



**POLITECNICO**  
**MILANO 1863**

**SCUOLA DI INGEGNERIA INDUSTRIALE  
E DELL'INFORMAZIONE**

EXECUTIVE SUMMARY OF THE THESIS

## Development of a modular cobot for E-Waste recycling

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

**Author:** FRANCESCO LAUDADIO

**Advisor:** PROF. MARCELLO FARINA

**Co-advisor:** ING. GIUSEPPE CANNIZZARO

**Academic year:** 2023-2024

### 1. Introduction

Industry 5.0 represents a new technological revolution that, rather than merely succeeding Industry 4.0, expands and complements it with a human-centric and sustainable approach. In an increasingly competitive and complex global landscape, this shift emphasizes the need for adaptation and innovation, fostering a holistic view that considers both technological and social impact. Industry 5.0 aims to transition from a paradigm focused solely on shareholder value to one that respects individual rights, prioritizes environmental sustainability, and fosters human-machine collaboration. Key technologies driving this change include human-machine interaction, bio-inspired materials, digital twins, AI, and renewable energy solutions, which together provide a foundation for intelligent, adaptive, and ethical industrial environments, as reported in [1].

Machines play a fundamental role in this revolution, especially regarding their relationship with humans, often framed within the "5C" model presented in [2]: Coexistence, Cooperation, Collaboration, Compassion, and Co-evolution. This new industrial approach enables intelligent systems to work closely with humans, exempli-

fied by collaborative robots (cobots) designed to assist in various sectors, including e-waste management. Cobots can operate safely alongside humans, enhancing productivity, safety, and adaptability in complex environments. By integrating automation with human skills, Industry 5.0 aims to create more resilient and flexible workflows across diverse industries.

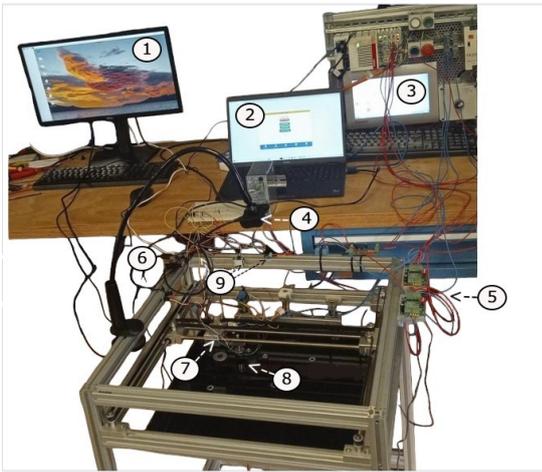
Cobots play a crucial role in circular e-waste management, taking on repetitive or hazardous tasks to optimize resource recovery and reduce human exposure to risks.

This thesis presents a modular cobot system aimed at improving e-waste recycling efficiency and safety. It explores the current challenges in cobot technology—such as safety in shared human-robot workspaces, performance, cost-effectiveness, and smart technology integration. The project focuses on designing a cobot with adaptable hardware that can handle various recycling tasks, supported by real-time control and predictive maintenance algorithms. This design allows for versatile tool integration, enabling the cobot to respond flexibly to different operational needs. Structured into chapters on system architecture, algorithms, testing, and future prospects, the thesis offers a scalable approach to collaborative robotics with applications be-

yond e-waste.

## 2. System architecture

Figure 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 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.

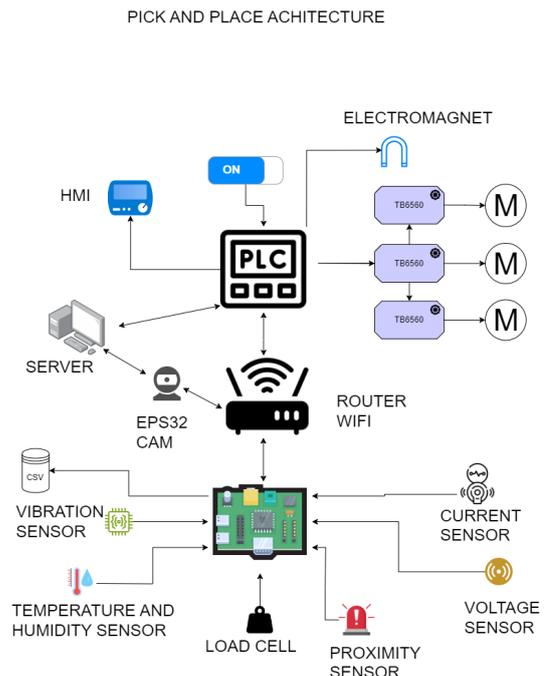


**Figure 1: 1. Raspberry Pi display**, showing real-time data for monitoring and control; **2. Server**, for data storage and processing; **3. Beckhoff PLC with HMI**[3], which serves as main controller connected to HMI for issuing commands; **4. ESP32 Cam**, for capturing images in object detection (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.

In Figure 2, the overall system architecture is presented in a schematic format, providing a representation of the connections and interactions between all components. All components are interconnected via a local area network

(LAN), facilitating seamless communication and data exchange. The Raspberry Pi board collects data in real-time from various sensors, such as those measuring temperature, weight, and vibration. These data points are continuously written to a shared folder accessible across the LAN, allowing for instantaneous access by the PLC and other devices on the network. The shared area, not only supports real-time data monitoring, but also enables functions such as motor speed optimization and data analysis for predictive maintenance.

The system includes a web-based HMI accessible from devices connected to the LAN. This HMI enables the operator to set monitoring thresholds for key operational parameters—such as current, voltage, vibration, weight, humidity, and temperature—triggering alerts when limits are exceeded. The operator can also define criteria for selecting objects to be recovered by configuring dimensional ranges, assigning batch codes, and organizing operations to support traceability and predictive maintenance analysis. Additionally, the HMI allows manual control to start or stop the system for immediate intervention, displays real-time sensor data to facilitate anomaly detection, and shows detected objects along with their diameter and conformity status.



**Figure 2: System architecture**

The system architecture is designed to ensure continuous data flow and high adaptability across various operational scenarios. The central server functions as a data hub, consolidating all essential information for system operation in a single location. Specifically, it stores and manages key files, including the user instruction file, which contains commands provided by the operator through the HMI; the image recognition results file, which records detected object values for automated retrieval by the PLC; production data, encompassing operational parameters collected from the cobots' PLCs; and sensor data, updated in real-time by the Raspberry Pi board, covering critical measurements such as vibration, temperature, current, and voltage. Upon detecting objects, the PLC instructs the cobot to process them and then logs the results in the production file. This centralized configuration enables real-time data accessibility, allowing each component to continuously update information on the server. This setup facilitates live monitoring and timely adjustments as needed, ensuring system responsiveness and efficiency in automated production environments.

### 3. Develop algorithms

In this thesis, four algorithms have been developed, each of which will be detailed in this section to illustrate its purpose, implementation, and role within the overall system architecture.

#### 3.1. Object detection algorithm

The object detection algorithm developed for the cobot system leverages an ESP32-CAM and custom-trained YOLO (You Only Look Once) models to identify and classify washers and nuts with high accuracy in real-time. This algorithm is designed to integrate seamlessly with a web-based HMI, which allows operators to configure detection criteria, such as the type of object (washer or nut), and set acceptable diameter ranges. Through this interface, operators can ensure only objects meeting specified dimensions and other compliance standards are processed, providing a customized approach for each batch. Images captured by the ESP32-CAM are processed by the algorithm, which performs various calculations to assess object geometry. Key supporting functions include overlap detection to reduce interference between objects, and geo-

metric measurements to determine each object's center, area, and diameter in millimeters, using a scaling factor based on the camera's field of view and distance. Each object detected is evaluated against the criteria set in the HMI, filtering out those that do not meet these specifications. Once detected, each object's type, center coordinates, area, diameter, and compliance status are recorded in a structured CSV file. This data format allows for streamlined tracking and future analysis of detected objects, as well as integration with other production processes. The detection process operates in a continuous loop managed by a server-side script that responds to commands issued from the HMI. The script monitors a file that signals when to initiate detection; once the process is complete, the system resets for the next request. This setup enables precise, on-demand object detection cycles, optimizing workflow efficiency and facilitating control over production timing.

#### 3.2. Predictive maintenance algorithm

The predictive maintenance algorithm monitors critical parameters, including current, voltage, vibration, humidity, temperature, and load weight, using a Random Forest model for classification. Initially trained on simulated data, the model is designed to transition to real sensor data to progressively enhance its accuracy. Each time the Raspberry Pi adds new data from the connected sensors, the algorithm immediately analyzes these inputs, allowing for continuous, real-time monitoring. This constant interaction ensures that the model is consistently updated with the latest operational data, enabling proactive detection of potential maintenance needs. The processed sensor data is recorded into a dedicated file within the shared LAN folder, to which the algorithm appends a specific column indicating whether maintenance is necessary based on current conditions. This information is accessible across the network, facilitating easy retrieval for real-time monitoring and timely maintenance alerts, ultimately reducing downtime and enhancing system performance. Periodic retraining further aligns the model with evolving operational conditions, keeping the predictive capabilities robust and adaptable.

### 3.3. Speed control algorithm

The speed control algorithm is derived from the motor model, represented by the equation

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

where the angular velocity  $\omega$  is influenced by parameters such as the motor current, load torque  $T_l$ , and viscous friction. To facilitate real-time optimization, a static model is employed, which simplifies the dynamics by assuming a steady-state relationship between speed and load torque, independent of time. This approach allows the algorithm to focus on achieving an immediate balance between speed and load torque, streamlining the computation process. The static model is derived from the full dynamic equations, focusing on the immediate relationship between speed, load, and current, while bypassing transient dynamics. By neglecting time-dependent fluctuations, this approach reduces computational overhead and enhances stability within operational limits. Most parameters in this model, including  $K_m$  and  $J$ , were obtained directly from the motor's datasheet, while the viscous friction coefficient  $K_v$  was estimated experimentally to reflect real-world operating conditions. The total current  $I$ , representing the resultant current from both phases, is calculated as:

$$I = \sqrt{i_a^2 + i_b^2}.$$

This current is then related to the total driving torque  $T_e$ , which includes the load torque  $T_l$  and the viscous friction torque  $K_v \cdot \omega$ , by the equation:

$$I = \frac{T_l + K_v \cdot \omega}{K_m}.$$

The Raspberry Pi, equipped with a load cell, continuously monitors the weight of the object being lifted and records this information in a shared file within the LAN directory. The optimization script accesses this file in real-time to retrieve the load weight as an input for calculating the optimal motor speed. This weight data is converted into load torque, enabling the algorithm to determine the optimal motor speed by minimizing a predefined cost function.

The cost function used in the optimization process is given by:

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

where  $\lambda_1$  and  $\lambda_2$  are weighting factors that balance the importance of minimizing the deviation from the maximum allowable speed  $\omega_{\max}$  and the current  $I$  used by the motor. The computed speed is then translated into a STEP pulse interval for the PLC, which dynamically adjusts the motor speed according to load variations, ensuring efficient and responsive operation under varying conditions.

### 3.4. Data acquisition and analysis infrastructure

A real-time infrastructure has been established to monitor and optimize cobot operations within an automated environment, ensuring efficient and responsive control over various tasks. Data from distributed PLCs, each responsible for different aspects of the cobot's operations, is collected by a centralized server where it is systematically stored in a standardized CSV format. This comprehensive dataset includes critical information such as execution time, weight, motor speed, and environmental parameters like temperature and humidity. The standardized format allows for seamless integration and analysis, supporting a continuous feedback loop that enhances system performance.

To further facilitate operational insights, an interactive dashboard has been implemented, allowing operators to track productivity metrics, analyze historical data, and identify trends or potential inefficiencies across multiple locations. The dashboard provides a real-time view of system status and performance, enabling operators to make data-driven decisions and quickly respond to any issues. This infrastructure not only improves transparency and oversight but also supports predictive maintenance by identifying early warning signs of potential failures, thereby minimizing downtime and optimizing cobot functionality across the entire automated network.

## 4. Experimental tests

This section presents the experimental tests conducted to thoroughly evaluate the system's overall performance. The tests were designed to validate multiple aspects of the system, including the accuracy of the object detection algorithm in identifying and differentiating washers and nuts, the predictive maintenance algorithm's effective-

ness in forecasting potential failures before they occur, and the optimal speed control algorithm's responsiveness to variations in load weight. By addressing these key areas, the tests aimed to confirm the system's efficiency, reliability, and adaptability under real-world operational conditions, ensuring that each component functions cohesively within the integrated robotic framework. The results provide a comprehensive assessment of the system's capabilities and highlight areas for further refinement and improvement.

#### 4.1. Object detection algorithm validation

The experimental results for the object detection algorithm demonstrated high accuracy in identifying washers and nuts, showcasing its reliability in standard operational conditions. Specifically, under typical lighting conditions and common object placements, the algorithm achieved impressive precision and recall rates, reaching nearly 98% for nuts and 93% for washers. However, in certain more challenging scenarios, such as low lighting, overlapping objects, and high object density, the algorithm encountered minor detection issues, resulting in some false positives and false negatives. These conditions highlighted potential areas for improvement, particularly in handling complex lighting setups and enhancing object separation algorithms.

Figure 3 illustrates an example of detection performance under high-density conditions, where multiple objects are closely spaced. Despite these challenges, the algorithm showed robust and consistent performance, suggesting that targeted refinements in lighting control and spatial separation could further boost its accuracy and reliability in demanding environments.

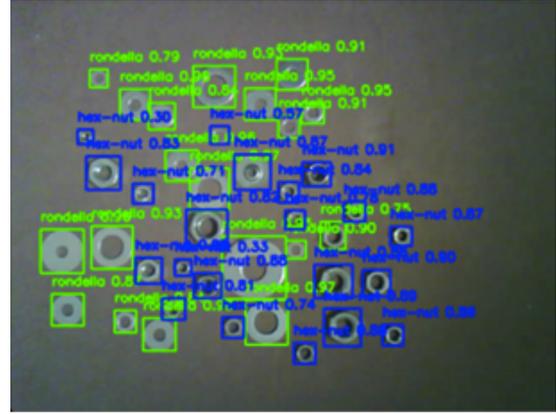


Figure 3: Nuts and washers recognition

#### 4.2. Predictive maintenance algorithm validation

The predictive maintenance algorithm demonstrated strong performance, achieving an accuracy of 92% in identifying fault conditions and successfully detecting 18 out of 20 simulated anomalies, including instances of overloads, temperature spikes, and other stress-related faults. The model exhibited excellent real-time responsiveness, allowing for timely maintenance alerts and ensuring robust performance across a range of varied operating conditions. This capability not only supports preventive actions but also minimizes potential downtime, contributing to overall system reliability and efficiency. However, minor limitations were observed when operating under extreme or borderline conditions, where the model occasionally missed or misclassified faults. These observations highlight specific areas for future refinement, with a focus on enhancing the algorithm's sensitivity and accuracy in detecting anomalies under high-stress scenarios. Such improvements could further solidify the predictive maintenance system's effectiveness in complex, demanding environments.

#### 4.3. Speed control algorithm validation

The speed control algorithm effectively calculated the optimal motor speed in response to real-time load variations, resulting in significant energy savings and highly responsive adaptation to changes in load weight. This responsiveness ensures that the motor operates efficiently, adjusting its speed dynamically to match the current load, thereby optimizing both performance and energy consumption.

Figure 4 illustrates the behavior of the cost function and optimal speed as weight changes, demonstrating the algorithm’s efficiency in balancing operational demands with minimal energy use. The results underscore the algorithm’s effectiveness in maintaining smooth operation across a range of loads.

While these initial results are promising, future enhancements could focus on incorporating dynamic feedback integration, enabling the algorithm to adapt to more rapid or unexpected fluctuations in load. This would improve system stability and further refine energy usage, especially under highly variable operating conditions. Additionally, incorporating advanced sensor data and real-time monitoring could allow the speed control algorithm to continuously fine-tune performance, leading to even greater reliability and efficiency in complex environments.

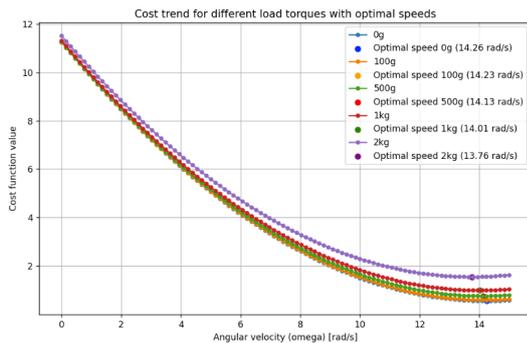


Figure 4: Cost trend for different load torques

#### 4.4. Overall analysis of experimental results

The experimental tests demonstrated the system’s overall effectiveness and readiness for practical applications. Each algorithm consistently met performance expectations, validating the cobot’s ability to operate reliably across a variety of conditions. The object detection, predictive maintenance, and speed control algorithms each contributed to enhancing the system’s adaptability, accuracy, and responsiveness, ensuring smooth and efficient operations. The results confirm that the system is well-suited for handling complex tasks autonomously, with minimal need for operator intervention. Minor limitations, observed in more challenging test scenarios, highlight potential areas for further refinement, particularly in adapting the

system to high-stress conditions or unexpected variations.

In summary, these tests underscore the robustness and reliability of the developed cobot architecture, supporting its potential to improve productivity and safety in automated recycling environments.

## 5. Future developments

Future developments could include integrating an electric gripper, allowing the cobot to handle a wider variety of materials, including non-ferrous components, which would expand its flexibility in sorting and assembly tasks. Adding an additional ESP32 CAM for enhanced monitoring could also provide real-time data to improve item selection and sorting efficiency. Further, refining the predictive maintenance model with real-world data would enable more accurate and timely scheduling, while upgrading sensors could enhance safety and responsiveness. Lastly, advanced data analysis could offer valuable insights into productivity, making the system more adaptable and efficient across a range of industrial applications.

## 6. Conclusion

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.

## References

- [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.
- [2] Jakub Pizoń and Arkadiusz Gola. Human-machine relationship—perspective and future roadmap for industry 5.0 solutions. *Machines*, 11(2):203, 2023.
- [3] Beckhoff Automation. Beckhoff official website, 2024.