–19. The first and final cycles were excluded from this comparison to avoid the boundary effects identified in the previous section. The resulting CV values for cycle duration and controller-derived Cartesian path length are shown in Figure 4.10.

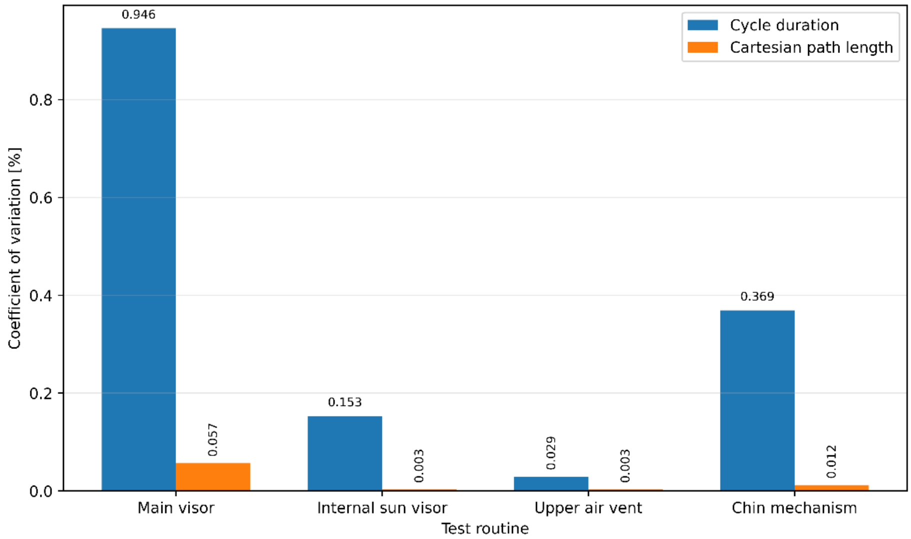

Figure 4.10. Coefficient of variation of steady-cycle duration and Cartesian path length for the four logged Helmet A routines

As shown in Figure 4.10, the coefficient of variation of cycle duration remains below 1% for all four routines. The main-visor routine exhibits the largest timing variation, with a CV of approximately 0.946%, followed by the chin mechanism at approximately 0.369%. Lower values are observed for the internal sun visor and upper air vent, with CV values of approximately 0.153% and 0.029%, respectively.

The relative variation in Cartesian path length is substantially smaller than the timing variation. The highest path-length CV is observed for the main visor at approximately 0.057%, followed by the chin mechanism at approximately 0.012%. The internal sun visor and upper air vent both show values of approximately 0.003%. These results indicate that the controller-reported geometric path length remained highly stable across the steady repeated cycles, even when small variations in execution time were present.

The Helmet B visor trial is not included in this quantitative comparison because it was performed separately as a functional reconfiguration demonstration. The CV values reported here describe short-run controller-level consistency and should not be interpreted as measurements of absolute physical positioning accuracy or long-term durability performance.

4.7 Controller-Reported Trajectory Consistency

The purpose of this analysis was to evaluate how closely the controller-reported TCP path was reproduced from one repetition to the next. Only cycles 2–19 were considered in order to focus on the steady repeated operation and to exclude possible transition effects associated with the first and final cycles.

The raw trajectories could not be compared directly point by point because different repetitions contained slightly different numbers of logged samples. Therefore, each cycle was first represented according to its progress along the Cartesian path. The beginning of the cycle was defined as 0% path progress and the end as 100%. Each trajectory was then interpolated onto 200 equally spaced path-progress points. This procedure made it possible to compare the TCP position at the same relative stage of motion across all eighteen steady cycles

For each test routine, a mean trajectory was then calculated. At each of the 200 normalized path-progress points, the X, Y, and Z coordinates of the eighteen steady cycles were averaged. The resulting sequence of mean Cartesian positions represents the reference trajectory for that routine.

The three-dimensional Cartesian distance between each individual cycle and the mean trajectory was calculated at every corresponding path-progress point. These distances were then combined using a root-mean-square (RMS) calculation to obtain one trajectory-consistency value for each routine. If all repeated trajectories were identical, the RMS value would be zero. Therefore, a lower RMS value indicates that the repeated controller-reported paths remained closer to the mean trajectory, while a higher value indicates greater cycle-to-cycle path dispersion.

The resulting RMS Cartesian deviations for the four logged Helmet A routines are shown in Figure 4.11.

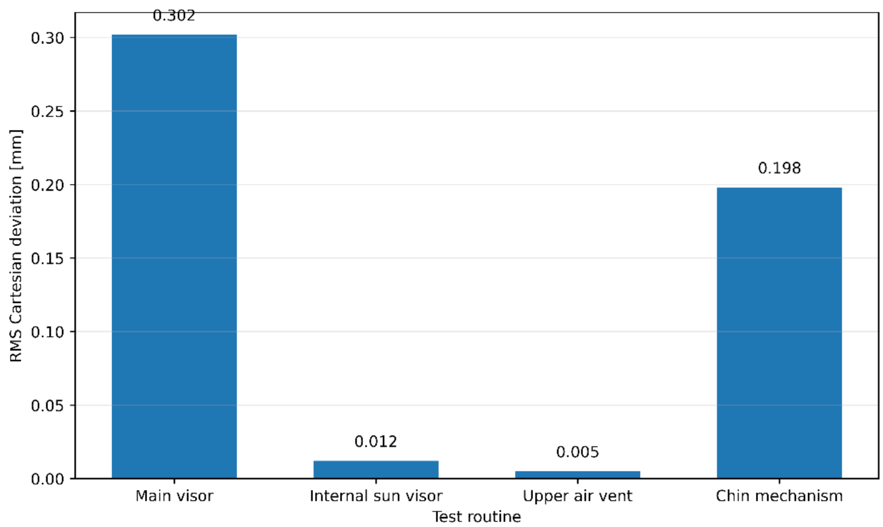

Figure 4.11. RMS Cartesian deviation from the mean controller-reported trajectory for steady cycles 2–19

As shown in Figure 4.11, the upper air-vent routine exhibits the lowest RMS deviation, approximately 0.005 mm, followed by the internal sun visor at approximately 0.012 mm. The chin-mechanism routine shows a higher deviation of approximately 0.198 mm, while the main-visor routine has the largest value, approximately 0.302 mm. These results indicate that the controller-reported trajectories of the upper air vent and internal sun visor remained particularly close to their respective mean paths, whereas greater cycle-to-cycle path dispersion was observed for the chin mechanism and main visor.

These values are derived exclusively from the Cartesian positions reported by the Dobot controller. They therefore describe relative consistency between repeated controller trajectories and should not be interpreted as independently measured physical TCP accuracy. In addition, the end-effector was operated in open loop and no external sensor was used to measure the actual helmet contact point. Consequently, small physical slips or missed motor steps would not necessarily be detected by this trajectory-consistency metric.

4.8 Robot Trajectory Analysis

The recorded three-dimensional TCP trajectories were examined to visualize the overall workspace used by the robot and the sequence of approach, interaction, transfer, and withdrawal motions. Before the repeated tests, low-speed trial movements were performed to verify adequate clearance between the robot, helmet fixture, and surrounding components.

A three-dimensional overview of the recorded robot motion is shown in Figure 4.12.

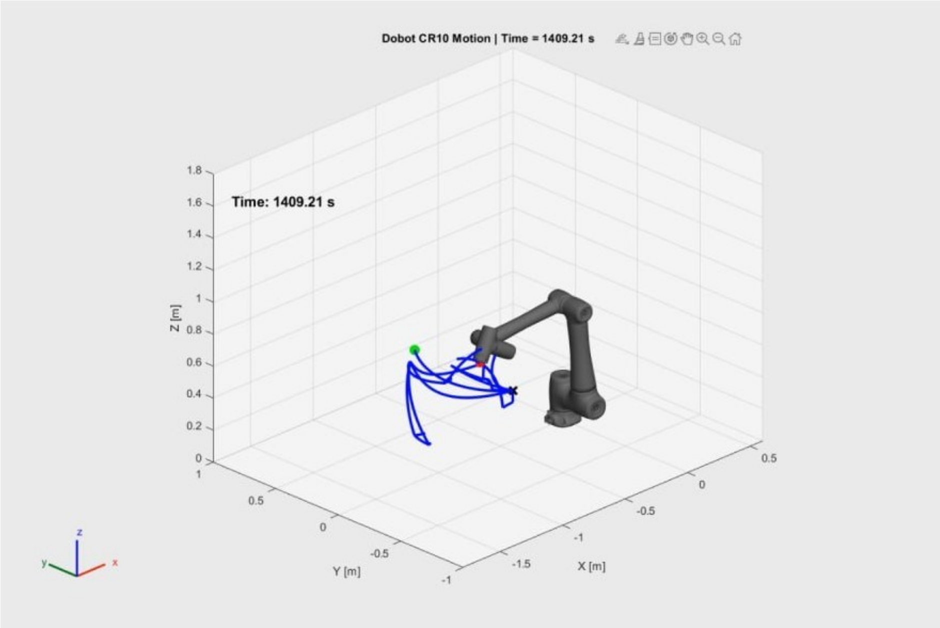

Figure 4.12. Three-dimensional overview of the recorded robot motion during the experimental campaign

The figure shows that the interaction points are located within a compact region of the robot workspace and that repeated motion segments visually overlap over successive cycles.

To make the temporal progression and direction of the recorded motion easier to interpret, the complete TCP trajectory was plotted using a continuous colour gradient based on elapsed time, together with directional arrows along the path, as shown in Figure 4.13.

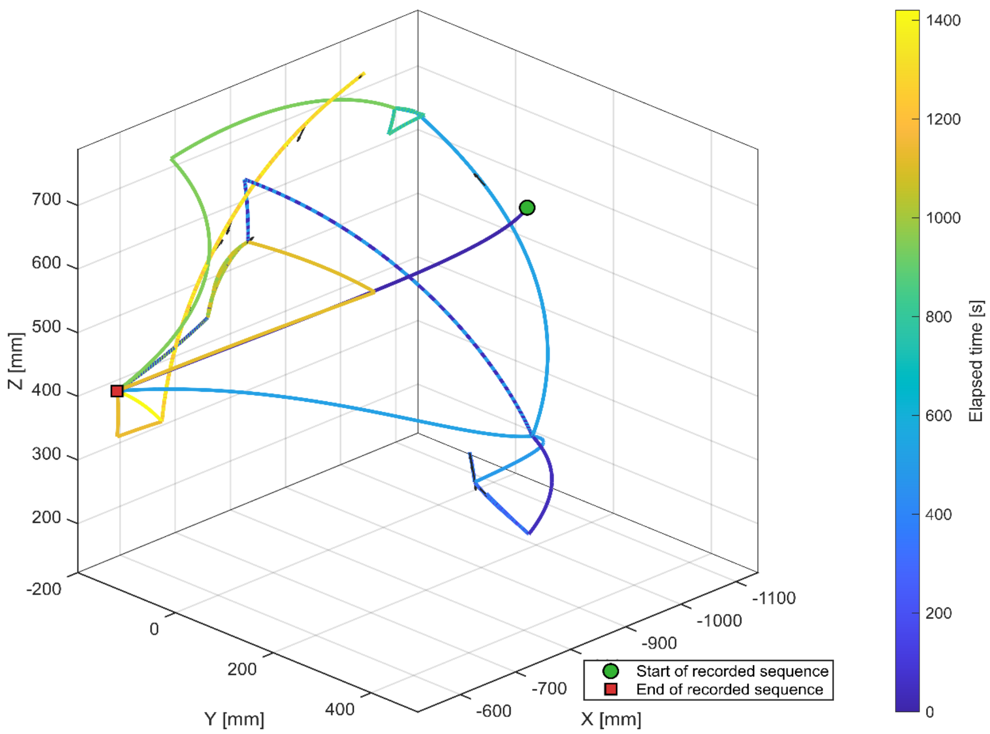

Figure 4.13. Complete controller-reported TCP trajectory with time-based color progression and motion-direction indicators

The colour progression represents the chronological evolution of the recorded motion, while the arrows indicate the local direction of TCP travel. This representation makes it possible to distinguish the progression through the repeated transfer, approach, interaction, and withdrawal segments more clearly than in the single-color trajectory plot.

No collision or emergency interruption was recorded during the reported experimental campaign. This confirms that the programmed trajectories could be executed safely for the tested fixture and helmet configurations. However, any change in fixture placement, TCP definition, or helmet geometry requires renewed low-speed verification before automatic execution.

4.9 Quantitative Comparison of Helmet A and Helmet B Main-Visor Tests

In addition to the functional reconfiguration demonstrated with Helmet B, controller trajectory data were recorded during its twenty-cycle main-visor test. This makes it possible to compare the two main-visor configurations quantitatively. To maintain consistency with the previous analyses, the steady-cycle metrics were calculated using cycles 2–19. The comparison is summarized in Table 4.1.

Table 4.1. Comparison of controller-derived steady-cycle metrics for the Helmet A and Helmet B main-visor tests

Metric	Helmet A Main Visor	Helmet B Main Visor
Mean cycle duration [s]	7.083	7.023
Cycle-duration CV [%]	0.946	0.859
Mean Cartesian path length [mm]	256.5	681.0
Path-length CV [%]	0.057	0.016
RMS trajectory deviation [mm]	0.302	0.083

The two configurations have very similar mean cycle durations, approximately 7.08 s for Helmet A and 7.02 s for Helmet B. However, their Cartesian path lengths are substantially different: the Helmet B visor routine covers approximately 681.0 mm per steady cycle, compared with approximately 256.5 mm for Helmet A. This difference reflects the different visor geometry and the corresponding change in the programmed robot trajectory.

Despite the substantially different motion geometry, the relative cycle-to-cycle variation remains low for both configurations. The duration CV is below 1% for both helmets, while the Cartesian path-length CV remains below 0.06%. The controller-reported RMS trajectory deviation is approximately 0.302 mm for Helmet A and 0.083 mm for Helmet B, indicating low short-run trajectory dispersion in both configurations.

These results provide quantitative support for the reconfigurability demonstrated with Helmet B. The second helmet required different target poses and a substantially different Cartesian trajectory, while the same robotic platform and principal end-effector architecture were retained. Therefore, the Helmet B experiment demonstrates not only functional operation on a second geometry but also stable controller-level execution over twenty repeated cycles.

The numerical values should not be interpreted as a direct ranking of robot accuracy between the two helmets, since the two routines follow different trajectory geometries. As in the previous analyses, all metrics are derived from internal controller feedback rather than independent external metrology.

4.10 General Discussion

The experimental campaign demonstrates the feasibility of using the developed robotic platform for short repetitive functional testing of motorcycle helmet mechanisms. The Dobot CR10, custom end-effector, Arduino-based motor control, and Python supervisory program operated as an integrated system and completed the programmed test sequences without continuous manual intervention.

A central result of the work is the reconfigurability of the proposed platform. The same principal end-effector was used to operate the main visor, internal sun visor, upper air vent, and chin mechanism of Helmet A. The same robotic platform and end-effector architecture were also retained for the main visor of Helmet B, while the robot target poses and trajectory were modified to accommodate the different visor geometry.

The quantitative comparison between the two main-visor configurations further supports this reconfigurability. Although the mean steady-cycle duration was similar for Helmet A and Helmet B, approximately 7.08 s and 7.02 s respectively, the corresponding Cartesian path lengths were substantially different, approximately 256.5 mm and 681.0 mm. Despite this change in motion geometry, both configurations exhibited low cycle-duration and path-length variation. This indicates that a substantial change in the programmed robot trajectory could be accommodated while retaining stable short-run controller-level execution.

The controller-data analysis also showed that each Helmet A routine had a distinct motion profile. Differences were observed in TCP speed, joint utilization, cycle duration, Cartesian path length, and RMS trajectory deviation. Within each individual routine, however, the steady-cycle duration and path length remained relatively stable, and the trajectory-consistency analysis showed limited cycle-to-cycle dispersion of the controller-reported TCP paths. These results support short-run consistency of the programmed motion sequences rather than absolute robot positioning accuracy.

From an automation perspective, the principal advantage of the robotic approach is that changes in interaction position and motion direction can be implemented primarily through robot programming rather than through a complete redesign of the mechanical test device. This flexibility is particularly relevant during prototype development or when several product geometries must be evaluated. Dedicated mechanical rigs may nevertheless remain more appropriate for standardized high-cycle durability testing when flexibility is not the primary requirement.

The results must nevertheless be interpreted within the limitations of the present experimental setup. Each configuration was evaluated over only twenty consecutive cycles, and the end-effector was operated in open loop. No force/torque sensor, external position-measurement system, or vision-based feedback was used. Consequently, the experiments do not quantify contact force, physical slip, wear, absolute TCP accuracy, or long-term durability.

The developed system should therefore be regarded as a functional proof-of-concept for flexible robotic repetition rather than as a certified durability-testing platform. Within this scope, the results demonstrate that one robotic cell and one principal end-effector architecture can be adapted to multiple helmet mechanisms and to a second helmet geometry while maintaining measurable short-run controller-level consistency.

Chapter 5: CONCLUSIONS AND FUTURE WORK

This chapter summarizes the outcomes of the proof-of-concept, clarifies the limits of the evidence, and identifies the work required before the platform can be used for long-duration or standardized durability testing.

The objective was to design and preliminarily validate a flexible robotic platform for repetitive functional testing of motorcycle helmet mechanisms. A Dobot CR10, a custom rack-and-pinion end-effector, a stepper motor, Arduino control, and computer-based supervisory software were integrated into one automated cell.

The end-effector was designed in Autodesk Inventor and manufactured by 3D printing. Its motor-driven jaws provided the local actuation stroke, while the robot positioned the tool at the mechanism-specific interaction points.

The electronic and software architecture implemented sequential synchronization: robot positioning was completed before stepper actuation, and the next robot motion began only after an Arduino acknowledgement.

Twenty-cycle trials were completed for the main visor, internal sun visor, upper air vent, and chin of Helmet A, together with a visor trial on Helmet B. The same principal end-effector was retained, while robot target poses were adapted to the mechanism geometry.

The controller logs showed stable steady-cycle durations and path lengths, and the trajectory RMS analysis provided a quantitative comparison of controller-reported path consistency. The experiments also confirmed collision-free execution for the tested fixture configurations.

The objectives of mechanical integration, automated execution, and preliminary consistency assessment were achieved. The work should nevertheless be understood as a functional proof-of-concept, not as a complete fatigue test or an independent validation of absolute robot accuracy.

The main contributions of this thesis can be summarized as follows:

- Design and manufacture of a compact rack-and-pinion end-effector for automated operation of several helmet mechanisms.
- Integration of robot motion, stepper actuation, Arduino control, and supervisory software in a single laboratory platform.
- Implementation of a sequential synchronization strategy between robot positioning and end-effector actuation.
- Preliminary functional validation on multiple mechanisms and two helmet geometries.
- Demonstration that reconfiguration can be achieved mainly through target-pose and trajectory changes while retaining the principal end-effector structure.

- Definition of controller-data metrics for cycle timing, path length, joint utilization, TCP speed, and steady-cycle trajectory deviation.

Although the proposed robotic platform achieved the intended objectives, several limitations should be acknowledged.

First, only two helmet models were considered, and the second model was used only for a separate visor reconfiguration trial. The results therefore do not establish general compatibility with commercial helmet geometries.

Second, each configuration was tested for twenty cycles. This is adequate for short-run feasibility but far below the cycle count required for fatigue-life or durability qualification.

Third, the robot follows predefined trajectories and the stepper-driven gripper is open-loop. There is no force/torque sensor, encoder-based jaw feedback, limit switch, or machine-vision correction.

Finally, the motion analysis relies on internal controller feedback. No external metrology was used to measure absolute TCP position, contact-point motion, or physical repeatability. The system also does not quantify actuation force, deformation, slip, or component wear.

Several improvements can be considered in future research to enhance the capabilities of the proposed robotic testing platform.

A wrist or end-effector force/torque sensor should be added to quantify interaction load, detect failed engagement, and compare mechanical resistance over long campaigns.

Machine vision and a reproducible fixture datum should be used to localize the helmet and compensate for small changes in specimen placement.

The end-effector should be upgraded with homing, limit switches, and encoder feedback before adaptive control is introduced. Closed-loop jaw position would make the commanded actuation verifiable rather than assumed from motor steps.

The test campaign should then be extended from tens to thousands of cycles, with planned inspection intervals and quantitative measurements of force, backlash, wear, and failure events.

Independent metrology should be used to validate TCP and contact-point repeatability under defined payload and speed conditions. Only after these additions should the platform be compared formally with dedicated industrial durability equipment.

APPENDIX A – PYTHON CONTROL CODE

A.1 Overview

This appendix presents the complete Python implementation used to control the robotic testing sequence and coordinate communication with the Arduino-based actuation system.

The code is responsible for:

- Defining the motion sequence
- Sending trigger signals to the Arduino
- Executing the full functional cycle

A.2 Python Code

```
import socket
import serial
import time

# ========================
# SETTINGS
# ========================
ROBOT_IP = "192.168.2.10"

DASHBOARD_PORT = 29999
MOVE_PORT = 30003

ARDUINO_PORT = "COM5"
ARDUINO_BAUD = 9600

# choose: "dark", "vent", "main", "chin", "all"
RUN_MODE = "all"

XYZ_TOL = 8.0          # mm
STABLE_XYZ_TOL = 0.8   # mm
STABLE_HITS_REQUIRED = 5
POSE_TIMEOUT = 90      # seconds
POLL_DELAY = 0.15      # seconds

# ========================
# POINTS
# ========================
P1 = [-758.8, 464.0, 462.83, -112.9, -53.30, 176.61]
P2 = [-974.0, 214.11, 153.74, -97.71, -19.14, 178.58]
P3 = [-970.5, 131.74, 190.37, -88.56, 0.3986, 176.77]
P4 = [-1024.0, 46.781, 165.00, -99.88, -6.646, -110.9]
P5 = [-1043.0, 14.937, 195.44, -99.88, -6.646, -110.9]
P7 = [-955.5, 228.14, 687.57, 165.08, -2.539, -5.671]
```

```
P8 = [-1107.0, -168.5, 650.05, 177.28, -0.615, 4.455]
P9 = [-1068.0, -131.8, 677.35, 171.14, 2.8079, 11.303]
P10 = [-1112.0, -154.3, 647.08, 172.57, 2.7185, 5.5055]
P11 = [-648.8, -151.4, 744.09, 171.44, -0.072, 11.365]
P12 = [-550.9, -151.0, 413.98, 175.94, 71.102, -1.892]
P13 = [-712.0, -147.6, 473.00, -175.1, -57.12, 178.47]
P14 = [-721.0, -146.4, 508.09, -179.0, -63.55, -177.4]
P15 = [-771, -151.5, 561.65, -178.8, -62.19, -177.5]
P16 = [-784.7, -151.4, 563.59, -178.7, -61.26, -177.7]
P17 = [-782.4, -151.5, 664.15, -178.7, -61.98, -177.6]
P18 = [-634, -152.9, 335.84, 166.88, -71.37, -163.8]
P19 = [-731.2, -141.8, 596.07, -178.0, -35.87, -176.5]
P20 = [-912.6, -142.8, 730.39, -175.4, -16.66, -177.7]
P21 = [-996.8, -147.0, 756.93, -175.4, -16.66, -177.7]
P22 = [-550.9, -151.0, 341.96, 167.61, -70.68, -164.5]
P23 = [-789.7, 107.04, 565.59, -178.7, -61.98, 163.69]

# =========================
# SOCKET HELPERS
# =========================
def create_socket(ip, port, timeout=5):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(timeout)
    s.connect((ip, port))
    return s

# =========================
# ROBOT HELPERS
# =========================
def connect_robot():
    dash = create_socket(ROBOT_IP, DASHBOARD_PORT, timeout=5)
    move = create_socket(ROBOT_IP, MOVE_PORT, timeout=5)
    print("[Robot] Dashboard connected")
    print("[Robot] Move connected")
    return dash, move

def dash_cmd(dash, cmd, verbose=True):
    dash.send((cmd + "\n").encode())
    reply = dash.recv(1024).decode(errors="ignore").strip()
    if verbose:
        print(f"[DASH] {cmd}")
        print(f"[DASH REPLY] {reply}")
    return reply

def move_cmd(move, cmd, verbose=True):
    move.send((cmd + "\n").encode())
    if verbose:
        print(f"[MOVE] {cmd}")
def prepare_robot(dash):
    dash_cmd(dash, "ClearError()")
```

```
    dash_cmd(dash, "EnableRobot()")
    dash_cmd(dash, "User(0)")
    dash_cmd(dash, "Tool(0)")
    dash_cmd(dash, "SpeedJ(10)")
    dash_cmd(dash, "AccJ(10)")
    dash_cmd(dash, "RobotMode()")
    dash_cmd(dash, "GetPose()")

def stop_robot(dash):
    try:
        dash_cmd(dash, "Stop()")
    except Exception:
        print("[Robot] Stop failed")

def parse_pose_reply(reply):
    try:
        start = reply.find("{")
        end = reply.find("}")
        if start == -1 or end == -1:
            return None

        values = reply[start + 1:end].split(",")
        if len(values) < 6:
            return None

        return [float(v) for v in values[:6]]
    except Exception:
        return None

def get_current_pose(dash, verbose=False):
    reply = dash_cmd(dash, "GetPose()", verbose=verbose)
    return parse_pose_reply(reply)

def is_xyz_close(current, target, tol=XYZ_TOL):
    if current is None:
        return False

    for i in range(3):
        if abs(current[i] - target[i]) > tol:
            return False
    return True

def is_xyz_stable(current, last_pose, tol=STABLE_XYZ_TOL):
    if current is None or last_pose is None:
        return False

    for i in range(3):
        if abs(current[i] - last_pose[i]) > tol:
            return False
    return True
```

```
def wait_until_xyz_stable(dash, target, name="", timeout=POSE_TIMEOUT):
    label = name if name else "POINT"
    start_time = time.time()
    stable_hits = 0
    last_pose = None
    last_print = 0

    while time.time() - start_time < timeout:
        current = get_current_pose(dash, verbose=False)

        close_to_target = is_xyz_close(current, target, XYZ_TOL)
        stable_vs_last = is_xyz_stable(current, last_pose, STABLE_XYZ_TOL)

        if close_to_target and stable_vs_last:
            stable_hits += 1
        else:
            stable_hits = 0

        now = time.time()
        if now - last_print > 1.0 and current is not None:
            print(f"[WAIT] {label} | xyz={current[:3]} |
stable={stable_hits}/{STABLE_HITS_REQUIRED}")
            last_print = now

        if stable_hits >= STABLE_HITS_REQUIRED:
            print(f"[Robot] Reached and settled at {label}")
            return True

        last_pose = current
        time.sleep(POLL_DELAY)

    raise RuntimeError(f"Timeout waiting for robot to reach and settle at {label}")

def movj(dash, move, p, name=""):
    label = name if name else "POINT"
    print(f"\n--- MoveJ to {label} ---")
    cmd = (
        f"MovJ({p[0]},{p[1]},{p[2]},{p[3]},{p[4]},{p[5]},"
        f"User=0,Tool=0,SpeedJ=30,AccJ=30)"
    )
    move_cmd(move, cmd)
    wait_until_xyz_stable(dash, p, label)

def arc(dash, move, p1, p2, name1="", name2=""):
    label1 = name1 if name1 else "P1"
    label2 = name2 if name2 else "P2"
    print(f"\n--- Arc from {label1} to {label2} ---")
    cmd = (
        f"Arc({p1[0]},{p1[1]},{p1[2]},{p1[3]},{p1[4]},{p1[5]},"
        f"{p2[0]},{p2[1]},{p2[2]},{p2[3]},{p2[4]},{p2[5]},"
        f"User=0,Tool=0,SpeedJ=4,AccJ=4)"
```

```
    )
    move_cmd(move, cmd)
    wait_until_xyz_stable(dash, p2, label2)

# ========================
# ARDUINO HELPERS
# ========================
def connect_arduino():
    arduino = serial.Serial(ARDUINO_PORT, ARDUINO_BAUD, timeout=2)
    time.sleep(3)
    arduino.reset_input_buffer()
    arduino.reset_output_buffer()
    print("[Arduino] Connected")
    return arduino

def send_arduino_cmd(arduino, cmd, done_word, timeout=20):
    arduino.reset_input_buffer()
    arduino.write((cmd + "\n").encode())
    print(f"[Arduino] {cmd}")

    start_time = time.time()
    while time.time() - start_time < timeout:
        if arduino.in_waiting > 0:
            msg = arduino.readline().decode(errors="ignore").strip()
            if msg:
                print(f"[Arduino Reply] {msg}")
                if msg == done_word:
                    return True
                if msg == "ERROR_UNKNOWN_COMMAND":
                    return False
        time.sleep(0.05)

    return False

def motor_forward(arduino, turns):
    ok = send_arduino_cmd(arduino, f"FORWARD {turns}", "DONE_FORWARD")
    if not ok:
        raise RuntimeError(f"Motor forward failed: {turns}")

def motor_backward(arduino, turns):
    ok = send_arduino_cmd(arduino, f"BACKWARD {turns}",
"DONE_BACKWARD")
    if not ok:
        raise RuntimeError(f"Motor backward failed: {turns}")

# ========================
# TEST SEQUENCES
# ========================
def test_dark_visor(dash, move, arduino):
    print("\n==============================")
    print("TEST: DARK VISOR TEST")
```

```
    print("==============================")

    prepare_robot(dash)
    movj(dash, move, P12, "P12")
    movj(dash, move, P13, "P13")
    motor_forward(arduino, 4.39)

    arc(dash, move, P14, P15, "P14", "P15")
    movj(dash, move, P16, "P16")
    motor_backward(arduino, 4.39)
    movj(dash, move, P17, "P17")

    movj(dash, move, P1, "P1")
    movj(dash, move, P2, "P2")
    movj(dash, move, P3, "P3")

    motor_forward(arduino, 2.9)
    movj(dash, move, P2, "P2")
    motor_backward(arduino, 2.5)

    movj(dash, move, P4, "P4")
    movj(dash, move, P5, "P5")
    movj(dash, move, P4, "P4")
    movj(dash, move, P7, "P7")
    movj(dash, move, P17, "P17")
    movj(dash, move, P16, "P16")
    motor_forward(arduino, 4.39)
    arc(dash, move, P15, P14, "P15", "P14")
    movj(dash, move, P13, "P13")
    motor_backward(arduino, 4.39)
    movj(dash, move, P12, "P12")
    movj(dash, move, P7, "P7")

    print("=== END: DARK VISOR TEST ===")

def test_helmet_vent(dash, move, arduino):
    print("\n==============================")
    print("TEST: HELMET VENT TEST")
    print("==============================")

    prepare_robot(dash)

    movj(dash, move, P8, "P8")
    motor_forward(arduino, 1.5)

    movj(dash, move, P9, "P9")
    movj(dash, move, P10, "P10")
    motor_backward(arduino, 1.6)

    movj(dash, move, P11, "P11")
```

```
    print("=== END: HELMET VENT TEST ===")

def test_main_visor(dash, move, arduino):
    print("\n==============================")
    print("TEST: MAIN VISOR TEST")
    print("==============================")

    prepare_robot(dash)

    movj(dash, move, P12, "P12")
    movj(dash, move, P13, "P13")
    motor_forward(arduino, 4.39)

    arc(dash, move, P14, P15, "P14", "P15")
    movj(dash, move, P16, "P16")
    arc(dash, move, P15, P14, "P15", "P14")
    movj(dash, move, P13, "P13")
    arc(dash, move, P14, P15, "P14", "P15")
    movj(dash, move, P16, "P16")
    motor_backward(arduino, 4.39)
    movj(dash, move, P23, "P23")
    movj(dash, move, P12, "P12")

    print("=== END: MAIN VISOR TEST ===")

def chin(dash, move, arduino):
    print("\n==============================")
    print("TEST: CHIN")
    print("==============================")

    prepare_robot(dash)

    movj(dash, move, P22, "P22")
    movj(dash, move, P18, "P18")

    motor_forward(arduino, 3.7)

    arc(dash, move, P19, P20, "P19", "P20")
    movj(dash, move, P21, "P21")
    movj(dash, move, P20, "P20")

    arc(dash, move, P19, P18, "P19", "P18")
    motor_backward(arduino, 3.2)

    movj(dash, move, P12, "P12")

    print("=== END: CHIN ===")

# =========================
# MAIN
# =========================
```

```
def main():
    dash = None
    move = None
    arduino = None

    try:
        dash, move = connect_robot()
        arduino = connect_arduino()

        if RUN_MODE == "dark":
            test_dark_visor(dash, move, arduino)

        elif RUN_MODE == "vent":
            test_helmet_vent(dash, move, arduino)

        elif RUN_MODE == "main":
            test_main_visor(dash, move, arduino)

        elif RUN_MODE == "chin":
            chin(dash, move, arduino)

        elif RUN_MODE == "all":
            test_dark_visor(dash, move, arduino)
            test_helmet_vent(dash, move, arduino)
            test_main_visor(dash, move, arduino)
            chin(dash, move, arduino)

        else:
            raise ValueError("RUN_MODE must be 'dark', 'vent', 'main', 'chin', or 'all'")

        print("\n=== PROGRAM FINISHED SUCCESSFULLY ===")

    except KeyboardInterrupt:
        print("\n[STOPPED] KeyboardInterrupt")
        if dash is not None:
            stop_robot(dash)

    except Exception as e:
        print(f"\n[ERROR] {e}")
        if dash is not None:
            stop_robot(dash)

    finally:
        if arduino is not None and arduino.is_open:
            arduino.close()
            print("[Arduino] Closed")

        if move is not None:
            try:
                move.close()
            except Exception:
```

```
            pass
        print("[Robot Move] Closed")

    if dash is not None:
        try:
            dash.close()
        except Exception:
            pass
        print("[Robot Dashboard] Closed")

if __name__ == "__main__":
    main()
```

APPENDIX B – ARDUINO CONTROL CODE

B.1 Overview

This appendix presents the Arduino code used to control the stepper motor integrated within the end-effector.

The Arduino system:

- Receives trigger signals from Python
- Executes opening and closing motion
- Ensures repeatable actuation

B.2 Arduino Code

```
const int stepPin = 2;
const int dirPin  = 5;
const int enPin   = 8;

const int stepsPerRevolution = 200;

void moveMotor(float turns, bool direction) {
  int totalSteps = (int)(stepsPerRevolution * turns);

  digitalWrite(enPin, LOW);   // driver enabled
  digitalWrite(dirPin, direction);

  for (int i = 0; i < totalSteps; i++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }

  // IMPORTANT:
  // do NOT disable driver here
  // digitalWrite(enPin, HIGH);
}

void setup() {
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(enPin, OUTPUT);

  digitalWrite(enPin, HIGH);

  Serial.begin(9600);
  delay(500);
```

```
  Serial.println("ARDUINO_READY");
}

void loop() {
  if (Serial.available() > 0) {
    String cmd = Serial.readStringUntil('\n');
    cmd.trim();

    Serial.print("RECEIVED: ");
    Serial.println(cmd);

    if (cmd.startsWith("FORWARD ")) {
      float turns = cmd.substring(8).toFloat();
      Serial.print("ACTION: FORWARD ");
      Serial.println(turns);
      moveMotor(turns, HIGH);
      Serial.println("DONE_FORWARD");
    }
    else if (cmd.startsWith("BACKWARD ")) {
      float turns = cmd.substring(9).toFloat();
      Serial.print("ACTION: BACKWARD ");
      Serial.println(turns);
      moveMotor(turns, LOW);
      Serial.println("DONE_BACKWARD");
    }
    else if (cmd == "HOLD_ON") {
      digitalWrite(enPin, LOW);
      Serial.println("DONE_HOLD_ON");
    }
    else if (cmd == "HOLD_OFF") {
      digitalWrite(enPin, HIGH);
      Serial.println("DONE_HOLD_OFF");
    }
    else {
      Serial.println("ERROR_UNKNOWN_COMMAND");
    }
  }
}
```

Bibliography

[1] European Committee for Standardization, *EN 13087: Protective Helmets—Test Methods*, Brussels, Belgium: CEN, 2000.

[2] International Organization for Standardization, *ISO 13232-1: Motorcycles—Test and Analysis Procedures for Research Evaluation of Rider Crash Protective Devices*, Geneva, Switzerland: ISO, 2005.

[3] D. Otte and C. Haasper, "Injury protection and helmet performance in powered two-wheeler accidents," *International Journal of Crashworthiness*, vol. 17, no. 4, pp. 435–448, 2012.

[4] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, 4th ed., Pearson, 2015.

[5] W. Bolton, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, 7th ed., Pearson, 2019.

[6] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer, 2010.

[7] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 4th ed., Pearson, 2018.

[8] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed., Springer, 2017.

[9] S. B. Niku, *Introduction to Robotics: Analysis, Control, Applications*, 3rd ed., Wiley, 2020.

[10] International Organization for Standardization, *ISO 9283: Manipulating Industrial Robots—Performance Criteria and Related Test Methods*, Geneva, Switzerland: ISO, 1998.

[11] Dobot Robotics, *CR Series Collaborative Robot User Manual*, Shenzhen, China: Dobot Robotics Co., Ltd., 2023.

[12] K. H. Lee and H. Hashimoto, "Intelligent robotic systems for flexible automation and testing applications," *Robotics and Computer-Integrated Manufacturing*, vol. 17, no. 3, pp. 253–262, 2001.

[13] S. Y. Nof, *Handbook of Industrial Robotics*, 2nd ed., Wiley, 1999.

[14] J. R. Hendershot and T. J. E. Miller, *Design of Brushless Permanent-Magnet Motors*, Oxford University Press, 2010.

[15] M. P. Groover, *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*, 7th ed., Wiley, 2020.

[16] W. Lu, Z. Yao, J. Yao, R. Tang, and S. He, “Experimental method research and miniature durability test-bench design of electric sliding door,” Advances in Mechanical Engineering, vol. 15, no. 3, pp. 1–11, 2023, doi: 10.1177/16878132231163679.

[17] A. A. Goldenberg, B. Benhabib, and R. G. Fenton, “A complete generalized solution to the inverse kinematics of industrial robots,” *IEEE Journal on Robotics and Automation*, vol. 1, no. 1, pp. 14–20, 1985.

[18] M. Sotgiu et al., “Automated robotic test bench for functional testing of helicopter sliding doors,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[19] M. K. Klankers, A. Rudloff, P. Mohammadi, N. Hoffmann, S. M. Mir Latifi, R. Gökay, R. Nagwekar, R. Weidner, and J. J. Steil, “Lessons Learned from Investigating Robotics-Based, Human-like Testing of an Upper-Body Exoskeleton,” Applied Sciences, vol. 14, no. 6, p. 2481, 2024, doi: 10.3390/app14062481.

[20] J. Hanke, C. Eymüller, J. Reichmann, A. Trauth, M. Sause, and W. Reif, “Software-defined testing facility for component testing with industrial robots,” in 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Stuttgart, Germany, 2022, pp. 1–8, doi: 10.1109/ETFA52439.2022.9921625.

[21] A. Valera, F. Benimeli, J. Solaz, H. De Rosario, A. Robertsson, K. Nilsson, R. Zotovic, and M. Mellado, “A car-seat example of automated anthropomorphic testing of fabrics using force-controlled robot motions,” IEEE Transactions on Automation Science and Engineering, vol. 8, no. 2, pp. 280–291, 2011, doi: 10.1109/TASE.2010.2079931.

[22] R. Müller, M. Vette, and M. Scholer, “Inspector Robot - a new collaborative testing system designed for the automotive final assembly line,” Procedia CIRP, vol. 23, pp. 59–64, 2014, doi: 10.1016/j.procir.2014.10.093.

[23] C. N. MacLeod et al., “Robotic ultrasonic inspection of complex components using automated scanning trajectories,” *NDT & E International*, vol. 102, pp. 1–10, 2019.

[24] J. Zhang, B. Wang, and Y. Chen, “Robot-assisted non-destructive testing: A review,” *Sensors*, vol. 20, no. 15, 2020.

[25] H. Ahmed, H. M. La, and N. Gucunski, “Review of non-destructive civil infrastructure evaluation for bridges: State-of-the-art robotic platforms, sensors and algorithms,” *Sensors*, vol. 20, no. 14, p. 3954, 2020.

[26] J. Zhao, “Robotic non-destructive testing of manmade structures: A review of the literature,” *arXiv preprint arXiv:2007.06080*, 2020.

[27] K. Aliev and D. Antonelli, “Proposal of a monitoring system for collaborative robots to predict outages and to assess reliability factors exploiting machine learning,” *Applied Sciences*, vol. 11, no. 4, p. 1621, 2021.

[28] C. Wang et al., "Feature sensing and robotic grasping of objects with uncertain information: A review," *Sensors*, vol. 20, no. 13, p. 3707, 2020.

[29] S. Puttero, E. Verna, G. Genta, and M. Galetto, "Collaborative robots for quality control: an overview of recent studies and emerging trends," Journal of Intelligent Manufacturing, vol. 37, pp. 1345–1381, 2026, doi: 10.1007/s10845-025-02600-w.

[30] E. Verna, S. Puttero, G. Genta, and M. Galetto, "Challenges and opportunities of collaborative robots for quality control in manufacturing: evidences from research and industry," in Proceedings of the 5th International Conference on Quality Engineering and Management (ICQEM22), Braga, Portugal, 2022, pp. 235–262.

[31] Y. Koren, X. Gu, and W. Guo, "Reconfigurable manufacturing systems: Principles, design, and future trends," *Frontiers of Mechanical Engineering*, vol. 13, pp. 121–136, 2018.

[32] H. A. ElMaraghy, "Flexible and reconfigurable manufacturing systems paradigms," *International Journal of Flexible Manufacturing Systems*, vol. 17, pp. 261–276, 2005.

[33] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, 2nd ed., Springer, 2016.

[34] V. Villani et al., "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[35] International Organization for Standardization, *ISO 10218-1: Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 1: Robots*, Geneva, Switzerland: ISO, 2011.

[36] Dobot Robotics, *CR Series Collaborative Robot User Manual*, Shenzhen, China, 2023.

[37] J. E. Shigley, C. R. Mischke, and R. G. Budynas, *Mechanical Engineering Design*, 10th ed., McGraw-Hill, 2015.

[38] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*, Springer, 2008.

[39] Arduino, *Arduino Uno Rev3 Technical Specifications*, Arduino S.r.l., Italy, 2024.

[40] I. Zeid, *Mastering CAD/CAM*, McGraw-Hill, 2005.

[41] G. G. Wang, "Definition and review of virtual prototyping," *Journal of Computing and Information Science in Engineering*, vol. 2, pp. 232–236, 2002.

[42] R. L. Norton, *Design of Machinery*, 6th ed., McGraw-Hill, 2020.

[43] I. Gibson, D. Rosen, and B. Stucker, *Additive Manufacturing Technologies*, 3rd ed., Springer, 2021.

[44] T. D. Ngo et al., “Additive manufacturing (3D printing): A review of materials, methods, applications and challenges,” *Composites Part B*, vol. 143, pp. 172–196, 2018.

[45] P. Acarnley, *Stepping Motors: A Guide to Modern Theory and Practice*, 4th ed., IET, 2002.

[46] M. Lutz, *Learning Python*, 5th ed., O’Reilly Media, 2013.

[47] I. Sommerville, *Software Engineering*, 10th ed., Pearson, 2015.

[48] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, Wiley, 2006.

[49] D. C. Montgomery, *Introduction to Statistical Quality Control*, 8th ed., Wiley, 2019.

Acknowledgements

Completing this thesis has been a challenging and rewarding experience, and I would like to express my sincere appreciation to everyone who supported me throughout this journey.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Hermes Giberti, for his guidance, encouragement, and valuable feedback during every stage of this research. His knowledge, patience, and willingness to share his experience have greatly contributed to the development of this work and to my growth as a student.

I am also sincerely grateful to my co-supervisor, Dr. Davide Zhou, for his continuous support, technical guidance, and constructive suggestions throughout the experimental activities. His availability and practical advice were invaluable during both the design and validation phases of this project.

I would also like to thank the staff of the Robotics and Automation Laboratory at the University of Pavia for providing access to the laboratory facilities and equipment that made this research possible. Working in such a supportive research environment has been an important part of this experience.

Finally, my heartfelt thanks go to my family. Their unconditional love, endless encouragement, and constant belief in me have given me the strength to overcome every challenge throughout my studies. Even from afar, their support has always been my greatest source of motivation, and I am truly grateful for everything they have done for me.

This thesis marks the end of an important chapter in my academic journey, and I will always be thankful to everyone who contributed, directly or indirectly, to making it possible.

List of Tables

List of Figures