



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Development of a modular cobot for E-Waste recycling

TESI DI LAUREA MAGISTRALE IN
AUTOMATION AND CONTROL ENGINEERING - INGEGNERIA
DELL'AUTOMAZIONE

Author: **Francesco Laudadio**

Student ID: 222807

Advisor: Prof. Marcello Farina

Co-advisors: Ing. Giuseppe Cannizzaro

Academic Year: 2023-24

Abstract

This thesis presents the design and realization of a modular collaborative robot (cobot) aimed at enhancing efficiency in electronic waste (e-waste) recycling. Positioned within the Industry 5.0 framework, the project leverages advanced automation and AI to address the increasing demands for sustainable e-waste management. The cobot's modular architecture allows the integration of various tools, facilitating flexible operations such as disassembly, sorting, and handling of e-waste materials. Key innovations include a predictive maintenance algorithm, an optimal speed control system, and an object detection model, each designed to enhance system adaptability and operational efficiency. The predictive maintenance component uses machine learning to anticipate potential failures, minimizing downtime, while the speed control system dynamically adjusts motor parameters in response to load conditions, optimizing both precision and energy efficiency. Experimental results confirm the cobot's capability to accurately identify and process waste components, demonstrating its potential to improve material recovery rates and reduce human exposure to hazardous environments.

This thesis contributes to the field of collaborative robotics, providing a versatile solution that can adapt to diverse recycling scenarios and other industrial applications, promoting sustainability and efficient resource use.

Abstract in lingua italiana

Questa tesi illustra la progettazione e lo sviluppo di un robot collaborativo (cobot) modulare, mirato a migliorare l'efficienza nel riciclo dei rifiuti elettronici (e-waste). Inserito nel contesto dell'Industria 5.0, il progetto sfrutta l'automazione avanzata e l'intelligenza artificiale per rispondere alle crescenti esigenze di una gestione sostenibile dei rifiuti elettronici. L'architettura modulare del cobot permette l'integrazione di vari strumenti, facilitando operazioni flessibili come lo smontaggio, la selezione e la gestione dei materiali elettronici di scarto. Le principali innovazioni comprendono un algoritmo di manutenzione predittiva, un sistema di controllo ottimale della velocità e un modello di rilevamento degli oggetti, ciascuno progettato per migliorare l'adattabilità del sistema e l'efficienza operativa. La manutenzione predittiva utilizza il machine learning per anticipare possibili guasti, minimizzando i tempi di inattività, mentre il sistema di controllo della velocità regola dinamicamente i parametri del motore in risposta alle condizioni di carico, ottimizzando precisione ed efficienza energetica. I risultati sperimentali confermano la capacità del cobot di identificare e processare accuratamente i componenti di scarto, dimostrando il suo potenziale nell'aumentare i tassi di recupero dei materiali e nel ridurre l'esposizione umana ad ambienti pericolosi.

Questa tesi contribuisce al campo della robotica collaborativa, fornendo una soluzione versatile in grado di adattarsi a diversi scenari di riciclo e altre applicazioni industriali, promuovendo sostenibilità ed efficienza nell'uso delle risorse.

Contents

Abstract	i
Abstract in lingua italiana	iii
Contents	v
1 Introduction	1
1.1 Motivations: Industry 5.0	1
1.1.1 <i>Industry 5.0 creative applications and technologies</i>	2
1.1.2 <i>Importance and diffusion of machinery related to industry 5.0</i>	3
1.2 State of the art	5
1.2.1 <i>Challenges for cobots</i>	5
1.2.2 <i>E-waste management</i>	7
1.2.3 <i>Cobots in e-waste management</i>	7
1.3 Objectives and contributions of the thesis	8
1.4 Structure of the thesis	10
2 System architecture: hardware and software	11
2.1 Hardware components	11
2.1.1 <i>Beckhoff PLC</i>	13
2.1.2 <i>NEMA 17 stepper motor</i>	13
2.1.3 <i>TB6560 Driver</i>	15
2.1.4 <i>Raspberry Pi board</i>	16
2.1.5 <i>ADC Converter</i>	17
2.1.6 <i>Current and voltage sensors</i>	17
2.1.7 <i>Temperature and humidity sensor</i>	18
2.1.8 <i>Load cell</i>	18
2.1.9 <i>Acceleration sensor</i>	19

2.1.10	<i>Motion sensor</i>	20
2.1.11	<i>Electromagnet</i>	20
2.1.12	<i>ESP32-CAM</i>	21
2.1.13	<i>Router Wi-Fi</i>	21
2.2	Software architecture	24
3	Developed algorithms	33
3.1	Object detection algorithm	33
3.1.1	Inputs from the HMI and image acquisition	33
3.1.2	Supporting functions for detection and measurement	34
3.1.3	Object detection and filtering	35
3.1.4	Coordinate conversion and motor step calculation	40
3.1.5	Continuous detection loop	40
3.2	Predictive maintenance algorithm	40
3.2.1	Simulated dataset	41
3.2.2	Exploratory Data Analysis	42
3.2.3	Model training	44
3.2.4	Real time inference for maintenance prediction	45
3.2.5	Periodic model retraining	46
3.3	Speed control algorithm	46
3.3.1	Motor model	47
3.3.2	Speed control law	51
3.3.3	Implementation of the speed controller	52
3.4	Data acquisition and analysis infrastructure	53
3.4.1	System architecture	53
4	Experimental tests	57
4.1	Object detection algorithm validation	57
4.1.1	Evaluation metrics: precision and recall	59
4.1.2	Model considerations and improvement suggestions	61
4.1.3	Execution times	62
4.1.4	Conclusion	64
4.2	Predictive maintenance algorithm validation	64
4.2.1	Results of the experimental tests	65
4.2.2	Conclusions	66
4.3	Optimal speed control algorithm validation	66
5	Future developments and conclusion	71

Bibliography	75
List of Figures	79
List of Tables	81
Ringraziamenti	83

1 | Introduction

1.1. Motivations: Industry 5.0

This thesis project is strongly connected with Industry 5.0, shortly introduced in this section.

The industry world is highly competitive and operates in an increasingly complex globalised economy. The only way to tackle the ever-changing world's challenges is through adaptation and innovation, leading to new technology-driven industrial revolutions. Industry 5.0 should not be regarded as an alternative or a chronological continuation of Industry 4.0. Rather, it complements and extends the hallmark features of the latter.

As reported in [1], Industry 5.0 was discussed for the first time on July 2020 amongst participants from research and technology organizations in two virtual workshops organised by Directorate "Prosperity" of DG Research and Innovation. There was a consensus on the need to make an effort to refocus industry from a paradigm that prioritizes shareholder values and return on investment to a paradigm that prioritizes the respect for the individual rights and sustainability for the environment and ecology.

Six main priority technologies have been identified, each of which can exploit its potential in combination with the others:

1. Individualised Human-Machine-Interaction
2. Bio-inspired technologies and smart materials
3. Digital twin and simulation
4. Data transmission, storage, and analysis technologies
5. Artificial intelligence
6. Technologies for energy efficiency, renewable, storage and autonomy.

Figure 1.1 shows the distribution of the Industry 5.0 market size across various technologies, along with its projected growth until 2032.

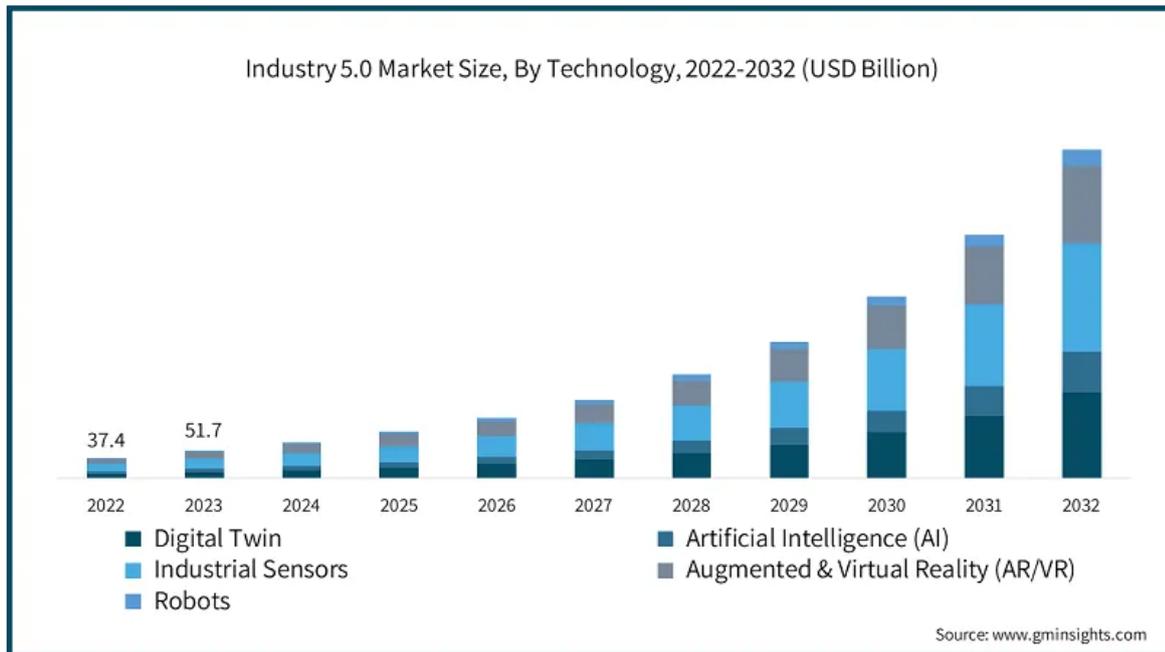


Figure 1.1: Industry 5.0 Market Size [2]

1.1.1. *Industry 5.0 creative applications and technologies*

As reported in [3], the fifth industrial revolution brings with it creative applications, some example of which are presented hereafter. An important example is the smart hospital with real-time capability. Thanks to smart healthcare technologies, doctors can be supported on focusing on patients and provide efficient data and students can more efficiently focus on their medical training. Machine learning, for instance, will enable automatic diseases' diagnosis, detection, and prediction.

Industry 5.0 is revolutionizing also the manufacturing systems. Intelligent robots and systems are penetrating supply chains to an unparalleled stage. Smart manufacturing allow designers to store manufacturing items in a cloud with robust access control and to use resources across various places.

Another Industry 5.0 paradigm is the Internet of Things, IoT. It is considered among other things, as a way to provide efficient solutions to data and information security problems. Big data analytics is a complex procedure aiming to examining a huge quantity of data to uncover hidden patterns, trends of the market, or to implement predictive maintenance. Most businesses use nowadays big data analytics to make strategic decisions.

1.1.2. *Importance and diffusion of machinery related to industry* 5.0

Machines have a fundamental role in the revolution, in particular in view of their possible strong relationship with humans. The latter, as stated in [4], is known in the literature as the 5C model: Coexistence, Cooperation, Collaboration, Compassion, Co-evolution.

Human-machine Coexistence - HMI

Humans and machines work in the same environment and share the same space, with the goal of balancing automation and productivity with flexibility and capability. In an HMI environment, for example, the operator can interact with semi-automated machines to realize the goal of human-enhanced automation.

The presence of machines and humans in the same space, however, creates the risk of collisions, and so there is the need to adopt appropriate safety features, such as emergency stops.

Human-machine Cooperation

Considering a group of agents (being they human or machines) in a collaborative situation, two situations may happen:

1. an agent (or a subgroup of agents) pursues goals that may conflict with those of others;
2. each agent seeks to incorporate its tasks and those of others in a common one.

In the latter case, the human operator and the machines are placed at the same decision-making level: agents work together to achieve a common goal through an interactive dialogue.

Human-machine Collaboration

In everyday life, there is an ever increasing level of interaction between robots and humans. Human-robot collaboration (HRC) focuses on the interaction between a human and a robot with a common goal. It has great potentials in case of risky decisions: machines could gather information and communicate key data to the human, i.e., the decision maker. In this respect, human-robot relationship is changing from "master-slave" to "master-collaborator" with ever increasing physical contact between humans and machines. This results in a new type of robot, the cobot, i.e., a collaborative robot capable of working and collaborating with humans. In this thesis, cobots will be considered in details.

Human-machine Compassion and Co-evolution

Since Industry 5.0 focuses on human-centricity, human mental health is deemed extremely important. This raises the need of developing empathetic machines that sense human emotions and needs and provide assistance. The "health" of empathetic machines will also be taken care of by humans, considering quantitative measures related to workload and tasks execution. An intimate human-machine interaction will enable the growth of both human and machine capabilities, leading to a continuous co-evolution in the future.

Machines in Industry 5.0

Industry 5.0 and the corresponding evolution of the machinery are having a great impact in different domains. For example, Industry 5.0 is revolutionizing the manufacturing systems across the globe by replacing human workers with robots in performing repetitive tasks.

Also, Industry 5.0 machines are designed to be more flexible, adaptable, and able to interact safely and intelligently with human operators.

Examples of such machines include:

- Cobots, which, as mentioned before, have advanced sensors and safety systems that allow them to interact directly with people. They grant the human operator with benefits, such as technical precision and heavy lifting capabilities. They help humans by leveraging advanced technologies, such as IoT, according to which a network of physical objects, embedded with sensors, software and other technologies, connect and exchange data over the internet.
- Predictive maintenance systems. Many Industry 5.0 machines include sensors that allow one to collect data on their operating conditions. These data are analyzed in real time by artificial intelligence algorithms, allowing for predictive maintenance aiming to avoid failures and optimize the production.
- Machines with digital twins. Many Industry 5.0 machines are equipped with digital twins, i.e. virtual representations that allow operations to be monitored, simulated and optimized. These systems enable predictive maintenance and continuous optimization of production processes, respecting the objectives of Industry 5.0 in terms of efficiency and sustainability.
- Machines for custom productions.
- Intelligent automation systems. These machines combine traditional automation with artificial intelligence and IoT.

1.2. State of the art

The objective of this thesis is to develop a collaborative robotic system within the context of electronic waste (e-waste) management. This section will first discuss the current general challenges and advancements in the field of cobots, before moving on to an overview of e-waste and its specific management challenges. Lastly, the role of cobots in e-waste management will be explored, emphasizing their potential to enhance both efficiency and safety in recycling operations.

1.2.1. Challenges for cobots

This subsection addresses the challenges of cobots, inspired by the insights and the findings presented in [5].

There are several type of cobots, classified as fixed, mobile, and hybrid solutions.

Robots are available with sensors as well as suitable user interfaces that allow the machine to recognize and react to unstructured environments. Cobots permit technologically advanced enterprises to obtain flexibility to accomplish productivity enhancements without compromising the low-volume production. Moreover, thanks to the sensors, data are gathered from machines at each phase of production and then aggregated and processed in order to optimize the entire production process.

Four main challenges have been identified for technologically advanced enterprises related to the introduction of collaborative robots: safety, performance, strategy, smart technology.

Safety

Safety is crucial, since human operators work closely with robots and risk serious injury. The main challenge here is to exploit as much as possible the robot's potentialities and keep the human safe at the same time.

A comprehensive and multidisciplinary approach is required for addressing safety issues. A thorough hazard analysis and risk evaluation must be performed according to specifically designed procedures.

The American National Standards Institute (ANSI) has published a draft safety standard for Intelligent Assist Devices (IAD), i.e., the ones defined as 'single-or multiple-axis devices that employ a hybrid, programmable, computer-human control system to provide human strength amplification, guiding surfaces or both'.

The standard includes the follow guidelines.

1. Risk assessments, rather than fixed rules, must be used to identify and mitigate risks in proportion to their severity and probability.
2. Safety-critical software is required, capable of resetting the system in a safe state, in case of component failures. This may be obtained using microprocessors redundancy and diversity.
3. The operator must be able to outrun, overpower, or turn off the IADs. For this reason, safe speed limits must be imposed.
4. IADs may operates in few different modes, well understood by human operators and easily commanded by the operator.

Performance

Other challenges are related to the performance of the robot. This is a concept strictly related with the human "trust-in-automation", concerning how comfortable the human is in sharing the robot workspace and collaborate with it. The trust between the human and the robotic automation process is pivotal, and it affects the productivity of the robot. Trust can be achieved by managing the task-specific expectations and actions of the robot, based on how a human operator expects the robot to behave.

The main challenge here is to combine the speed with the quality of the work.

Strategy

There are challenges also related to strategic issues, e.g., cost-effectiveness with small production volumes. It is important also to choose a proper cobot solution for the requested task and a reliable supplier. The biggest challenge here is to make a profit despite small volumes production. Many companies lack the expertise for successful cobot implementation and tend to choose too complex automation tasks and too high levels of interaction.

Smart technology

Systems must be integrated with intelligent solutions, which represent another challenge. Flexible solutions are called for, with sensors and vision systems for quality control, handling unexpected behaviour, and offering information to the human operator. The challenge is planning operations, for which a digital twin can provide a substantial help. Finally, the cobot should be integrated into a machine learning environment, where AI solutions may allow one to achieve higher performance.

1.2.2. *E-waste management*

One of the primary goals in today's industrial landscape is achieving sustainability in production systems and eliminating waste. According to [6], circular manufacturing represents an innovative approach that fosters the development of sustainable businesses. In this context, Industry 5.0 plays a pivotal role, driving advancements towards more resource-efficient and environmentally friendly practices. Studies show that companies adopting Industry 5.0 tools enhance the efficiency of product reuse and provide better evaluations of post-use handling options, potentially extending product lifespan and improving recycling processes. For example, machine learning can be used in controlling production, maintenance, recycling and remanufacturing processes.

As stated in [7], digitalising all types of activities promotes circular e-waste management, including prevention, collection, and treatment. Unfortunately, the amount of e-waste continues to grow, becoming an environmental problem. In 2022, 62 billion kg of e-waste was generated globally, equivalent to approximately 7.8 kg per person. Of this amount, only 22.3% was collected and sustainably recycled in a documented manner. The production of e-waste is growing faster than the capacity to adequately collect and recycle it.

Key challenges in the regulation of e-waste include the growing variety of electronic and electrical equipment (EEE) and the complexity of defining and tracking producers responsible for e-waste management. Current e-waste management primarily focuses on collection and recycling targets, rather than waste prevention. It is crucial to shift priorities towards waste prevention by promoting the reuse and repair of electronic devices, although these goals are often absent in current policies.

Given these challenges, it is essential to implement innovative solutions that address the growing complexity of e-waste and reduce its generation. In this context, predictive maintenance emerges as a key strategy to extend the lifespan of electronic devices and prevent premature disposal. By leveraging advanced tools such as artificial intelligence and machine learning, predictive maintenance can monitor the condition of devices in real-time, identify potential failures before they occur, and promote repair or upgrades instead of replacement. This approach not only reduces the volume of e-waste but also fosters more efficient resource use, aligning with circular economy goals.

1.2.3. *Cobots in e-waste management*

According to [8], e-waste disassembly, which is the operation of extracting valuable components for recycling purposes, has received ever increasing attention as it can serve both

the economy and the environment. Traditionally, e-waste disassembly is labor-intensive with significant occupational hazards. To reduce labor costs and enhance working efficiency, cobots might be a viable option. The feasibility of deploying cobots in high-risk or low added value e-waste disassembly operations is of tremendous significance to be investigated.

A large body of the literature focuses on human-robot collaboration during the disassembly process. However, enabling safe and seamless collaboration between human workers and robotic assistants in a shared workspace is a complex task, requiring, for instance, real-time object detection. From the study conducted in [8], it is confirmed that work with the cobot reduces the overall human workload and physical stress, improving ergonomics. Nevertheless, the task took longer to complete when using the cobot, highlighting a trade-off between increased duration and improvements in human workload and ergonomics.

In [9], the growing challenge of managing e-waste is addressed. The integration of cobots is explored into the disassembly processes of electronic devices. The proposed solution incorporates human-robot collaboration, where cobots are assigned repetitive and hazardous tasks, allowing human operators to focus on more complex, decision-driven ones. The use of cobots has demonstrated economic viability, resulting in a more efficient material recovery process and higher-quality separation of valuable and recyclable materials, particularly plastics. The results demonstrate substantial improvements in both the economic performance and safety conditions of the recycling process. By assigning repetitive and hazardous tasks to cobots, the disassembly line achieved a productivity increase from 30 units per hour (in manual processes) to 48 units per hour. Additionally, the cost of treatment per unit was reduced to € 2.57, indicating improved operational efficiency. This automated collaboration also led to a higher recovery rate of valuable materials, particularly plastics, with a recovery percentage of 90.24% in weight. The refined process not only increased the economic value of the recycled materials but also minimized the exposure of human workers to dangerous substances, enhancing workplace safety. Cobots demonstrate significant operational versatility, seamlessly shifting between roles that involve repetitive, labor-intensive tasks, and enabling human operators to focus on decision-making and complex activities. This adaptability allows them to dynamically adjust to the varying demands of the disassembly process, contributing to greater overall process efficiency and productivity.

1.3. Objectives and contributions of the thesis

The primary objective of this thesis is to develop a highly modular cobot, aimed at improving the efficiency and adaptability of processes within the e-waste recycling industry.

The project focuses on creating a flexible system that can accommodate a variety of tools and components to perform a wide range of specialized tasks, depending on the specific needs of the application. One of the main challenges in the design and development of cobots lies in their ability to adapt to different operational scenarios. This work addresses this challenge by prioritizing flexibility in the hardware design of the proposed cobot. The design allows for the integration of various interchangeable tools to perform different functions. For instance, the system can operate with an electromagnet for the manipulation of ferrous objects, but it can just as easily be equipped with an electric screwdriver for disassembly operations, or a soft gripper for handling delicate components. This versatility ensures that the system can potentially be applied to a broad range of scenarios within recycling as well as in other industries.

A significant objective of this work is to demonstrate how a modular robotic system can be built with limited hardware resources while remaining flexible and scalable. The design allows for quick and efficient reconfiguration, overcoming the typical constraints of fixed robotic systems. Furthermore, the integration of advanced control algorithms and Artificial Intelligence (AI) aims to optimize operations in real-time, adjusting movement patterns, analyzing sensor data, and implementing predictive maintenance to ensure the system's functionality and efficiency over extended periods. In addition, all components of the system are interconnected within a local network, creating a unified communication framework that allows data exchange between all the system's elements. This networked approach not only enhances the system's responsiveness and adaptability but also facilitates centralized control and monitoring, enabling real-time optimization and efficient management of resources.

Building upon this, the system is designed to incorporate optimal speed control, allowing it to dynamically adjust speed based on real-time sensor feedback on, e.g., weight, position, temperature, and vibration. By applying advanced control strategies, the cobot seeks to balance operational speed and precision, reducing errors and improving efficiency. This control mechanism is intended not only to ensure smoother operations but also to minimize energy consumption and reduce wear on mechanical components, thereby extending the cobot's lifespan and enhancing its overall performance. Furthermore, the system will feature a dedicated Human-Machine Interface (HMI), providing a user-friendly platform for supervising and managing operations. This addition will support the effective interaction between the operator and the cobot, streamlining workflow and enhancing the overall productivity.

The contribution of this thesis lies in the development of a versatile and adaptable platform that can be adapted for various industrial applications beyond e-waste recycling. By creating a cobot system that can be equipped with different tools, this work demonstrates

the potential for a single robotic system to handle diverse tasks. This modular approach, not only enhances the versatility of the cobot, but also shows how scalable solutions can be developed even with hardware constraints. Furthermore, the implementation of a local network for centralized communication and real-time optimization lays the foundation for future improvements in robotic efficiency and reliability. This thesis thus contributes to the field of collaborative robotics by offering a design and methodology that promote adaptability, sustainability, and operational efficiency.

1.4. Structure of the thesis

Following the Introduction, which establishes the background, motivation, and objectives of the work, the remainder of this thesis is structured into four main chapters:

1. System architecture. This chapter provides an in-depth description of the overall design and configuration of the cobot system, covering both the hardware and software components. It provides details on the integration of sensors, control systems, and the software environment used for the development and testing of the cobot.
2. Developed algorithms. In this chapter, the algorithms designed and implemented for the cobot are presented. Specifically, it covers the recognition algorithm for nuts and washers, the optimal speed control mechanism, and the predictive maintenance approach. Each algorithm is explained in detail, with a focus on its implementation and role within the system.
3. Experimental tests. This chapter discusses the experimental trials carried out to validate the cobot's performance. It presents the testing environment, methodologies, and obtained results, providing an analysis of the effectiveness and limitations of the system.
4. Conclusions. The final chapter summarizes the findings of the thesis, reflecting on the achieved objectives and contributions. It also suggests potential future developments and improvements for the cobot system, considering its scalability and adaptability to other industrial applications.

2 | System architecture: hardware and software

This chapter provides a detailed description of the system architecture. The analysis focuses on both hardware and software aspects, illustrating how the various components integrate to achieve the project's goals.

The cobot described in this thesis is specifically designed to recover discarded nuts and washers, sorting them by size. It is worth noting, however, that this is only one of its many potential applications. The modular nature of the system allows the cobot to be adapted for a wide range of tasks in various industrial contexts.

We describe the hardware components of the system, forming the backbone of the automated process. To begin with, Section 2.1 will detail the physical elements, including sensors, actuators, and control units, and describe their roles in ensuring the system's functionality and performance. Then, in Section 2.2 will be presented the software architecture and a brief description of the operating principles.

2.1. Hardware components

The hardware of the system includes a robust combination of motors, sensors, control units, and a custom network setup. The key hardware components include a 3D printer frame, NEMA 17 stepper motors, TB6560 drivers, the Beckhoff PLC, a Raspberry Pi board, an electromagnet, an ESP32-Cam, and various sensors for current, voltage, acceleration, weight, temperature, and humidity measurements, along with a pyroelectric infrared (PIR) motion sensor.

The Raspberry Pi board uses an ADS1115 Analog-to-Digital (ADC) converter to handle the analog signals from the current and voltage sensors, converting them with high precision from analog to digital. Additionally, the system integrates a router to establish a local area network (LAN), connecting all devices for seamless communication. Another crucial component of the system is the server, which plays a critical role in the optimization process. While the Raspberry Pi board is responsible for collecting sensor data in real

time, the server performs the computationally intensive task of motor speed optimization. This division ensures that the system remains efficient and responsive during operation. The following subsections will explore the specific components and its specific role within the system architecture.

Figure 2.1 presents the complete architecture of the system, detailing the interconnected components and hardware configuration essential for the collaborative robotic setup. This figure illustrates the seamless integration of various sensors, control units, and computational devices, each playing a critical role in ensuring efficient and safe human-robot interaction. The accompanying image description provides additional insights into the functionality and placement of individual components, highlighting their specific roles within the system to support optimal performance and adaptability across various tasks.

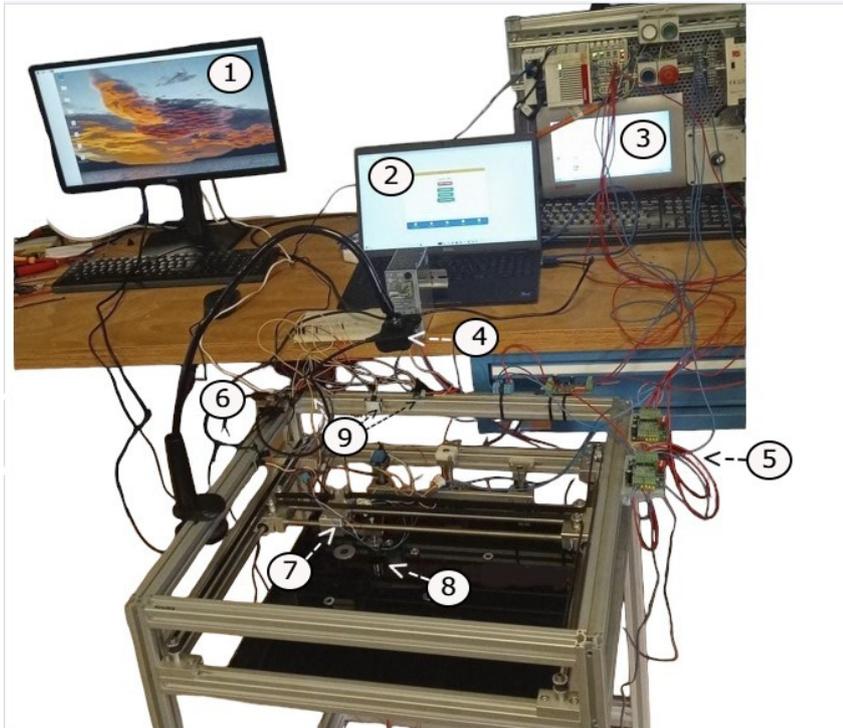


Figure 2.1: 1. **Raspberry Pi display**, showing real-time data for monitoring and control; 2. **Server**, for data storage and processing; 3. **Beckhoff PLC with HMI**[10], which serves as main controller connected to HMI for issuing commands; 4. **ESP32 Cam**, for capturing images in object detection (washers and nuts); 5. **Driver TB6560**, to control motor speed and direction; 6. **Raspberry Pi Board**, to collect and transmit sensor data to the server; 7. **Load cell**, providing real-time weight data for speed control; 8. **Electromagnet**, to lift and move ferrous objects; 9. **Various sensors**, including temperature, humidity, vibration, human proximity, current, and voltage.

2.1.1. *Beckhoff PLC*

Figure 2.2 illustrates a Beckhoff central control unit with various communication and power modules. On the left, Ethernet ports are visible, used for communication within the system. On the right, I/O modules are shown, which handle power management for the automation system.



Figure 2.2: Beckhoff PLC[10]

The Beckhoff PLC system is integral to the project's control architecture, providing reliable automation. The EL1008 and EL2008 modules offer standard digital input and output functionality, facilitating the integration of sensors and actuators for essential control operations.

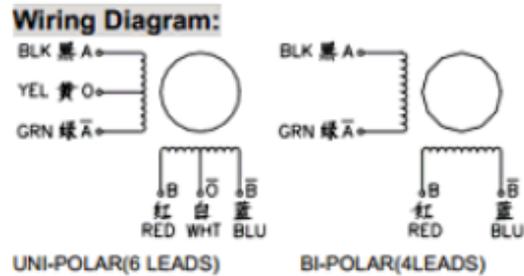
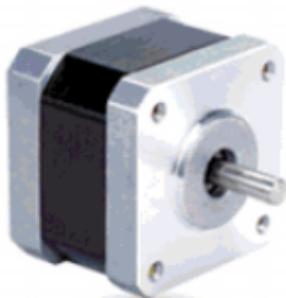
2.1.2. *NEMA 17 stepper motor*

The NEMA 17 HS4401 is a 4-wire bipolar stepper motor, widely used in precision applications such as 3D printers and CNC machines. It has a holding torque of 40 N·cm and a step angle of 1.8 degrees, providing 200 steps per revolution. With a standard frame size of 42x42 mm², it offers a reliable balance of torque and compactness, making it suitable for tasks requiring accurate control of position and speed. The datasheet of the motor is presented in Figure 2.3.

MotionKing (China) Motor Industry Co., Ltd.

17HS4401

2 Phase Hybrid Stepper Motor 17HS series-Size 42mm(1.8 degree)

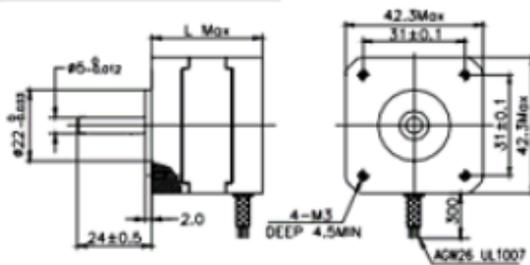


Electrical Specifications:

Series Model	Step Angle (deg)	Motor Length (mm)	Rated Current (A)	Phase Resistance (ohm)	Phase Inductance (mH)	Holding Torque (N.cm Min)	Detent Torque (N.cm Max)	Rotor Inertia (g.cm ²)	Lead Wire (No.)	Motor Weight (g)
17HS2408	1.8	28	0.6	8	10	12	1.6	34	4	150
17HS3401	1.8	34	1.3	2.4	2.8	28	1.6	34	4	220
17HS3410	1.8	34	1.7	1.2	1.8	28	1.6	34	4	220
17HS3430	1.8	34	0.4	30	35	28	1.6	34	4	220
17HS3630	1.8	34	0.4	30	18	21	1.6	34	6	220
17HS3616	1.8	34	0.16	75	40	14	1.6	34	6	220
17HS4401	1.8	40	1.7	1.5	2.8	40	2.2	54	4	280
17HS4402	1.8	40	1.3	2.5	5.0	40	2.2	54	4	280
17HS4602	1.8	40	1.2	3.2	2.8	28	2.2	54	6	280
17HS4630	1.8	40	0.4	30	28	28	2.2	54	6	280
17HS8401	1.8	48	1.7	1.8	3.2	52	2.6	68	4	350
17HS8402	1.8	48	1.3	3.2	5.5	52	2.6	68	4	350
17HS8403	1.8	48	2.3	1.2	1.6	46	2.6	68	4	350
17HS8630	1.8	48	0.4	30	38	34	2.6	68	6	350

*Note: We can manufacture products according to customer's requirements.

Dimensions: unit=mm



Motor Length:

Model	Length
17HS2XXX	28 mm
17HS3XXX	34 mm
16HS4XXX	40 mm
16HS8XXX	48 mm

Figure 2.3: Stepper motor NEMA17 data sheet[11]

2.1.3. *TB6560 Driver*

The motors described in Section 2.1.2 are controlled by a TB6560 stepper motor driver, a versatile and reliable device designed to control 2-phase and 4-phase stepper motors with high precision. This driver employs PWM chopper technology to ensure smooth operation, supporting voltages up to 32V and output currents up to 3A. Figure 2.4 provides an illustration of the device.

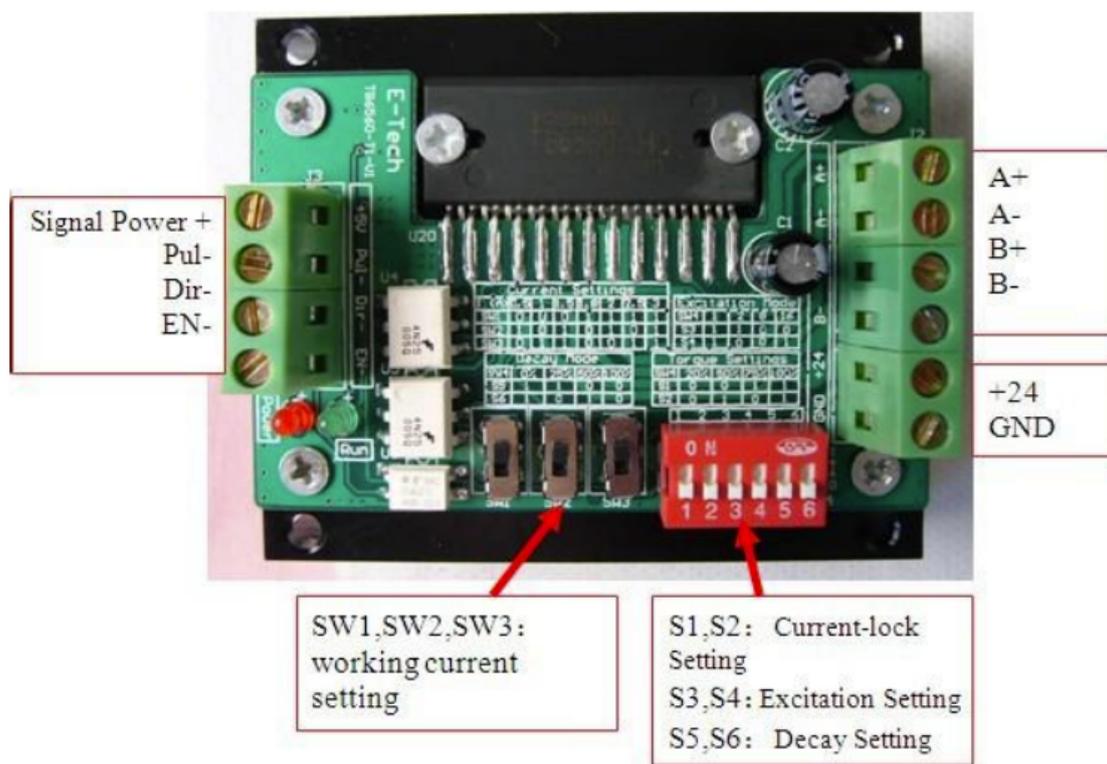


Figure 2.4: TB650 Driver [12]

Additionally, the TB6560 driver is equipped with a set of DIP switches that allow for fine-tuning of various operational parameters. These switches enable users to adjust:

- Current settings, which can be adjusted to align with the motor's specific current requirements, offering a range from 0.5A to 3A.
- Current-lock settings, which allow the motor to maintain torque when idle by controlling the percentage of current supplied during inactivity.
- Excitation mode settings, which provide options for different microstepping levels, such as 2, 8, or 16 microsteps, which enhance precision.

- Decay mode settings, which control the rate of current decay in the motor's windings, helping to reduce noise and improve stability at higher speeds.

These configurable options make the TB6560 driver a flexible choice for a wide range of applications, such as CNC machines and automated systems requiring precise motor control.

2.1.4. *Raspberry Pi board*

The Raspberry Pi 2 Model B, shown in Figure 2.5, is equipped with a 900MHz quad-core ARM Cortex-A7 CPU and 1GB of RAM, offering significantly improved performance over previous models, with up to six times the processing power. It includes 40 GPIO pins, 4 USB 2.0 ports, and supports HDMI output, Ethernet connectivity, and a micro SD card for booting the operating system. The board dimensions are 85x56x17 mm³, and it operates with a 5V, 2A micro USB power source.



Figure 2.5: Raspberry Pi 2 Model B [13]

The microprocessor is connected to all the sensors used in the system, in order to collect and process the data in real-time. This setup allows for efficient data acquisition from multiple sources, such as current, voltage, temperature, humidity, and vibration sensors, making the Raspberry Pi board a central unit for monitoring and controlling various parameters. Its processing power is sufficient to handle the real-time processing and analysis tasks, enabling smooth operation and integration with the broader automation system.

2.1.5. *ADC Converter*

The ADS1115 is a 16-bit analog-to-digital converter (ADC), represented in Figure 2.6, that communicates via the I2C interface, offering high precision for converting analog signals. It supports four single-ended input channels or two differential inputs and operates with a programmable gain amplifier, making it ideal for applications requiring accurate analog signal measurements. In this setup, it is used with a Raspberry Pi board to handle the conversion of analog signals from sensors such as current and voltage.

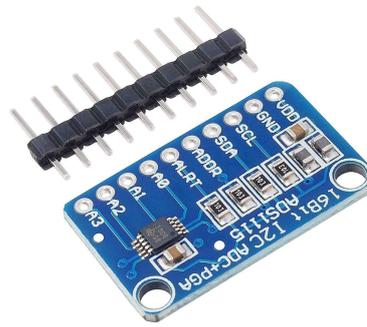


Figure 2.6: ADS1115 [14]

2.1.6. *Current and voltage sensors*

The sensors shown in Figure 2.7 are used to measure current and voltage in the motor phases. The collected data will be processed in real-time for both speed control and performance monitoring of the motors. Additionally, these data will be used for predictive maintenance, allowing for the early detection of potential issues, minimizing downtime, and optimizing overall system efficiency.



Figure 2.7: Current and voltage sensors [15]

2.1.7. *Temperature and humidity sensor*

For temperature control, the DHT11 sensor shown in Figure 2.8 is employed to measure environmental conditions with precision. It operates within a temperature range of 0°C to 50°C and a humidity range of 20% to 90%, with an accuracy of $\pm 2^\circ\text{C}$ for temperature and $\pm 5\%$ for humidity. The sensor outputs data in a digital format, making it easy to interface with microcontrollers. The data gathered are integral to the system's predictive maintenance strategy, enabling the early detection of potential anomalies.

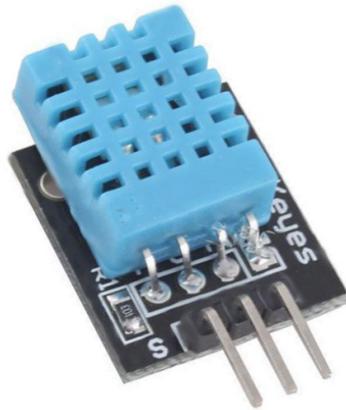


Figure 2.8: DHT, temperature and humidity sensor [16]

2.1.8. *Load cell*

In this project, a load cell is used to measure the real-time weight of objects lifted by the electromagnet. The load cell is connected to the Raspberry Pi board, enabling data collection and monitoring of the weight of the objects in real time. The sensor, presented in Figure 2.9, features the following characteristics:

- Capacity: 20 kg
- Operating Voltage: 3.3V to 5V (compatible with the Raspberry Pi board)
- Sensitivity: high precision, providing accurate weight readings with minimal error.
- Output: analog signal, processed using the HX711 amplifier board for digital conversion before being sent to the Raspberry Pi board.

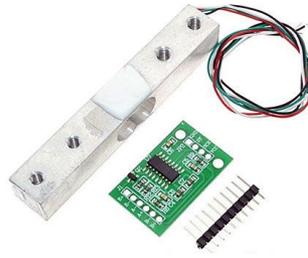


Figure 2.9: Load cell [16]

2.1.9. Acceleration sensor

MPU-6050 is a compact, versatile sensor that combines a 3-axis accelerometer and a 3-axis gyroscope, allowing for precise motion tracking and vibration measurement. Key features include:

- 3-axis accelerometer, which measures acceleration along the X, Y, and Z axes.
- 3-axis gyroscope, which tracks angular velocity for rotational movement along three axes.
- Communication interface, which uses I2C, making it easy to interface with the Raspberry board and other microcontrollers.
- Small form factor, ideal for compact projects.

The MPU-6050, visible in Figure 2.10, is used in this project to measure vibrations, providing critical data to monitor and optimize the system performance. Its precise sensing capabilities make it an excellent choice for real-time vibration analysis and motion tracking.

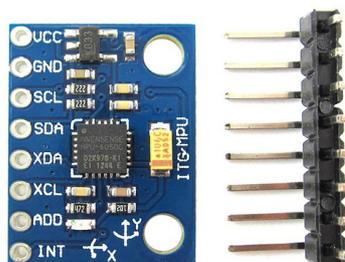


Figure 2.10: MPU-6050 [17]

2.1.10. *Motion sensor*

The pyroelectric infrared (PIR) motion sensor, displayed in Figure 2.11, detects human presence by sensing changes in infrared radiation. It serves as a safety system, ensuring that, when the motors are active and human movement is detected nearby, the cobot immediately halts. This feature helps to prevent accidents by stopping operations whenever a person is in close proximity during motor activity, enhancing the overall safety of the working environment.



Figure 2.11: PIR sensor [18]

2.1.11. *Electromagnet*

The electromagnet, visible in Figure 2.12, operates at 12V and has a lifting force of 5 kg. It generates a magnetic field when current is applied, allowing it to attract and hold metallic objects. It is controlled by the Beckhoff PLC, which manages the timing and activation based on the system's operational needs. This enables precise handling of metallic items, such as nuts and washers, as part of the cobot's pick-and-place operations.



Figure 2.12: Electromagnet HS-P25X20 [19]

2.1.12. *ESP32-CAM*

The ESP32-CAM (depicted in Figure 2.13) plays a crucial role in the system, as it is responsible for capturing images of nuts and washers. Also, the CAM has built-in Wi-Fi capabilities allowing it to connect to the local area network (LAN) and facilitating wireless communication. Additionally, it has the ability to create a web server, enabling remote access to its data via a browser. These features are highly beneficial for real-time monitoring and data collection. Further sections will detail how the ESP32-CAM is used to its full potential in this project.



Figure 2.13: ESP32-CAM [20]

2.1.13. *Router Wi-Fi*

The Teltonika RUT955 router, visible in Figure 2.14, is used to establish a local area network (LAN) for the system. The ESP32-CAM is connected wirelessly via Wi-Fi, while the Raspberry Pi board and PLC are connected through Ethernet cables. This LAN setup ensures efficient communication between all devices, allowing for real-time data transfer and seamless operation across the network. This configuration provides stable and reliable connectivity, which is essential for the effective coordination of the system components.



Figure 2.14: MPU-6050 [21]

In Figure 2.15 we depict the overall system architecture, integrating various essential components, including the PLC, stepper motor drivers, sensors, Raspberry Pi board, and electromagnet, all interconnected via a local area network (LAN). A shared area, physically hosted on the server, is accessible via LAN by all the devices involved, including the PLC and the Raspberry Pi board. The latter collects data from various analog and digital sensors, storing them in the shared area, which the server uses for tasks such as motor speed optimization and data analysis. The PLC's primary function is to control the cobot, executing the instructions received through the HMI. These instructions are first processed by the server and then stored in the shared area in a format compatible with the PLC, which executes them to operate the cobot. Concurrently, the PLC records data related to the operations performed, such as tracking nuts or washers, within the shared area, making them available to the server for monitoring and logging production activities. The system includes two HMI interfaces: a simplified version directly connected to the PLC for local management and a complete web-based version, accessible via LAN, which allows for remote monitoring and control of the system. This architecture ensures efficient communication, accurate data tracking, and a rapid system response.

SYSTEM ARCHITECTURE

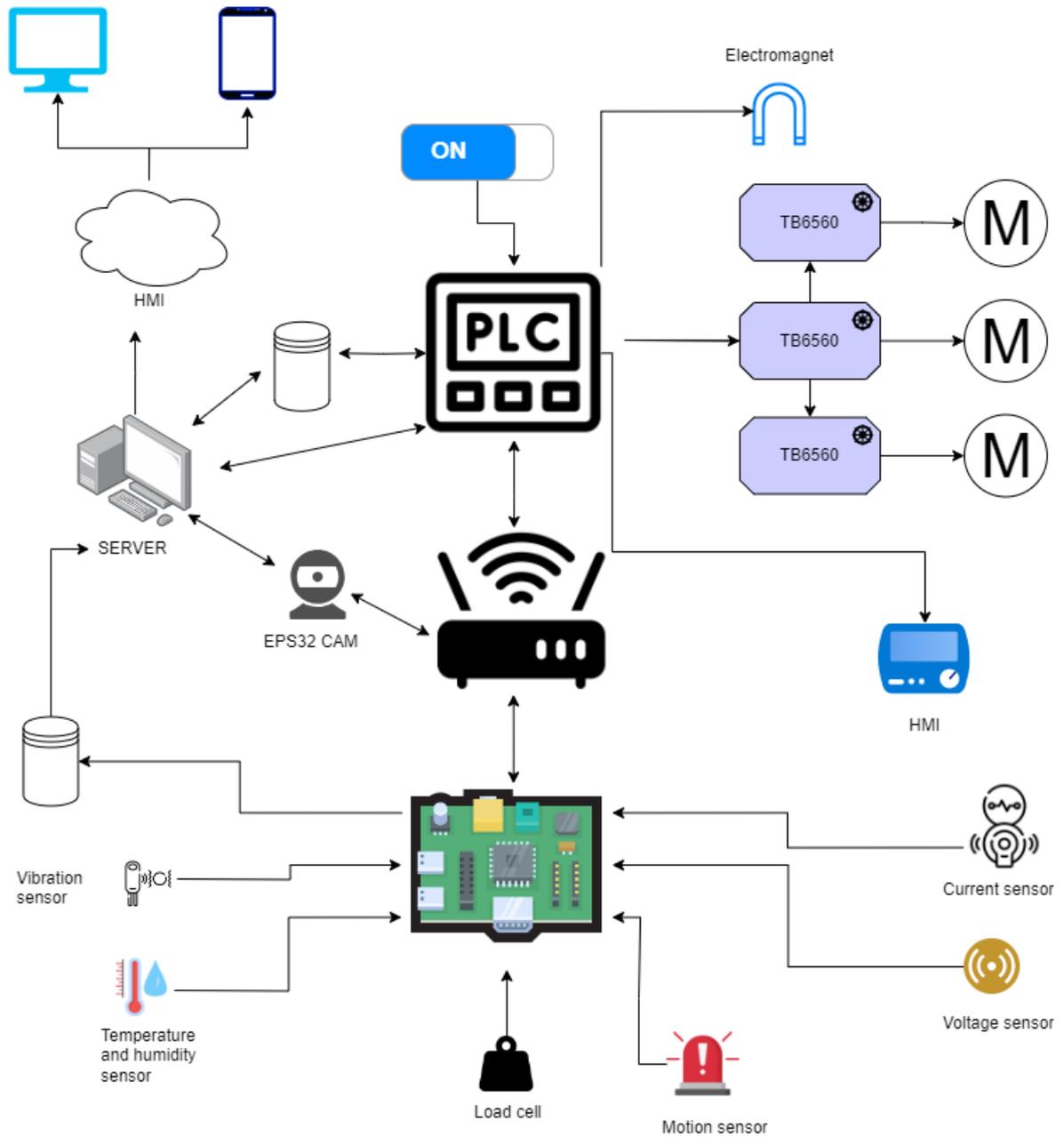


Figure 2.15: System architecture

2.2. Software architecture

The software architecture plays a crucial role in coordinating communication between the various hardware components and optimizing the overall performance. The system makes use of a computer vision model for object recognition, which runs on the server and communicates with an ESP32-CAM by sending requests to the web server hosted on it. The model receives the image from the camera as input, processes the data to recognize objects such as nuts and washers, and generates an output file. This output file is generated based, not only on the results of object recognition, but also on the instructions received via the HMI. The file is then used by the PLC as the starting point to control the cobot.

The server plays a fundamental role in the dynamic optimization of the system. It receives data from the Raspberry Pi board through the shared area and, using specially developed Python scripts, calculates the optimal speed for the stepper motors in real-time based on the current load conditions. This calculation is crucial to ensure the correct functioning of the system under variable conditions, optimizing performance and reducing energy consumption. Once calculated, the optimal speed is immediately saved in a CSV file accessible by the PLC through the shared network. The Beckhoff PLC reads this file to dynamically and efficiently adjust the speed of the NEMA 17 motors, allowing the system to adapt in real-time to load variations. This continuous optimization process is essential for ensuring optimal cobot performance, guaranteeing reliable and precise operation in all stages.

In real-time, the PLC stores a file in the shared area containing detailed information on production processes, including timestamp, object type, weight, compliance, speed, and batch number. These data are processed by an artificial intelligence model that, not only generates process statistics, but also provides operational insights to improve material recovery. The model's analysis extends beyond a single batch, considering correlated data to provide a comprehensive view of the production cycle. Furthermore, the model can analyze similar files from other production lines or locations, allowing a comparative evaluation of performance and the identification of common issues. The integration of the artificial intelligence model, running on the server, enables real-time analysis of sensor data, allowing the system to predict failures and reduce downtime. Initially trained with simulated data, the model is periodically updated using real historical data, thus improving the accuracy of its predictions and enhancing the system's overall reliability. Additionally, the artificial intelligence can establish alert thresholds for each monitored parameter, allowing the system to detect and signal abnormal or critical conditions in advance. This enables timely interventions to prevent potential failures or malfunctions,

further improving operational efficiency.

Finally, the system is equipped with a human-machine interface (HMI). The local HMI allows operators to manage key operations on-site, while the web-based HMI, accessible via LAN, offers full and flexible remote management. The web-based HMI is designed to be fully responsive, ensuring optimal functionality and user experience across both desktop and mobile devices. This allows operators to interact with the system seamlessly, regardless of the platform they are using. This system enables monitoring and intervention not only from fixed stations but also from mobile devices, providing a high degree of operational flexibility. To further illustrate the capabilities of the web-based HMI, four main illustrations are provided:

1. **COBOT control interface:** Figure 2.16 displays the main control panel for the cobot, allowing operators to power on or off the cobot, initiate image processing, and input camera distance in millimeters. It also includes an alert bar at the top to indicate any active alerts, ensuring immediate visibility of system status.

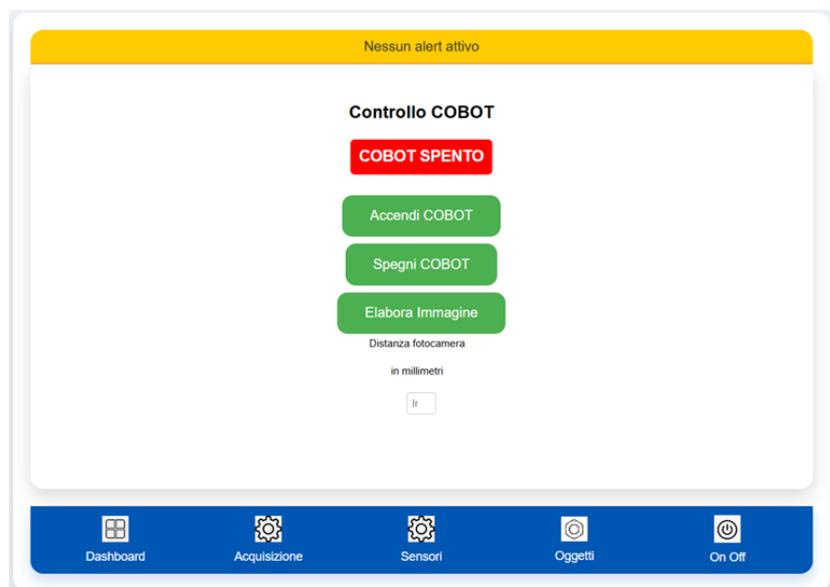


Figure 2.16: COBOT control interface

2. **Detected objects overview:** Figure 2.17 shows a table collecting the recognized objects, detailing the type (e.g., washer or nut), diameter, and information evaluating whether each item meets the compliance criteria. This organized list provides operators with a real-time overview of the objects detected by the computer vision model.

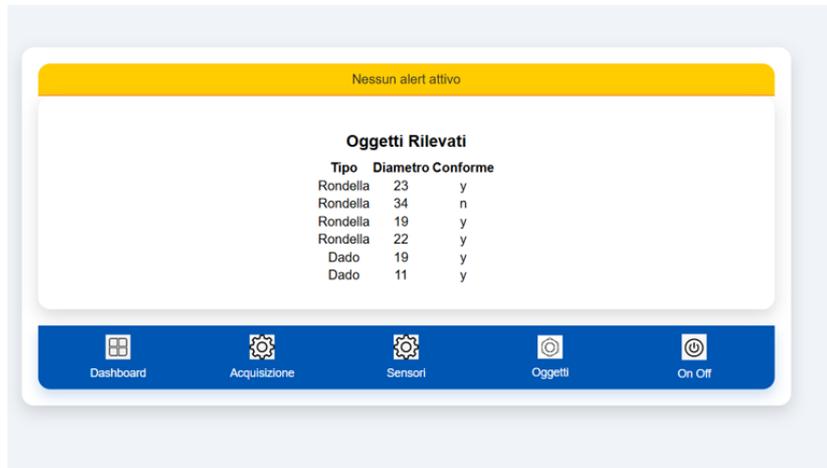


Figure 2.17: Detected objects overview

3. **Sensor threshold settings:** the interface displayed in Figure 2.18, operators can configure minimum and maximum threshold values for various monitored parameters, including humidity, vibration, temperature, current, voltage, and weight. These thresholds ensure that the system operates within safe limits, triggering alerts if any parameter exceeds the set bounds.

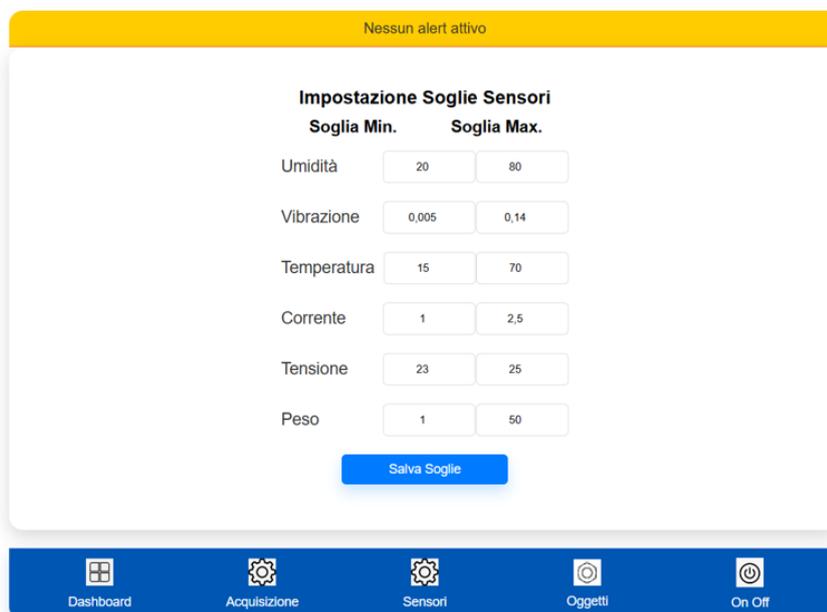


Figure 2.18: Sensor threshold settings

4. **Sensor dashboard with alerts:** the dashboard in Figure 2.19 provides a view of sensor readings, displaying current values relative to their respective thresholds.

If a parameter exceeds its limit, an alert message is shown in red, indicating that maintenance is necessary. This feature supports proactive maintenance, helping operators to quickly identify and address potential issues.

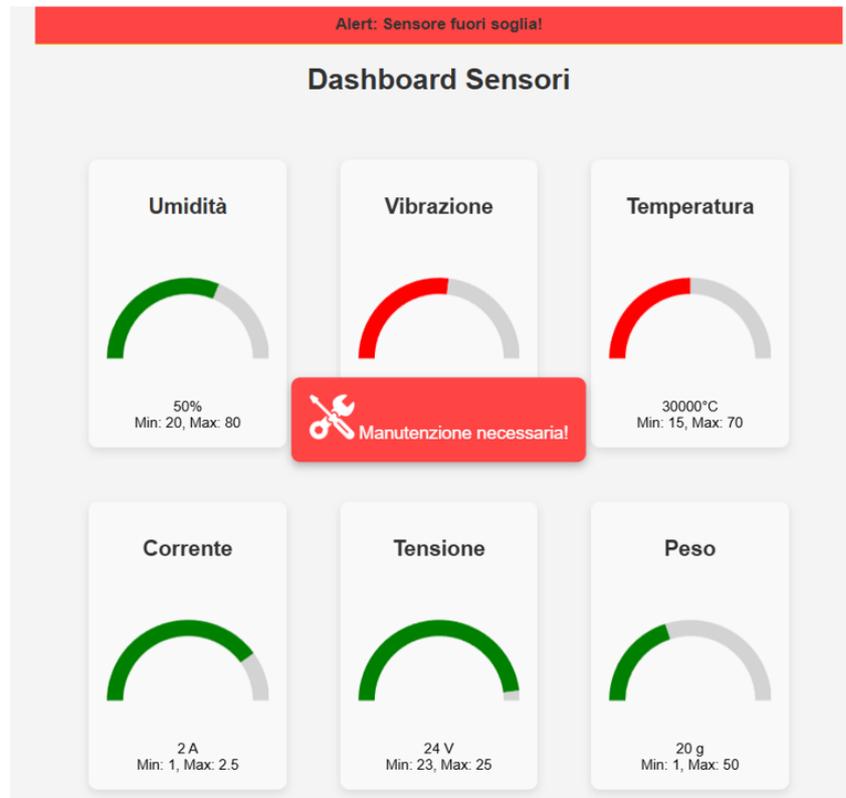
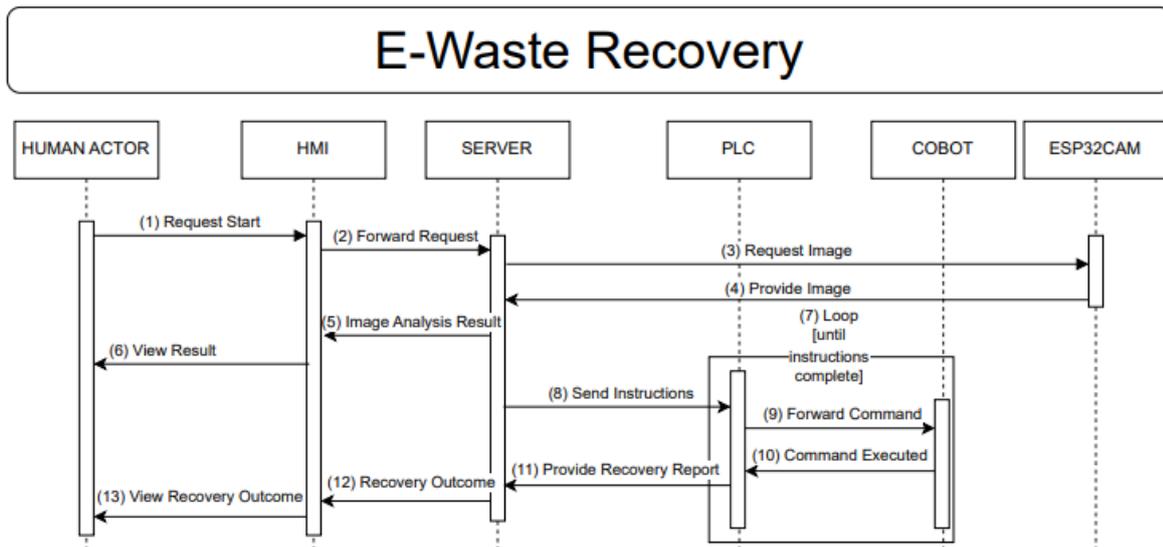


Figure 2.19: Sensor dashboard with alerts

Each of these screenshots illustrates a different aspect of the HMI's functionality, emphasizing its user-friendly design and flexibility for both on-site and remote monitoring. Together, they highlight the system's capability to adapt to operational needs and ensure efficient, reliable control over the cobot and associated hardware.

In addition, several UML diagrams are provided to offer a clearer understanding of the system's workflow and interactions. These diagrams include:

- **E-Waste recovery:** Figure 2.20 illustrates the main sequence of actions for the e-waste recovery process, from initial request to recovery report.



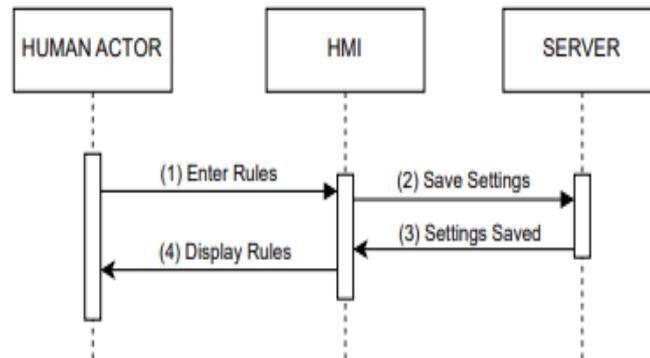
Message Description

1. Request Start: The Human Actor (user) initiates the recovery process through the HMI (human-machine interface).
2. Forward Request: The HMI forwards the start request to the server to initiate the recovery process.
3. Request Image: The server requests an image from the ESP32CAM to view the recovery area.
4. Provide Image: The ESP32CAM captures and sends the image to the server for analysis.
5. Image Analysis Result: The server analyzes the image with an AI model, identifying the coordinates of washers and nuts present, and, based on instructions provided by the user, determines which objects should be recovered. It then creates a file containing a set of instructions (records) for each object to be recovered.
6. View Result: The HMI displays the image analysis result to the user, specifying the objects to be recovered.
7. Loop [until instructions complete]: The PLC loops through the instruction file, sending each record of instructions to the cobot until all operations are completed.
8. Send Instructions: The server saves the instruction file and makes it available to the PLC for execution.
9. Forward Command: The PLC, reading each record from the file, forwards the commands one by one to the cobot to guide the recovery.
10. Command Executed: The cobot notifies the PLC that it has completed the operation related to a single command.
11. Provide Recovery Report: The PLC sends a final recovery report to the server, including details such as weight, dimensions, and time taken.
12. Recovery Outcome: The server transmits the overall recovery outcome to the HMI to update the user.
13. View Recovery Outcome: The HMI displays the final recovery outcome to the user, including the details provided in the report.

Figure 2.20: E-Waste recovery UML diagram

- **Recovery rule configuration:** Figure 2.21 depicts the setup of specific rules that guide the recovery operations.

RECOVERY RULE CONFIGURATION



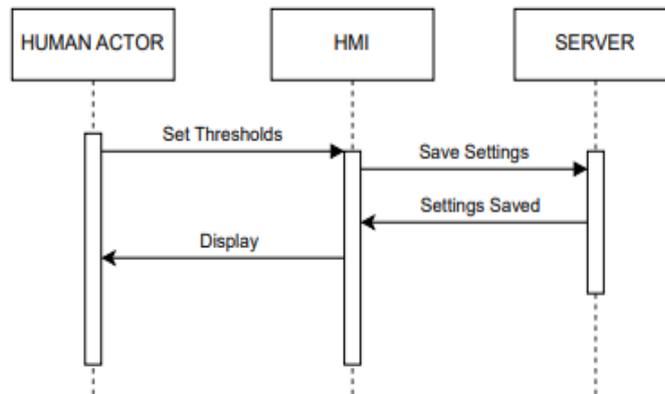
Message Description

1. **Enter Rules:** The **user** enters recovery rules through the **HMI**, specifying criteria (e.g., recover only washers, only nuts, or both of a certain diameter).
2. **Save Settings:** The **HMI** sends the entered rules to the **server**, which saves them for later use in the recovery process.
3. **Settings Saved:** The **server** confirms to the **HMI** that the rules have been saved successfully, providing feedback to ensure the process is complete.
4. **Display Rules:** The **HMI** displays the saved rules to the **user**, allowing a visual verification of the

Figure 2.21: Recovery rule configuration UML diagram

- **Alert threshold setup:** Figure 2.22 shows the configuration of threshold values for various sensors, enabling proactive alert management.

ALERT THRESHOLD SETUP

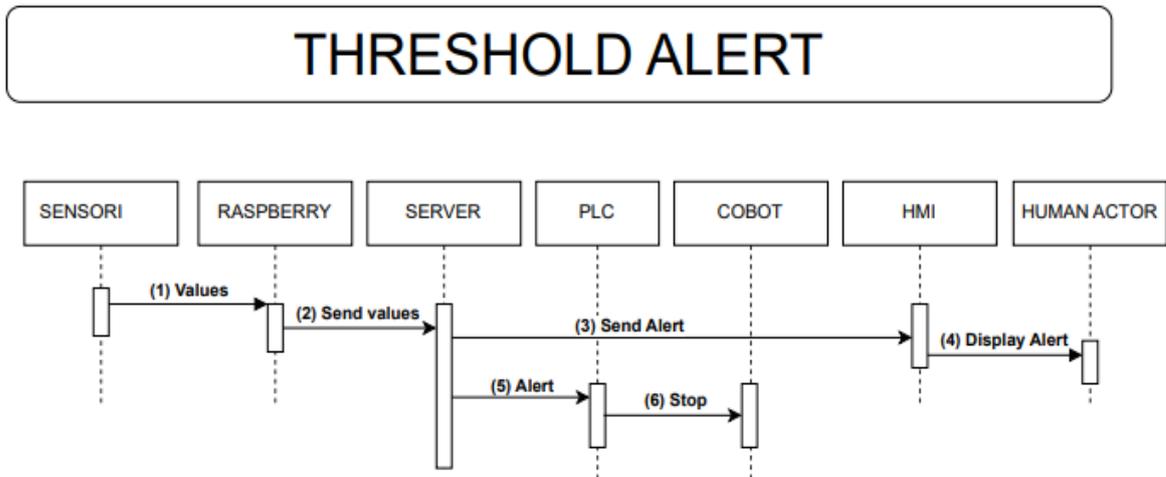


Message Description

1. **Set Thresholds:** The **user** defines alert thresholds through the **HMI**, specifying the limits at which alerts should be triggered.
2. **Save Settings:** The **HMI** sends the set thresholds to the **server**, which saves them to activate alerts when values exceed the defined thresholds.
3. **Settings Saved:** The **server** confirms to the **HMI** that the thresholds have been saved successfully, providing feedback to ensure the process is complete.
4. **Display:** The **HMI** displays the saved thresholds to the **user**, allowing a visual verification of the configured settings.

Figure 2.22: Alert threshold setup UML diagram

- **Threshold alert:** Figure 2.23 details the alert process when a threshold is exceeded, triggering maintenance notifications.



Message Description

1. **(1) Values** – (*Sensors* → *Raspberry*): The **sensors** monitor relevant parameters (such as temperature, pressure, weight, etc.) and send the detected values to the **Raspberry** for preliminary processing.
2. **(2) Send values** – (*Raspberry* → *Server*): The **Raspberry** transmits the collected sensor values to the **server**, which will use them to compare against the set thresholds.
3. **(3) Send Alert** – (*Server* → *PLC*): The **server** checks the values against the configured thresholds, and if one or more values exceed the allowed limits, it sends an alert signal to the **PLC**.
4. **(4) Display Alert** – (*PLC* → *HMI* → *Human Actor*): The **PLC** sends the alert to the **HMI**, which displays it to inform the **Human Actor** that a threshold exceedance has occurred.
5. **(5) Alert** – (*PLC* → *Cobot*): The **PLC** forwards the alert to the **cobot**, notifying it to stop the current operation to avoid potential damage or to manage the threshold exceedance situation.
6. **(6) Stop** – (*Cobot* → *PLC*): The **cobot** confirms to the **PLC** that it has halted the operation in response to the received alert, indicating that it is in a stopped state.

Figure 2.23: Threshold alert UML diagram

- **Predictive maintenance:** Figure 2.24 outlines the predictive maintenance workflow, using sensor data to anticipate failures and schedule interventions.

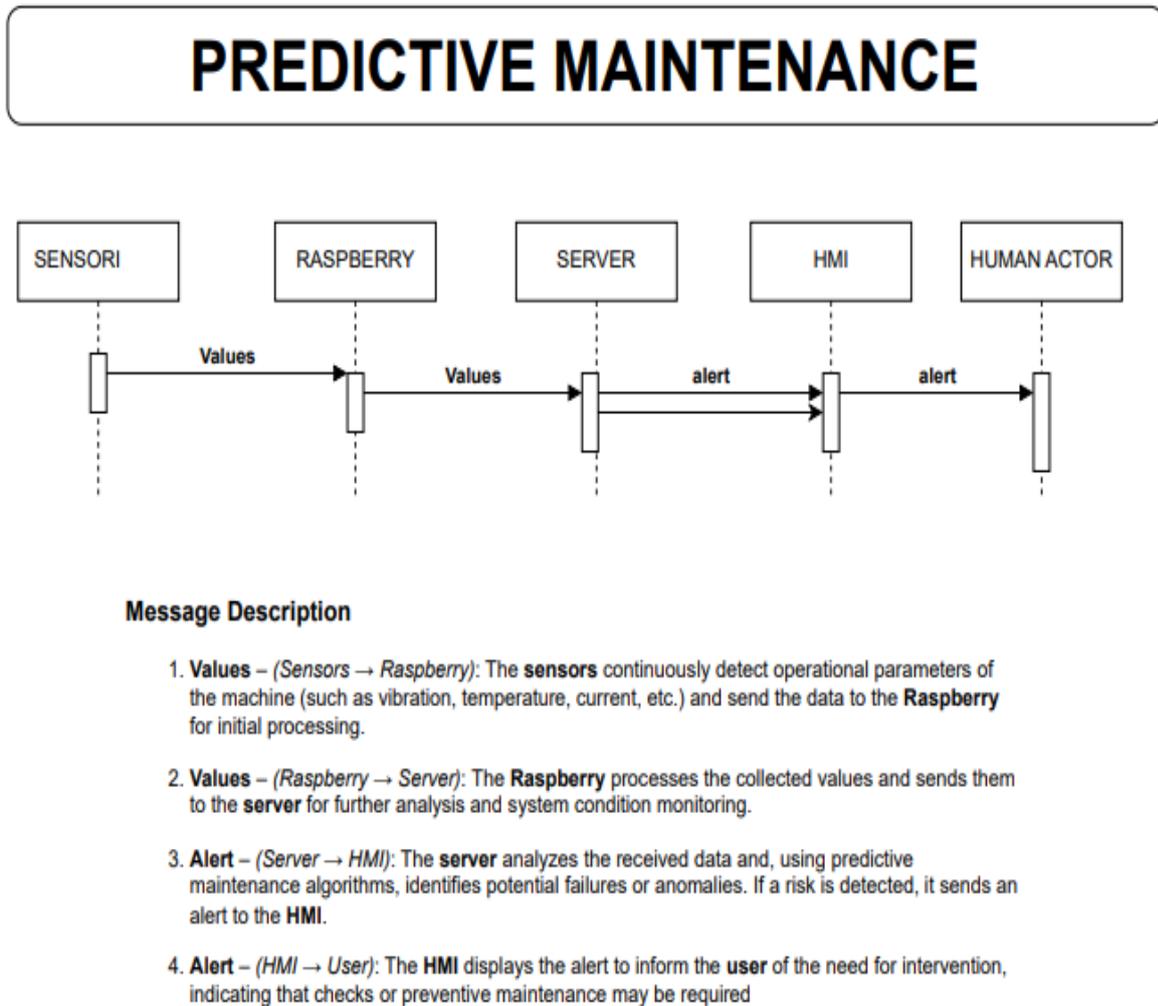


Figure 2.24: Predictive maintenance UML diagram

These diagrams serve to enhance the reader’s understanding of the system’s architecture and control logic, with particular regard to automation, monitoring, and predictive capabilities.

The overall software architecture ensures smooth integration between system components, enabling efficient process control, optimized motor speed management, and rapid response to changing operational conditions.

3 | Developed algorithms

This chapter describes the main algorithms developed for the cobot system, and in particular the object detection approach, the optimal speed controller, and the predictive maintenance scheme. Each algorithm is explained in detail, with a description of its operation, implementation, and role within the overall system architecture.

3.1. Object detection algorithm

The object detection algorithm is designed for identifying washers and nuts using an ESP32-CAM and YOLO object detection models. It consists of multiple core components and functions. These elements collectively enable real-time monitoring, geometric analysis, and precise positioning of detected objects within an automated process. The system uses customized YOLO models initially obtained from the Roboflow Universe and further trained to specialize in detecting washers and nuts with high accuracy.

3.1.1. Inputs from the HMI and image acquisition

The algorithm receives object selection criteria directly from a web-based Human-Machine Interface (HMI), enabling operators to define target dimensions and types, such as minimum and maximum diameter values for washers and nuts. This HMI supports real-time visual monitoring of detected objects and indicates whether they meet specified requirements, providing a responsive feedback mechanism that enhances control over the detection process.

As shown in Figure 3.1, the HMI allows operators to configure detection settings, including options to identify washers or nuts and set acceptable diameter ranges. It also includes fields for batch number entry and a submission button to send instructions to the robot, ensuring a streamlined interaction with the system and enabling customized detection for each batch.

Nessun alert attivo

Impostazione Acquisizione



Individua Rondelle

Diametro Minimo Rondelle (mm)

Diametro Massimo Rondelle (mm)



Individua Dadi

Diametro Minimo Dadi (mm)

Diametro Massimo Dadi (mm)

Numero Lotto

Invia Istruzioni al Robot

Dashboard
Acquisizione
Sensori
On Off

Figure 3.1: Instructions interface

Additionally, the algorithm includes functions to:

- **Read conformity parameters from the HMI:** a JSON file is accessed to retrieve operator-defined dimensional parameters, ensuring compliance by filtering objects that fall within the acceptable range.
- **Capture images from the ESP32-CAM:** using an HTTP request, images are retrieved directly from the ESP32-CAM and prepared for analysis, enabling real-time image acquisition essential for automated monitoring.

3.1.2. Supporting functions for detection and measurement

The algorithm's core logic incorporates several support functions essential for accurate detection and measurement. Key functionalities include:

- **Overlap detection**, which calculates the degree of overlap between two bounding boxes, using a predefined threshold to determine if the boxes are sufficiently separate. By doing so, it minimizes interference between nearby objects, ensuring clear differentiation between individual items.
- **Geometric calculations**: a dedicated function determines each bounding box's center, area, and diameter in millimeters by applying a scale factor based on image resolution and camera distance. This function enables accurate geometric data collection for each detected object, adjusted to the camera's distance and field of view.

3.1.3. Object detection and filtering

The algorithm's detection process is driven by two YOLO models trained specifically for identifying washers and nuts. Each model returns bounding box coordinates for detected objects, along with a confidence level indicating detection reliability. The detection function performs several critical tasks:

- **Dimensional compliance check**, in which each object is evaluated against the dimensional parameters provided via HMI, confirming whether the detected washers and nuts meet the specified tolerances.
- **Overlap filtering**, in which by calculating bounding box overlap, the function discards instances where objects may be ambiguously positioned, thus enhancing accuracy.
- **Recording results in CSV format**, in which each detected object's data (type, center coordinates, area, diameter, and compliance status) is recorded in a CSV file. This structured output format enables subsequent analysis and archival of detection results.

To provide a clearer understanding of the YOLO models' performance and their role in the detection process, the following figures illustrate key metrics and examples. These visual representations offer insights into how the models balance precision and recall, accurately classify washers and nuts, and operate effectively in diverse real-world conditions.

Figures 3.2 and 3.3 display the Precision-Recall (PR) curves for each model, highlighting differences in their performance for hex nuts and washers. Model 1 (Figure 3.2), trained specifically for hex nuts, achieves near-perfect precision and recall, indicating exceptionally high accuracy in detecting hex nuts with minimal false positives and negatives. This high performance reflects the fact that Model 1 was trained more extensively, with additional

data and epochs, to optimize detection of hex nuts. This extra training was necessary because hex nuts are generally more challenging to recognize due to their complex shape, which includes multiple edges and facets that can vary based on orientation and lighting.

In contrast, Model 2 (Figure 3.3) trained for washers, shows a slightly less steep curve, with a small drop in precision at high recall levels. Washers, having a simpler and more distinct circular shape, are inherently easier to recognize, requiring less intensive training. While Model 2 performs well, the comparison reveals the impact of additional training on Model 1, as its extensive training has led to near-perfect accuracy for the more complex hex nuts. This highlights how adjusting training depth based on object complexity can improve detection performance.

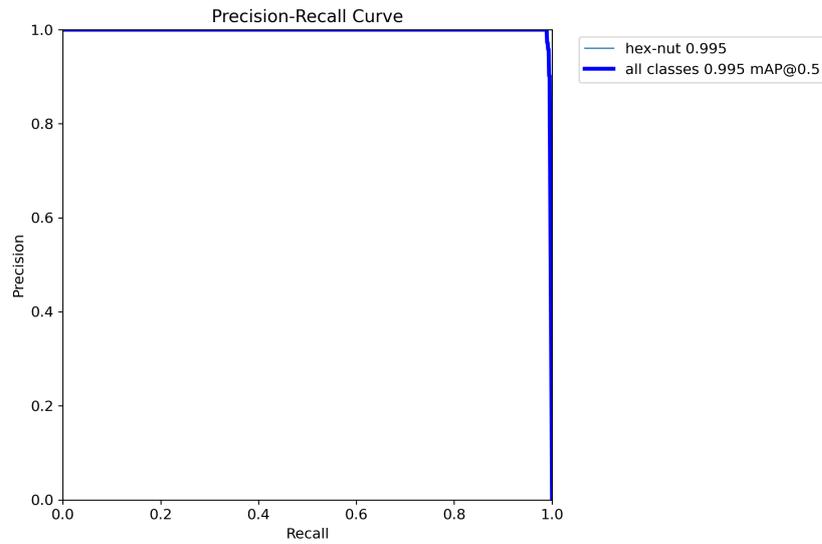


Figure 3.2: Precision-Recall curve for hex-nuts

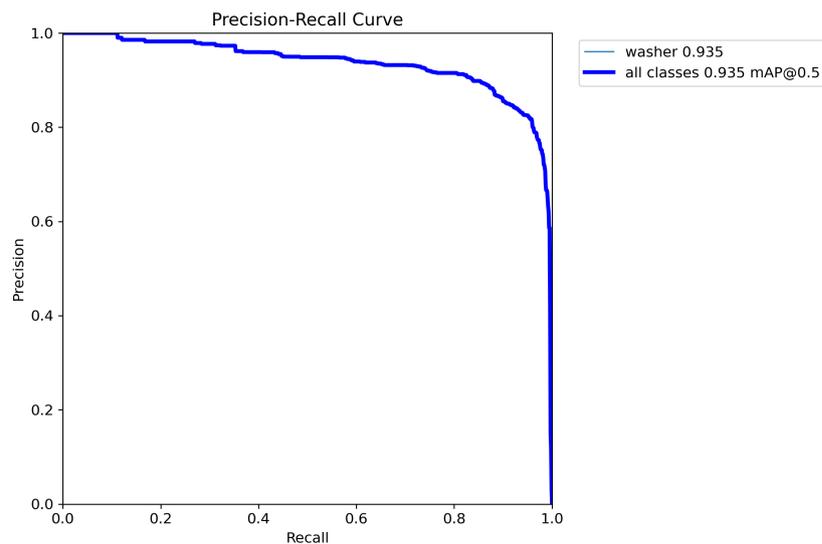


Figure 3.3: Precision-Recall curve for washers

Figures 3.4 and 3.5 show the normalized confusion matrices for hex nuts and washers, respectively. In Figure 3.4, the model demonstrates high accuracy for hex nuts, correctly identifying 99% of them, with only a minor error where 1% of hex nuts are misclassified as background. Additionally, the model achieves 100% accuracy in distinguishing the background, with no instances of background being misclassified as hex nuts. This result suggests that the model is highly effective at recognizing hex nuts, likely due to intensive training tailored to handle the complex shapes and features typical of hex nuts. In Figure 3.5, which represents the model trained for washers, we see that 96% of washers are correctly identified, with a small 4% misclassified as background. The background category

is again perfectly distinguished with 100% accuracy. Washers, with their simpler circular shape, are inherently easier for the model to recognize, though the minor misclassification indicates occasional challenges, potentially due to partial occlusion or specific variations in appearance. Comparing the two matrices, it is clear that both models perform exceptionally well; however, the model trained for hex nuts required more focused training to effectively manage the additional complexity of hex nut shapes.

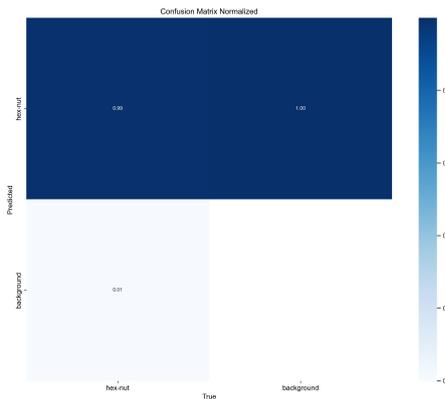


Figure 3.4: Normalized confusion matrix for hex-nuts

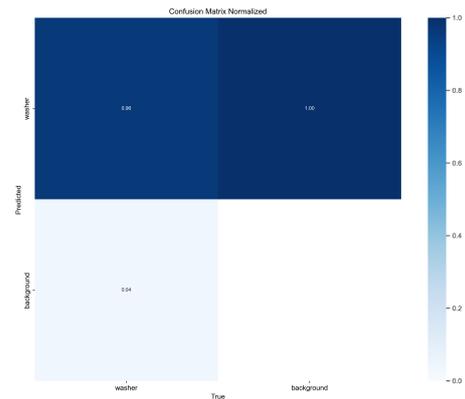


Figure 3.5: Normalized confusion matrix for washers

In Figure 3.6, we see the detection results for hex nuts, highlighted with blue bounding boxes and confidence scores. The model accurately identifies all hex nuts present in the image, with confidence scores around 0.91 to 0.92. This high level of confidence across multiple instances reflects the model's ability to detect hex nuts effectively, even when they vary in size and orientation. In Figure 3.7, the detection results for washers are shown, marked with green bounding boxes and corresponding confidence scores. The model successfully identifies each washer, achieving high confidence levels ranging from 0.89 to 0.99. The distinct shape of the washers likely contributes to these strong results, allowing the model to consistently differentiate washers from hex nuts and other objects in the background. These images illustrate the effectiveness of the model in recognizing both hex nuts and washers with high accuracy.

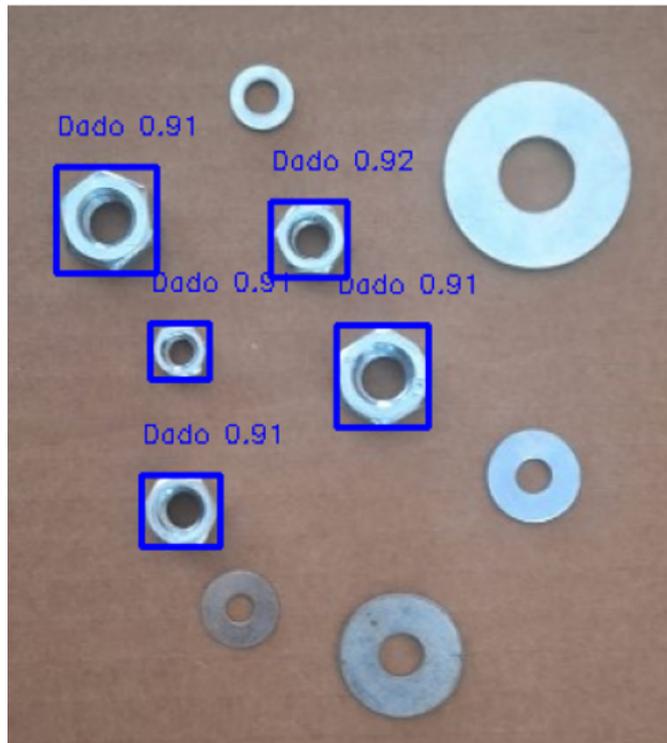


Figure 3.6: Nuts detection

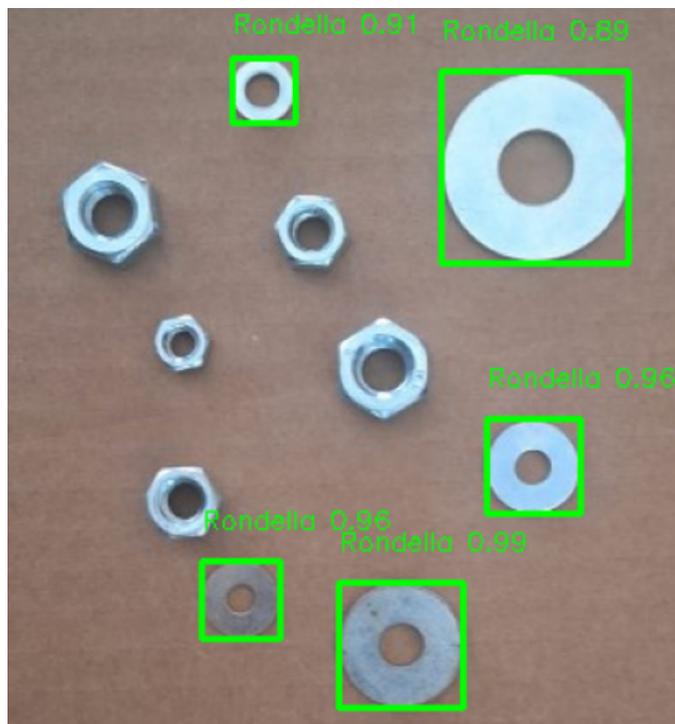


Figure 3.7: Washers detection

3.1.4. Coordinate conversion and motor step calculation

After detection and measurement, the system translates pixel-based coordinates into centimeters using a scaling function derived from the camera's distance and field of view. This conversion is crucial for calculating exact physical positions of objects. Knowing these real-world coordinates, the system determines the required motor steps to accurately reach or manipulate the objects, enabling precise and efficient control over motor movement in real-time.

3.1.5. Continuous detection loop

The continuous detection loop is managed by a script running on the server that monitors a file named "command.txt." This file serves as a communication link between the operator, who uses a Human-Machine Interface (HMI), and the PLC.

- **Command Activation:** when the operator requests image capture and detection through the HMI, the PLC sets the value of "command.txt" to "1." This action signals the server-side script to activate the detection system.
- **Detection completion and reset:** upon detecting that "command.txt" has been set to "1," the script initiates the image capture, performs object detection, and logs the relevant data. Once the detection cycle is complete, the script resets the value in "command.txt" back to "0," indicating that the process is finished and the system is ready for the next request.
- **Next detection cycle:** the PLC can initiate a new detection cycle at any time by setting "command.txt" back to "1" upon a new request from the operator. This allows for precise control over the detection timing, ensuring that the system only captures and processes images when needed according to the production workflow.

In this setup, the "command.txt" file functions as a dynamic control switch, with the server script and the PLC working in tandem based on real-time needs. This approach minimizes resource usage by activating the detection loop only when necessary and ensures seamless integration between the operator's requests via the HMI, the PLC, and the detection system, aligning with the timing demands of the production line.

3.2. Predictive maintenance algorithm

One of the goal of this work is the development of a predictive maintenance algorithm designed to monitor and anticipate potential failures. The initial step of the algorithm

is based on simulated data generated to represent realistic operating conditions for key parameters such as current, voltage, vibration, humidity, temperature, and load weight. This synthetic dataset provides a solid foundation for the preliminary training and validation of the model, allowing for the identification of the most relevant parameters and optimization of model configurations in the absence of real data. The approach involves a gradual evolution: the simulated data will be progressively replaced by real sensor readings collected directly from the operating system. This transition will enable the model to adapt to the specific conditions of the real operational environment, enhancing the accuracy of predictions and the ability to proactively identify failures. Consequently, the algorithm will be capable of reliably managing data variability over time and providing timely maintenance alerts, thereby reducing downtime and optimizing overall system efficiency. The specifications for the simulated parameters are as follows:

3.2.1. Simulated dataset

The simulated data used to develop the predictive maintenance algorithm is generated using a custom Python script. This script creates a synthetic dataset of key operational parameters—such as current, voltage, vibration, temperature, humidity, and load weight—based on realistic operating ranges for the system.

- **Current (A)**, which is uniformly distributed between 1.0 A and 2.0 A, with some anomalies extending beyond this range;
- **Voltage (V)**, whose values are concentrated between 23 V and 25 V, with occasional anomalies dropping to as low as 20 V;
- **Weight (g)**, with most values range between 1 g and 50 g, with some samples reaching up to 500 g;
- **Vibration (G)**, which is distributed between 0.005 G and 0.05 G, with some higher values that may indicate anomalies.
- **Humidity (%)**, which Values are distributed between 20% and 60%.
- **Temperature (°C)**, with most samples fall between 25°C and 70°C, with some anomalies reaching up to 100°C.

More specifically, the dataset comprises 20,000 samples with a one-minute interval between each other, providing a robust foundation for training the algorithm. This simulated dataset closely approximates real-world operating conditions, allowing for initial training and testing of the predictive maintenance model before real data becomes available.

3.2.2. Exploratory Data Analysis

The subsequent step involves the *eda.py* script, designed to perform Exploratory Data Analysis (EDA) on the simulated dataset. This script uses key Python libraries, such as *pandas* for data manipulation, *matplotlib* and *seaborn* for data visualization, and *logging* to track the analysis process. Its main functions are the following:

- **Dataset loading and error handling:** the script loads the simulated dataset from a CSV file and logs success or error messages, documenting any issues encountered with data access.
- **Data type conversion:** this phase ensures that all numeric columns are formatted as floating-point numbers to maintain consistency in the analysis.
- **Visualization setup:** this step applies a clean, readable plot style optimized to highlight trends and relationships within the data.
- **Visualizations and analysis:** this function generates distribution plots and correlation maps to explore the dataset's key characteristics, identifying patterns or anomalies before proceeding with model training.

In summary, the *eda.py* script is an essential tool for preparing and conducting exploratory data analysis on the simulated dataset, providing a solid foundation for the subsequent development of the predictive maintenance model. The script methodically performs data loading, validation, and visualization, enabling the early detection of any irregularities that might affect later stages of the process. For instance, a distribution plot is generated for the weight variable (see Figure 3.8) to analyze the range and variability of values, and similar plots are created for other key variables. This provides a comprehensive view of each parameter's behavior, facilitating an understanding of the dataset's main characteristics. Additionally, a correlation matrix (Figure 3.9) is computed to explore relationships between variables, identifying key predictive variables for the maintenance algorithm. These visualizations not only improve dataset quality by highlighting patterns, outliers, and anomalies but also enable model refinement, ensuring that the algorithm can better adapt to real-world operational conditions.

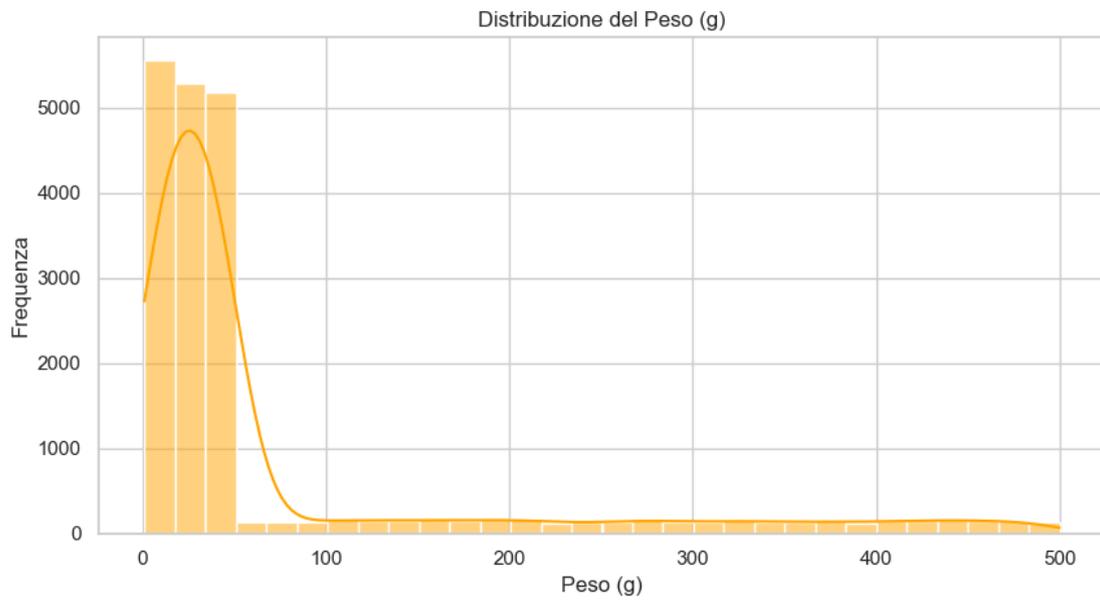


Figure 3.8: Weight distribution



Figure 3.9: Correlation matrix

3.2.3. Model training

The training process is carried out using a specific script that employs a Random Forest classifier to predict the need for maintenance in the system, based on the previously generated simulated dataset. Below a detailed breakdown of the main steps executed by the script is given.

1. **Dataset loading:** the script loads the simulated dataset using the pandas library. This dataset includes various operational variables (such as current, voltage, weight, vibration, humidity, and temperature) along with a target label that indicates whether maintenance is required.
2. **Data pre-processing:**
 - **Separating features and target:** the dataset is divided into a vector X , containing the independent variables (operational parameters) and y , which contains the target label.
 - **Splitting into training and test sets:** using ‘train_test_split’ from sklearn, the data is divided into a training set (80%) and a test set (20%) to validate the model’s performance. This split ensures that the model is evaluated on data it has not seen during training, providing an unbiased measure of its predictive accuracy.
 - **Feature standardization:** the script applies a ‘StandardScaler’ to normalize the values of the independent variables in X . This standardization is essential for models sensitive to the scale of input features. The scaler is saved to a file for future use, enabling consistent transformation of new data before it is fed into the model.
3. **Training the Random Forest model:** the script trains a Random Forest classifier on the training set. This model is well-suited for binary classification tasks like predictive maintenance due to its robustness in handling correlated variables and its tendency to reduce overfitting. Additionally, Random Forest is effective in managing complex datasets with non-linear relationships, which are common in predictive maintenance scenarios.
4. **Model evaluation:** after training, the model’s performance is evaluated using the test set, yielding outstanding results:
 - **Classification report:** precision, recall, and F1-score for both classes (0 and 1) are all 1.00, indicating that the model accurately classified every test sample.

-

$$\text{Confusion Matrix: } \begin{bmatrix} 3803 & 0 \\ 0 & 197 \end{bmatrix}$$

All samples from class 0 and 1 were correctly classified. These results, with an accuracy of 100%, suggest that the Random Forest model achieved perfect classification on the test set, demonstrating its efficacy in predicting maintenance needs with high reliability.

5. **Model persistence:** once trained and evaluated, the Random Forest model is saved to a file, allowing it to be deployed in real-time applications. The saved model can be loaded and applied to new data as it becomes available, enabling continuous, data-driven maintenance predictions in the operational environment.

This structured training process establishes a robust predictive maintenance model that adapts effectively to real-world operational conditions, ensuring both accuracy and consistency in anticipating maintenance needs. The model's exceptional performance on the test set further underscores its suitability for deployment in predictive maintenance tasks.

3.2.4. Real time inference for maintenance prediction

A third Python script implements an inference pipeline for the practical application of the predictive maintenance model. Each time the *sensori.csv* file, located in a shared area on the server, is updated, the model analysis is automatically triggered to process the new record. The main steps of the script are as follows:

1. **Loading the model and scaler:** the script loads the previously trained Random Forest model along with the standard scaler, ensuring that data transformations are consistent with those applied during the training phase.
2. **Loading the real-time data:** the script reads the new record in the *sensori.csv* file, specifically the latest update that has not yet been processed.
3. **Model inference and CSV file update:** after normalizing the operational parameters, the script uses the model to predict whether maintenance is required. The prediction result is added as the last column in the record *manutenzione_necessaria*, indicating whether intervention is needed. The *sensori.csv* file is then updated with this information, enabling historical tracking of operational conditions and maintenance predictions.
4. **Generating alerts:** if the model predicts the need for maintenance, the script overwrites the *alert_manutenzione.json* file with the sensor readings and a notification

to proceed with maintenance, serving as a real-time alert system.

This inference pipeline completes the predictive maintenance system by applying the trained model to data collected automatically each time the file is updated, providing timely alerts when maintenance interventions are necessary. The integration with the *sensori.csv* and *alert_manutenzione.json* files ensures reliable and continuous system monitoring. This proactive approach enables the anticipation of maintenance needs, reducing downtime and enhancing overall system efficiency. The automation of the inference pipeline makes the predictive maintenance model a reliable and adaptable solution for complex operational environments.

3.2.5. Periodic model retraining

A final python script periodically retrains the predictive maintenance model using accumulated historical data to maintain accuracy. It consolidates data from various production lines, normalizes it, and retrains the Random Forest model and scaler each month. This automated process, managed by a Task Scheduler, ensures the model stays aligned with current operational conditions.

3.3. Speed control algorithm

The TB6560 driver uses an **H-bridge** combined with **Pulse Width Modulation (PWM)** and a **feedback control loop** to precisely regulate the current in each motor phase. Additionally, the H-bridge enables control over the direction of current flow, allowing the motor to rotate clockwise or counterclockwise based on the **DIR signal**.

The **STEP signal** serves as a pulse input, instructing the driver to move the motor by one step for each received pulse. The frequency of these pulses directly influences the motor's speed: an increased step frequency leads to a higher rotational speed, whereas a lower frequency results in slower movement. Consequently, the pulse rate of the step signal is effectively translated by the driver into the rotational speed of the motor.

The driver applies a PWM signal to each phase of the motor to control the average current. The **duty cycle** of the PWM (the percentage of time the signal is "on" during each cycle) determines the effective current in the motor windings. A higher duty cycle (e.g., 80%) results in a higher average current, while a lower duty cycle (e.g., 20%) reduces the current.

In the current setup, the TB6560 continuously measures the actual current in each motor phase and compares it to the target current required. This target current depends on the microstepping profile, which varies according to the chosen stepping mode (e.g., full step,

half step, or finer microstepping like 1/8 or 1/16 steps). Typically, a sinusoidal approximation is used in higher microstepping modes to achieve smoother movement and finer position control. If the measured current is lower than the target, the driver increases the PWM duty cycle to raise the current. If the current is too high, the driver reduces the duty cycle. This real-time feedback loop ensures that the current stays at the desired level for each microstep, which is crucial for precise and stable motor control.

In summary, the TB6560 driver is able to:

- **regulate current intensity** through PWM modulation, adjusting the duty cycle to maintain the target current for each microstep;
- **control rotation direction** based on the DIR signal, allowing the motor to rotate clockwise or counterclockwise by changing the direction of current flow in the windings;
- **control motor speed** according to the frequency of the STEP signal.

The combination of precise current control, direction switching, and speed regulation provides smooth, stable microstepping and accurate positioning for stepper motors.

In view of this, using the current driver, direct control of the current is not possible, as it is managed internally by the driver itself, as just discussed. Only a maximum current limit can be set. This limitation implies that real-time control is based on adjusting the motor's speed while keeping the current within the established limits. The control strategy is therefore oriented toward optimizing speed as a function of the applied load (weight), respecting current limitations to avoid overload.

3.3.1. Motor model

To design a speed control algorithm that considers the load weight, we consider the motor model. The main equations governing the motor's behavior are the following:

$$\frac{di_a}{dt} = \frac{1}{L} (v_a - Ri_a + K_m\omega \sin(N\theta)) \quad (3.1a)$$

$$\frac{di_b}{dt} = \frac{1}{L} (v_b - Ri_b - K_m\omega \cos(N\theta)) \quad (3.1b)$$

$$\frac{d\omega}{dt} = \frac{1}{J} (-K_m i_a \sin(N\theta) + K_m i_b \cos(N\theta) - K_v\omega - T_l) \quad (3.1c)$$

$$\frac{d\theta}{dt} = \omega \quad (3.1d)$$

where:

- i_a and i_b are the phase currents,
- v_a and v_b are the phase voltages,
- ω is the angular velocity,
- θ is the angular position of the motor,
- T_l represents the load torque,
- L is the inductance, R is the resistance, K_m is the torque constant, K_v is the viscous friction coefficient, and J is the inertia,
- N is the number of rotor teeth, which is 50 for the Nema 17.

The values for inductance L , resistance R , inertia J , and torque constant K_m are obtained from the motor's datasheet. The viscous friction coefficient K_v was instead estimated experimentally to accurately reflect the motor's behavior under static conditions. The torque constant K_m is obtained through the relationship between the torque produced and the current supplied to the motor. This constant represents an approximation, as it assumes linearity between current and torque, neglecting non-linearities such as magnetic saturation or minor variations in the motor's behavior under different operational conditions.

Estimation of parameter K_v

The viscous friction coefficient K_v has been experimentally estimated by assessing the performance of the motor under no-load conditions. To achieve this, we first measured the frictional torque, which was calculated using the motor's torque constant K_m and the current flowing through the motor phases.

Under no-load conditions and at a constant angular velocity, we can assume that the frictional torque is equivalent to the torque generated by the motor. This assumption is valid because, at constant velocity, the motor must generate enough torque to overcome the internal friction forces without any additional load, allowing us to consider the torque of the motor as equal to the frictional torque. The calculations have been conducted using various angular velocity values, allowing us to perform an interpolation to estimate K_v more accurately. The results of these calculations are summarized in Table 3.1.

Using the values obtained, a linear regression analysis was performed to estimate the viscous friction coefficient K_v . The linear fit provides an estimate of K_v , representing the slope of the best-fit line through the data points, as shown in Figure 3.10. The resulting coefficient captures the dynamic performance of the motor under the specified experimental conditions.

Table 3.1: Velocities and corresponding currents

Angular Velocity (rad/s)	Torque (Nm)	Current (mA)
2	0.0247	105
7	0.085	360
10	0.126	535
15	0.187	795

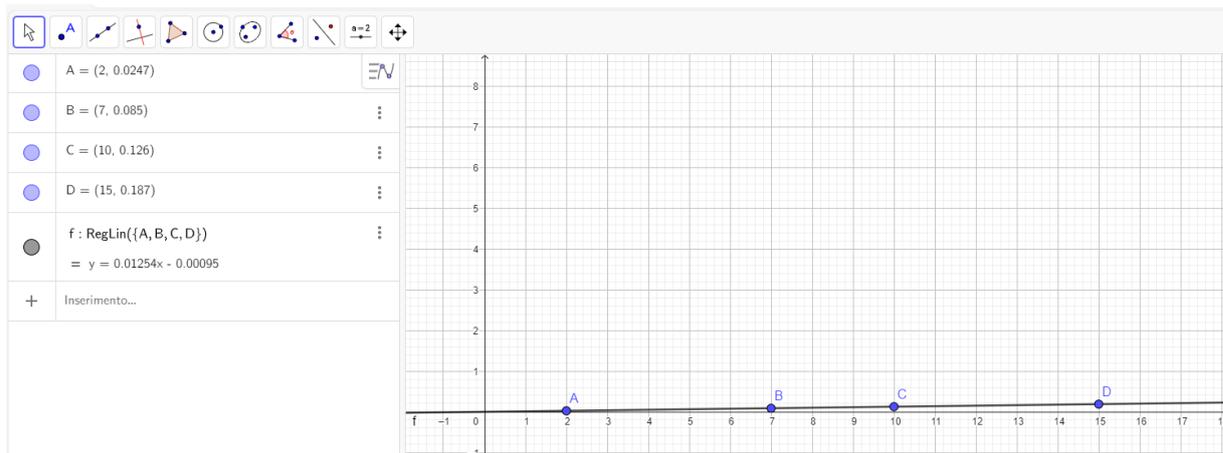


Figure 3.10: Linear regression

Model simulation

The simulation results presented in Figure 3.11 illustrate the response of the stepper motor, which is controlled by a simulated driver implemented in Simulink.

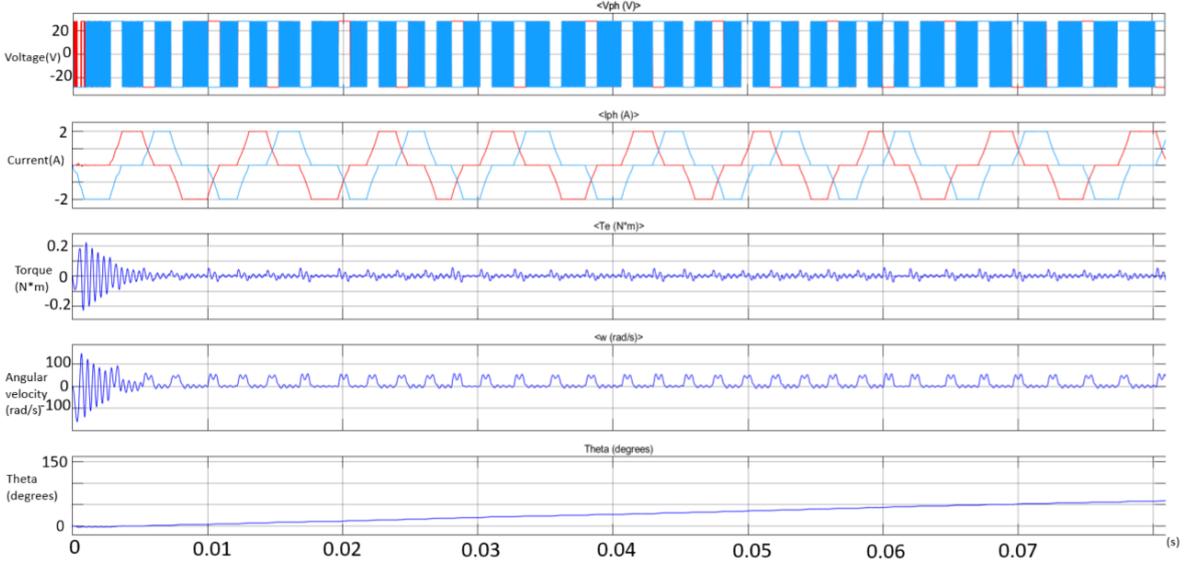


Figure 3.11: Simulation results of the stepper motor model controlled by a simulated driver, showing the phase voltage (V_{ph}), phase current (I_{ph}), torque (T_e), angular velocity (ω), and angular position (θ).

Static model

For optimization purposes, a simplified static model derived from equations (3.1) will be used, rather than the full dynamic equations. This approach allows for faster and real-time calculations by focusing on the immediate relationship between speed, load, and current. By bypassing the time-dependent dynamics associated with current fluctuations and angular velocity, this methodology minimizes computational overhead and ensures stable performance within prescribed operational limits.

The static model is obtained by considering equation (3.1c) and the total current I , that represents the resultant current, calculated as $I = \sqrt{i_a^2 + i_b^2}$, which combines the phase currents i_a and i_b to generate the torque $T_e = K_m \cdot I$.

The current I is therefore obtained as:

$$I = \frac{T_l + K_v \cdot \omega}{K_m} \quad (3.2)$$

where:

- T_e is the total driving torque needed (sum of load torque T_l and the torque due to viscous friction, $K_v \cdot \omega$),
- K_m is the motor's torque constant.

Note that, the current hardware configuration uses the full-step mode (200 steps per revolution) due to hardware limitations, further discussed in the next chapter. In full-step mode, the current waveform is a square wave, switching between discrete values at each step rather than varying continuously as in microstepping. This characteristic makes the static model not only adequate but particularly well-suited for this application, as it aligns closely with the actual behavior of the motor in this mode.

3.3.2. Speed control law

An algorithm has been developed to determine the optimal motor speed based on real-time load weight. We define a cost function J to balance speed and current effectively, expressed as follows:

$$J = \lambda_1(\omega_{\max} - \omega)^2 + \lambda_2 I^2$$

where:

- ω_{\max} is the maximum permissible angular velocity,
- ω is the current angular velocity,
- I is the current required to maintain ω with the given load,
- λ_1 and λ_2 are weight factors balancing the emphasis on speed versus current.

The weights λ_1 and λ_2 can be chosen, e.g., to prioritize current savings over high speeds. This configuration helps to ensure energy-efficient operation while keeping within safe current limits. However, if higher speeds are needed, the weights can be adjusted to increase the importance of speed relative to current consumption.

The optimization process aims to minimize J by adjusting ω , such that the speed is maximized within the limits of current draw. During optimization, the following constraints are also applied:

- **Speed constraint:** $\omega \leq \omega_{\max}$, ensuring that the motor does not exceed the safe operational speed.
- **Current constraint:** $I \leq I_{\max}$, where I_{\max} is the maximum allowable current, as defined by the driver's settings.

These constraints guarantee that the motor operates safely within the predefined limits, balancing efficiency with system protection.

3.3.3. Implementation of the speed controller

The speed control function implemented in the plant is composed of the following main steps:

1. Input: load weight

The script begins by receiving, in real time, the weight of the object that the motor system must lift.

2. Calculation of the load torque

Considering the load weight as input, the script calculates the corresponding *load torque* T_l , as:

$$T_l = Fr$$

where:

- F is the force exerted by the weight (i.e., $F = \text{weight} \times g$, where g is the gravitational constant),
- r is the radius of the pulley.

Since two identical motors are used in the system, both running at the same time to lift the load, the *effective load torque is divided in two*. Each motor is indeed responsible for half the total load torque, thus ensuring balanced load distribution and reducing the required torque per motor.

3. Optimization of the motor speed

The script calculates the optimal *angular velocity* ω for each motor to lift the load effectively while staying within current and speed limits, i.e., by formulating an optimization problem through a cost function that adjusts ω to balance efficiency with load requirements. The objective is to derive the ω that achieves this balance without exceeding the predefined constraints.

4. Conversion to clock period

The calculated optimal angular velocity ω is then *converted into a STEP pulse interval* that can be used by the PLC to control the driver. This pulse determines the timing signal for the motor's control, translating speed requirements into precise inputs for the motor driver. By adjusting the STEP pulse interval, the PLC ensures that the motor runs at the calculated optimal speed, allowing the load to be lifted efficiently, with minimal strain, and within safe operational limits.

5. PLC integration

Finally, the calculated STEP pulse is sent to the *PLC*, which controls the driver responsible for powering the motor. The PLC uses this pulse to adjust the motor's speed in real time, aligning motor performance with load requirements as the weight changes.

3.4. Data acquisition and analysis infrastructure

This section describes the data acquisition and analysis infrastructure developed to monitor and optimize the operation of cobots in an automated industrial environment. The system is based on an architecture that integrates distributed PLCs for each cobot and a centralized server, designed for the real-time collection and monitoring of production parameters. The data, acquired from the cobots and sent to the server, are stored in a standardized format that allows for continuous, in-depth analysis, supported by an interactive dashboard for operational control.

To test and optimize the data acquisition and analysis infrastructure, a simulated dataset was created that faithfully replicates the structure and operational dynamics of actual production data. In this simulation, a distributed system with six cobots operating in three different international locations was assumed: Hong Kong (1), London (2), and Milan (3). Each location was configured to handle different product lines, including washing machines, ovens, hair dryers, and toasters. This test dataset enabled a thorough trial of the dashboard and monitoring and analysis functionalities, thus preparing the infrastructure for real data application.

3.4.1. System architecture

The data collection infrastructure comprises the following key components:

1. **Central server and cobot PLCs:** the PLCs of the various cobots periodically send the collected data to the central server, where all information is stored in a single file, *cobot_production_data.csv*. This architecture, with a centralized server and distributed PLCs, enables centralized management of production data, keeping all information accessible in a single location. Each file record contains a unique identifier (CobotCode) that allows individual tracking and analysis of each cobot's performance.
2. **Production data format** THE data are stored in the CSV file *cobot_production_data.csv*, containing the following main information:

- **Identification data:** LotCode (lot code), Type (type of object, such as bolt or washer), and Diameter (object diameter in millimeters).
- **Operational parameters:** ExecutionTime (time taken to complete the task) and WeightMeasured (weight of the processed object), which allow monitoring of cobot performance and workload.
- **Operational and environmental conditions:** sensor readings (vibration, temperature, current, voltage, and humidity) provide relevant data for evaluating environmental conditions and identifying potential anomalies. These values are essential to ensure that cobots operate within optimal parameters and for supporting predictive analysis.

3. **Interactive web dashboard for monitoring:** an interactive web dashboard has been developed using Dash, accessible through the local area network (LAN) from various devices, including desktop computers, tablets, and smartphones. The dashboard enables real-time data visualization and facilitates monitoring of key production process metrics, making it accessible to operators even on the move.

The dashboard was created with *Dash* and *Plotly*, tools that allow for the development of interactive web applications in Python and advanced, customizable data visualizations. Currently, the dashboard features four main charts: distribution of objects processed by cobot, daily total weight, productivity by cobot and location, and a timeline of execution time. Data is loaded directly from the centralized CSV file, and interactive components, such as filters, allow users to view specific data selections by cobot, production line, and location. This modular structure makes the dashboard easily expandable, allowing new charts and features to be added without substantial changes to the existing architecture. LAN operation ensures that the dashboard is accessible on various devices connected to the local network, maintaining the integrity and security of production data. Figure 3.12 displays the dashboard, showing the execution time graph with a filter ("Execution Time Over Time") and the count by object type ("Type Count per Cobot"). This visualization enables monitoring of each cobot's operating times, as well as the distribution of processed objects.

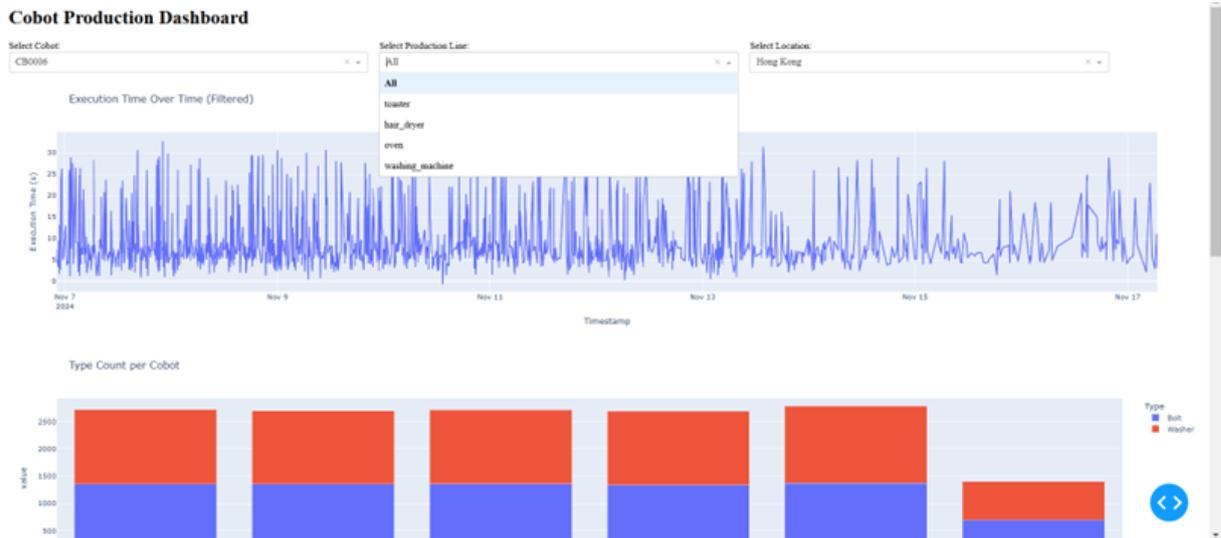


Figure 3.12: Execution time and object count per cobot

Figure 3.13 shows the daily total weight chart ("Total Weight per Day") and productivity by cobot and location ("Productivity by Cobot and Location"). These charts provide an overview of system productivity, showing the workload distribution across cobots and various operational sites.

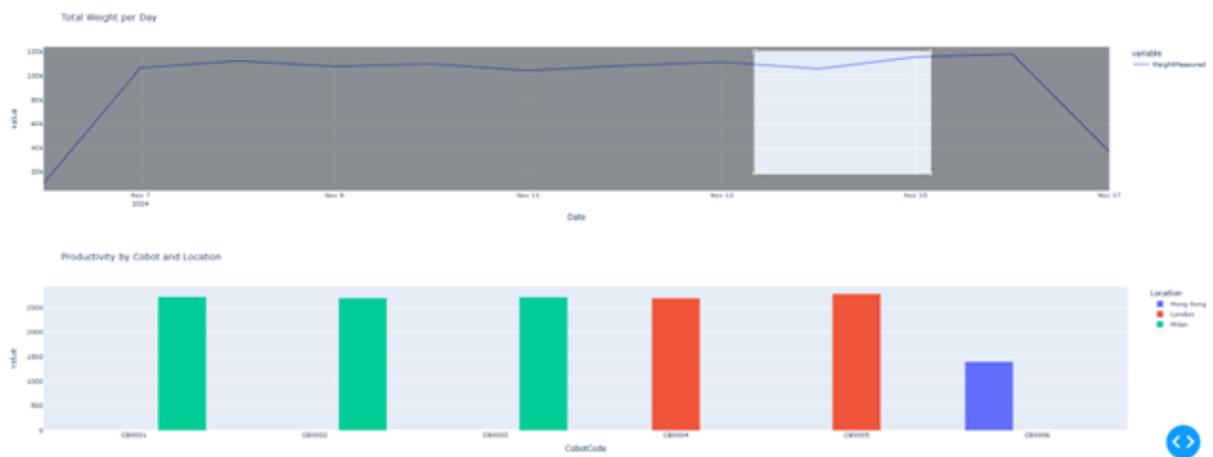


Figure 3.13: Daily total weight and productivity by cobot and location

The described infrastructure provides a solid foundation for implementing advanced analysis modules that could be developed to enhance operational management, a topic that will be discussed in detail in the last chapter.

4 | Experimental tests

This chapter presents the experimental tests conducted to assess the performance of the developed cobot system. The experiments aim to evaluate the effectiveness of the three main algorithms: object detection, predictive maintenance, and optimal speed control, under real operating conditions. Each test has been designed to analyze different aspects of the system, ensuring its ability to operate reliably and safely.

The objectives of the experimental tests are:

1. **to validate the object detection algorithm:** by testing the accuracy in recognizing nuts and washers under varied lighting and positioning conditions.
2. **to assess the predictive maintenance algorithm:** verifying its effectiveness in forecasting potential system failures and reducing downtime by monitoring parameters such as temperature, current, and vibration.
3. **to verify the optimal speed control algorithm:** evaluating its responsiveness and accuracy in adapting motor speed based on real-time load measurements, thus ensuring both performance efficiency and energy savings.

These experiments collectively aim to confirm the viability and robustness of the cobot's design, while identifying any areas for potential improvement. Through this testing phase, key insights into the system's operational capabilities and limitations were gathered, contributing to the overall validation of the cobot's design and control strategies.

4.1. Object detection algorithm validation

To evaluate the performance of the recognition model, eight photos were taken, each containing between 8 and 40 nuts and washers (with two additional different objects in one case). The objects were arranged in various configurations to test the model's robustness in variable situations. The images feature:

- Lighting variations to simulate realistic conditions (image 4 in Figure 4.1 in low-light conditions);

- Objects in different positions, with some clearly visible and others partially overlapping (image 3 in Figure 4.1), nuts placed vertically instead of horizontally (image 8 in Figure 4.2), or high object density (image 7 in Figure 4.1), to test the model's ability to distinguish objects in realistic scenarios.

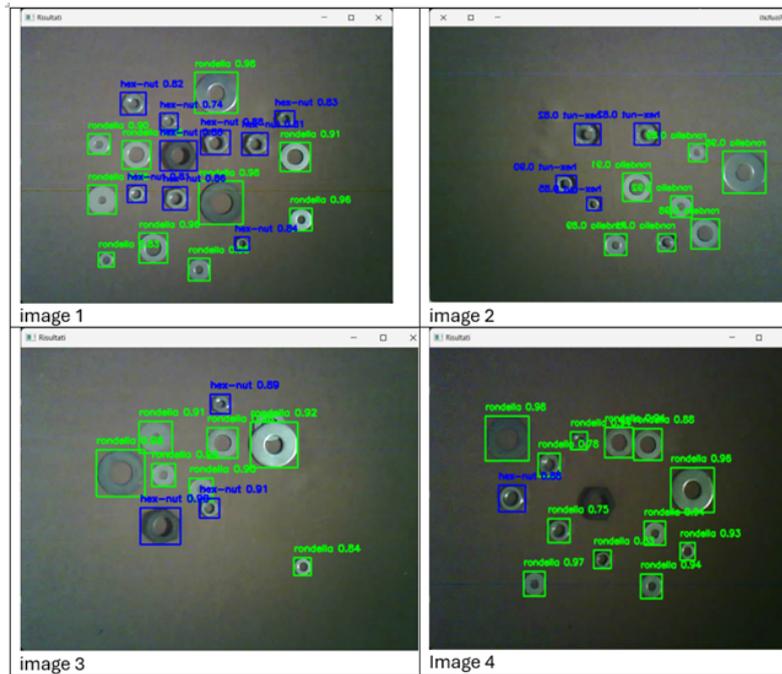


Figure 4.1: Nuts and washers recognition

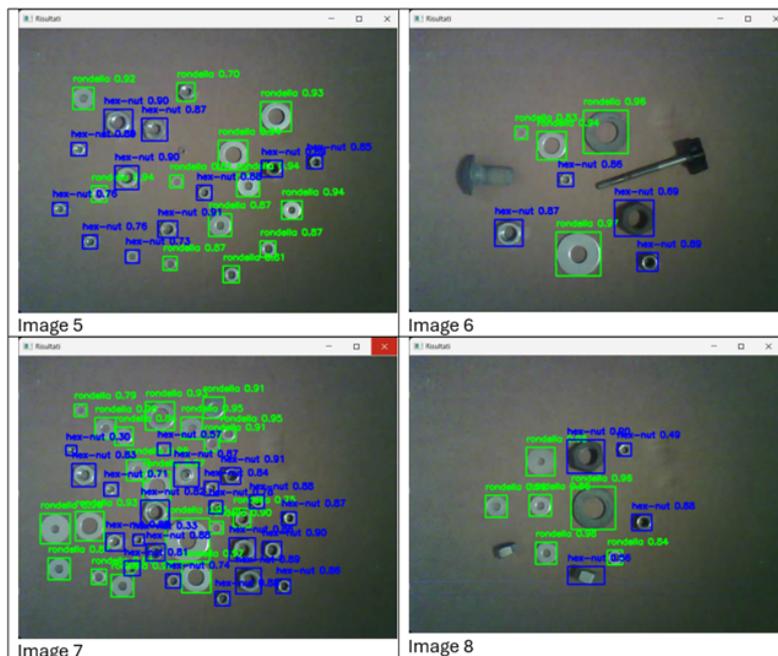


Figure 4.2: Nuts and washers recognition

Image	Real			Detected		
	Objects	Nuts	Washers	Objects	Nuts	Washers
1	19	9	10	19	9	10
2	11	5	6	11	4	7
3	11	3	8	10	3	7
4	14	6	8	13	1	12
5	24	12	12	23	11	12
6	8	4	4	8	4	4
7	40	20	20	40	21	19
8	11	5	6	10	4	6
Total	138	64	74	134	57	77

Table 4.1: Comparison of real and detected nuts and washers

Image	Nuts			Washers			Notes
	True Positives	False Positives	False Negatives	True Positives	False Positives	False Negatives	
1	9	0	0	10	0	0	
2	4	0	1	6	1	0	
3	3	0	0	7	0	1	
4	1	0	5	8	4	0	low-light conditions
5	11	0	1	12	0	0	
6	4	0	0	4	0	0	
7	20	1	0	19	0	1	high object density
8	4	0	1	6	0	0	nuts placed vertically
Total	56	1	8	72	5	2	

Table 4.2: Detection results with notes on specific conditions

4.1.1. Evaluation metrics: precision and recall

For each object class, precision and recall metrics were calculated. The results obtained and the corresponding calculations are shown below.

1. Class: nuts

- True Positives (TP): 56 nuts correctly identified.
- False Positives (FP): 1 object incorrectly classified as nut.
- False Negatives (FN): 8 nut not correctly detected.

Using these values, it is possible to calculate precision and recall for the "Nuts" class.

- **Precision:**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{56}{56 + 1} = \frac{56}{57} = 0.98 \text{ or } 98\%$$

The precision value indicates that the model correctly identifies nuts in 98% of cases, with only one false positive.

- **Recall:**

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{56}{56 + 8} = \frac{56}{64} = 0.87 \text{ or } 87\%$$

The recall value shows that the model is able to identify 87% of the nuts present; false negatives mostly occurred under low-light conditions and in a case where the nut was positioned vertically.

Error analysis: under ideal lighting conditions, the model proved to be nearly flawless. The only false positive occurred in an image with a particularly high object density (40 in total). False negatives mainly occurred in conditions of particularly low light.

2. Class: washers

- True Positives (TP): 72 washers correctly identified.
- False Positives (FP): 5 objects incorrectly classified as washers.
- False Negatives (FN): 2 washers not correctly detected.

Here as well, we can calculate precision and recall for the "Washers" class.

- **Precision:**

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{72}{72 + 5} = \frac{72}{77} = 0.93 \text{ or } 93\%$$

The precision indicates that the model correctly identifies washers in 93% of cases, with false positives primarily due to low lighting conditions (4 out of 5).

- **Recall:**

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{72}{72 + 2} = \frac{72}{74} = 0.97 \text{ or } 97\%$$

The recall demonstrates that the model successfully detected 97% of the washers, with only two cases of false negatives (one of which occurred in a high

object density scenario).

Error analysis: errors were limited, with some false positives mostly due to low lighting conditions.

Confusion matrix for nuts and washers The confusion matrix provides a visual representation of the results for each class:

	Predicted nuts	Predicted washers
nuts	56	5
washers	1	72

Table 4.3: Confusion matrix for nut and washer prediction

Interpretation:

- **True Positives (TP):** 56 nuts and 72 washers were correctly classified.
- **False Positives (FP):** 1 washers was incorrectly identified as a nut.
- **False Negatives (FN):** 5 nuts were incorrectly classified as washers.

These results indicate a good balance between precision and recall, with overall high performance and few classification errors, mainly due to low lighting or unusual object arrangements.

4.1.2. Model considerations and improvement suggestions

Model considerations

The recognition model has shown excellent performance in distinguishing nuts and washers, with high precision and recall values for both classes. However, the error analysis highlights some areas for improvement:

- the model may struggle under low-light conditions, making some objects less recognizable.
- the presence of overlapping objects can reduce overall accuracy.

Improvement suggestions

- **Verification of expected weight:** during the transfer phase, implementing a check of the measured weight against the expected weight for the object type could be beneficial. For instance, a nut, given the same diameter, will inevitably weigh

more than a washer. This additional check would confirm object identification based on weight, enhancing system reliability.

- **Lighting management:** a lighting management system is necessary, especially in environments where light conditions may vary. Controlled and uniform lighting can reduce detection errors caused by shadows and reflections, ensuring greater accuracy in object classification.
- **Lens cleaning:** regular cleaning of the camera lens used for detection is essential. Dust, fingerprints, or other residues on the lens can compromise the quality of the captured images, leading to inaccuracies in object detection and classification. A clean, unobstructed lens ensures better image clarity, reducing the risk of identification errors and maintaining high system performance.

4.1.3. Execution times

The recognition model was tested on a series of 8 images to determine the average execution time required for detecting and classifying nuts and washers in each individual capture. The processing time was measured separately for the two classes, providing a detailed assessment of the system's effectiveness in detecting each type of object.

Nuts						
Image	Objects	Preprocess (ms)	Inference (ms)	Postprocess (ms)	Total (ms)	Average time (ms)
Image 1	9	3	150.1	1	154.1	17.02
Image 2	4	5	162.1	1	168.1	33.53
Image 3	3	2	123.6	2	127.6	42.53
Image 4	1	3	125.2	1	129.2	129.2
Image 5	11	3	133.3	2	138.3	12.36
Image 6	4	39.6	137.3	1	177.9	44.73
Image 7	21	3	150.5	2	155.5	7.36
Image 8	4	3	135.2	2.1	140.3	35.08
Total	57				1191	40.23
Average			148.875			

Table 4.4: Execution times for nuts

Washers						
Image	Objects	Preprocess (ms)	Inference (ms)	Postprocess (ms)	Total (ms)	Average time (ms)
Image 1	10	10.5	231.8	40.9	283.2	28.32
Image 2	7	5	114.5	2	121.5	17.93
Image 3	7	5	136.5	1	142.5	20.79
Image 4	12	4	140.8	2	146.8	12.27
Image 5	12	3.2	151.8	2	157.0	13.02
Image 6	4	6	140.4	2	148.4	37.1
Image 7	19	5	149.1	2	156.1	8.24
Image 8	6	6.4	133.6	1	141.0	23.83
Total	77				1296.5	20.19
Average			162.06			

Table 4.5: Execution time for washers

			IMAGE	
Object Type	Objects	Average time (ms)	Object Type	Average time (ms)
Washers	77	20,19	Washers	162
Nuts	57	40,22	Nuts	149
Total	134	30,21	Total	311

Table 4.6: Average times for nuts and washers

Average result per image:

- Total average time per image: approximately 0.311 seconds (311 milliseconds)
- Average time for washer detection: approximately 0.162 seconds (162 milliseconds)
- Average time for nut detection: approximately 0.311 seconds (311 milliseconds)

These times include the complete processing cycle for each image, from loading to object recognition and classification. *Estimated time per single object for each class:*

- Estimated time per washer: approximately 20.19 milliseconds
- Estimated time per nut: approximately 40.22 milliseconds
- Estimated average time per object: approximately 30.21 milliseconds

These results provide an indication of the model's speed in individually recognizing nuts and washers, reflecting the system's capability to operate under real-world application conditions.

Discussion of results

The model demonstrated the capability to process each image in a short amount of time,

with an average of 311 milliseconds per image. The values obtained for recognition time per single object and for the overall cycle suggest that the model is sufficiently fast to meet the operational requirements of the automation system. The model's overall performance, in terms of accuracy, recall, and execution times, indicates an adequate level of reliability for automated identification applications in industrial environments. With short processing times and high accuracy for both classes, the system meets the required speed and accuracy standards. However, the model's robustness could be affected by significant variations in lighting conditions or by partially occluded objects, highlighting the need for a controlled environment to ensure optimal performance.

4.1.4. Conclusion

The experimental tests demonstrated that the recognition model achieves high levels of precision and recall, making it suitable for automated identification applications in controlled environments. The results are promising, and performance can be further optimized with the suggested improvements.

4.2. Predictive maintenance algorithm validation

In the testing phase of the predictive maintenance algorithm, several fault conditions were simulated to evaluate the model's effectiveness. The following techniques were employed:

1. **Inducing vibration in the cobot:** to simulate mechanical issues, the cobot was manually moved to induce vibrations in the system. This approach allowed for the recording of vibration levels, simulating conditions malfunction.
2. **Opposing resistance to movement:** it was simulated an increase in current by opposing resistance to the movement of the system. This was accomplished by partially blocking the movement mechanism to simulate the load of an excessively heavy object.
3. **Forced temperature increase:** to test the algorithm's response to elevated temperature conditions, a controlled heat source was placed near the temperature sensors. This allowed for the simulation of overheating conditions, enabling the observation of whether the algorithm correctly identified the risk of failure.
4. **Lifting overweight objects:** finally, tests were conducted by lifting objects significantly heavier than the nominal load.

4.2.1. Results of the experimental tests

This subsection presents the results obtained from the application of the predictive maintenance algorithm, based on the fault condition tests executed with parameters forcibly altered by human intervention. The trials confirmed the model's effectiveness in detecting fault conditions and predicting anomalies in the monitored parameters.

Model Accuracy

During the tests, the algorithm demonstrated an overall accuracy of **92%** in correctly predicting fault conditions, based on the simulated anomalies.

Anomaly Detection

The model successfully identified **18** simulated fault events out of a total of **20** programmed events, including overload and temperature increases. Notably, anomalies were detected in real-time, allowing for timely predictions that could be beneficial for maintenance.

Performance under varying conditions

The tests revealed that the algorithm maintained satisfactory performance even under different input conditions, indicating that the model is robust and adaptable to variable scenarios.

Precision, recall, and accuracy

The performance of the predictive maintenance algorithm is evaluated using three key metrics: precision and recall, as aforementioned, along with accuracy.

- **Accuracy:** it measures the proportion of correct predictions out of the total number of samples. It is calculated as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

where:

- **TP** (True Positives): the number of correctly identified faults.
- **TN** (True Negatives): the number of correctly identified normal states.
- **FP** (False Positives): the number of normal states incorrectly identified as faults.
- **FN** (False Negatives): the number of faults not identified.

The following results were obtained from the tests conducted:

- **TP:** 18 (correctly identified faults)

- **TN:** 60 (correctly identified normal states)
- **FP:** 0 (normal states incorrectly identified as faults)
- **FN:** 2 (unidentified faults)

Utilizing these data:

- **Accuracy:**

$$\text{Accuracy} = \frac{18 + 60}{18 + 60 + 5 + 2} = \frac{78}{85} \approx 0.917 \text{ or } 91.7\%$$

- **Precision:**

$$\text{Precision} = \frac{18}{18 + 0} = \frac{18}{18} = 1 \text{ or } 100\%$$

- **Recall:**

$$\text{Recall} = \frac{18}{18 + 2} = \frac{18}{20} = 0.900 \text{ or } 90.0\%$$

These metrics provide a comprehensive evaluation of the algorithm's effectiveness in identifying faults and predicting anomalies in the monitored parameters.

Limitations and considerations

While the results are promising, it was noted that the model exhibits some limitations, such as sensitivity to abrupt variations in input data, which resulted in occasional false negatives under stress conditions. This underscores the need for further research to enhance accuracy in such situations.

4.2.2. Conclusions

The experimental tests have demonstrated that the predictive maintenance algorithm is effective in detecting anomalies and predicting faults in a simulated context. With an accuracy of **92%** and a recall of **90%**, the model yields very satisfactory results that justify further optimizations. However, it is crucial to continue refining the model and validating it further with real data to ensure its reliability and applicability in operational scenarios. These results provide a solid foundation for future implementations and optimizations.

4.3. Optimal speed control algorithm validation

The algorithm is currently capable of calculating the optimal speed with notable precision and efficiency in response to variations in load. This capacity demonstrates the model's effectiveness within the current setup.

Figure 4.3 illustrates the behavior of the cost function across different test scenarios involving varying load torques, specifically with weights from 0g to 2kg. Each curve represents the cost function's response as a function of angular velocity for a given weight. The plot demonstrates how the cost decreases with increasing angular velocity up to a point of optimal efficiency, after which further increases in speed yield minimal changes in cost. The legend highlights the optimal speed for each load, indicating the specific angular velocity at which the cost function reaches its minimum value for each load scenario. This optimal speed varies with load weight, underscoring the relationship between load torque and the most efficient operational speed necessary to minimize the cost function within the system constraints.

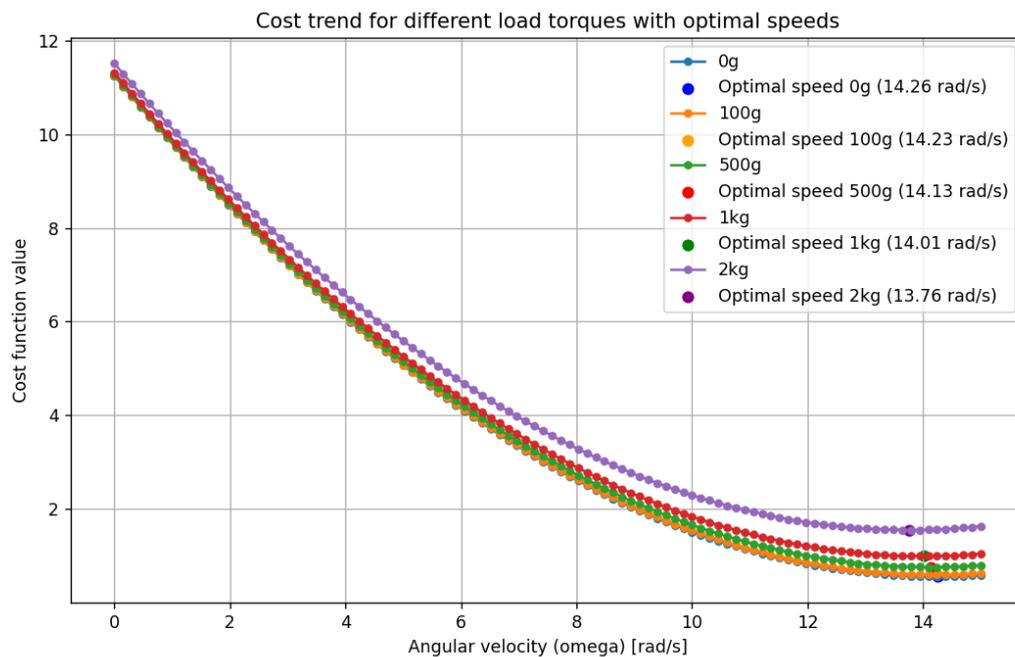


Figure 4.3: Cost trend for different load torques

Figure 4.4 presents the current profiles for three trials where the cobot operates with different load weights: no load, 500g, and 2000g. This analysis demonstrates that, in each case, the transport process is performed without overshoot, and efficiently from an energy perspective. The evaluation is based on a static model that assumes the current profile directly reflects the velocity profile, as shown in Equation (3.2). This relationship allows for the inference of speed behavior through the analysis of current fluctuations. The graph shows consistent current values with minimal oscillations across different load conditions, indicating that the system effectively maintains stable and energy-efficient

operation. These results highlight the robustness of the speed control in adapting to varying transport scenarios.

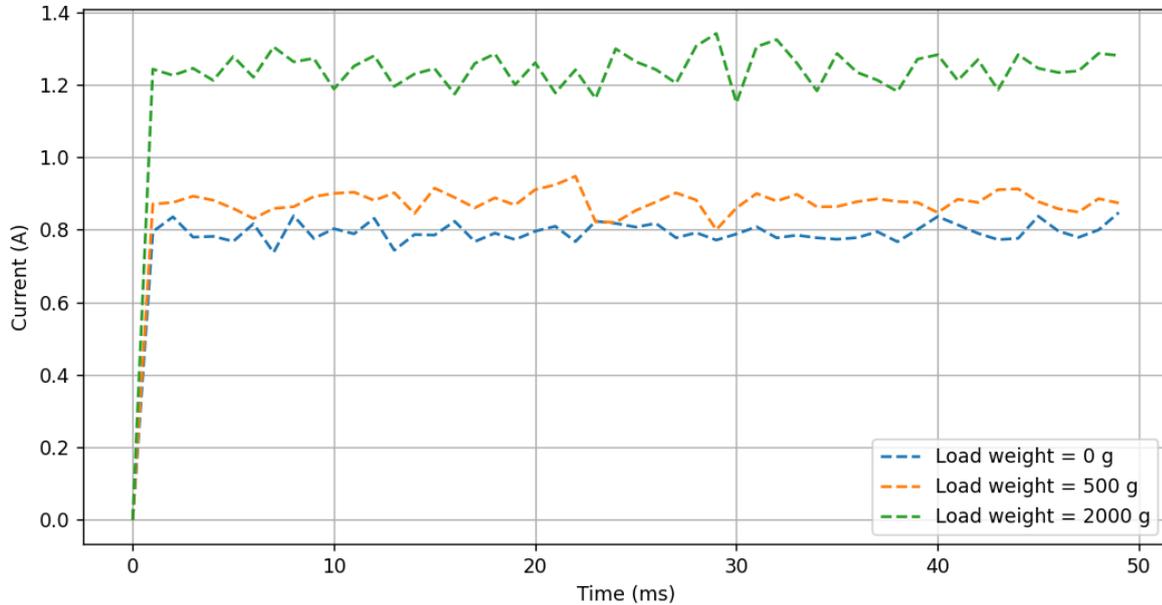


Figure 4.4: Current profiles for different load torques

However, several assumptions and simplifications limit the performance in dynamic, real-world scenarios. For instance, the speed is treated as a constant value, disregarding the influence of external factors such as vibrations and other operational fluctuations. Moreover, the control system operates in open-loop mode, lacking real-time feedback, which restricts its adaptability and resilience under variable conditions. These limitations suggest considerable potential for improvement, particularly in the following areas:

1. **Incorporating a dynamic model and using a more advanced driver:** transitioning from a static to a dynamic model would enable the system to respond more accurately to variations in load and operational disturbances. By selecting a driver that permits direct current control, it would be feasible to implement a dynamic optimal control strategy. This approach would allow the system to adapt its performance in real-time, thereby enhancing responsiveness to load changes and improving overall stability and precision.
2. **Employing a Beckhoff module specifically designed for stepper motor control:** the current system utilizes an EL2008 module, which, while functional, is not optimized for the precise demands of stepper motor applications. Dedicated Beckhoff modules for stepper motors are available and could significantly improve

the system's operational efficiency. These specialized modules would enable the motor to achieve higher speeds with enhanced accuracy and stability, thus broadening the application scope of the system in more demanding or high-speed contexts.

- 3. Integrating an encoder or implementing speed estimation for closed-loop feedback control:** introducing an encoder or adopting advanced speed estimation methods would enable closed-loop control, allowing the system to respond dynamically to fluctuations in load, vibrations, and other external disturbances. With feedback control in place, the system could adjust its performance in real-time, compensating for deviations from the target speed and thus achieving a higher degree of reliability and accuracy. This improvement would address the limitations of the current open-loop configuration by ensuring continuous monitoring and adjustment of the motor's speed, ultimately leading to a more robust and versatile control system.

These enhancements collectively represent a pathway to a more sophisticated and resilient control system, capable of adapting to real-world operating conditions with greater accuracy and efficiency. By addressing the identified limitations, the system could achieve a level of performance more closely aligned with the demands of complex, variable applications, further establishing its viability for industrial and automation contexts.

5 | Future developments and conclusion

This thesis presented the development of a modular collaborative robot designed to enhance efficiency in e-waste recycling within the framework of Industry 5.0. The modular design of the collaborative robot allowed for the integration of various tools, supporting tasks such as disassembly, sorting, and handling of electronic waste. Key innovations included a predictive maintenance algorithm, an object detection algorithm, and an optimal speed control system, each contributing to the adaptability and operational efficiency of the collaborative robot. The predictive maintenance system used machine learning to anticipate potential failures and reduce downtime, while the object detection algorithm accurately identified components in real time to improve sorting precision. Additionally, the speed control system dynamically adjusted motor parameters based on load conditions to optimize energy usage. Experimental validation confirmed the effectiveness of the collaborative robot in enhancing material recovery rates and reducing safety risks for human operators, demonstrating its potential for versatile applications in recycling and other industrial contexts.

While areas for improvement exist, the current system represents a robust and adaptable foundation, showcasing the impactful role that collaborative robotics can play in modern recycling processes and industrial automation. This system's modularity, combined with its capacity for data-driven optimization, illustrates the potential of cobots to streamline workflows, enhance precision, and reduce resource waste in demanding industrial environments.

The system developed in this thesis could be enhanced with some improvements aimed at increasing its versatility, precision, and safety. Below are some potential future developments:

- **Integration of new tools and components:** An electric gripper could replace the current electromagnet, enabling precise handling of a wider range of materials, including non-ferrous components. In fact, the use of an electromagnet presents a few challenges. First, the magnet may struggle to pick up small nuts that are

positioned closely together, requiring adjustments tailored to each specific scenario. Additionally, when the magnet releases an object, there is a tendency for the item to remain attached due to magnetic hysteresis, which can make the release process less consistent. These issues can be mitigated by adapting the magnet as necessary or by opting for an electric gripper instead. Such an upgrade would extend the cobot's capabilities, allowing it to handle diverse tasks such as small-scale assembly or sorting, in addition to its primary recycling applications. In addition to this upgrade, integrating a second ESP32 CAM along the material collection line, just before the cobot, would enable the capture of images of components within the workspace. These images would be processed in real-time by an AI model, which would provide the operator with detailed statistics on the incoming material, including the type, size, and quantity of components. With this information, the operator could give precise instructions to the cobot, which would be thus able to automatically select specific items, optimizing the sorting process and enhancing the overall operational efficiency of the recycling system.

- **Expanding predictive maintenance with real-world data:** as the system operates, an increasing amount of real-world data can be collected, supplementing the initially simulated data used in the predictive maintenance model. With a larger, more diverse dataset based on actual operating conditions, the model can be continuously refined to improve the accuracy of maintenance predictions, adapting more precisely to the system's specific requirements over time.
- **Hardware upgrades and improved sensors:** future iterations of the system could benefit from upgrading existing sensors to more advanced models. For instance, a more sensitive temperature sensor could enhance thermal monitoring capabilities, while improved safety sensors could help detect and respond to potential risks in real time, ensuring safer interactions between the cobot and its environment.
- **Enhancing safety measures:** To improve safety, advanced protocols such as *AI-driven proximity detection* could be implemented to automatically adjust the cobot's movements when operators are nearby. Intelligent safety protocols could allow the cobot to slow down or modify its actions in real-time, aligning with regulatory standards for collaborative robotics and creating a safer working environment.
- **Potential enhancements to production analysis:** as outlined in Chapter 3, the infrastructure described there provides a robust foundation for implementing advanced analysis modules.

1. **Optimization of cobot operational efficiency**

The infrastructure could be enhanced to improve the operational efficiency of cobots by reducing downtime and optimizing production cycles. Key areas include:

- **Productivity retrospective analysis:** analyze historical data to identify inefficiency patterns, such as slowdowns or unnecessary phases, that could be reduced or eliminated. This analysis provides insights to dynamically adjust processing times, optimizing resource utilization.
- **Optimization of operational sequence:** leverage real-time data to manage the sequence of cobot activities based on workload and environmental conditions. This approach improves task allocation, optimally balances load, and reduces cycle times.

2. Integration of models for production quality improvement

Optimization extends beyond productivity to include product quality. A potential evolution of the infrastructure involves integrating systems for production quality control and improvement by leveraging process data to identify critical areas and prevent defects.

- **Advanced traceability system:** develop traceability tools to analyze historical batch data and identify periods or production lines with higher defect rates. This system supports targeted preventive interventions.
- **Statistical Process Control (SPC):** implement statistical control models to monitor critical parameters—such as weight, cycle time, and standard conformity—continuously, with automatic alerts for significant deviations. This approach ensures ongoing quality verification, enabling timely process adjustments.

3. Development of a comparative analysis module across production lines or facilities

For companies with multiple production lines or facilities, the infrastructure could evolve to provide comparative analysis tools, supporting strategic decision-making and large-scale optimization.

- **Production benchmarking:** implement a benchmarking module across production lines or facilities to compare productivity, conformity, and downtime, thereby identifying best practices and areas for improvement.
- **Comparative reporting:** integrate weekly or monthly report generation, highlighting the efficiency and performance of various production lines,

to facilitate information sharing and adoption of optimal solutions across different operational sites.

4. **Optimization of recovery techniques by product type:** integrate historical data analysis within the infrastructure to identify optimal disassembly and recovery methods for each product type (e.g., hair dryer, oven), improving processes for different material categories. Some products may require a specific disassembly sequence or specialized tools, optimizing recovery rates and reducing processing times.
5. **Conformity and quality reporting of recovered materials by batch:** batch data allows for quality control of recovered materials, detecting discrepancies in specific product batches. This analysis helps ensure the quality of recovered components, standardizing control and increasing the value and reliability of materials for subsequent resale or reuse.

This infrastructure for data acquisition and analysis forms a robust foundation for continuous, reliable production monitoring, with the potential to offer valuable operational insights and support real-time strategic decision-making once real data is integrated.

Future enhancements promise to further refine and elevate this cobot system, broadening its versatility and value. With a foundation that supports adaptability and continuous improvement, this system is well-positioned to become an important tool in collaborative robotics, driving innovation and efficiency in industrial automation and recycling. As development continues, this system is poised to become a critical tool across a wide range of applications, underscoring its capacity for continuous innovation and adaptability within the ever-evolving field of collaborative robotics.

Bibliography

- [1] Athanasios Petridis Maija Breque, Lars De Nul. Industry 5.0: Towards a sustainable, human-centric and resilient european industry. *R&I Paper Series, Policy Brief*, 2021. doi: 10.2777/308407.
- [2] Global Market Insights. Global market insights - market research reports & consulting, 2024. URL <https://www.gminsights.com>.
- [3] Amr Adel. Survey on industry 5.0: Potential opportunities, adoption challenges and future research directions. *Journal of Cloud Computing: Advances, Systems and Applications*, 11(40), 2022. doi: 10.1186/s13677-022-00314-5. URL <https://doi.org/10.1186/s13677-022-00314-5>.
- [4] Jakub Pizoń and Arkadiusz Gola. Human-machine relationship—perspective and future roadmap for industry 5.0 solutions. *Machines*, 11(2):203, 2023. doi: 10.3390/machines11020203. URL <https://doi.org/10.3390/machines11020203>.
- [5] Marie Schnell and Magnus Holm. Challenges for manufacturing smes in the introduction of collaborative robots. In *Proceedings of SPS2022*, pages 173–183. IOS Press, 2022. doi: 10.3233/ATDE220137. URL <https://doi.org/10.3233/ATDE220137>.
- [6] Agata Mesjasz-Lech, Ágnes Kemendi, and Pál Michelberger. Circular manufacturing and industry 5.0: Assessing material flows in the manufacturing process in relation to e-waste streams. *Engineering Management in Production and Services*, 16(1): 114–133, 2024. doi: 10.2478/emj-2024-0009.
- [7] Cornelis P. Baldé, Ruediger Kuehr, Tales Yamamoto, Rosie McDonald, Elena D’Angelo, Shahana Althaf, Garam Bel, Otmar Deubzer, Elena Fernandez-Cubillo, Vanessa Forti, Vanessa Gray, Sunil Herat, Shunichi Honda, Giulia Iattoni, Deepali S. Khetriwal, Vittoria Luda di Cortemiglia, Yuliya Lobuntsova, Innocent Nnorom, Noémie Pralat, and Michelle Wagner. *Global E-waste Monitor 2024*. International Telecommunication Union (ITU) and United Nations Institute for Training and Research (UNITAR), Geneva/Bonn, 2024. ISBN 978-92-61-38781-5. URL <https://www.itu.int/itu-d/sites/environment>.

- [8] Yuhao Chen, Yue Luo, Mustafa Ozkan Yerebakan, Shuyan Xia, Sara Behdad, and Boyi Hu. Human workload and ergonomics during human-robot collaborative electronic waste disassembly. In *2022 IEEE 3rd International Conference on Human-Machine Systems (ICHMS)*, pages 1–6, 2022. doi: 10.1109/ICHMS56717.2022.9980828.
- [9] Esther Álvarez-de-los Mozos, Arantxa Rentería-Bilbao, and Fernando Díaz-Martín. Weee recycling and circular economy assisted by collaborative robots. *Applied Sciences*, 10(14):4800, 2020. doi: 10.3390/app10144800. URL <https://www.mdpi.com/2076-3417/10/14/4800>.
- [10] Beckhoff Automation. Beckhoff official website, 2024. URL <https://www.beckhoff.com>.
- [11] DatasheetCafe. 17hs4401 datasheet - 40mm, 2 phase stepper motor, 2024. URL <https://www.datasheetcafe.com/17hs4401-datasheet-stepper-motor/>.
- [12] Alldatasheet.com. Tb6560 stepper motor driver datasheet, 2024. URL <https://www.alldatasheet.com/datasheet-pdf/pdf/535313/TOSHIBA/TB6560.html>. Accessed: 17-10-2024.
- [13] RS Components. Raspberry pi 2 model b datasheet, 2024. URL <https://docs.rs-online.com/790d/0900766b8139232d.pdf>.
- [14] AZ-Delivery. Analog to digital converter ads1115 with i2c interface, 2024. URL <https://www.az-delivery.de/products/analog-digitalwandler-ads1115-mit-i2c-interface?shpxid=c429ac3d-1b95-4866-873b-e928b2388e30>.
- [15] Amazon. Arceli modulo di sensore di corrente e tensione, 2024. URL <https://www.amazon.it/dp/B07379NNS3>.
- [16] Components101. Dht11 temperature and humidity sensor, 2024. URL <https://components101.com/sensors/dht11-temperature-sensor>.
- [17] Geekbuying. Gy-521 mpu-6050 module three axis accelerometer sensor module for arduino raspberry pi avr arm, 2024. URL <https://it.geekbuying.com/item/GY-521-MPU-6050-Module-Three-Axis-Accelerometer-Sensor-Module-For-Arduino-Raspberry.html>.
- [18] Mini ir pyroelectric infrared pir motion human body sensor detector module, 2024. URL <https://www.elecbee.com/it-26020-10pcs-Mini-IR-Pyroelectric-Infrared-PIR-Motion-Human-Body-Sensor-Detector->

- [19] Heschen. Heschen elettromagnete solenoide, 2024. URL <https://www.amazon.it/Heschen-Elettromagnete-Solenoide-diametro-kilogram/dp/B078KSYN1S>.
- [20] Hobby Components. Esp32-cam programming adapter board, 2024. URL <https://hobbycomponents.com/other-dev-boards/1130-esp32-cam-programming-adapter-board>.
- [21] 4GLTE.eu. Teltonika rut955 4g lte router, 2024. URL <https://www.4glte.eu/it/4g-lte-routers/1726-teltonika-rut955-4g-lte-router-0708747432275.html>.

List of Figures

1.1	Industry 5.0 Market Size [2]	2
2.1	1. Raspberry Pi display , showing real-time data for monitoring and control; 2. Server , for data storage and processing; 3. Beckhoff PLC with HMI[10] , which serves as main controller connected to HMI for issuing commands; 4. ESP32 Cam , for capturing images in object detection (washers and nuts); 5. Driver TB6560 , to control motor speed and direction; 6. Raspberry Pi Board , to collect and transmit sensor data to the server; 7. Load cell , providing real-time weight data for speed control; 8. Electromagnet , to lift and move ferrous objects; 9. Various sensors , including temperature, humidity, vibration, human proximity, current, and voltage.	12
2.2	Beckhoff PLC[10]	13
2.3	Stepper motor NEMA17 data sheet[11]	14
2.4	TB650 Driver [12]	15
2.5	Raspberry Pi 2 Model B [13]	16
2.6	ADS1115 [14]	17
2.7	Current and voltage sensors [15]	17
2.8	DHT, temperature and humidity sensor [16]	18
2.9	Load cell [16]	19
2.10	MPU-6050 [17]	19
2.11	PIR sensor [18]	20
2.12	Electromagnet HS-P25X20 [19]	20
2.13	ESP32-CAM [20]	21
2.14	MPU-6050 [21]	21
2.15	System architecture	23
2.16	COBOT control interface	25
2.17	Detected objects overview	26
2.18	Sensor threshold settings	26
2.19	Sensor dashboard with alerts	27

2.20	E-Waste recovery UML diagram	28
2.21	Recovery rule configuration UML diagram	29
2.22	Alert threshold setup UML diagram	30
2.23	Threshold alert UML diagram	31
2.24	Predictive maintenance UML diagram	32
3.1	Instructions interface	34
3.2	Precision-Recall curve for hex-nuts	37
3.3	Precision-Recall curve for washers	37
3.4	Normalized confusion matrix for hex-nuts	38
3.5	Normalized confusion matrix for washers	38
3.6	Nuts detection	39
3.7	Washers detection	39
3.8	Weight distribution	43
3.9	Correlation matrix	43
3.10	Linear regression	49
3.11	Simulation results of the stepper motor model controlled by a simulated driver, showing the phase voltage (V_{ph}), phase current (I_{ph}), torque (T_e), angular velocity (w), and angular position ($theta$).	50
3.12	Execution time and object count per cobot	55
3.13	Daily total weight and productivity by cobot and location	55
4.1	Nuts and washers recognition	58
4.2	Nuts and washers recognition	58
4.3	Cost trend for different load torques	67
4.4	Current profiles for different load torques	68

List of Tables

3.1	Velocities and corresponding currents	49
4.1	Comparison of real and detected nuts and washers	59
4.2	Detection results with notes on specific conditions	59
4.3	Confusion matrix for nut and washer prediction	61
4.4	Execution times for nuts	62
4.5	Execution time for washers	63
4.6	Average times for nuts and washers	63

Ringraziamenti

Vorrei dedicare questo spazio a chi ha reso possibile la realizzazione di questo elaborato e a chi mi è stato costantemente vicino.

Un ringraziamento speciale va al mio relatore, il Prof. Marcello Farina, per aver creduto in questo progetto sin dal primo momento in cui gli è stato presentato. La sua fiducia nell'idea, l'infinita disponibilità e i consigli preziosi sono stati determinanti in ogni fase del lavoro. In un momento in cui non tutti credevano nel potenziale di questo lavoro, il suo sostegno ha rappresentato una spinta fondamentale, permettendomi di raggiungere questo importante traguardo.

Desidero esprimere un sincero ringraziamento ad ALTEN Italia, per avermi dato l'opportunità di svolgere questo progetto di tesi magistrale in collaborazione con la loro realtà. La disponibilità, il supporto tecnico e la professionalità del team hanno rappresentato un contributo fondamentale per il raggiungimento dei risultati ottenuti. In particolare, un grazie speciale va al mio correlatore e tutor, Giuseppe Cannizzaro, per la guida e il costante sostegno durante tutto il percorso. Ci tengo a ringraziare anche tutti i colleghi con cui ho condiviso, anche in piccola parte, questo percorso.

Un grazie dal profondo del cuore va a Ludovica, la mia compagna di vita. Con te ho condiviso ogni passo di questo percorso, dai primi giorni al Politecnico di Milano fino a oggi, un cammino fatto di sfide, sacrifici e conquiste. Sei stata la mia luce e la mia forza nei momenti di difficoltà, in cui ho avuto paura di essere sovrastato. Sei la mia più grande motivazione a crescere e migliorare, non solo come studente, ma soprattutto come persona. Senza il tuo amore, la tua presenza e il tuo sostegno incondizionato, nulla di tutto questo sarebbe stato possibile. Grazie, di cuore, per essere al mio fianco e per credere in me ogni giorno. Questo traguardo è anche tuo.

Desidero esprimere la mia più profonda gratitudine ai miei genitori, per il loro amore incondizionato, il sostegno morale ed economico e per non aver mai smesso di credere in me. Grazie per avermi insegnato l'importanza dell'impegno e della perseveranza. Un ringraziamento speciale va a papà, per aver condiviso con me le sue conoscenze tecniche e per il tempo dedicato a guidarmi e supportarmi anche negli aspetti più pratici del mio percorso. Un grazie altrettanto speciale va a mamma, per essere sempre stata la mia

roccia, un esempio di forza e comprensione. È soprattutto grazie a voi se oggi sono qui. Un ringraziamento speciale va a mia sorella, una presenza fondamentale nella mia vita. Anche se spesso fatico a esprimere ciò che provo, la tua pazienza e il tuo esempio sono stati una guida preziosa per me. La forza del nostro legame va oltre le parole, e quello che sono oggi è anche il frutto di quello che mi hai trasmesso, direttamente o indirettamente, con il tuo modo di essere. Grazie di cuore.

Un sincero ringraziamento va ai miei compagni di corso, Paolo, Guglielmo, Luca, Thomas, con cui ho condiviso questo percorso. La loro presenza è stata fondamentale per affrontare ogni sfida con prospettive diverse e ha reso ogni difficoltà più affrontabile, trasformandola in un'esperienza più leggera e arricchente.

Un ringraziamento speciale va a Ivan, che è stato non solo un compagno di corso, ma anche un punto di riferimento costante. Grazie alla sua dedizione, disponibilità e al suo approccio sempre positivo, ha saputo offrire supporto nei momenti più complessi e rendere il nostro cammino insieme ancora più significativo. La sua capacità di motivare e il suo entusiasmo sono stati un esempio per tutti noi, e non posso che essergli grato. Sono particolarmente felice di poter condividere con lui questo giorno speciale e festeggiare insieme il raggiungimento di un traguardo tanto importante.

Vorrei poi esprimere la mia gratitudine ai miei amici d'infanzia, con cui ho condiviso momenti indimenticabili e un legame che va ben oltre le scelte di vita o i percorsi intrapresi. Anche se la vita ci ha portati su strade diverse, la vostra amicizia è rimasta una costante preziosa, un punto fermo che mi ha sempre sostenuto. La vostra presenza nella mia vita è per me una fonte di forza e serenità.

Infine, un ringraziamento va anche a me stesso, per la determinazione e la forza dimostrate lungo questo percorso. Non è sempre stato facile, ma ho trovato il coraggio di affrontare le difficoltà, superare i momenti di incertezza e continuare a credere nei miei obiettivi. Sono spesso avaro di complimenti verso me stesso e fin troppo severo nel giudicarmi, ma oggi voglio riconoscere con orgoglio l'impegno, la passione e la costanza che mi hanno permesso di arrivare fin qui. Questo traguardo è il frutto di tutto ciò che ho costruito, passo dopo passo, e merita di essere celebrato.