

POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering



**Politecnico
di Torino**

Master Thesis

Design and implementation of a zero errors Automated Servo-System for skateboard manufacturing, using collaborative robotics and AGV integration

Supervisor: Dario ANTONELLI

Assistant Supervisor: Paolo BRIZZI

Francesco SUCCURRO

Identification number 280004

Academic Year 2022/2023

Abstract

The manufacturing sector accounts for one-fifth of Italy's gross national product, and has always been a spearhead of excellence in our economic system. The advent of the Industry 4.0 technologies, represents an opportunity to revitalize the entrepreneurial fabric of our country, which is mostly made up of small and medium-sized companies. The challenge for Italian industry is to guide and accompany SMEs in the process of digital transition to a smart, connected and sustainable factory, based on the integration of robotic process automation systems and Big Data & Analytics solutions and techniques, in order to manage products, processes and resources more intelligently. The goal is to make companies more productive, flexible, competitive, in order to make them global, regardless of their size, and increasingly projected into international markets.

In this context, this master's thesis aims at developing an automated servo-system for a skateboard production line.

The line consists of two stations:

- a collaborative robotics one, which includes a Fanuc CRX-10iA cobot mounted on a cart and a robot-mounted vision system. The station is intended to pick up the components of a skateboard such as wheels, trucks, and deposit them in special trays arranged on the Mizubot 4.0 AGV from New Value Group company. The AGV is used to transport components from this station to the next one;
- a final assembly station where there is the operator, who, once the components arrive, will assemble them to make the skateboard.

To do this, new gripping fingertips were investigated to pick up all skateboard components in a stable manner, even though the grip did not perfectly fit all objects equally. The new jaws were designed through SolidWorks, a 3D modelling software. Then, the components were printed in PLA through additive manufacturing with the UltiMaker S5 printer. Similarly, special trays have been designed and manufactured, to be placed on the AGV channels, ready to accommodate the placement of skateboard components.

After that, the robot was programmed so that:

- it could recognize and pick up, through the vision system, all the different skate components, regardless of how they were positioned;
- it could deposit the components into custom made trays on the AGV, which were free to be positioned along the channels at each production cycle, but also free to slide along the channels once the components were placed upon them. The correct placing of the components by the robot was guided by the vision system, which searched for appropriate markers placed on the trays, that could provide an offset on position and orientation.

Teach pendant, the HMI device of the robot, and Fanuc programming language were used for the programming part.

The last phase focused on AGV programming. It is important to know that the Mizubot 4.0 consists of a MiR 200 mobile robot, on which a physical structure with a work surface is integrated, custom made for this type of application. In this way it is possible, using the same core, to create a system that can be adapted according to the type of materials it has to transport and of the delivery of the material on the unloading station, changing only its physical structure. Specifically, programming focused first on creating a predetermined pathway, which the MiR was able to recognize independently, to go alongside the final assembly station. In the last step, using the software created by New Value Group, some parameters regarding the AGV's unloading channels were configured in order to appropriately calibrate the push paddles that unloaded the trays onto the final station.

The results were a successful implementation of the entire skateboard production line, resulting from a proper integration between the collaborative robotics station and the AGV. In particular, the technological challenge of depositing the components inside the trays with millimeter precision was overcome, even though the trays could vary their position along the channels at each cycle. In addition, it was verified that the application worked the same way, even if the robot was moved slightly from its original location. This line demonstrates that smart automation of this type can simplify processes that may

be prone to human errors, thereby improving defect detection and providing a higher quality product to customers, but also a lower cost to the company.

Summary

Introduction	1
1.1 Industry 4.0	1
1.2 Smart Factory	4
1.2.1 System Architecture	4
1.3 Lean Smart Production	6
1.3.1 Pull Logic	7
1.3.2 Takt time	7
1.3.3 One piece flow	8
1.3.4 Zero defects	9
1.4 Competence Center	9
1.5 Thesis objective	11
1.6 Organization of the chapters	12
Chapter 2 – Robotics: general concepts.....	15
2.1 Industrial robotics	15
2.1.1 The robot	16
2.1.1.1 Mechanical structure	17
2.1.1.2 Control unit.....	18
2.2 Robot types	19
2.2.1 Cartesian robots.....	19
2.2.2 Cylindrical robots.....	20
2.2.3 Spherical robots.....	21
2.2.4 Scara robots	22
2.2.5 Anthropomorphic robots	23
2.2.6 Collaborative robots	24
Chapter 3 – Additive Manufacturing	27
3.1 Basic principles.....	27

3.2 AM Process.....	29
3.2.1 CAD	30
3.2.2 Conversion to STL	30
3.2.3 STL File Manipulation and transfer to AM machine	31
3.2.4 Machine setup	32
3.2.5 Build.....	32
3.2.6 Removal	33
3.2.7 Post-processing.....	33
3.2.8 Application.....	33
Chapter 4 – Case study.....	35
4.1 Overview.....	35
4.2 Robot gripping system	36
4.2.1 Gripper selection	37
4.3 CAD modelling.....	40
4.3.1 Interface plate.....	40
4.3.2 Trays.....	43
4.3.2.1 Wheels tray	43
4.3.2.2 Trucks tray	44
4.3.2.3 Boards tray.....	44
4.3.3 Custom fingertips	45
4.4 Realization of components by Additive Manufacturing.....	48
4.4.1 Printing of the components	51
4.4.2 Coating of the new gripper fingertips	54
4.5 Implementation of the line	56
4.6 Configuration of the collaborative robotics station	57
4.6.1 Collaborative robotics station.....	57
4.6.1.1 Robot CRX-10iA.....	57
4.6.1.2 Tablet Teach Pendant	59
4.6.1.3 R-30iB Mini Plus controller	60
4.6.1.4 iRVision 2D system.....	61
4.6.2 Payload settings.....	63
4.6.3 Creation of the frames	67
4.6.3.1 User Frames.....	67

4.6.3.2 Tool Frame	71
4.6.4 Calibration vision system	72
4.6.5 Robot Program creation	80
4.6.5.1 Main Program	81
4.6.5.1.1 Trucks program	81
4.6.5.1.2 Wheels program	96
4.6.5.1.3 Board program	103
4.7 Mizubot 4.0 programming	107
4.7.1 Introduction	107
4.7.2 Mission of the AGV	111
4.7.2.1 Map creation	113
4.7.2.2 Mission Creation	114
4.7.3 Configuration of the unloading channels	117
Chapter 5 – Conclusions and future developments	121
References	125

List of figures

Figure 1.1: 4.0 Key Enabling Technologies.....	3
Figure 1.2: System Architecture of the Smart Factory	4
Figure 1.3: Smart factory software	6
Figure 1.4: 4 JIT principles.....	7
Figure 1.5: Pull System	7
Figure 1.6: Variation in demand.....	8
Figure 1.7: One piece flow	9
Figure 1.8: Competence Centers in Italy	10
Figure 1.9: Partners CIM 4.0.....	10
Figure 1.10: Pilot Line “Digital Factory”	11
Figure 1.11: Pilot Line “Additive Manufacturing”	11
Figure 2.1: Mechanical structure of a robot.....	16
Figure 2.2: Control unit of a robot.....	16
Figure 2.3: Schematic drawing of the main constituent elements of a robotic manipulator.....	17
Figure 2.4: Revolute joint	18
Figure 2.5: Prismatic joint	18
Figure 2.6: Tool frame.....	18
Figure 2.7: Example of cartesian robot.....	20
Figure 2.8: Scheme of the cartesian robot	20
Figure 2.9: Example of a cylindrical robot.....	20
Figure 2.10: Scheme of the cylindrical robot	20
Figure 2.11: Example of a spherical robot.....	21
Figure 2.12: Scheme of the spherical robot.....	21
Figure 2.13: Example of a scara robot.....	22
Figure 2.14: Scheme of the scara robot	22
Figure 2.15: Example of an anthropomorphic robot	23

Figure 2.16: Scheme of the anthropomorphic robot.....	23
Figure 2.17: Example of a collaborative robot	24
Figure 2.18: Robot-operator collaboration.	24
Figure 3.1: Images showing the CAD model of a component and the printing effects of using different layer thicknesses	28
Figure 3.2: AM process..	29
Figure 3.3: An original CAD model converted into an STL file using different offset height (cusp) values	31
Figure 3.4: Supports generated for printing the component	32
Figure 4.1: Overview of the skateboard production line	35
Figure 4.2: Parallel Gripper	36
Figure 4.3: Self-centring gripper	36
Figure 4.4: Radial gripper.....	36
Figure 4.5: OnRobot RG6 gripper	38
Figure 4.6: RG6 Datasheet	39
Figure 4.7: RG6 work range	39
Figure 4.8: End effector interface	40
Figure 4.9: Quick-changer for I/O – Robot side.....	41
Figure 4.10: 2D iRVision camera package.....	41
Figure 4.11: Gripper side interface plate	42
Figure 4.12: Robot side interface plate.....	42
Figure 4.13: assembly drawing of the gripper with the camera	42
Figure 4.14: CAD model of the wheels tray.....	43
Figure 4.15: CAD model of the trucks tray	44
Figure 4.16: CAD model of the trucks tray	45
Figure 4.17: Wheels.....	45
Figure 4.18: Trucks.....	45
Figure 4.19: Board.....	46

Figure 4.20: Outer part of the gripper jaw	46
Figure 4.21: Inner part of the gripper jaw	46
Figure 4.22: Technical drawing of the hooks for the gripper jaws.....	47
Figure 4.23: UltiMaker S5	48
Figure 4.24: Printing simulation of the gripper jaws	49
Figure 4.25: Rear view of the printing simulation of the gripper jaws, with their inner part resting on the printing plate.....	50
Figure 4.26: Front view of the printing simulation of the gripper jaws, with their inner part resting on the printing plate.....	50
Figure 4.27: Rear view of the printing simulation of the gripper jaws resting on its side	51
Figure 4.28: Front view of the printing simulation of the gripper jaws resting on its side	51
Figure 4.29: Printing stage of the new gripper fingertips.....	51
Figure 4.30: New gripper fingertips mounted on the gripper	52
Figure 4.31: Printing simulation of the interface plate	52
Figure 4.32: Interface plate mounted on the end effector of the robot.....	52
Figure 4.33: Printing simulation of the wheels tray	53
Figure 4.34: Printed wheels tray.....	53
Figure 4.35: Printing simulation of the boards tray	53
Figure 4.36: Printed boards tray	53
Figure 4.37: Printing simulation of the trucks tray.....	54
Figure 4.38: Printed trucks tray	54
Figure 4.39: Adhesive neoprene sheet.....	55
Figure 4.40: Detail of the grip fingertip coated with the adhesive neoprene	55
Figure 4.41: The collaborative robotics station and the MIZUBOT 4.0	56
Figure 4.42: Assembly station	56
Figure 4.43: Collaborative robotics station	57
Figure 4.44: Kinematics of the robotic arm.....	57
Figure 4.45: Fanuc CRX-10iA collaborative robot	58
Figure 4.46: Tablet Teach Pendant.....	59
Figure 4.47: R-30iB Mini Plus controller.....	60

Figure 4.48: Elements of the iRVision 2D system	61
Figure 4.49: Camera in fixed position	61
Figure 4.50: Camera mounted on the robot	62
Figure 4.51: Specifications of the Fanuc iRVision 2D system.....	63
Figure 4.52: Reference system placed on the flange centre	63
Figure 4.53: Coordinate system on the counter flange	64
Figure 4.54: Mass properties of the system including interface plate, gripper, and camera.....	64
Figure 4.55: Payload sheet number 5	65
Figure 4.56: Payload sheet number 6, 7, 8.	66
Figure 4.57: Calibration grid.	67
Figure 4.58: Tip of the Tool Master	68
Figure 4.59: Tailstock of the Tool Master	68
Figure 4.60: Four different orientations in which the tip of the Tool Master touches the tailstock.....	68
Figure 4.61: Calibration grid on the cobot cart.....	69
Figure 4.62: Calibration grid on the AGV.....	69
Figure 4.63: Overview of all User Frames	70
Figure 4.64: User Frame 1 – Banchetto.....	70
Figure 4.65: User Frame 2 – AGVschermo.....	70
Figure 4.66: User Frame 3 – AGVfondo	70
Figure 4.67: TCP calculator file	71
Figure 4.68: TCP offset	71
Figure 4.69: Tool frame 1	72
Figure 4.70: A calibration grid	73
Figure 4.71: Calibration grid 45mm×45mm.....	74
Figure 4.72: Calibration grid 15mm×15mm.....	74
Figure 4.73: Calibration grid 20mm×20mm.....	74
Figure 4.74: Calibration on the robot cart	74
Figure 4.75: Camera setup.....	75
Figure 4.76: Calibration setup	75
Figure 4.77: Calibration grid location	76

Figure 4.78: Calibration.....	76
Figure 4.79: Capture of the calibration grid image.....	77
Figure 4.80: Grid calibration found.....	77
Figure 4.81: Calibration points.....	78
Figure 4.82: CALIBRAZIONEBANCHETTO	78
Figure 4.83: CALIBRAZIONEAGVLATOSCHERMO	79
Figure 4.84: CALIBRAZIONEAGVTAVOLESKATE.....	79
Figure 4.85: Main program.....	81
Figure 4.86: FINDPICKPLACETRUCK program.....	82
Figure 4.87: Position register.....	83
Figure 4.88: Banquet of the cobot on which the trucks, wheels, and board were located	83
Figure 4.89: Order in which the photos were taken.....	84
Figure 4.90: CERCATRUCK vision function.....	85
Figure 4.91: GPM Locator Tool for the CERCATRUCK vision function.....	86
Figure 4.92: Reference position of the CERCATRUCK vision function	88
Figure 4.93: Truck pick phase	89
Figure 4.94: SVINCOLOTRUCK program.	90
Figure 4.95: Midpoint between the cobot cart and the AGV	90
Figure 4.96: SCARICOTRUCKSDEF program	91
Figure 4.97: Marker	92
Figure 4.98: Trucks tray with the marker applied	92
Figure 4.99: P[2] Snap Position.....	92
Figure 4.100: VASSOIOTRUCKS vision function	93
Figure 4.101: GPM Locator Tool for the VASSOIOTRUCKS vision function	93
Figure 4.102: Reference position of the VASSOIOTRUCKS vision function	94
Figure 4.103: Truck place phase.....	95
Figure 4.104: FINDPICKPLACERUOTA program	96
Figure 4.105: CERCARUOTA vision function	97
Figure 4.106: GPM Locator Tool for the CERCARUOTA vision function	97
Figure 4.107: Reference position of the VASSOIOTRUCKS vision function	98
Figure 4.108: Wheel pick phase	98

Figure 4.109: SVINCOLORUOTA program	99
Figure 4.110: Midpoint between the cobot cart and the AGV	99
Figure 4.111: SCARICORUOTE program	100
Figure 4.112: Marker	100
Figure 4.113: Wheels tray with the marker applied	100
Figure 4.114: RICERCAVASSOIORUOTA vision function	101
Figure 4.115: GPM Locator Tool for the RICERCAVASSOIORUOTA vision function	101
Figure 4.116: Reference position of the RICERCAVASSOIORUOTA vision function	102
Figure 4.117: Wheel place phase.....	102
Figure 4.118: Photo to the boards tray.....	103
Figure 4.119: Boards tray with the marker applied	103
Figure 4.120: RICERCAVASSOIoTAVOLA program	104
Figure 4.121: RICERCAVASSOIoTAVOLE vision function.....	104
Figure 4.122: Reference position of the RICERCAVASSOIoTAVOLE vision function	105
Figure 4.123: PICKTAVOLA program.....	105
Figure 4.124: Board pick phase	105
Figure 4.125: SVINCOLOTAVOLA program.....	106
Figure 4.126: SCARICOTAVOLA program	106
Figure 4.127: Board place phase	106
Figure 4.128: Mizubot 4.0	107
Figure 4.129: Intralogistics 4.0.....	107
Figure 4.130: Push paddles.....	108
Figure 4.131: Push paddles with vertical orientation	108
Figure 4.132: Push paddles with horizontal orientation	108
Figure 4.133: Internal structure of the Mizubot 4.0.....	109
Figure 4.134: Mizusumashi roundtrip	110
Figure 4.135: MiR 200	111
Figure 4.136: The two security laser scanners.....	112
Figure 4.137: There are four ultrasonic sensors on the robot: two in the front (left) and	

two in the back (right)	112
Figure 4.138: Created map	113
Figure 4.139: VL marker	114
Figure 4.140: Created mission.....	115
Figure 4.141: Unload side.....	117
Figure 4.142: Order in which channels are unloaded and delivery mode	118
Figure 4.143: Configuration of different delivery modes.....	119

Introduction

The manufacturing industry has always been the backbone of the made-in-Italy economy. In fact, our country represents the sixth largest manufacturing power in the world and the second largest manufacturing country in Europe after Germany. The spread of digital technologies in manufacturing represents a great opportunity for an industrial revolution for our country, which aims toward Industry 4.0, that is, a production model characterized by a strong technological component, a strong digital component. All this is to improve human working conditions, to increase production efficiency and to the quality of the final output, that is, the product [1].

Industry 4.0 is the way to enable our companies to intercept the change and not be left out of the competitive trajectories of global industry. Open innovation and digitalization, which permeate every sphere of the enterprise and production process, redefine both value chains and business models.

1.1 Industry 4.0

The term Industry 4.0 refers to the “transformation of the entire sphere of industrial production that has occurred through the fusion of digital technology and the Internet with conventional manufacturing” [2]. At the base of Industry 4.0 there are innovative digital technologies, also called as Smart Technologies.

These can be categorized into two major classes:

1. the first one, which is more related to the world of Information Technology (IT – covers data storage, retrieval, transmission, manipulation and protection) and includes Industrial IoT, Industrial Analytics and Cloud Manufacturing;
2. the second one, closer to the Operational Technologies (OT – refers to the network of devices and software used in industrial, manufacturing and process control systems) and includes Advanced Automation, Advanced Human Machine Interface and Additive

Manufacturing.

Smart Technologies find use in the entire production chain of a manufacturing company: from the prototyping of a product to the management of its lifecycle, from the planning of physical and financial flows of the entire supply chain to the heart of the production process, internal and external logistics, maintenance, quality and safety [3].

The 4.0 Key Enabling Technologies are:

- **Advanced Manufacturing Solution:** this term refers to advanced robotic systems capable of interacting with their surroundings and operators (collaborative robots), or being autonomously guided (AGVs), and integrating the use of vision systems and pattern recognition techniques [4];
- **Additive Manufacturing:** also called 3D printing, refers to the production of three-dimensional objects from a CAD model representing their design. This type of printing is based on a layered deposition of material;
- **Augmented Reality:** is the evolution of virtual reality, and integrates reality with textual, graphic, and sound information in real time. Augmented reality aims to simplify the user's activity and interaction with the physical environment [5];
- **Simulation/Digital twin:** digital twin is the virtual simulation of a real plant, machine, process or system, developed from data collected by IoT sensors [6]. The goal is to test and optimize production systems before they are built, identifying potential failures and having more information about their life cycle, thus enabling predictive maintenance [5];
- **Horizontal/Vertical Integration:** horizontal integration refers to the integration of production processes. It can take place on multiple levels: within the same production line, between multiple production plants or along the entire supply chain. Vertical integration links all logical levels within the organization, from the production area to R&D, quality control, product management, sales, and marketing [7];
- **Industrial Internet:** the Industrial Internet of Things (IIoT) represents a well

delineated area of the common IoT and refers to a network of smart objects communicating with each other in the industrial environment. There is an increasing amount of data from process machinery, equipment, infrastructure, and the introduction of IIoT aims to connect the sources of this data together, to create an interconnected network that communicates constantly, so that companies can monitor, analyze data, and outline strategies for optimizing production processes;

- **Cloud:** infrastructure defined as common, flexible, scalable, and open by design, for sharing data, information and applications across the network [5];
- **Cybersecurity:** the set of technologies useful for protecting computer systems from loss or breaches of data and information [5];
- **Big Data & Analytics:** it refers to the collection and analysis of data coming from different resources, areas, which pass through the Internet. These are an added value for companies that are thus able to optimize products and production processes, but also to identify possible future scenarios [5].

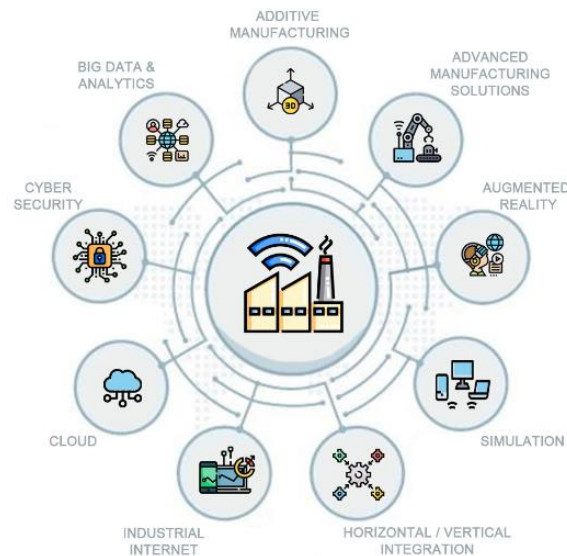


Figure 1.1: 4.0 Key Enabling Technologies. Reference: urly.it/3x-5q

1.2 Smart Factory

The smart factory is the implementation of Industry 4.0 technologies: automation, the Internet, the ability to store data (Big Data) and be able to interpret it. Thanks to the latest generation of sensor technology that makes the factory always connected and communicating, it is possible to collect, manipulate and share data, which has always been the beating heart of the smart factory [8]. In this way, data-driven decision-making processes can be implemented to optimize product production and distribution and improve related services. Smarter management of products, processes and people is thus one of the greatest benefits of the smart factory. By being able to increase production and drastically reduce waste and human errors, it also becomes a sustainable factory that can deliver a higher quality product to customers at a lower cost to the company.

1.2.1 System Architecture

The structure of the Smart Factory is represented below in the Figure 1.2.

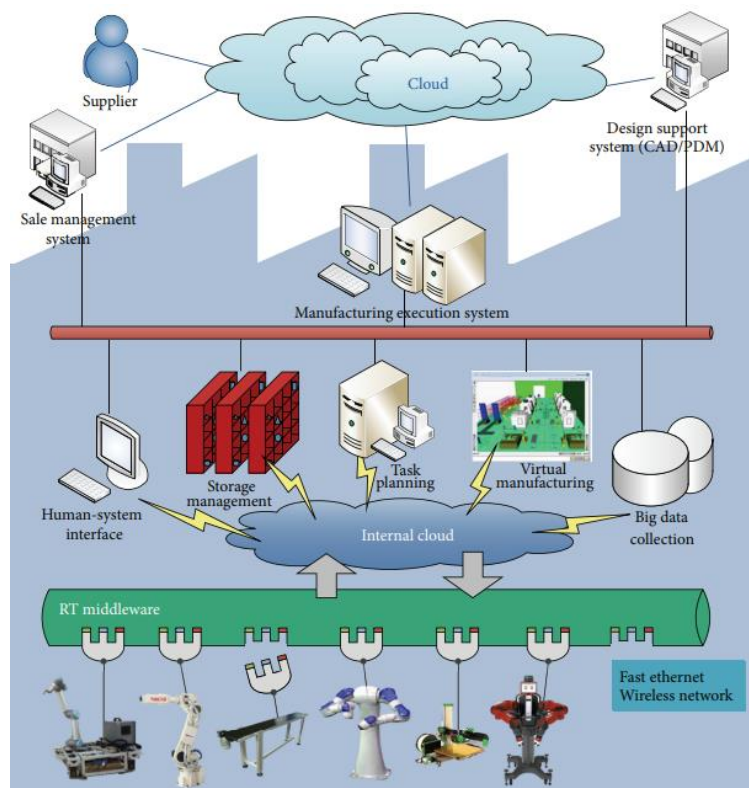


Figure 1.2: System Architecture of the Smart Factory. Reference: urly.it/3ymhm

In the low level, there are robotic devices like cobot, AMR robots, 3D printers, that can be plug & play into the system, so that they can be repurposed and reused for other production processes. These robotic systems differ in operating systems, programming languages. A common standard platform called RT-Middleware (RTM) is used to construct the robot system and to enable dynamic communication and coordination between any two of these robotic modules [9].

In the middle level, there are several functionalities implemented in the internal cloud like human-system interface, storage management, task planning, virtual manufacturing, big data collection. When the customer makes the order of personalized products through a human-machine interface (HMI), this request is sent to the task planning module. The planner converts the customers' orders into a sequence of subtask, that must be carried out by the corresponding robotic components.

In the upper level there are software that are used to control and optimize production processes and to facilitate effective sales workflow, such as:

- Manufacturing Execution System (MES): a software that manages, controls, and optimizes production processes [10];
- Advanced Planning & Scheduling (APS): for production planning and scheduling along the supply chain [11];
- Customer Relationship Management (CRM): which helps companies manage, analyse, and optimize interactions between the company and customers, through all related data [12];
- Enterprise Resources Planning (ERP): a software based on enterprise resource planning, but also and especially on business processes , both as a whole and in detail [13];
- Warehouse Management System (WMS): to manage and optimize warehouse logistics [14].

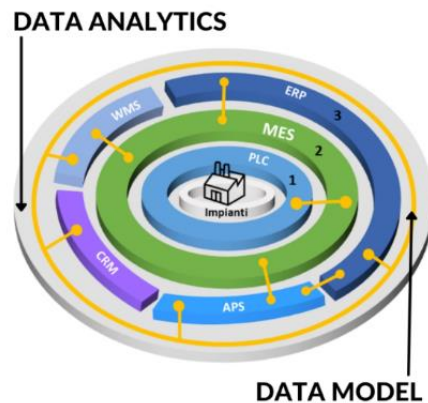


Figure 1.3: Smart factory software. Reference: urly.it/3ymkr

It is therefore possible to realize a self-configuring factory, capable through its automated systems of dynamically sequencing and scheduling production orders, of transporting components from storage locations to the appropriate stations, of collecting and monitoring in real time all data and information coming from machines, people, or business areas, with the goal of improving plant performance, organizational productivity, products, and customer service levels.

1.3 Lean Smart Production

To lead a company to achieve important efficiency goals and operational excellence, production within a smart factory must necessarily be optimized through the principles of lean logic [15]. The goals are to eliminate waste, to produce in the shortest possible time and to start production only from customer input.

At the foundation of lean philosophy there is the concept of just-in-time (JIT) production, that is, producing what is needed at the time it is needed. Just-in-time management of materials, of orders, allows minimization of stored material and greater responsiveness to possible market fluctuations.

There are four principles on which just-in-time production is based:

1. Pull System;
2. Takt Time;
3. One piece flow;
4. Zero defects.

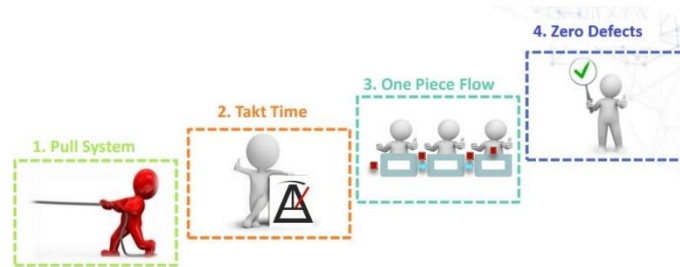


Figure 1.4: 4 JIT principles. Reference: New Value Group s.r.l.

1.3.1 Pull Logic

In such a view, the entire production process is commanded by customer demand. In the pull system, it is the need of the downstream process, that is the customer order, that moves the upstream ones. This ensures that production is based on actual demand and not on market forecasts.



Figure 1.5: Pull System. Reference: New Value Group s.r.l.

1.3.2 Takt time

The takt time is the cadence of production for the customer and is used to synchronize the pace of production with the pace of sales.

$$TAKT\ TIME = \frac{\text{Total daily available time}}{\text{Average daily demand during the period under consideration}}$$

Producing according to takt time avoids overproduction. The attempt is to follow the fluctuation of demand in a timely manner, but this is an impossible condition to achieve. Production may follow the fluctuation to a good approximation, but always with a lag that may be variable, like shown in the Figure 1.6.

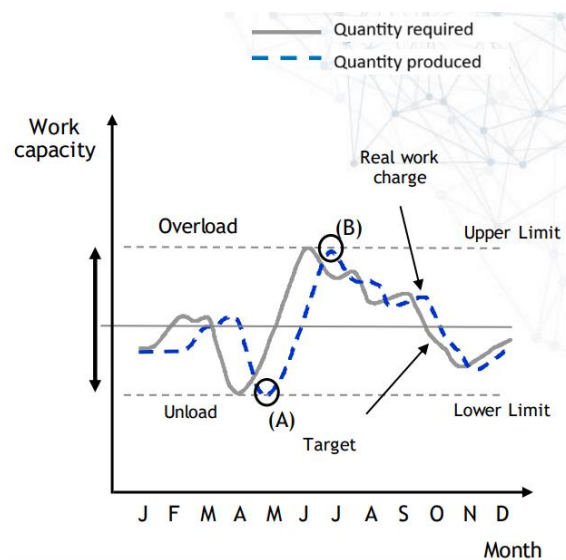


Figure 1.6: Variation in demand. Reference: New Value Group s.r.l.

There are operational strategies to reduce and solve the problem of uncontrolled lag, such as production levelling, in which a target is identified to keep the quantity to be produced constant in the medium-long term. In this way fluctuations in demand are covered by a controlled stock level of ready products.

1.3.3 One piece flow

It consists of producing one item at a time. This results in no handling of material from one location to another, and therefore no stock between the various stages of the production process. Another advantage of this production flow is the reduction of lead

time, which is the time to produce the product, downstream of the customer's request for it.

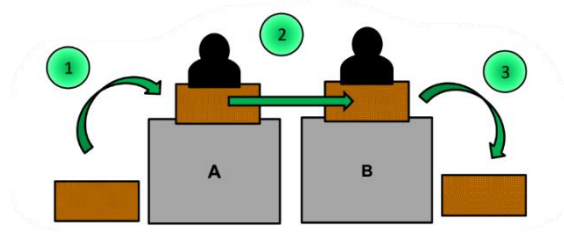


Figure 1.7: One piece flow. Reference: New Value Group s.r.l.

1.3.4 Zero defects

Defects are the results of mistakes. Several methods and tools can be used to improve process stability and avoid human errors. A proper quality system, which moves from product control to process control, is essential to avoid non-conformities and thus to detect defects before they propagate to subsequent process steps. Of no less importance is maintenance, both autonomous and preventive, which enables proper operation of machinery, reducing possible failures, and maximizes the technical availability of equipment.

1.4 Competence Center

The Competence Center is a place where skills and technologies coexist, to make them available to the territory, to SMEs, and to all companies operating in Italy.

The competence center has a dual purpose:

- of **technology transfer**, with Italian companies on Industry 4.0 themes;
- of **training**, because competence centers must support companies to transfer skills within their organizations.

The Competence Industry Manufacturing 4.0 (CIM 4.0) is one of the eight highly specialized competence centers established by the Industry 4.0 plan and is located in Turin. CIM 4.0 is composed of stakeholders from multiple sectors: its founding members include public academic institutions and 22 companies operating nationally and internationally, representing multiple sectors (automotive, aerospace, machinery, technology Hardware/Software, integrators, and service/energy/telecommunications providers) [16].



Figure 1.8: Competence Centers in Italy.
Reference: urly.it/3xaw9



Figure 1.9: Partners CIM 4.0. Reference: urly.it/3xawd

CIM 4.0 consists of two pilot lines:

- a **Digital Factory**, dedicated to all the aspects of digital manufacturing technologies;
- a High Tech Center, dedicated to the industrialization of **Additive Manufacturing**;

These two lines provide a technological showcase, where companies can explore digital technologies that can increase production efficiency and business competitiveness, but at the same time they are functional for training activities with on-the-job training. In addition, with these testing laboratories it is possible to accompany companies to translate

a proof of concept, a functional object, into a prototype, an object that can be easily taken into production.

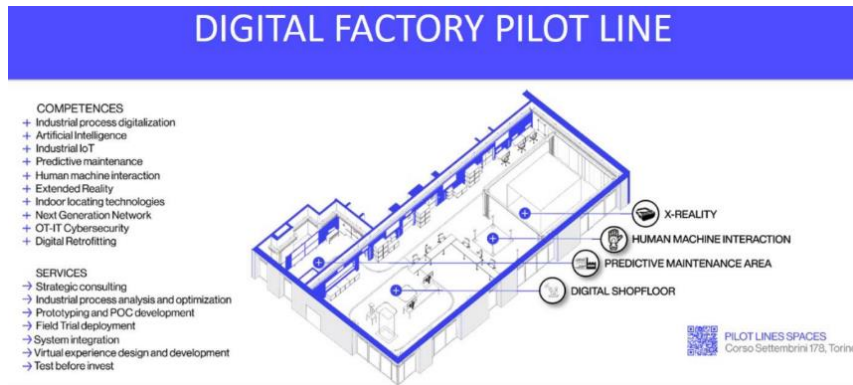


Figure 1.10: Pilot Line “Digital Factory”. Reference: urly.it/3xaxz

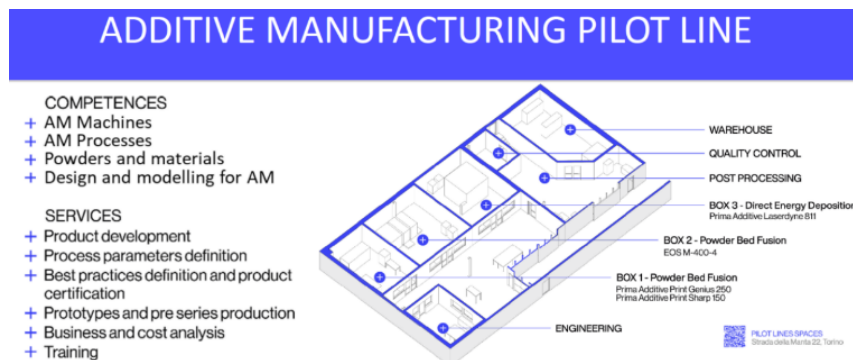


Figure 1.11: Pilot Line “Additive Manufacturing”. Reference: urly.it/3xaxz

1.5 Thesis objective

The objective of this thesis is to implement an automated servo system for a skateboard production line. The system consists of a FANUC CRX-10iA collaborative robot, with iRVision 2D vision system on board, and the Mizubot 4.0, a self-driving vehicle made by the company New Value Group. The robot has been programmed to perform the picking of skateboard components: specifically, it must be able to locate and pick up parts, regardless of size, shape or position, and to deposit them on the Mizubot

4.0 handling system, that is, a mobile line or process delivery system. At this point the Mizubot 4.0 was programmed so that, by following a certain path, it could reach the final assembly station, where it could unload the components.

1.6 Organization of the chapters

The thesis consists of this introduction in which the general concepts of Industry 4.0, Smart Factory and Lean manufacturing are included.

Chapter 2 describes Robotics in more detail with an overview of the main robotic systems used in industry, starting with Cartesian robots, and ending with the most innovative ones, which are collaborative robots.

Chapter 3 discusses in more detail one of the enabling technologies of Industry 4.0 which is Additive Manufacturing: the process that leads a CAD model to be transformed into a real printed object is described in more detail.

Chapter 4 gives a detailed description of the steps that were necessary in developing the case study: from CAD modelling, to programming the cobot, to configuring the Mizubot 4.0 system.

Finally, Chapter 5 presents the conclusions that were drawn and possible future developments of the system.

Chapter 2

Robotics: general concepts

Robotics is the discipline of engineering that studies and develops methods that allow a robot to perform specific tasks, by reproducing in an automatic way the human work.

Robotics is divided into two major classes:

- **Advanced Robotics**, which includes applications in hostile environments such as military, nuclear, space, submarine, or applications in medical, domestic, agriculture, and many other fields. These applications from the technological point of view are still not very robust and not yet mature;
- **Industrial Robotics**, with application of robots in the industrial field. This turns out to be well-established, mature, and reliable applications [17].

2.1 Industrial robotics

Industrial robotics is the integration of robots within a factory, to make production lines automated [18]. In recent years, the market for industrial robots has experienced a sudden increase [19]: the reasons are increased productivity, the goal of simplifying the workflows to increase efficiency and reduce the risk of errors in production. Another reason for their strong diffusion in the industry is that they perfectly fulfil tasks within a manufacturing process: handling, manipulation, and measurement. Within a manufacturing process of an artifact, it starts from the raw material to arrive at the finished object after a series of processing steps: robots are able to manipulate objects whose shape and characteristics vary during this cycle, to transfer them from one location to another

to be machined, assembled, welded, rather than sealed, without damaging them. In addition, robots are capable of performing quality control of finished products, of searching for nonconformities using accurate robot-mounted vision systems [20].

Thus, the use of robots in the industrial field makes it possible to automate production lines, and have them be programmable, by easily reprogramming robot operating sequences, and flexible, since robots are capable of performing different processes.

2.1.1 The robot

According to the Robot Institute of America, “a robot is a reprogrammable multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks” [20].

The robot consists of:

- a mechanical structure, with actuators and sensors;
- a control unit.



Figure 2.1: Mechanical structure of a robot.
Reference: urly.it/3yn9d



Figure 2.2: Control unit of a robot.
Reference: urly.it/3yn98

2.1.1.1 Mechanical structure

The robotic manipulator is a kinematic chain, consisting of a series of ideal arms, also called links, connected by ideal joints. One end of the chain is the base of the robot, usually fixed on the ground, on a pedestal, or sometimes even on a mobile stand. At the other end is the end-effector, which is the gripper or working tool.

The links are idealized geometric bars connecting two or more joints. Joints are idealized physical components allowing a relative motion between the attached links. Each joint allows a single degree of motion between the connected links.

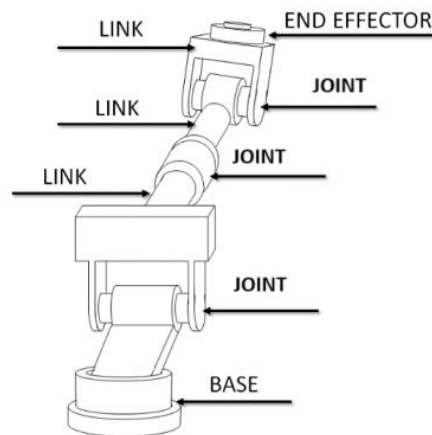


Figure 2.3: Schematic drawing of the main constituent elements of a robotic manipulator.
Reference: urlly.it/3yncs

Joints may be of two types:

- Revolute (or rotational) - **R**, that allows a rotation between the connected links;
- Prismatic (or translational) - **P**, that allows a translation between the connected links.

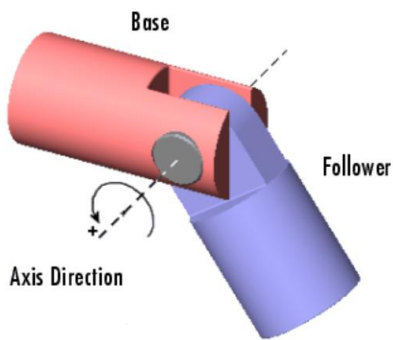


Figure 2.4: Revolute joint.
Reference: urly.it/3ynjf

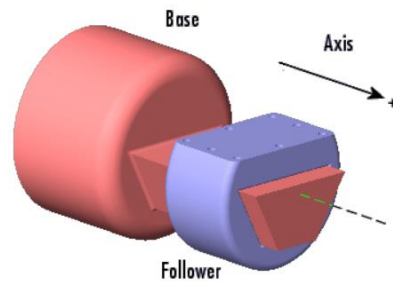


Figure 2.5: Prismatic joint.
Reference: urly.it/3ynjf

The task to be performed by the robot and its movement in space are usually expressed in terms of the **TCP** (Tool Center Point), which is an ideal point on the end-effector, typically in its middle. The TCP constitutes the origin of the tool frame where the z-axis points outward, along the axis of the last joint, and the x- and y-axes are obtained by construction through the right-hand rule.

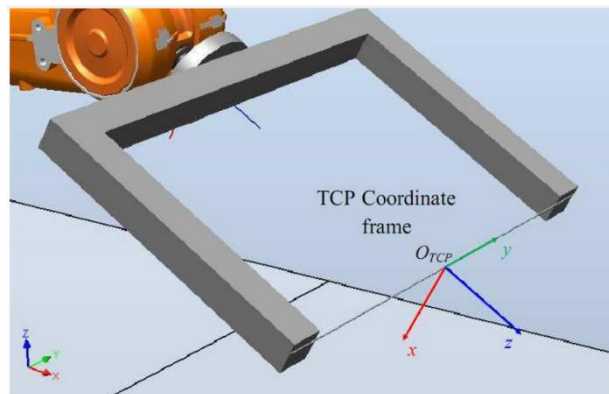


Figure 2.6: Tool frame. Reference: urly.it/3ynf0

2.1.1.2 Control unit

A robot control system is a device that can manage the movement and behaviour of a robot. Its main tasks are to Interface with the operator (HMI), trajectory planning, real-time control of joint motion, data storage, management of external devices installed

on the robot, such as vision systems, grippers, and diagnostics and malfunction management [17].

2.2 Robot types

The robot types are defined by the arm configuration, based on the sequence of its joints. There are six different types of robots:

- Cartesian robots;
- Cylindrical robots;
- Spherical robots;
- Scara robots;
- Anthropomorphic robots;
- Collaborative robots.

2.2.1 Cartesian robots

Cartesian = 3P = P – P – P

Cartesian robots consist of three prismatic joints, with mutually orthogonal axes, and are used in industrial production, with reference to molding. Each axis has only one degree of freedom, and with the combination of the motion of all axes it is possible to define and reach the required working area. Cartesian 3-axis robots, in particular, consist of a vertical Z-axis, a Y-axis, which moves the vertical arm in line with the machine, and an X-axis, represented by the horizontal crossbar, which generally extends above the mold.

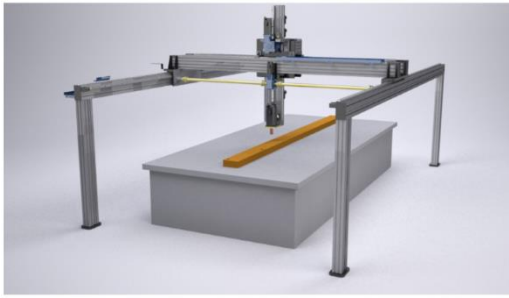


Figure 2.7: Example of cartesian robot.
Reference: urly.it/3yngt

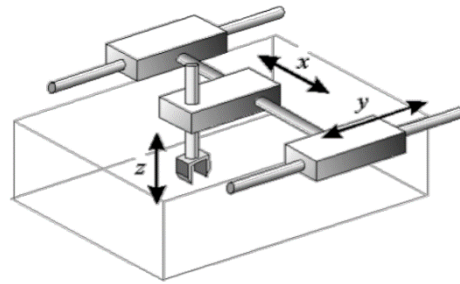


Figure 2.8: Scheme of the cartesian robot. Reference: Robotics course slides

2.2.2 Cylindrical robots

$$\text{Cylindrical} = 1R - 2P = R - P - P$$

The shoulder has one revolute joint with vertical axis followed by two prismatic joints, one of which is vertical and the other horizontal. The horizontal prismatic joint allows to reach horizontal spaces, but the accuracy decreases toward the arm ends [21].

Some applications of cylindrical robots include [22]:

- Spot welding automation;
- Robotic casting and molding machine handling;
- General machine handling applications;
- Material handling applications including package assembly and palletizing;
- Robotic coating and finishing applications;
- Automated assembly operations;

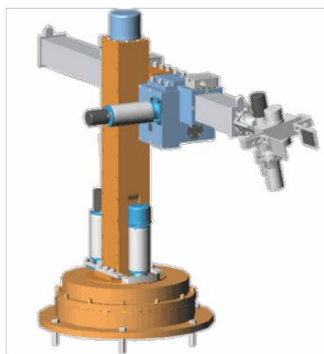


Figure 2.9: Example of a cylindrical robot

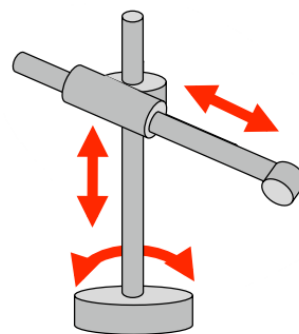


Figure 2.10: Scheme of the cylindrical robot. Fonte: urly.it/3ynn0

2.2.3 Spherical robots

$$Spherical = 2R - 1P = R - R - P$$

The shoulder has two revolute joints (one vertical, one horizontal axis) followed by a prismatic joint with axis orthogonal to the last revolute joint. The structure is less rigid than the cylindrical one, and the accuracy decreases with the elongation of the prismatic arm [21].

Some applications of spherical robots include [23]:

- Machine tool tending;
- Spot welding automation cells;
- Assembly operations;
- Casting and material handling automation applications;
- Diecast process tending;
- Gas welding applications;
- Arc welding automation;
- Painting systems and paint line tending applications.



Figure 2.11: Example of a spherical robot. Fonte: urly.it/3ynj9

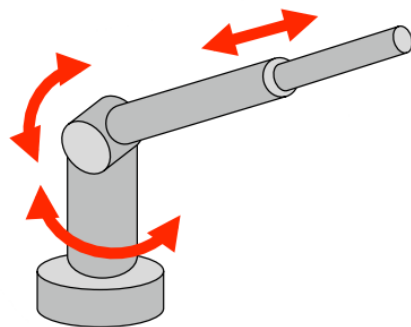


Figure 2.12: Scheme of the spherical robot. Fonte: urly.it/3ynn0

2.2.4 Scara robots

$$\text{Scara} = 2R - 1P = R - R - P$$

The shoulder has two revolute joints followed by one prismatic joint, all with parallel/vertical axes. The structure is rigid in the vertical component and compliant in the horizontal components [21].

This type of robot was developed to improve speed and repeatability in pick-and-place operations or to speed up and improve assembly steps. These robots are widely used in the automotive industry, as well as in the industrial electronics sector. The kinematics of scara robots make them particularly suitable for performing assembly operations with tight tolerances. The scara arm mechanics are generally quite strong and able to withstand fatigue and collisions without problems.

Some applications of spherical robots include [24]:

- Pick and place operations;
- Small assembly operations;
- Sealant application;
- Machine tool handling;
- Dispensing applications;
- Electric appliance manufacturers;
- Home appliance manufacturers.



Figure 2.13: Example of a scara robot.
Fonte: urly.it/3ynn9

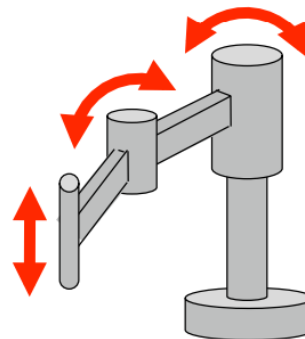


Figure 2.14: Scheme of the scara robot.
Fonte: urly.it/3ynn0

2.2.5 Anthropomorphic robots

Anthropomorphic = 3R = R – R – R

The shoulder has three revolute joints: the first one is vertical, the other two are horizontal and parallel. The structure is like the human body, with trunk, arm and forearm, with a final wrist. In fact, they have been studied and designed to aid and replace humans in certain stages of the assembly line, the most dangerous ones specifically. It is one of the most common structures in industry, since it provides the best dexterity [21].

Some applications of anthropomorphic robots include [25]:

- Assembly;
- Palletizing;
- Material handling;
- Machine servicing;
- Packaging solutions;
- Processing;
- Foundry and forging;
- Robotic welding and cutting of all materials;
- Pick and Place;
- Die casting;
- Welding.



Figure 2.15: Example of an anthropomorphic robot. Fonte: urly.it/3ynmg

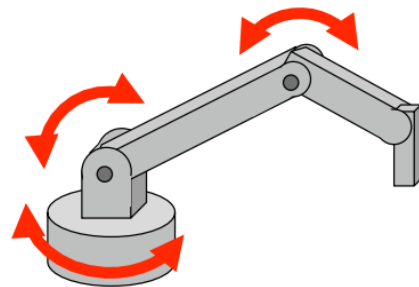


Figure 2.16: Scheme of the anthropomorphic robot. Fonte: urly.it/3ynn0

2.2.6 Collaborative robots

Collaborative robots are anthropomorphic robots with six-axis movements designed to meet safety, flexibility and compactness criteria, and designed to work closely with the operator.

Any form of interaction between humans and robots that involves their coexistence, cooperation and assistance is collaborative. As the name itself suggests, “Collaborative Robotics” means sharing work [26].

Collaborative robots, often called cobots, are automated systems capable of working in the same workspace shared by the operator. They thus allow the strengths of both of these actors to be combined: on the one hand, the flexibility and reasoning ability of humans; on the other hand, the precision, reliability and repeatability of robots. With built-in torque sensors to control external forces, cobots can sense their surroundings and adapt to the context, stopping their movement and producing no harm to humans, in case of accidental contact with the operator. This avoids the installation of protective cages in the working environment, reducing space occupancy and optimizing factory space. Cobots are built to be easily programmable and trainable, with increasingly intuitive interfaces and programming modes, so that even the factory operator is able to adapt or modify trajectories or operations on the part on line.



Figure 2.17: Example of a collaborative robot.
Fonte: urlv.it/3vnr7



Figure 2.18: Robot-operator collaboration.
Fonte: urly.it/3ynr1

All these features make cobots extremely versatile and perfect for a wide range of applications: from palletizing to assembly, from quality control to blasting, from handling to machine servicing. They represent a ready-to-use solution capable of enhancing production lines and processes because of their safety, flexibility, and speed of deployment.

Chapter 3

Additive Manufacturing

Additive manufacturing is the term used to refer to what used to be rapid prototyping and is now commonly referred to as 3D printing. Specifically, rapid prototyping is a term used in many fields to refer to a process of rapidly making an initial model of a part or component, i.e., a prototype, which will be followed by modifications and variations before it is marketed. However, more recent applications of printing, which are increasingly geared toward the realization of a final product rather than a prototype, have shown that the term rapid prototyping is no longer correct. It also does not take into account the principles on which the latest printing processes are based, that is the creation of a part by additive technique. According to recently adopted ASTM consensus standards, nowadays the term being used is the term additive manufacturing (AM) [27].

3.1 Basic principles

The principle behind additive manufacturing is that starting from a 3D model of a part, made using CAD (Computer-Aided Design) modelling software, it is possible to directly print it, without the need to resort to programming the different stages of the manufacturing process. The only knowledge that additive manufacturing requires is just a few dimensional and orientation details of the part before printing, the functioning of the machine and the materials it uses in printing. This makes it possible to print very complex objects in a very simple way.

The working principle of additive manufacturing is that a component is made by a layered deposition of material. Each layer is a thin cross-section of the part obtained from the CAD model created. In reality each layer has a finite thickness, and therefore the component produced will be an approximation of the original model, like shown in the Figure 3.1. It is clear that the thinner these layers are, the more faithful to the model

the final printed product will be. Depending on the type of AM machine used, the materials that can be used may vary, as well as how the layers are created and bonded together. These differences affect the accuracy of the part produced and its mechanical properties, but also the speed of printing, the amount of post-processing required, and the overall cost of the part.



Figure 3.1: Images showing the CAD model of a component and the printing effects of using different layer thicknesses.

Fonte: I. Gibson, D. Rosen and B. Stucker, *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, Springer,

3.2 AM Process

Additive manufacturing consists of a series of stages through which from the design stage, i.e., CAD modelling of the component, to the final application, i.e., production of the actual, physical component.

In general, most AM processes involve eight steps, as shown in Figure 3.2, which are:

1. CAD;
2. Conversion to STL;
3. STL File Manipulation and transfer to AM machine;
4. Machine Setup;
5. Build;
6. Removal;
7. Post-processing;
8. Application.

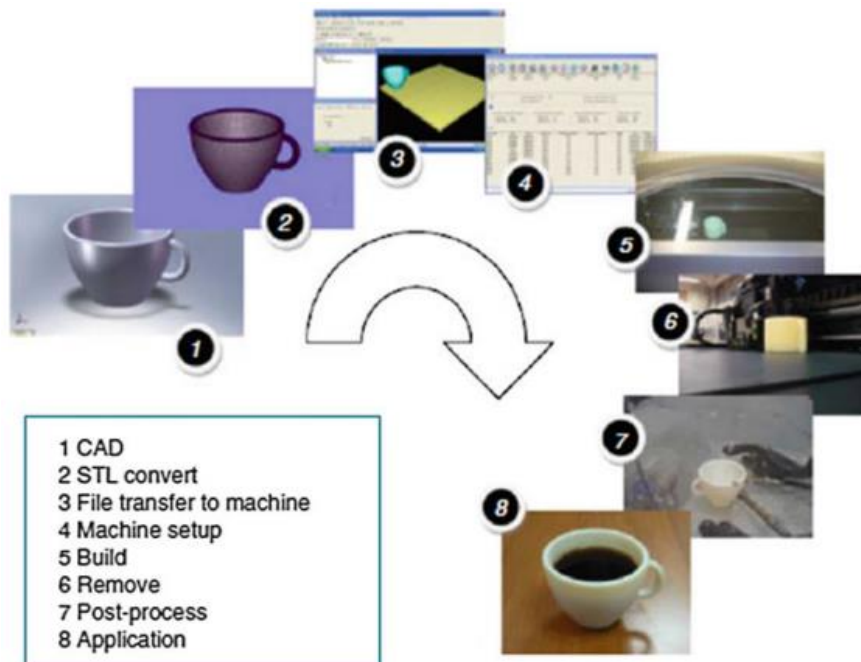


Figure 3.2: AM process.

Fonte: I. Gibson, D. Rosen and B. Stucker, Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing, Springer, 2010.

3.2.1 CAD

The first step in any printing process is to figure out what the product should look like, how it should work, and what special features it should have. To develop an idea several sketches and representational models are conceptualized. Once the geometry we want to give our component is chosen, an attempt is made to translate it into a 3D model, using any professional CAD solid modelling software. It is important to remember that additive manufacturing of a component always begins with a 3D model, which must describe its geometry. In some cases, it is possible to obtain a CAD model in reverse, that is, starting from the real object and using an optical scanner [27].

3.2.2 Conversion to STL

Each AM machine uses the STL file format. STL is a file format originated for CAD STereoLithography software [28]. The STL format describes a cad model in terms of geometry only, eliminating all construction data, different modelling steps, and approximating each model surface with a series of triangular faces. The minimum size of these triangles can be set within most CAD software and the objective is to ensure the models created do not show any obvious triangles on the surface. The triangle size is calculated as the minimum distance between the plane represented by the triangle and the surface it is supposed to represent. A basic rule of thumb is to ensure that the minimum triangle offset is smaller than the resolution of the AM machine. The STL file is thus a collection of the vertices of triangles and surface normal vectors, and it has no units, colour, material, or other feature information. In addition, this file format is essential for calculating the slices and layers according to which the component will be produced. Conversion to the STL format typically can be done directly through some save settings of the CAD modelling software, used to create the 3D model.

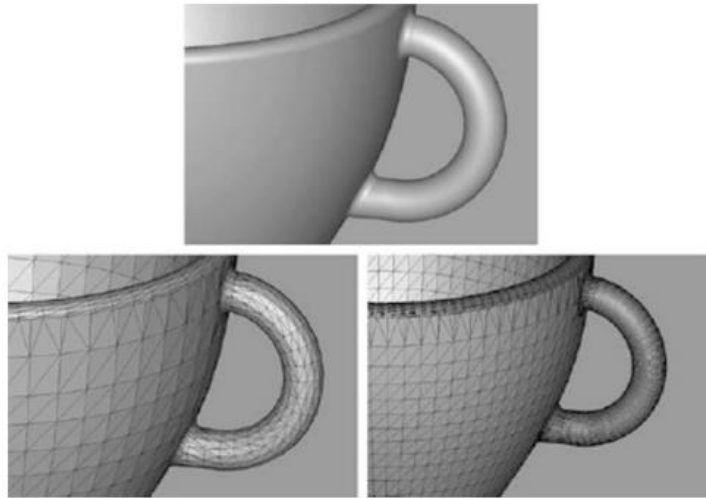


Figure 3.3: An original CAD model converted into an STL file using different offset height (cusp) values.
Fonte: I. Gibson, D. Rosen and B. Stucker, Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing, Springer, 2010.

3.2.3 STL File Manipulation and transfer to AM machine

Once the STL file has been created, it is manipulated on the printer's system software before being transferred to the AM machine. In particular through the software, it is possible to verify that the part is correct, and to check that its size is congruous. In addition, the part can be repositioned in a different location within the machine and its orientation can be varied as well for building. Orientation can affect several parameters such as the accuracy of the part, the printing time of the part, and the amount of support structures necessary to help keep the part from collapsing or warping during the build process for printing [27].

Typically, the choice of an orientation can improve the characteristics of certain parameters such as increasing the accuracy of the part produced, but at the same time this can greatly slow down the printing process and require more support structures to print. In addition, the orientation must take into account the fact that the mechanical strength

properties of the produced part are worse in the z-direction of the part extrusion, so it is recommended to choose the orientation so that this direction is the least stressed by mechanical stresses during its use. Improvements in these parameters are thus in contrast with each other, and so improving some may make others worse. Therefore, it is essential to find an orientation that provides the right compromise between all these parameters.



Figure 3.4: Supports generated for printing the component.
Fonte: I. Gibson, D. Rosen and B. Stucker, Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital

3.2.4 Machine setup

The AM machine must be properly configured before the build process, so settings involving material constraints, power source, layer thickness, and timings are adjusted. In addition, many machines must be physically prepared for printing, so the operator must check that there is enough material loaded into the machine to complete the printing [27].

3.2.5 Build

The build is the phase in which the part is produced, with the layer-based deposition material approach. This step is an automated procedure, requiring no special

attention from the operator, except that he must ensure that during the printing phase the material does not run out, or that there are no problems related to machine power supply or software glitches.

3.2.6 Removal

Once the part has been made, it is removed from the AM machine. Before performing this operation, the operator must be careful to ensure that the operating temperatures are sufficiently low or that there are no actively moving parts.

3.2.7 Post-processing

Once the part has been removed from the machine, depending on the application, it may require a more or less extensive post-processing phase before it can be used. All support structures for printing are removed, and depending on where they are located, this process can require a lot of time and a careful, experienced manual manipulation. This step also includes a component cleaning process, such as abrasive finishing, like polishing and sandpapering, or application of coatings [27]. Other applications may require chemical or thermal treatment of the part to achieve final part properties.

3.2.8 Application

The processed parts can now be used for applications. It has to be considered that possible discrepancy may occur between the expected behaviour from standard specifications and the real one. In fact, some AM processes can create parts with small voids trapped inside, causing failures under mechanical structures. In some cases, some processes may cause the material to degrade quickly, bond or crystallize in a worse way. In almost every case, the properties are anisotropic, which mean that there are different properties in different directions. So, AM produced parts behave differently than those produced by using a more conventional approach. For this reason, designers must consider the possible differences between applications and be aware of them during the design phase.

Chapter 4

Case study

4.1 Overview

The aim of this case study is the development of an automatic servo system for a skateboard production line. The line consists of a collaborative robotics station, which includes a Fanuc cobot CRX-10iA mounted on a mobile cart, the Mizubot 4.0 AGV from New Value Group, and a final component assembly station. The workflow begins with the cobot, which through a gripper and a vision system mounted on the end-effector, is tasked with recognizing and picking up the different skate components such as wheels, boards, and trucks, regardless of how they are positioned. After that, the cobot task is to unload these components into special trays placed on the AGV, that are free to slide on linear guides. At this stage the AGV is positioned close to the collaborative robotics station, so that the cobot has sufficient outreach to load it. Afterwards, the AGV begins its mission, which starts from the loading position and leads it, through a predetermined path, to the final component assembly station. Here the AGV unloads the trays thanks to push paddles, free to slide along the rails and able to push them on a chute, which will lead them to the work surface, where the operator will manually assemble the different components to make the skateboard.



Figure 4.1: Overview of the skateboard production line.
Reference: edited by Francesco Succurro

4.2 Robot gripping system

Grippers differ based on the gripping technology they adopt. The main classes considered for this case study were:

- **Parallel grippers:** grippers with two parallel jaws with flat edges, suitable for small components, for safe and precise handling of parts. These grippers are not suitable for objects of irregular shape and size [29];
- **Self-centring grippers:** equipped with three jaws, for internal and external clamping of the part, to keep it centred. They are most used for round or cylindrical components [30];
- **Angle/radial grippers:** with two or three jaws, for safe handling of parts with different weights. They are used in applications ranging from packaging and assembly techniques for electronic components, to loading and unloading machine tools [31].



Figure 4.2: Parallel Gripper.
Reference: urly.it/3ypf0



Figure 4.3: Self-centring gripper.
Reference: urly.it/3ypf0



Figure 4.4: Radial gripper.
Reference: urly.it/3ypf0

In turn, each gripper is divided according to the type of drive to be adopted, **pneumatic** or **electric**. Pneumatic grippers use compressed air and pistons to operate. They therefore require an air supply system, which causes the gripper to close and grip the object, and instead open when the pressure is released [32]. These grippers therefore do not allow a programmable intermediate opening but can either close or open to their maximum jaw stroke, at their maximum force. Servo-electric grippers are equipped with an electric motor and by a control unit. The controller sends a signal to the gripper, with

parameters regarding force, stroke setting per jaw, and desired opening speed, required by the robot. Once the gripper receives this signal, it drives the electric motor, which brings the gripper into the desired configuration. Some types of electric grippers are also able to exchange other information with the robot, through a feedback mechanism, such as figuring out if the gripper is busy, is grip, is safety on, is connected, or getting depth, relative depth, width. This information can be very useful to implement possible control logic within a robot program, to ensure the safety of the application.

Among electrically driven grippers, there is a category of special grippers called **collaborative** grippers. These grippers, in addition to having the characteristics of the electric ones described earlier, are equipped with certain safety systems that can ensure the safety of the operator in case of accidental insertion of fingers between the jaws during gripping. This is made possible by a limitation on the gripping force. Therefore, collaborative grippers are extremely safe and reliable for performing pick & place operations in collaborative applications, but at the same time they are limited in terms of maximum gripping force.

4.2.1 Gripper selection

This initial design phase focused on choosing an appropriate robot gripping system, that could perform the tasks required by our application. The first element that was considered in the selection of the gripper was the type of drive: the choice fell on an electric type of drive. The reason for this was that the collaborative robotics station was not designed to have an air system setup, so the pneumatic gripper was discarded. In addition, the possibility of customizable selection of parameters such as force and opening of the gripper leaned toward this choice. The second element that was considered concerns the handling technology: the robot must pick up the skateboard components, i.e., wheels, trucks, boards, and deposit them on special trays placed on the AGV. A two-jaw parallel gripper was chosen, as despite the fact that the three objects had different shapes from each other, it was able to clamp the parts and keep them stable with force. To this end, special gripper jaws were made specifically for our application and were able to make the gripping of all objects stable and firm in the same way. The description of

the process of making gripping fingers will be covered in the following paragraphs.

Another factor to consider when choosing a gripping system is its weight. Each robot has within its datasheet a parameter called payload: it represents the maximum weight the arm can lift, including that of the gripper. Therefore, it is important to choose a gripper whose weight, combined with the weight of the objects to lift, does not exceed the robot's payload.

But a decisive aspect in the choice of gripper was the robot used for the application: being a collaborative type of robot, the desire was to make the entire robot plus gripper system a fully collaborative station. After careful evaluations regarding the feasibility of gripping objects despite a limited maximum force, a collaborative-type gripper was chosen. The robotic gripper chosen is the RG6 from the company OnRobot.



Figure 4.5: OnRobot RG6 gripper.
Reference: urly.it/3ypfc

It is an electric collaborative gripper, with built-in Quick Changer, a device that allows quick and easy replacement of one tool performing a machining operation with another. The RG6 has a maximum gripping stroke of 150 mm, which is adjustable. This feature makes it a general-purpose gripper that can be repurposed and scaled up in other applications where it may be required to grip bulkier objects. It is capable of lifting objects

up to a weight of 6 kg in vertical gripping, and up to 10 kg in horizontal gripping. It is equipped with an automatic depth compensation system, which allows the fingers of the gripper to run parallel to the work surface. It is also able to make detections regarding the state of gripping, whether it has failed or is in progress, or get information regarding the real-time opening of the gripper fingers, thus having the ability to measure the width of the objects to be picked up. This last piece of information can be very important when creating the robot's pick & place program because it allows to precisely calibrate the gripper's closure and thus pick up the object correctly. Another strength of this gripper is the ability to customize the gripper fingers to fit the shape of the objects to be lifted and to have a wide flexibility of gripping. As for the software integration of the gripper with the robot, on the other hand, this is very simple, as the gripper has plug & play software on board, which any person without a technical background can install in the robot controller.



General Properties	Minimum	Typical	Maximum	Unit
 Payload Force Fit 6 Kg	-	-	6	[kg]
	-	-	13.2	[lb]
 Payload Form Fit 10 Kg	-	-	10	[kg]
	-	-	22.04	[lb]
Total stroke (adjustable)	0	-	160	[mm]
	0	-	6.3	[inch]
Finger position resolution	-	0.1	-	[mm]
	-	0.004	-	[inch]
Repetition accuracy	-	0.1	0.2	[mm]
	-	0.004	0.007	[inch]
Reversing backlash	0.1	-	0.3	[mm]
	0.004	-	0.011	[inch]
Gripping force (adjustable)	25	-	120	[N]
Gripping force deviation		±25		%
Gripping speed *	51	-	140	[mm/s]
Gripping time **	0.05	-	0.15	[s]
Adjustable bracket tilting accuracy	-	< 1	-	°
Storage temperature	0	-	60	[°C]
	32	-	140	[°F]
Motor	Integrated, electric BLDC			
IP Classification	IP54			
Dimensions	262 x 212 x 42			[mm]
	10.3 x 8.3 x 1.6			[inch]
Weight	1.25			[kg]
	2.76			[lb]

Figure 4.6: RG6 Datasheet.
Reference: urly.it/3ypj

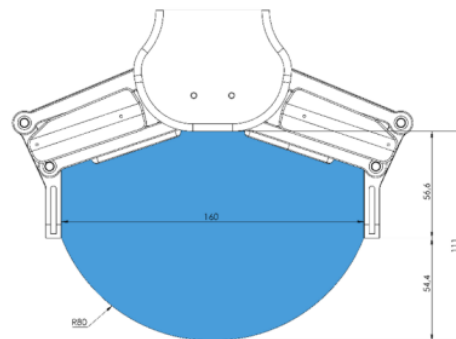


Figure 4.7: RG6 work range.
Reference: urly.it/3ypj

4.3 CAD modelling

4.3.1 Interface plate

The first thing that was done to enable the gripper to be mounted on the robot end-effector was to design an interface adapter plate that would mechanically fit both the desired robot flange and the gripper quick-changer. In addition, to allow the vision system to be installed on the end effector and ensure its co-presence with the gripper, it was planned to make a plate with an L-shaped structure, which would allow the camera to move in an integral manner with the robot, and not be installed on a fixed structure instead. To start in the design of the plate, it was necessary to begin from the technical drawings of the robot flange, the robot-side quick-changer and the camera.

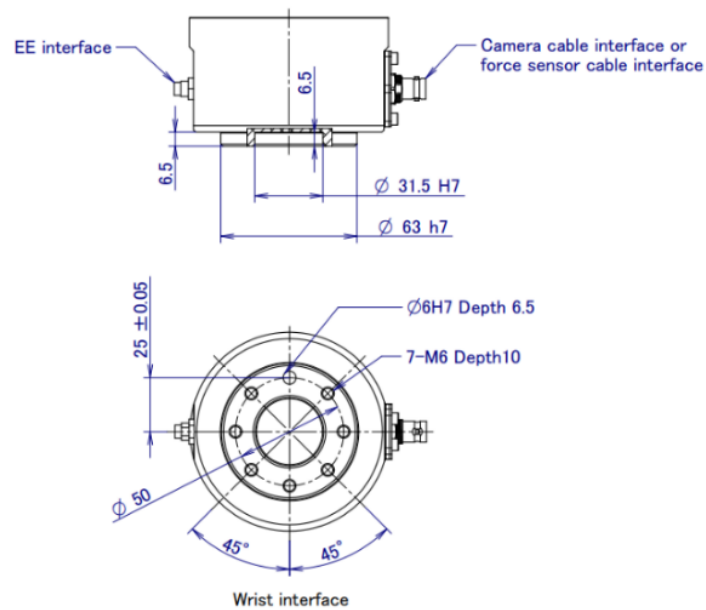


Figure 4.8: End effector interface. Reference: Fanuc Italia S.r.l

Case Study

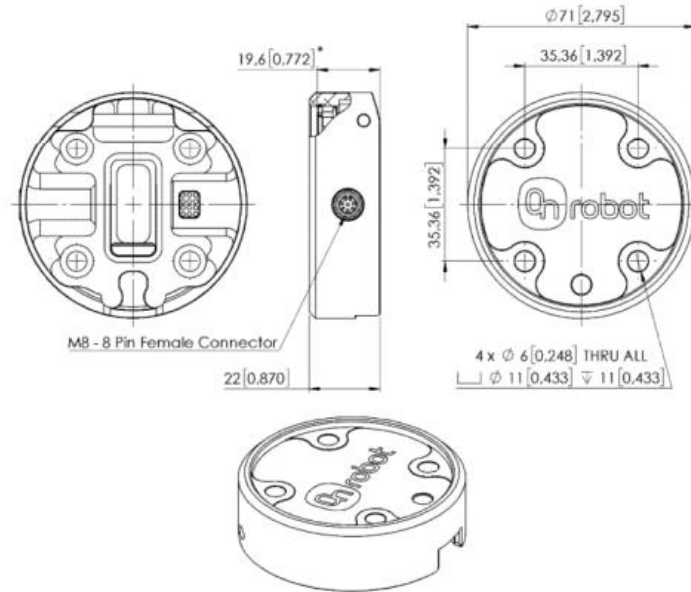


Figure 4.9: Quick-changer for I/O – Robot side. Reference: Fanuc Italia S.r.l.

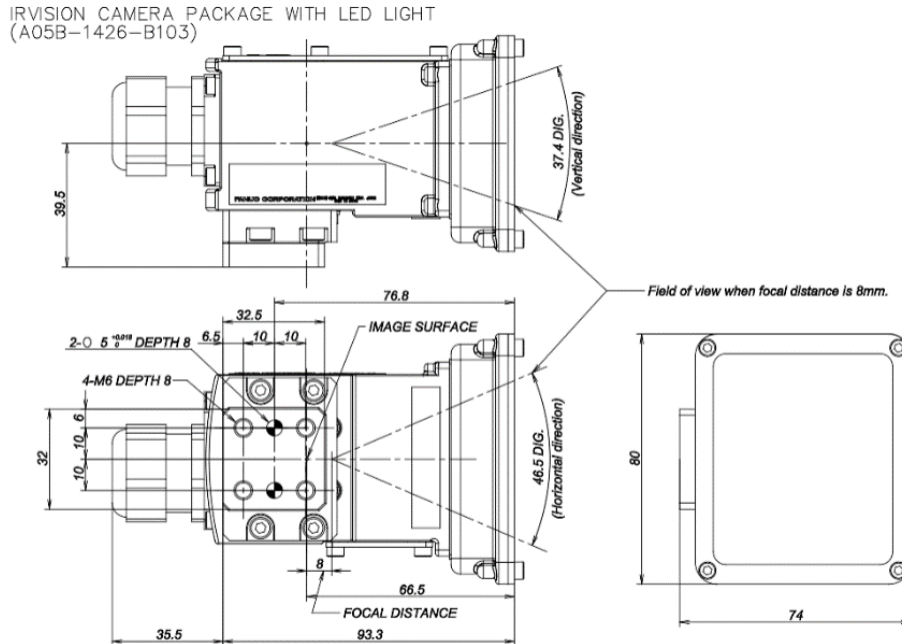


Figure 4.10: 2D iRVision camera package. Reference: Fanuc Italia S.r.l.

The design of the component was based on SolidWorks, a 3D CAD modelling tool.



Figure 4.11: Gripper side interface plate.
Reference: edited by Francesco Succurro

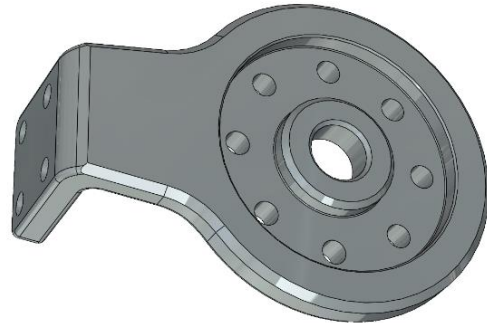


Figure 4.12: Robot side interface plate.
Reference: edited by Francesco Succurro

The designed plate has an L-shaped structure, allowing for camera mounting, and at the gripper-robot coupling, it has a circular shape that follows the geometry of the robot flange and quick changer. It has eight through holes, although it can be seen from the Figure 4.9 that the quick-changer only needs four anchor screws. This redundancy in the holes was done to provide more freedom when mounting the gripper, allowing it to be positioned in two possible configurations, rotated 90° from each other. In this case, the choice was made to mount the gripper with the opening direction of its jaws perpendicular to the extension direction of the L-shaped structure. This was done to prevent the camera from interfering with the gripper during maximum opening of its jaws.

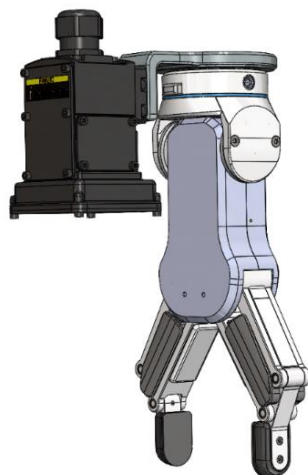


Figure 4.13: Assembly drawing of the gripper with the camera.
Reference: edited by Francesco

4.3.2 Trays

Appropriate trays were shaped and made to accommodate the placement of wheels, trucks, and boards of the skateboard.

4.3.2.1 Wheels tray

The wheels tray was designed to hold a complete kit of four wheels, useful for making a single skateboard. With this solution, the wheels are arranged in two columns, each consisting of two wheels. In this way, the wheels arrangement is compact and safe from an anchoring point of view, so that there is no possibility of them tipping over or falling off even when unloading on the assembly station. In particular, two centring rods with chamfered invitations were made, which not only provide stability to the wheel arrangement, but also act as a guide to the positioning of the wheels when the robot unloads them. In addition, the tray has a raised initial part, which will be home to a marker, which each cycle will be recognized by the robot's vision system and will allow the position offset of the tray to be known.

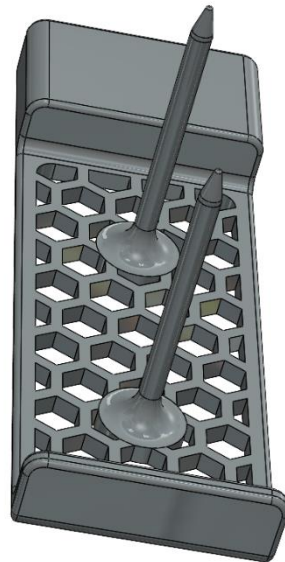


Figure 4.14: CAD model of the wheels tray.
Reference: edited by Francesco Succurro

4.3.2.2 Trucks tray

The designed truck tray can accommodate two components, and has a toothed structure, the task of which is to represent an invitation in the positioning of the trucks when they are being deposited by the robot, but also to give stability to the components when they are unloaded from the AGV onto the assembly station, so that they cannot tip over.

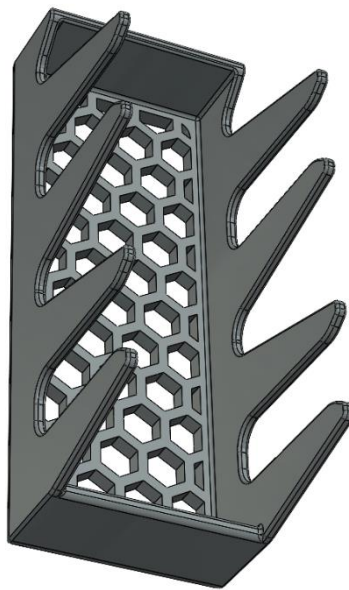


Figure 4.15: CAD model of the trucks tray.
Reference: edited by Francesco Succurro

4.3.2.3 Boards tray

The board tray can hold up to three components, which can be arranged parallel to each other and laid each on its own long side. It has a notched structure with chamfered invitations, which act as a guide when the robot deposits the components. In this case, as shown in the Figure 4.16, the width of this tray is greater than that of the wheel or truck tray. This is because the boards are bulky objects, and therefore they need a larger tray running on a wider width guide.

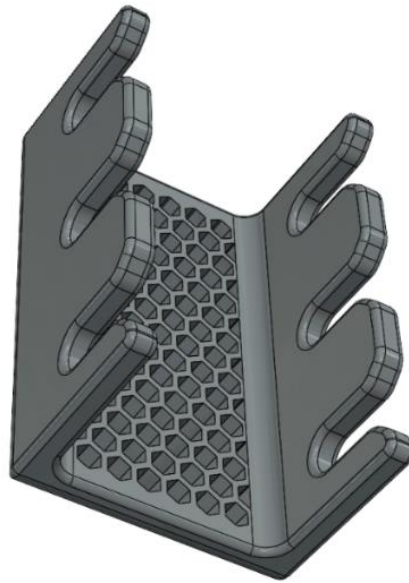


Figure 4.16: CAD model of the trucks tray.
Reference: edited by Francesco Succurro

4.3.3 Custom fingertips

The gripper jaws that come standard with the gripper are not able to pick up all the components of the skateboard but are designed to pick up objects with regular shapes. For this reason, new gripper jaws were investigated that would be able to pick up all skateboard components in a stable manner, even though the grip did not perfectly fit all objects equally. First, considerations were made about the objects to be picked up: wheels, trucks and boards. The geometric characteristics of the objects were studied, trying in each object to figure out what surfaces might be suitable for gripping, and with what geometries these might fit best.



Figure 4.17: Wheels.
Reference: urly.it/3ypry



Figure 4.18: Trucks. Reference: urly.it/3yys0



Figure 4.19: Board.
Reference: urly.it/3yps2

It was noticed that both the wheel and one side of the truck hanger have round shapes. Therefore, efforts were made to make gripper fingertips that would have a circular recess, in such a way to best accommodate the wheel and the truck. As for the board, it presents a flat upper surface, while the lower part has an irregular shape composed of several ribs, of different cross-section from each other, in relief. Therefore, it was thought to make flat surfaces on the jaws, despite the circular recess, that could allow when in grip, to contact the flat parts of the objects: in the case of the skate, the jaws grip the flat top and the lower rib near the edge of the board. As for the truck, the grip is provided on one side by the circular recess, and on the other one by the flat part of the jaw that contacts the flat side of the truck.

After these preliminary studies, the CAD model of the new gripper jaws was made.

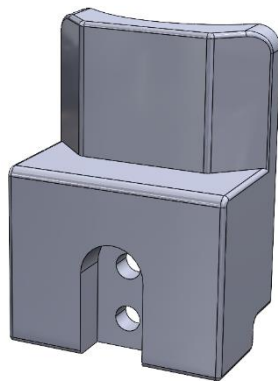


Figure 4.20: Outer part of the gripper jaw.
Reference: edited by Francesco Succurro

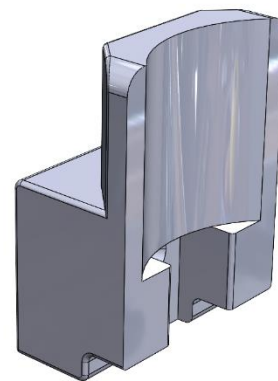


Figure 4.21: Inner part of the gripper jaw.
Reference: edited by Francesco Succurro

It can be seen in Figure 4.21 that the inner part of the jaw has a semi-cylindrical cavity, with a height and radius equal to that of the wheel. This is to make sure that the two complementary jaws of the gripper are able during gripping to self-centre the wheel and fit its shape perfectly. At the same time, the semi-cylindrical cavity ensures a good grip on the circular side of the truck hanger. We can also see that there are also flat surfaces on the inside of the jaw, which are useful, as described earlier, for stable gripping of boards and trucks. As for the outer part of the jaw, on the other hand, it can be seen in Figure 4.20 how a custom-made recess was made, which was able to allow for the proper installation of the new gripper fingertips on the finger hooks. For the realization of this anchoring system, the technical drawings of the gripper provided by OnRobot were consulted, as shown in Figure 4.22. Specifically, two M5 cylindrical threaded countersunk screws were used so that they could engage the threaded through holes in the gripper fingers.

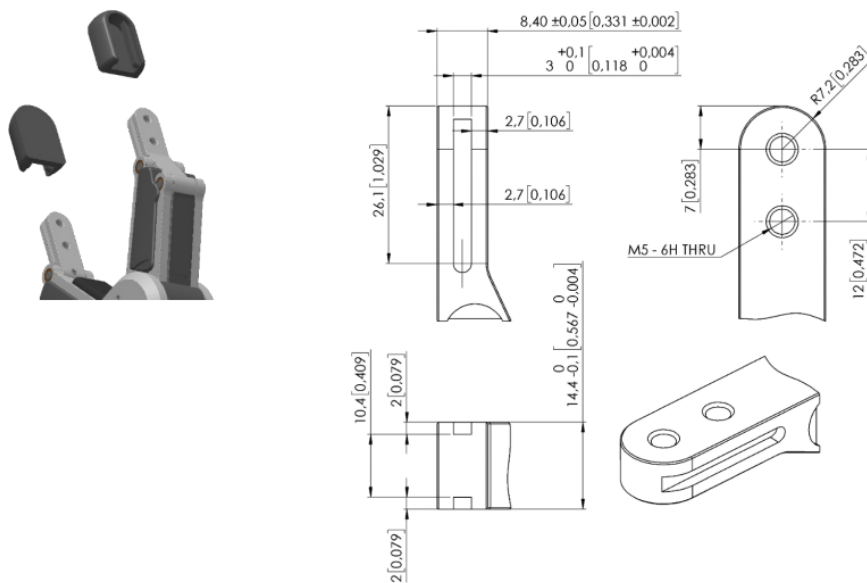


Figure 4.22: Technical drawing of the hooks for the gripper jaws.

Reference: urly.it/3yppsp

4.4 Realization of components by Additive Manufacturing

In order to properly proceed in the process of printing the components through the additive manufacturing machine, it was initially necessary to convert the CAD modelling files of the components into the .STL format. The STL file represents an intermediate step between the geometry described by the CAD and the machine language of the 3D printer [33]. In this step, the surface of the object is discretized into a set of triangles that form a lattice, and thus an approximation of the continuous surfaces of a 3D model is generated [34]. In the conversion to .STL files, a chordal tolerance of 0.0025 mm and an angular tolerance of 5° were chosen. By adjusting these two parameters, it is possible to adapt the mesh creation to our component, thus creating a less dense triangulation in the case of flat surfaces, while denser in the case of curved surfaces [33]. Lower values of these two parameters can truly and accurately approximate component surfaces, but this comes at the expense of speed in the printing process. Therefore, it is good to find a compromise between fidelity in approximating a surface and speed in the printing process, depending on the characteristics of the component to be printed.

After that, came the preparatory phase of putting the components into the machine. This stage depends on the machine chosen for the component molding process, the type of additive manufacturing process used, and the material chosen. In this case, the machine chosen for printing the components is the UltiMaker S5, which has the following specifications:



Figure 4.23: UltiMaker S5.
Reference: urly.it/3ypt1

- Technology: Fusion Filament Fabrication (FFF) [44]
- Printhead: Double extrusion print head with a unique lifting system automatic nozzles and interchangeable print cores [44]
- Printing volume (XYZ): 330 x 240 x 300 mm [44]
- Materials supported: PLA, Tough PLA, Nylon, ABS, CPE, CPE , PC, TPU 95A, PP, PVA, Breakaway [44]
- Connectivity: Wi-Fi, LAN, USB port [44]
- Software: UltiMaker Cura [44]

It is possible to note that the printing technology used is Fusion Filament Fabrication (FFF), also called as Fused Deposition Modelling (FDM), which involves extruding the molten material through a nozzle in a controlled manner: specifically in these systems there is a pre-heating chamber, where the material is brought to its melting point, and then it is deposited layer by layer on the printing surface until the object is formed. The order of the layers in which the material is deposited depends on the orientation that is given to the part within the print volume of our machine. It is therefore necessary to carry out, through the machine's standard software, simulations of the positioning of the object within the print volume, before proceeding with the printing of the part. These simulations allow to figure out which orientation of the part provides greater printing accuracy, or whether one orientation of the part requires more supports than another when printing.

In this case, UltiMaker Cura software was used to perform the components printing simulations. They will now be examined in order, starting with the gripper jaws.

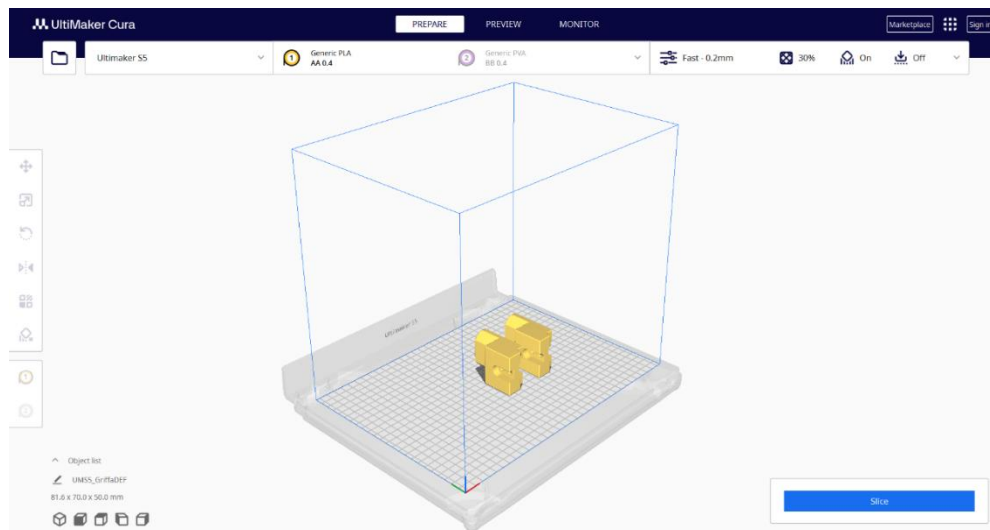


Figure 4.24: Printing simulation of the gripper jaws. Reference: edited by Francesco Succurro

Some printing parameters were set, such as the material and the infill density. The chosen material was PLA (PolyLactic Acid), a thermoplastic material made from renewable resources such as corn starch. It is the easiest material to print with, does not

tend to warp and shows good mechanical strength. Aesthetically, it has a well-maintained surface, and the part produced is easy to machine or cut [35]. Infill density is a percentage value that indicates how much material our part is filled with. In this case a 30 % infill was chosen.

After this step, it was analysed how to lay and orient the parts in the printing process. It started in testing the configuration in which the jaws rested on the printing plane on their inner side, where the cylindrical concavity was present.

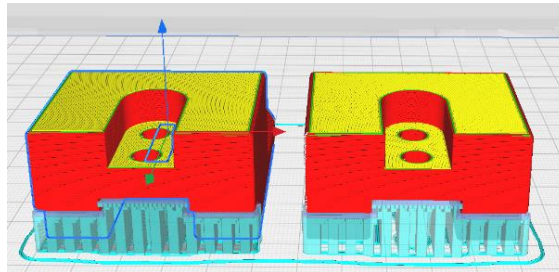


Figure 4.25: Rear view of the printing simulation of the gripper jaws, with their inner part resting on the printing plate.
Reference: edited by Francesco Succurro

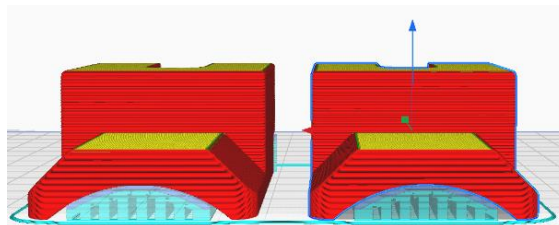


Figure 4.26: Front view of the printing simulation of the gripper jaws, with their inner part resting on the printing plate.
Reference: edited by Francesco Succurro

In this configuration, printing would take about five hours and would involve installing a large amount of supports at the cylindrical cavity and at the initial part of the jaw where there was a fold. Therefore, it was decided to place the jaws on their side to print them.

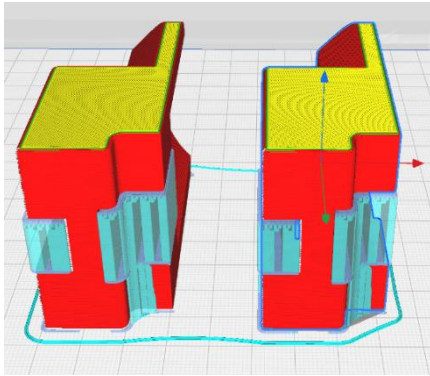


Figure 4.27: Rear view of the printing simulation of the gripper jaws resting on its side.
Reference: edited by Francesco Succurro

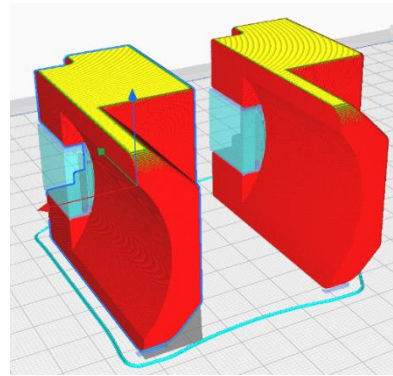


Figure 4.28: Front view of the printing simulation of the gripper jaws resting on its side.
Reference: edited by Francesco Succurro

This orientation allowed less support structures to be installed during printing: only a few at the gripper jaws coupling area, while none at the cylindrical recess. In addition, another factor that was considered in choosing the orientation, was the mechanical strength property of the produced part: this was worse in the z-direction of extrusion. As shown in Figure 4.27, in this configuration the z-direction, represented by the blue line, was the one with the worst strength properties, but also the one affected by less mechanical stress during gripping in this application.

4.4.1 Printing of the components

After deciding the orientation of the part, the print configuration file was saved on a USB stick. The latter was connected to the machine so that printing could take place. The printing phase took approximately four hours and thirty minutes. When the printing was completed, the support structures were removed.



Figure 4.29: Printing stage of the new gripper fingertips.
Reference: edited by Francesco Succurro

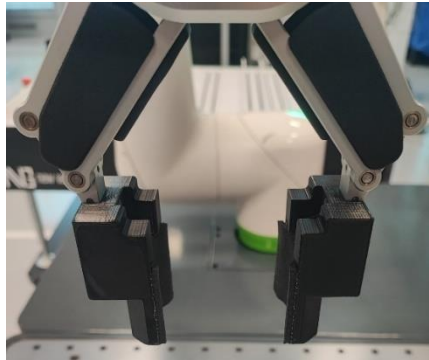


Figure 4.30: New gripper fingertips mounted on the gripper.
Reference: edited by Francesco Succurro

For printing the other components, the procedure to be followed was similar. The material used was the same, but a different infill density was chosen based on the component that was to be manufactured. For each component, an analysis was performed on the correct positioning of the object within the print volume, before proceeding with its fabrication.

Printing time: 5 hours 3 minutes

Infill density: 90%

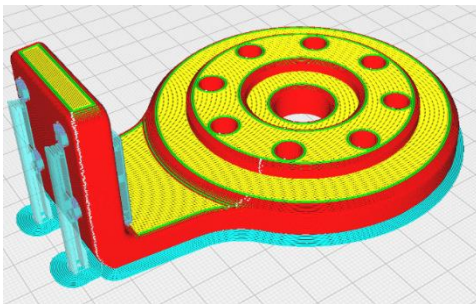


Figure 4.31: Printing simulation of the interface plate. Reference: edited by Francesco Succurro



Figure 4.32: Interface plate mounted on the end effector of the robot. Reference: edited by Francesco Succurro

- Printing time: 4 hours 4 minutes
- Infill density: 30%

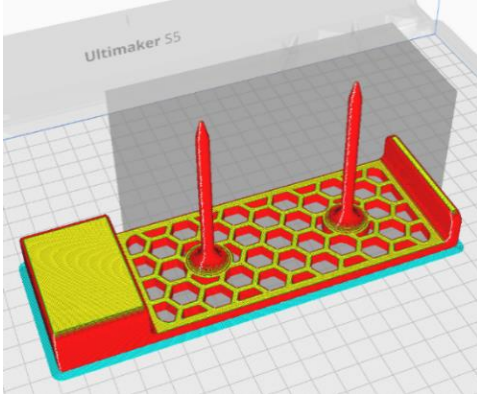


Figure 4.33: Printing simulation of the wheels tray. Reference: edited by Francesco Succurro



Figure 4.34: Printed wheels tray. Reference: edited by Francesco Succurro

- Printing time: 20 hours 37 minutes
- Infill density: 30%

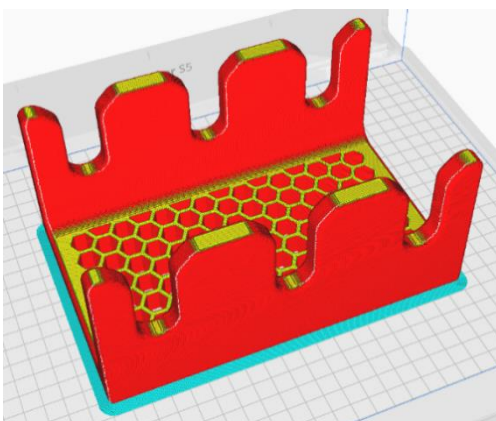


Figure 4.35: Printing simulation of the boards tray. Reference: edited by Francesco Succurro



Figure 4.36: Printed boards tray. Reference: edited by Francesco Succurro

- Printing time: 5 hours 24 minutes
- Infill density: 30%

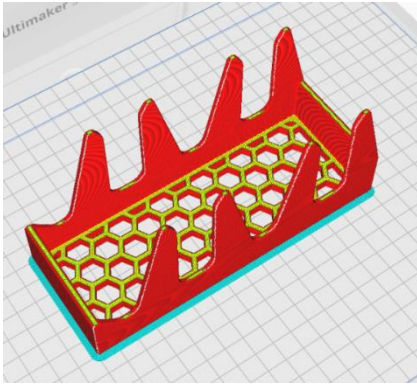


Figure 4.37: Printing simulation of the trucks tray. Reference: edited by Francesco Succurro



Figure 4.38: Printed trucks tray. Reference: edited by Francesco Succurro

4.4.2 Coating of the new gripper fingertips

After printing all the components, a possible material to be integrated on the inside gripping part of the jaw was studied to improve the grip. In the case of contact between the PLA of the jaw and the plastic of the board, or between the PLA and the metal of the truck, the grip adhesion was very low and thus the part would be in danger of slipping off. In addition, a low grip adhesion between object and jaw required more force that gripper must apply to try to hold the object firmly.

The force required to grip an object is expressed by the following formula:

$$F = \frac{m * (g + a) * f}{d}$$

Where:

- F is the gripping force of the gripper in N;
- m is the mass of the object to be moved in kg;
- a is the acceleration set for handling in m/s^2 ;
- g is the acceleration of gravity in m/s^2 ;
- f is a correction coefficient;

➤ μ is the friction coefficient, which thus takes into account the ratio between the two materials in contact, that of the gripper and that of the jaws.

The coefficient of friction μ and the gripping force F are inversely proportional. Therefore, an attempt was made to coat the inner part of the jaws with a material that could make the objects gripping stable without ruining them and increase the coefficient of friction between the two materials, so that less gripping force was needed to hold the pieces.

The chosen material was **adhesive neoprene**, with a thickness of 3 mm. It is a type of waterproof closed-cell foam rubber, perfect for realizing friction while gripping, and able to model itself according to the geometry of the object when in grip and expand and return to its original state when not. It has a very good adhesive, which provides excellent grip on any smooth surface it is applied. It can also be easily cut with a cutter and shaped to fit its function.



Figure 4.39: Adhesive neoprene sheet.
Reference: urly.it/3yq0c



Figure 4.40: Detail of the grip fingertip coated with the adhesive neoprene. Reference: edited by Francesco Succurro

4.5 Implementation of the line

As mentioned at the beginning of the case study, the line consisted of two stations:

- the first was the collaborative robotics station, where the skateboard components to be loaded onto the AGV were located;
- the second one was where the operator resided and was dedicated to assembling the components to form the skateboard.

Transportation of the components from one location to the other was carried out by the autonomously guided vehicle Mizubot 4.0.



Figure 4.41: The collaborative robotics station and the MIZUBOT 4.0. Reference: edited by Francesco Succurro



Figure 4.42: Assembly station. Reference: edited by Francesco Succurro

The main steps that were taken to make the line work were:

1. Configuration of the collaborative robotics station, that is the robot and the vision system;
2. Robot program creation, in order to pick up the skate components placed on the robot cart and deposit them on the trays placed on the Mizubot 4.0;
3. Creation of the mission of the AGV to transport the components from the loading station to the final assembly station;
4. Configuration of the different Mizubot 4.0 unloading channels.

4.6 Configuration of the collaborative robotics station

4.6.1 Collaborative robotics station

The collaborative robotics station was composed of several devices:

- Fanuc CRX-10iA cobot;
- R-30iB Mini Plus controller;
- Tablet Teach Pendant;
- Mobile robot support cart;
- Fanuc *iR*Vision 2D system;
- OnRobot RG6 gripper.



Figure 4.43: Collaborative robotics station.
Reference: edited by Francesco Succuro

4.6.1.1 Robot CRX-10iA

The CRX-10iA robot belongs to the Fanuc family of collaborative robots. It is equipped with six axes of motion, each of which determines a degree of freedom by rotation around its axis [36]. It is possible to see in Figure 4.44 that the combination of these rotations characterizes the kinematics of the robotic arm. It has a maximum reach of 1249 mm and can carry up to a maximum of 10 kg on its wrist, making it a flexible robot that can be used in various application fields. Due to its low weight, the CRX model can be easily installed on mobile systems, such as automated guided vehicles (AGVs) or on wheeled platforms, making it a solution that can be readjusted and scaled up according to the task it needs to perform.

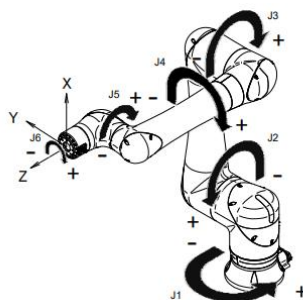


Figure 4.44: Kinematics of the robotic arm. Reference: Fanuc Italia S.r.l.

It is equipped with a LED indication system through which it is possible to recognize what state the robot is in: if the light is green, the robot is in collaborative mode; yellow in non-collaborative mode with doubled maximum speed; red if the robot is stopped in alarm. The maximum speed of the robot in collaborative mode is 1000 mm/sec, while in non-collaborative mode it is 2000 mm/sec [37]. Another state that is useful to recognize is when the light, in addition to being green, is also intermittent: this means that the robot is prepared to be moved manually by the operator. Rounded surfaces and wide space between the forearm and the back arm allow the operator to manoeuvre the robot as safely as possible. Thanks to internal torque sensors, the robot can detect any collision with the operator and stop automatically. In case the robot gets stuck in contact with a station or operator, it can be pushed away from the collision, making it impossible to get stuck.

In these types of collaborative robots, peripheral devices such as grippers, vision systems, and sensors can be easily integrated both on the software side, through simple plugins to be installed on the controller, and on the hardware side through two connection ports on J6, one of which is dedicated to the vision system and the other to any other type of tool. This offers quick connections and interchangeability for end of arm tooling (EOAT) [38].



Figure 4.45: Fanuc CRX-10iA collaborative robot.
Reference: urlly.it/3yq2k

4.6.1.2 Tablet Teach Pendant

The interface with the robot is a Teach Pendant touchscreen tablet equipped with a capacitive pen, which allows easy programming. It enables even those who have never programmed a robot before, to realize their own application easily and in a short time. Even complex robotic tasks are made simple thanks to the intuitive interfaces: for example, by using icon drag-and-drop, the operator can enter operations or movements that will then be executed in sequence by the robot. In addition, thanks to the wizards, it is possible to build reference systems around the robot, or to define the parameters of the load carried by the robot in a simpler and more intuitive way than in traditional robotics. The teach pendant is also equipped with a red emergency stop button, to immediately stop the robot at any time, but also with a deadman switch on the back, that is used as an enabling device. For example, as mentioned earlier, it is possible to manually move the robot to teach the points of processing or record the entire trajectory: in this mode, robot motion is allowed only while deadman switch is gripped, while it is stopped immediately if the switch is released.



Figure 4.46: Tablet Teach Pendant.
Reference: urlly.it/3yq2n

4.6.1.3 R-30iB Mini Plus controller

For the operation of the robotic arm and direct communication with any external hardware or software, connection through a controller is required. The controller is responsible for properly managing the inputs and outputs to be transferred to the robot during the execution of an application or program.

There are six main internal elements of the controller [39]:

- a breaker, where the power terminals are located;
- a main CPU, where the processors and memories are located. The memory circuit can store programs and data set by the user in the C-MOS RAM, on the main CPU printed circuit board;
- a power supply unit (PSU), where the power supply is connected;
- an operation control circuit, that controls the servo amplifier which moves all the robot axes, including any additional axes, via the main CPU printed circuit board;
- an emergency board, to manage safeties and motor power.
- an input/output (I/O) circuit, that interfaces the controller with the peripheral units by receiving and sending signals via the I/O modules (I/O printed circuit boards).

The remote I/O signal is used for communication with the remote controller.

The user should use a teach pendant and operator's box to operate the controller.

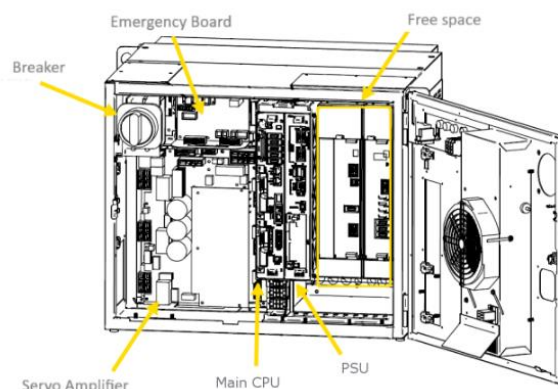


Figure 4.47: R-30iB Mini Plus controller.
Reference: Fanuc Italia S.r.l.

4.6.1.4 *iR*Vision 2D system

*iR*Vision is an intelligent vision system, developed by Fanuc, that can be fully integrated with the robot and allows it to guide the robot in its tasks.

The *iR*Vision system consists of the following elements:

- Camera and lens;
- Camera cable;
- Lighting devices, that are able to enhance and optimize the images to be taken;
- Camera multiplexer (if needed);

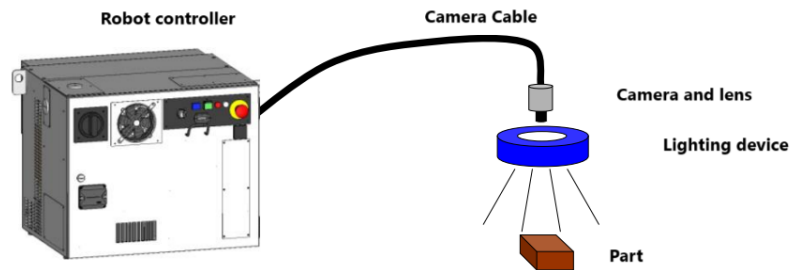


Figure 4.48: Elements of the *iR*Vision 2D system. Reference: Fanuc Italia S.r.l.

A camera can be installed either permanently or on the robot. The *iR*Vision system supports both of these camera installation methods.

A camera is defined as fixed when it is installed on a structure that cannot move. With this installation method, the camera's field of view is always the same, as is its distance from the framed objects. An advantage related to fixed camera installation is that the cycle time of the robot is not affected by the vision process because the *iR*Vision system performs it while the robot is working on another task [40].

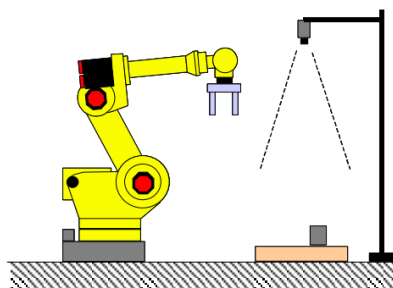


Figure 4.49: Camera in fixed position. Reference: Fanuc Italia S.r.l.

On the other hand, having the camera on board the robot implies that the camera is mounted on the robot's wrist. As the robot moves, detection of the position of the workpiece can be made at different points in space or at different distances. When using a camera mounted on the robot, the *iRVision* system calculates the position of the workpiece by taking into account the fact that the camera moves along with the robot [40].

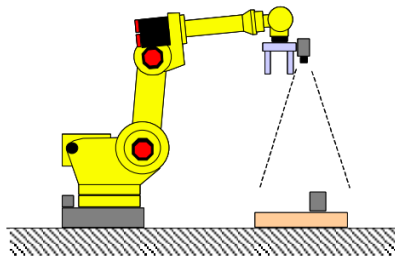


Figure 4.50: Camera mounted on the robot. Reference: Fanuc Italia S.r.l.

Fanuc robots typically reproduce programs created by learning performed with a programming device. This method of operation requires that the operations performed by the robot have been programmed in advance. This type of robot, therefore, can only perform operations that have been defined before the machine is commissioned. This means that if you want a robot to be able to manipulate all parts in the same way, they must always be placed in the same position. The *iRVision* system makes it possible to eliminate this restriction. *iRVision* detects the position of each part by means of cameras and intervenes in the robot's movement so that the robot can manipulate each part exactly as programmed, even if its position is different from that of the workpiece used when the robot program was created.

This 2D vision system can be used by the robot to [41]:

- perform visual image processing;
- perform picking and placing of randomly positioned and oriented parts;
- sort by colour, shape or many other features;
- read 1D and 2D barcodes;
- control completeness and dimensions;

- perform quality controls;

The specifications of the vision system are shown in Figure 4.51.

2D Camera	
Image Type	Grayscale / Color
LED Light for 2D Detection	Red / White / None
Image Resolution [pixel]	Grayscale : 1280 x 1024 Color : 640 x 512
Focal Length [mm]	8 / 12 / 16 / 25
Outer Dimension [mm]	80 x 131.8 x 74
Weight [kg]	0.6
Matching controller	R-30iB, R-30iB Plus R-30iB Mate, R-30iB Mate Plus R-30iB Compact Plus R-30iB Mini Plus

Figure 4.51: Specifications of the Fanuc iRVision 2D system. Reference: urly.it/3yq41

4.6.2 Payload settings

Since the application to be realized had to be able to lift and deposit different components of different weights, the creation and configuration of payload sheets was necessary.

Payload, or wrist load, is determined by:

- Mass;
- Centre of gravity;
- Inertia;

Payload data are expressed in reference to the triplet at the centre flange, as depicted in the Figure 4.52.

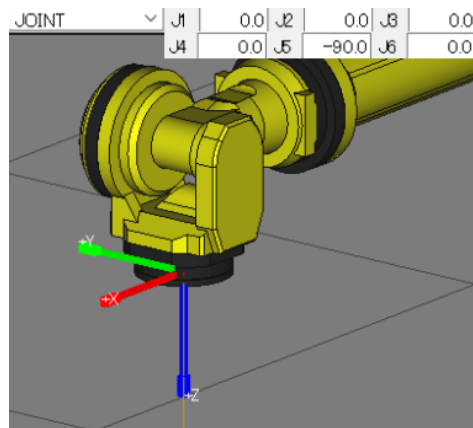


Figure 4.52: Reference system placed on the flange centre.
Fonte: Fanuc Italia S.r.l.

They are important to verify the transportability of the load, to make the simulation more reliable and to optimize the durability of the robot. A new payload sheet was created whenever the wrist load varied. Specifically, in this application there are 4 scenarios in which the payload varies: camera plus gripper unloaded, camera plus gripper with wheel, camera plus gripper with truck, camera plus gripper with board. It is worth mentioning that there are ten payload sheets that can be set, which by default are all set to zero with regard to parameters such as centre of gravity and inertia, while with regard

to weight they are set to the maximum supported load, which is 10 kg.

The first step was the creation of the payload sheet, with camera plus unloaded gripper. Initially, all the objects involved in the wrist load were weighed, i.e. the camera, the gripper, the quick changer and the interface plate. The sum of these weights was calculated, and it amounted to 2.07 kg, which was recorded in sheet 5, named gripper plus camera.

On the other hand, regarding the calculation of the new centre of gravity and inertias, a special procedure was required in this case. The steps were as follows:

1. Open the 3D drawing of the tool with a CAD software. In this case, SolidWorks was used;
2. Place a coordinate system on the counter flange, which was coincident with the backhoe in the flange, like shown in Figure 4.53;
3. Select the complete assembly, consisting of gripper, camera and interface plate;
4. Open the window of the mass properties, like shown in Figure 4.54;
5. Report the calculation to the reference system in flange, rather than to the origin of the drawing (see the orange box);
6. Make a note of the payload data highlighted (see the green boxes).

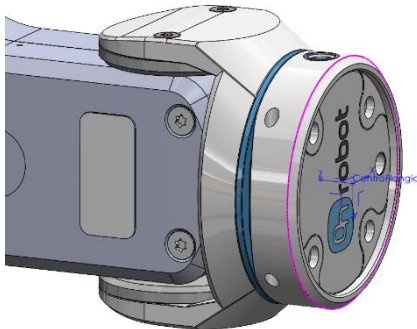


Figure 4.53: Coordinate system on the counter flange. Reference: edited by Francesco Succurro

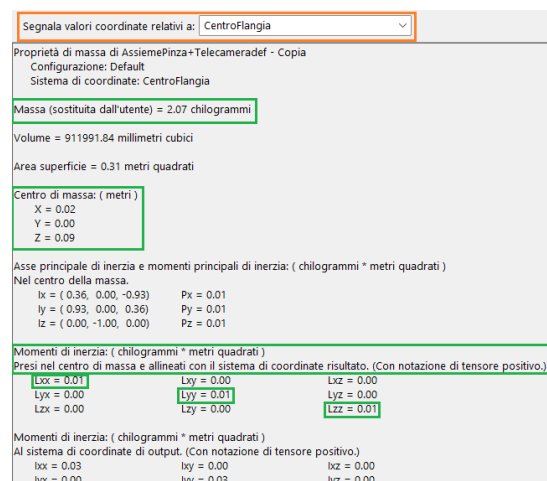


Figure 4.54: Mass properties of the system including interface plate, gripper, and camera. Reference: edited by Francesco Succurro

At this point, it was possible to complete the payload sheet and register it, as shown in the blue box in the Figure 4.55.

No	Payload[kg]	Comment	BarycentricPositionOfLoad[m]			LoadInertia[kgm ²]		
			X	Y	Z	X	Y	Z
1	0.00	EOAT w/o part	0.00	0.00	0.00	0.00	0.00	0.00
2	0.75	Staffe+telecamera	0.16	0.00	0.00	0.00	0.00	0.00
3	1.31	Pinza	0.00	0.00	0.12	0.00	0.00	0.00
4	2.00	Pinza camera	0.00	0.00	0.00	0.00	0.00	0.00
5	2.07	Pinza+camera	0.02	0.00	0.09	0.01	0.01	0.01
6	2.20	Ruota+PinzaCamera	0.03	0.00	0.10	0.01	0.01	0.01
7	2.36	Track+PinzaCamera	0.03	0.00	0.12	0.01	0.01	0.01
8	2.78	Tavola+PinzaCamer	0.02	0.00	0.15	0.00	0.01	0.01
9	10.00		0.00	0.00	0.00	0.00	0.00	0.00
10	10.00		0.00	0.00	0.00	0.00	0.00	0.00

Figure 4.55: Payload sheet number 5. Reference: edited by Francesco Succurro

For the other three payload sheets, however, a different method was used:

1. the weights of the different components were measured:
 - Wheel weight: 0.130 kg ;
 - Truck weight: 0.295 kg ;
 - Board weight: 0.705 kg .

Based on the payload sheet that was being created, the weight of the component was added to the weight of the gripper plus camera system, of 2.07 kg.

2. for the calculation of the inertias and the centre of gravity, a wizard available as a function of the robot was used: first the input of the payload, for example camera plus gripper with wheel, was requested. After that, the gripping of the object in question, in this case the wheel, was performed, and the robot was brought into three different configurations, required by the procedure itself. These three positions were recorded, and automatically the system was able to provide an accurate calculation regarding inertias and centre of gravity.

Case Study

No	Payload[kg]	Comment	BarycentricPositionOfLoad[m]			LoadInertia[kgm ²]		
			X	Y	Z	X	Y	Z
1	0.00	EOAT w/o part	0.00	0.00	0.00	0.00	0.00	0.00
2	0.75	Staffe+telecamera	0.16	0.00	0.00	0.00	0.00	0.00
3	1.31	Pinza	0.00	0.00	0.12	0.00	0.00	0.00
4	2.00	Pinaza camera	0.00	0.00	0.00	0.00	0.00	0.00
5	2.07	Pinza+camera	0.02	0.00	0.09	0.01	0.01	0.01
6	2.20	Ruota+PinzaCamera	0.03	0.00	0.10	0.01	0.01	0.01
7	2.36	Track+PinzaCamera	0.03	0.00	0.12	0.01	0.01	0.01
8	2.78	Tavola+PinzaCamer	0.02	0.00	0.15	0.00	0.01	0.01
9	10.00		0.00	0.00	0.00	0.00	0.00	0.00
10	10.00		0.00	0.00	0.00	0.00	0.00	0.00

Figure 4.56: Payload sheet number 6, 7, 8. Reference: edited by Francesco Succurro

This procedure was repeated for the truck and the board. The three payload sheets are shown in the blue box in the Figure 4.56.

4.6.3 Creation of the frames

In order to achieve the desired precision in movements in the Cartesian world and to use a well-defined system of reference with respect to which computing the working positions, it is necessary to correctly declare the dimensions of the tool (Tool Frame) and define the User Frame.

4.6.3.1 User Frames

The User frames were created using the calibration grid, with the four-points method. In the lab, the robot was driven manually upon four points of the grid so that it could register their position and automatically it was able to determine the corresponding User Frame. The points to be taken into account were: the origin of the x-axis parallel to the frame (Orient Origin Point), the point which specifies the positive direction of the x-axis (X Direction point), the point which specifies the positive direction of the y-axis (Y Direction Point), and the origin of the frame (System Origin).

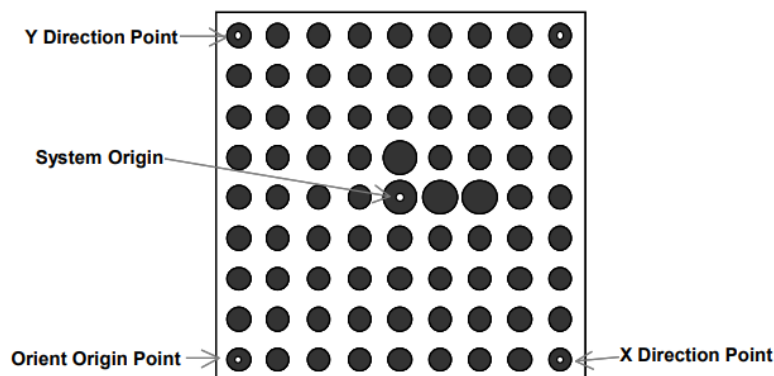


Figure 4.57: Calibration grid. Fonte: Fanuc Italia S.r.l.

To be as accurate as possible, the TCP was positioned on a tip, so that this could ideally be a well-defined point with which to record these four points. To do this, a Tool Frame was created with the four-point method so that it had its origin (TCP) on the tip of the Tool Master. The Tool Master supplied by Fanuc was used, which consists of a tip and a tailstock. This method was different from the four-points User Frame method.



Figure 4.58: Tip of the Tool Master. Reference: edited by Francesco Succurro



Figure 4.59: Tailstock of the Tool Master. Reference: edited by Francesco Succurro

In this case the four approach points must be taught with the tip of the Tool Master in grip between the two jaws of the gripper, touching a common point, that in this case is the tailstock, from four different approach statuses. The more different orientations of the point to be learned, the better the results will be in the calculation steps. Usually, four points are a significant amount to enable tool computation, but more may also need to be learned. In Figure 4.60 four different orientations of the reference point are shown as examples.



Figure 4.60: Four different orientations in which the tip of the Tool Master touches the tailstock. Reference: Fanuc Italia S.r.l.

As a result, not only the location (x, y, z) of the tool frame but also the deviation of positions of the tool tip at each approach point are automatically calculated. Users can improve inaccuracy of TCP, attributed to inaccurate teaching, by iteratively correcting indicated outliers. The setting value of the tool orientation (w, p, r) is the standard value (0, 0, 0).

Once the tool frame with origin (TCP) on the tip was originated, three different User Frames were created, using the four points method with calibration grid:

- the first one, called Banchetto, on the cart where the cobot is mounted, and the different components of the skateboard were present;
- the second one, called AGVscherma, on the portion of the AGV where the wheels and trucks trays were present;
- the third one, called AGVfondo, on the portion of the AGV where the boards tray was present;

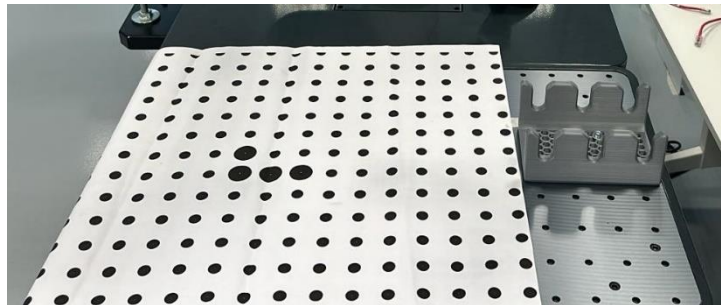


Figure 4.61: Calibration grid on the cobot cart.
Reference: edited by Francesco Succurro

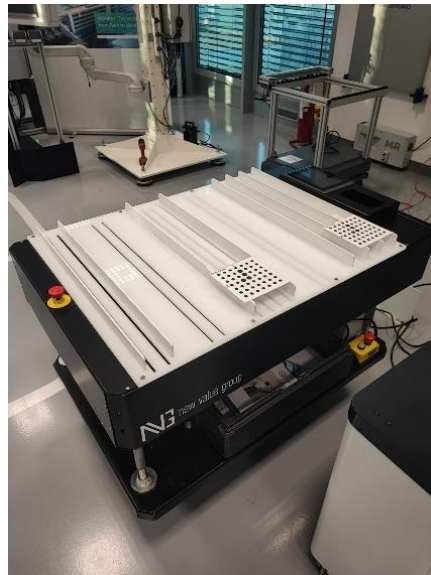


Figure 4.62: Calibration grid on the AGV.
Reference: edited by Francesco Succurro

Below, the Figure 4.63 shows an overview of all three different User Frames created.

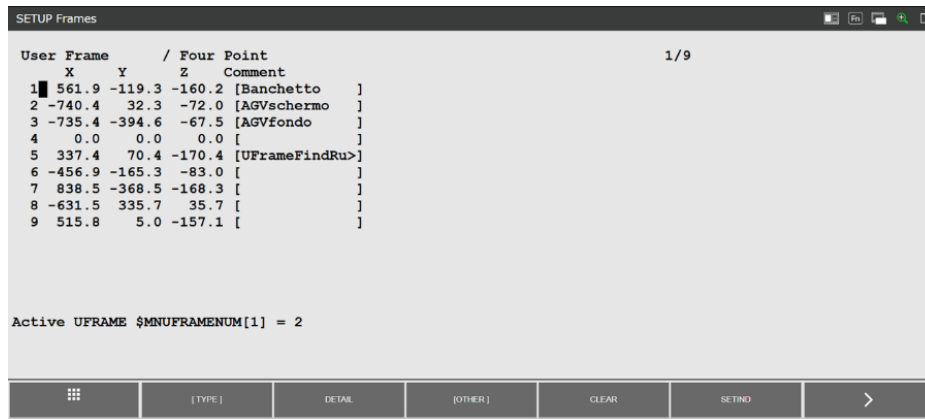


Figure 4.63: Overview of all User Frames. Reference: edited by Francesco Succurro

More details of each User Frame can be accessed, such as the location (x, y, z) of the User Frame, but also its orientation (w, p, r). In addition, it is possible to see the method used to create the frame, and the points recorded for its creation. Below in Figure 4.64 are reported, for example, details of the User Frame called Banchetto.

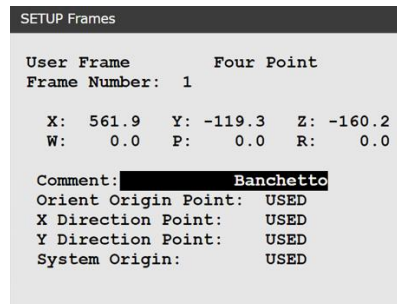


Figure 4.64: User Frame 1 – Banchetto. Reference: edited by Francesco Succurro

Likewise, the details of the other user frames created on the AGV, can be consulted, as shown in Figures 4.65 and Figure 4.66.

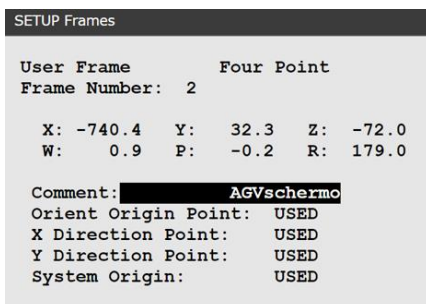


Figure 4.65: User Frame 2 – AGVscherma. Reference: edited by Francesco Succurro

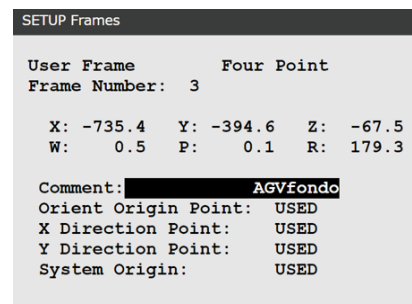


Figure 4.66: User Frame 3 – AGVfondo. Reference: edited by Francesco Succurro

4.6.3.2 Tool Frame

A tool coordinate system is a Cartesian coordinate system that defines the position of the tool centre point (TCP) and the attitude of the tool. On the tool coordinate system, the zero point usually represents the TCP and the Z-axis usually represents the tool axis. Tool coordinates include (x, y, z) indicating the position of the tool centre point (TCP), and (w, p, r) indicating the attitude of the tool. The tool centre point is used to specify the position data. The attitude of the tool is required to perform tool attitude control.

A special Tool Frame was created for this application, so that TCP was placed on the tip of the new gripping fingertips.

To do this, two steps were made:

1. First, it was figured out how much the offset was between the flange centre and the TCP positioned on the tip of the standard gripper fingertips. This information was obtained from a file provided by the OnRobot company (Figure 4.67), which, once filled in some of its parameters, provided the TCP offset directly, like shown in Figure 4.68.

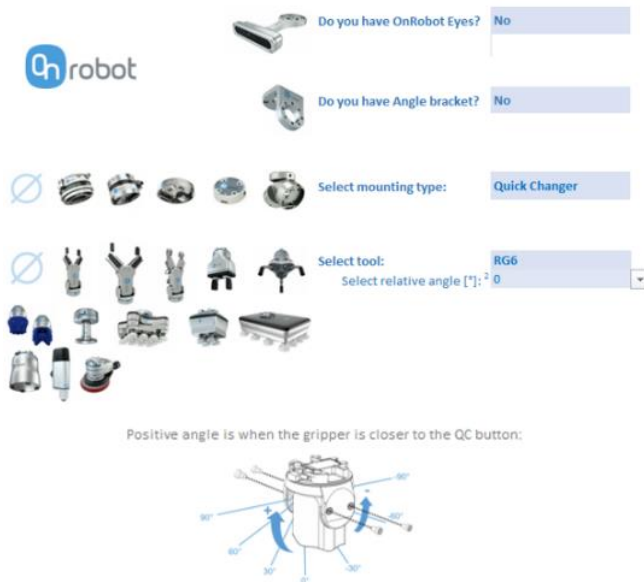


Figure 4.67: TCP calculator file.
Reference: OnRobot

	TCP Offset					
	x [mm]	y [mm]	z [mm]	Rx [rad]	Ry [rad]	Rz [rad]
Tool	0,0	0,0	278,6	0,000	0,000	0,000

Figure 4.68: TCP offset.
Reference: OnRobot

2. The distance along the z-axis (tool axis) between the tip of the old fingertips and the new ones was measured using a calliper. This distance was added to the TCP offset calculated in the first step, to have the offset of the new TCP, positioned at the tip of the new gripper fingertips, calculated with respect to the flange centre.

At this point Tool Frame 1 was created with the updated data, as shown in Figure 4.69.

SETUP Frames	
Tool Frame	Direct Entry
Frame Number:	1
1 Comment:	Toolframeruota
2 X:	0.000
3 Y:	0.000
4 Z:	322.000
5 W:	0.000
6 P:	0.000
7 R:	0.000
Configuration:	N D B, 0, 0, 0

Figure 4.69: Tool frame 1. Reference:
edited by Francesco Succurro

4.6.4 Calibration vision system

The *iR*Vision system locates the parts that appear in an image captured with the camera. To correct the position of the robot based on the information produced by the *iR*Vision system, this must be converted from the camera coordinate system to the robot coordinate system, user or tool coordinates. For this conversion to take place, data describing the position of the camera and its field of view are needed. This data constitutes the calibration of the camera. Calibrating the camera accurately is very important, since the accuracy with which the robot will manipulate the localized part depends on it. Grid calibration is the standard method of calibrating the camera and can be used for many vision applications. When a grid like the one shown in Figure 4.70 is framed by the camera, the *iR*Vision system automatically recognizes the positional relationship existing between the grid itself and the camera, as well as the lens distortion, focal distance, and other parameters.

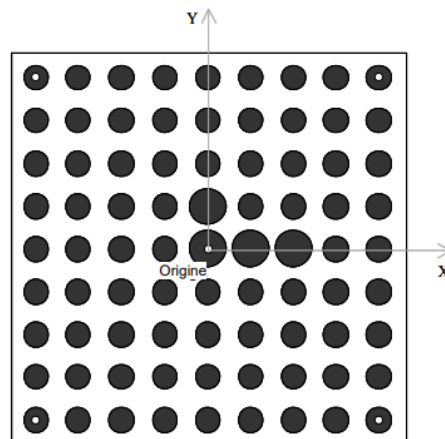


Figure 4.70: A calibration grid.
Reference: Fanuc Italia S.r.l.

All the black circles are arranged so that they are evenly distributed, both horizontally and vertically. Four of the circles are larger than the others and are positioned near the centre of the grid, referred to in the Figure 4.70 as the Origin, and indicate the directions of the coordinate system of the grid itself. The ratio of the diameter of a large circle to that of a small one is about 10:6. In addition, the black circles in the grid have a distance between the centres from each other that could be chosen according to the type of field being framed. A smaller distance between centres, means having more circles within the grid, so the camera stores more position data, and this results in higher accuracy in terms of locating a part.

Three calibrations were necessary for this application, corresponding to the areas where the three different User Frames were previously created:

- a calibration on the robot cart;
- a calibration on the portion of the AGV where the wheels and trucks trays were located;
- a calibration on the portion of the AGV where the boards tray was present.

Three calibration grids of different sizes were made using DraftSight software, to fit the field that the camera was to frame during the application. Specifically, the grid for the robot cart had a centre-to-centre distance of 45 mm, while those for the AGV were 15 mm one and 20 mm the other.

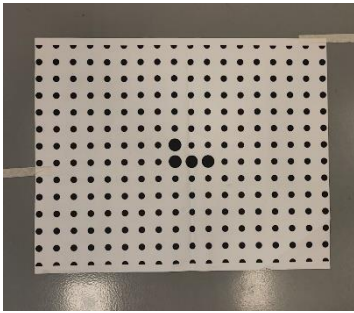


Figure 4.71: Calibration grid 45mm×45mm. Reference: Edited by Francesco Succurro

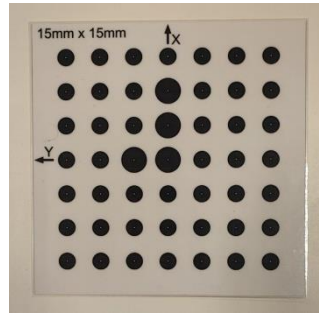


Figure 4.72: Calibration grid 15mm×15mm. Reference: Edited by Francesco Succurro

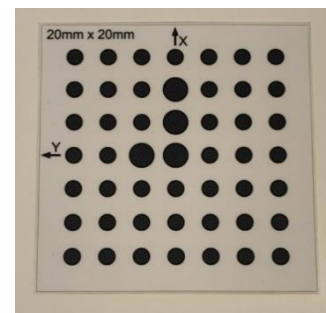


Figure 4.73: Calibration grid 20mm×20mm. Reference: Edited by Francesco Succurro

The position and height of the camera from which to perform each calibration was decided: typically, the choice of camera placement to perform the calibration coincides with the position where the photo will be taken during the robot program.

At this point the calibration wizard began. It started with the calibration on the robot cart.

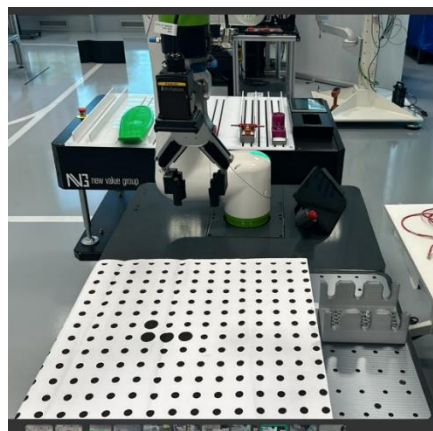


Figure 4.74: Calibration on the robot cart. Reference: edited by Francesco Succurro

In the first screen (Figure 4.75), the camera connected to the robot controller was selected, and the fact that it was mounted on board the robot. An appropriate exposure was chosen to make the image bright at the right point. It is important to remember that the larger the value is, the brighter a snapped image is. It was chosen whether to enable the LED light integrated in the camera package, and the aperture and focus of the camera were adjusted. Finally, it was selected a Grid Pattern Calibration method to perform the

camera calibration.

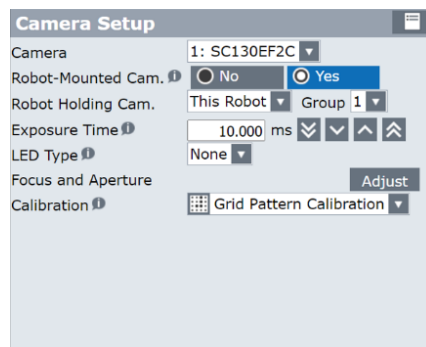


Figure 4.75: Camera setup.
Reference: edited by Francesco Succurro

In the second screen (Figure 4.76), the distance between the black circles of the created grid was selected, which in this case was 45 mm, and also that the calibration grid was not moved with respect to the user frame. It was necessary to set the arrangement position and direction of the calibration grid when performing calibration for the camera. To do this, it was chosen to use the value set in the user frame, to set the calibration grid frame.

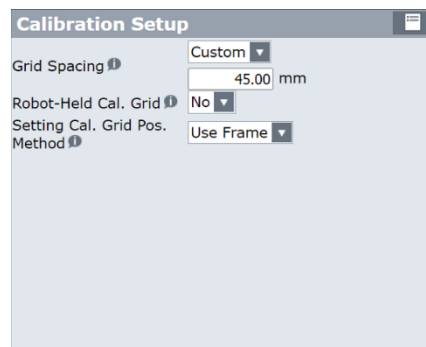


Figure 4.76: Calibration setup.
Reference: edited by Francesco Succurro

In the third screen (Figure 4.77), it was selected the User Frame number 1, called Banchetto, where the calibration grid frame was recorded. The position of the calibration grid relative to the User Frame selected was displayed.



Figure 4.77: Calibration grid location. Reference: edited by Francesco Succurro

In the fourth screen (Figure 4.78), the number of planes to be calibrated were chosen. In this specific case, a one-plane calibration was used because the photos during the robot program were always taken at the same positions with the camera always at the same height. It was selected an Override Focal Distance method to setup the camera distance: when calibration is performed using one plane by installing the grid pattern perpendicular to the camera's optical axis, the correct focal distance cannot be calculated in principle, so it is recommended to select and input the nominal focal distance of the lens, that for this camera was 8 mm.



Figure 4.78: Calibration. Reference: edited by Francesco Succurro

At this point the goal was to capture the image of the calibration grid, specifying the grid interval with the red rectangle displayed.

Case Study

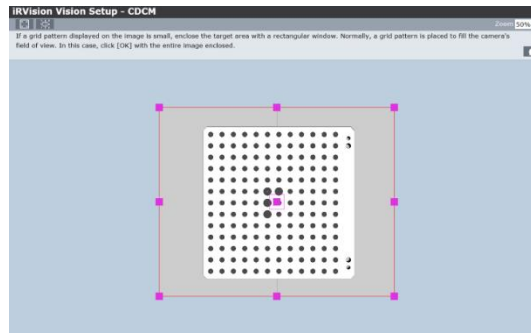


Figure 4.79: Capture of the calibration grid image. Reference: edited by Francesco Succurro

When the grid pattern was found successfully, crosshairs (+) appeared at the center of each of the found circles.

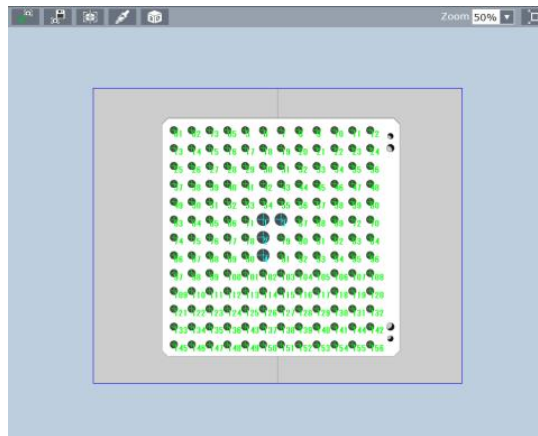


Figure 4.80: Grid calibration found. Reference: edited by Francesco Succurro

It was checked that blue crosshairs (+) appeared in the four large circles, and that green crosshairs (+) appear in small circles. Also red crosshairs (+) were displayed, showing the calculated position of where the calibration point should have been. Since green crosshairs (+) were plotted after red crosshairs (+), only a green crosshair (+) was visible if a green and a red crosshair (+) were plotted at the same position. The found calibration were checked. Calibration points that were incongruent with grid points could be eliminated at this stage.

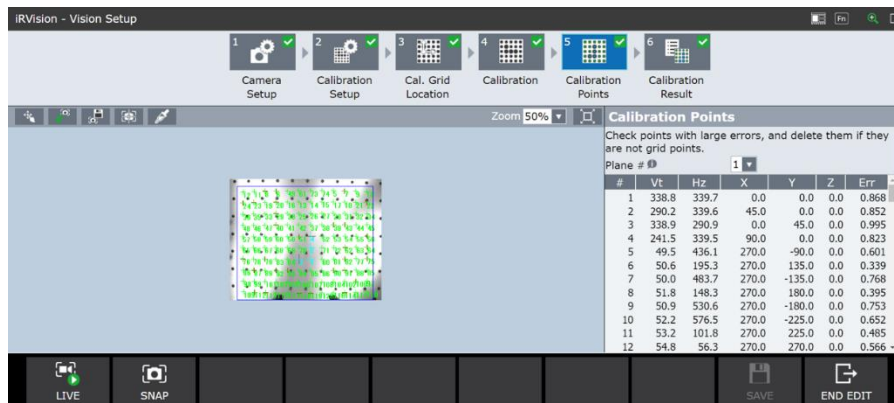


Figure 4.81: Calibration points. Reference: edited by Francesco Succurro

In the last screen (Figure 4.82), some important calibration results were displayed, such as:

- the calculated Standoff Distance, which is the distance between the centre of the length direction of the lens and the top surface of the grid pattern along the camera's optical axis;
- the calculated Lens Distortion coefficient. A larger absolute value indicates greater lens distortion;
- the Maximum/Mean error value, that refers to the average and maximum errors of all the calibration points.
- some data related to positioning of camera relative to the calibration grid, of calibration grid relative to the application User Frame used, of the robot holding the camera.

It was important to check if the standoff distance had an appropriate value, but also that the mean error value and the maximum error value were not too large. At this point the procedure for calibration was completed.

Calibration Result	
Focal Distance	8.00 mm
Standoff Distance	695.4 mm
Scale	0.921 mm/pix
Mean error value	0.440 pix
Maximum error value	1.038 pix
Position of Camera Relative to Cal. Grid	
X	78.5
Y	14.3
Z	695.3
W	0.3
P	0.1
R	-0.1
Position of Cal. Grid Relative to App. UFrame	
X	561.9
Y	-119.3
Z	-160.2
W	0.0
P	0.0
R	0.0
Position of Robot Holding Camera	
X	531.3
Y	-110.3
Z	608.6
W	-179.8
P	-1.1
R	-0.0

Figure 4.82: CALIBRAZIONE BANCHETTO. Reference: edited by Francesco Succurro

Through the same procedure, calibrations were performed on the AGV, which resulted in the following calibration results:

Calibration Result	
Focal Distance	8.00 mm
Standoff Distance	325.9 mm
Scale	0.431 mm/pix
Mean error value	0.164 pix
Maximum error value	0.409 pix
Position of Camera Relative to Cal. Grid	
X	-13.4 Y -1.6 Z 326.0
W	0.0 P 0.2 R -0.3
Position of Cal. Grid Relative to App. UFrame	
X	-740.4 Y 32.3 Z -72.0
W	0.9 P -0.2 R 179.0
Position of Robot Holding Camera	
X	-616.0 Y 35.5 Z 325.7
W	-179.7 P -1.1 R 178.7

Figure 4.83:
 CALIBRAZIONEAGVLATOSCHERMO.
 Reference: edited by Francesco Succurro

Calibration Result	
Focal Distance	8.00 mm
Standoff Distance	465.4 mm
Scale	0.616 mm/pix
Mean error value	0.109 pix
Maximum error value	0.283 pix
Position of Camera Relative to Cal. Grid	
X	0.9 Y -2.3 Z 465.4
W	0.1 P 0.1 R -0.2
Position of Cal. Grid Relative to App. UFrame	
X	-735.4 Y -394.6 Z -67.5
W	0.5 P 0.1 R 179.3
Position of Robot Holding Camera	
X	-626.3 Y -388.1 Z 471.2
W	-179.7 P -1.1 R 179.0

Figure 4.84:
 CALIBRAZIONEAGVTAVOLESKATE.
 Reference: edited by Francesco Succurro

4.6.5 Robot Program creation

At this stage, the work focused on writing the robot program. The first phase was the design phase, in which the goals of the program were defined in detail, and moreover the critical aspects of such an application and how they could be overcome were analysed. In this specific case, the robot had to be able, through the vision system, to locate four wheels and two trucks of the skateboard, placed randomly on the cart, and deposit them correctly on the trays placed on the AGV. Similar subject for the pick & place of the board, except that it was placed on the cart in a fixed, and not random position each cycle, and thus did not require localization by the vision system.

The critical aspect of the application lied in the deposition of parts on the MIZUBOT 4.0: the trays on which the parts were deposited, were free to slide along the channels of the AGV, and thus after each deposition, following the release of the part, the tray may have moved from its original position. In addition, the AGV, after unloading the parts to the operator, returned to the loading position next to the robot at each cycle, with a small offset in position and orientation. It was therefore necessary to compensate, through the vision system, these two offsets, to ensure accurate and precise depositing of parts into the trays.

The second phase was the implementation phase, the one in which, in fact, the program came to life. Programming was carried out according to a top-down development methodology: the highest or TOP level is the description of the main problem-solving process, by fundamental description of its basic steps called subproblems. Each subproblem is detailed separately. It means going down from the general to the particular. Such a program development made the work of analysis and design more streamlined: it was better to divide the whole application into simpler subprograms, so that in case some changes had to be made, one would only modify the subproblem concerned without touching the rest. Moreover, such a development methodology makes it easier for any person to read and understand the solving algorithm.

4.6.5.1 Main Program

The main of the program is shown in the Figure 4.85: there are calls to other subprograms and decision constructs, which depend on some counters, and which are able to repeat some parts of the main iteratively, or to switch to the execution of other parts of the program.

```
THEISISPROGRAM
1: R[20]=0
2: LBL[10]
3: PR[26,3]=100
4: PR[23,1]=0
5: CALL FINDPICKPLACETRUCK
6: CALL SVINCOLOTRUCK
7: CALL SCARICOTRUCKSDEF
8: R[20]=R[20]+1
9: IF R[20]<2,JMP LBL[10]
10: R[11]=0
11: LBL[2]
12: PR[23,1]=0
13: PR[23,3]=120
14: CALL FINDPICKPLACERUOTA
15: CALL SVINCOLORUOTA
16: CALL SCARICORUOTE
17: R[11]=R[11]+1
18: IF R[11]<4,JMP LBL[2]
19: CALL RICERCAVASSOIOTAVOLA
20: CALL PICKTAVOLA
21: CALL SVINCOLOTAVOLA
22: CALL SCARICOTAVOLA
[End]
```

Figure 4.85: Main program.
Reference: edited by Francesco
Succurro

It can be seen that the application was handled in three macro subproblems, in which it was decided to handle separately and in a specific order the picking up and depositing of components: first the trucks, then the wheels, and finally the board.

4.6.5.1.1 Trucks program

Starting with the trucks, programming focused on creating a system that would be able to recognize the components, randomly arranged on the cobot cart, and subsequently pick them up once their positions were located. Indeed, it is possible to notice a CALL instruction to a subprogram called FINDPICKPLACETRUCK. The CALL instruction is responsible for moving the flow of code execution, from the program in which the

instruction is present, to the first line of another program (subprogram). Below in the

Figure 4.86, the FINDPICKPLACETRUCK subprogram is shown.

```

FINDPICKPLACETRUCK
1: PAYLOAD[5]
2: UFRAME_NUM=1
3: UTOOL_NUM=1
4: PR[1]=LPOS-LPOS
5: R[12]=0
6: R[13]=0
7: LBL[100]
8: PR[1,1]=150*R[13]
9: PR[1,2]=120*R[12]
10: J P[1:Snap position] 100% FINE Offset,PR[1]
11: R[1]=120
12: R[2]=80
13: R[3]=0
14: CALL COMMAND_GRIP
15: VISION RUN_FIND 'CERCATRUCK'
16: VISION GET_OFFSET 'CERCATRUCK' VR[2] JMP LBL[1]
17: L P[2] 100mm/sec FINE VOFFSET,VR[2] Offset,PR[23]
18: L P[2] 100mm/sec FINE VOFFSET,VR[2]
19:
20: PAYLOAD[7]
21: R[1]=80.4
22: R[2]=120
23: R[3]=0
24: CALL COMMAND_GRIP
25: WAIT 1.25 (Sec)
26: L P[2] 100mm/sec FINE VOFFSET,VR[2] Offset,PR[23]
27: JMP LBL[5]
28:
29:
30: LBL[1]
31: R[12]=R[12]+1
32: IF R[12]<3,JMP LBL[100]
33: R[12]=0
34: R[13]=R[13]+1
35: IF R[13]<3,JMP LBL[100]
36:
37:
38:
39:
40:
41: LBL[5]
[End]

```

Figure 4.86: FINDPICKPLACETRUCK program. Reference: edited by Francesco Succurro

The first important thing to do was to set the correct payload carried by the robot and the frames in which to record the robot's movement instructions. At first, payload 5 was selected, which corresponded to that of the camera plus the unloaded gripper, because the robot did not pick up any components yet. As for the frames, Tool Frame 1, with TCP at the centre of the gripper fingertips, and User Frame 1, which corresponded to the reference system created on the plane of the cobot cart, were selected. It is important to remember that the robot positions that are recorded within a program always refer to the positioning of the tool with respect to the chosen User Frame. In this way it was very intuitive to check whether the coordinates of the gripper with respect to the plane corresponded to a recorded position.

It is possible to see at line 4 that a position register PR[1] was initialized by LPOS-LPOS subtraction, so that all position register elements were equal to zero. A position register is a variable composed of a series of position register elements, that can hold position data (x, y, z, w, p, r). A position register axis instruction PR[i,j] performs an arithmetic operation on position register elements. i of PR[i,j] represents a position register number, and j of PR[i,j] represents a position register element number [42].

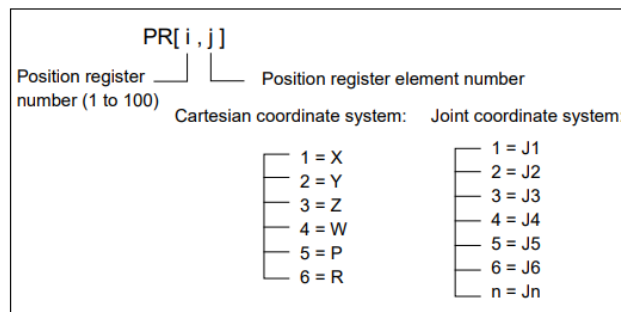


Figure 4.87: Position register. Reference: Fanuc Italia S.r.l.

After that, two registers R[12] and R[13], that are variables to hold integers, were also initialized to zero. At this point the work focused on recognizing and locating the trucks on the cart plane. Since part of the plane was occupied by the tray containing the boards, like shown in the Figure 4.88, and since the boards always had the same and fixed position at each cycle, it was decided to restrict the area of truck recognition to the remaining portion of the cart.

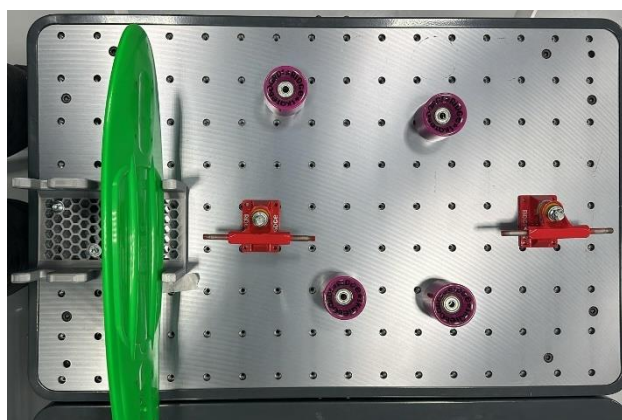


Figure 4.88: Banquet of the cobot on which the trucks, wheels, and board were located. Reference: edited by Francesco Succurro

It was decided to ideally divide the portion where the trucks and wheels were arranged, into nine rectangles, equal with each other, and to take a picture for each of these rectangles of space, at the same height. In the Figure 4.89, the order in which photos were taken is shown.

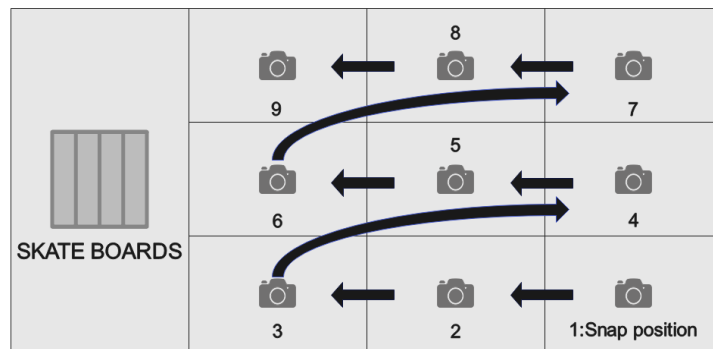


Figure 4.89: Order in which the photos were taken.
Reference: edited by Francesco Succurro

The goal was to locate, within this area, the two trucks and to pick them up correctly. The choice of taking multiple photos, as opposed to a single one that framed the entire field, stemmed from the fact that in this way the location of the piece was found to be extremely precise and accurate.

To do this, a loop was implemented, always starting at LBL[100]. Specific values were assigned to the first and second elements of PR[1], as a product of a constant and a register.

$$PR[1,1] = 150 * R[13]$$

$$PR[1,2] = 120 * R[12]$$

After that, the initial position, where to take the first photo, was recorded as P[1:Snap position]. The robot had to go to this position through one motion per joint (J), at its maximum execution speed (100 %), and stopping for a moment, trough the instruction FINE, before continuing to the next point. To this position, as suggested by the instruction Offset, PR[1], a positional offset was added, which consisted of going to sum at each cycle what was contained in the registers elements numbers of PR[1] to the original recorded snap position P[1]. The position register elements of PR[1] were based

on the two registers R[12] and R[13]. Depending on their values, they were able to modify and change the snap position. In this case, during the first three cycles R[13] remained fixed at 0, while R[12] varied from 1 to 3, thus allowing three pictures to be taken on the same line 120 mm apart. After that, for the next three cycles, R[13] was incremented to 1 and then kept fixed, causing R[12] to vary from 1 to 3, thus allowing three more photos to be taken 120 mm apart again, on the row placed 150 mm from the one before. And again for the last three cycles, R[13] was increased to 2 and then kept fixed, causing R[12] to vary from 1 to 3. To summarize, R[12] was responsible for a shift along the same row, R[13] for a shift from one row to the next one.

Each time the photo was taken, a vision process was executed, as suggested by the line 15, with the command VISION RUN_FIND: in particular, a vision function called CERCATRUCK was created, so that depending on the geometry of the object, it was able to locate the part.

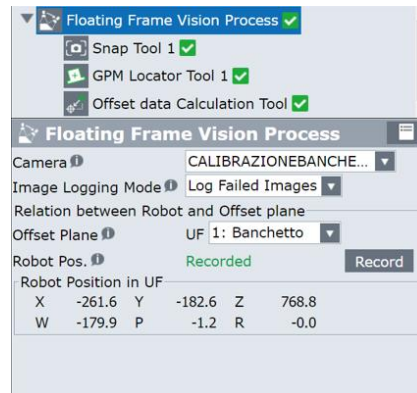


Figure 4.90: CERCATRUCK vision function. Reference: edited by Francesco Succurro

At first, the type of viewing process used was selected. In this case, since multiple photos were being taken at different positions in the plane, a Floating Frame Vision process was chosen. The Floating Frame Vision Process can find targets with the robot-mounted camera and offsets the robot. By setting the relationship between the offset plane and the camera in advance and carefully keeping it while in measurement, it allows measuring the targets with various robot postures. Since the vision process is

applied on the cart on which the cobot is mounted, in our case User Frame number 1, called Banchetto, was chosen as the offset plane. For the same reason, the calibration used for the camera was the one called CALIBRAZIONEBANCHETTO.

At this point after adjusting the brightness of the image, through the exposure and LED lighting system, the camera was taught the model pattern of the workpiece it needed to be able to recognize. To do this a GPM Locator Tool was used. The GPM locator tool employs the image processing tool that is the core of *iR*Vision. It checks a camera-captured image for the same pattern as a model pattern taught in advance and outputs its location.



Figure 4.91: GPM Locator Tool for the CERCATRUCK vision function.
Reference: edited by Francesco Succurro

It is possible to see from the Figure 4.91 that the reference model chosen to locate the truck was its rectangular base. A Training Mask was used to remove from the pattern any unnecessary items in the background, any unwanted or incorrect features of the model, or any blemishes.

To understand whether the selected model pattern was good for component recognition, evaluation results of different parameters regarding Location, Orientation and Scale of the taught model pattern were displayed on the following three levels:

- Good: Can be found stably;
- Poor: Cannot be found very stably;
- None: Cannot be found.

After determining that the model was good, some search parameters for the part in the image were set, such as:

- the Score Threshold, that is the accuracy of the found result expressed by a score. The target line is successfully found if its score is equal to or higher than this threshold value. If the score is lower, the target line is not found. The range of this parameter is between 10 and 100. The default value is 70. Setting a small value might lead to inaccurate location;
- the Contrast Threshold, that specifies the threshold for how much contrast (difference between light and dark) is required in the image to identify something as a feature. The default value is 50. If a small value is set, objects that cannot be seen clearly can be detected, but processes will take time. On the other hand, if something with low contrast is mistakenly detected, it is recommended to increase this value. Image features, whose contrast is lower than the threshold, are ignored;
- the Orientation, that specifies the range of orientation subject to be searched. The tool searches for a model pattern rotated in the range specified by [Min.] and [Max.], with the orientation of the taught model pattern being 0 degrees. The narrower the orientation range is, the faster the search process ends. By default, the orientation search is enabled and the range is from -180 to $+180$ degrees.

In this case by setting a Score Threshold of 70% and a Contrast Threshold of 50 it was possible to accurately find trucks on the plane. Orientation was enabled, since the trucks could be placed on the plane in different orientations.

In the final stage of the vision function, the reference position was set, as shown in the Figure 4.92.

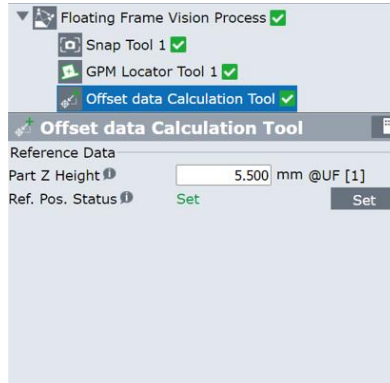


Figure 4.92: Reference position of the CERCATRUCK vision function. Reference: edited by Francesco Succurro

The workpiece was placed in a reference position, which was recorded. The positional offset was calculated based on the relationship existing between the reference position set here and the position where the workpiece was located. The height of the reference model that had to be searched and located was also entered. In this case, the height of the truck base was measured, and it was 5.5 mm relative to User Frame 1, which in this case was the offset plane.

At this point, having finished creating the vision function, keeping the part at the reference position just registered, the approach and pick positions of the component were recorded. The approach position precedes the pick position and is useful to prevent that the picked object collides in the ascent against some other component positioned nearby. The approach position is obtained by using the same pick position $P[2]$, but shifted along the positive z-axis, due to the positional offset of the $PR[23]$. It can be seen from lines 17 and 18, how both the pick and approach positions contain the $VOFFSET$, $VR[2]$ instruction. Each time the robot took the picture, and located a component, it recorded a positional offset, that is the difference between the found object with respect to its reference position in a vision register. The vision register has the same structure as a position register with the difference that it stores positional data derived from vision.

Thus, by applying the positional offset stored in the VR[2], the robot was able to correct the pick and approach position whenever a part was located.

As can be seen from the program, the opening and closing of the gripper is handled through a CALL to a specific drive program, called COMMAND_GRIP. Before making this call, however, three registers R[1], R[2], R[3] must be set, which corresponded to:

- $R[1] = \textit{width}$
- $R[2] = \textit{force}$
- $R[3] = \textit{wait}$

R[1] corresponds to the desired opening/closing of our gripper, R[2] corresponds to the force with which we want the gripper to open/close the jaws, and R[3] corresponds to a wait for a certain condition, before performing the opening or closing of the gripper.

In this case, a gripper opening of 80.4 mm was chosen to grasp the truck, at the maximum of its available force 120 N, in order to be sure that it held the objects firmly during its path to the AGV. No wait condition was enabled, so R[3] was set to 0. It was necessary to enable the PAYLOAD [7], like shown at line 20, corresponding to the one with the camera plus gripper system with the truck, to ensure proper transport of the part. In addition, a WAIT 1.25 (sec) instruction was added to line 25: this condition is used to ensure that the moment the gripper arrives at the pick position, it grabs the part and waits a specified amount of time before ascending back to the approach position, thus preventing the gripper from ascending without having picked up the part.

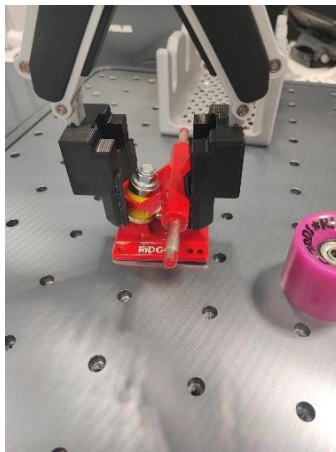


Figure 4.93: Truck pick phase. Reference: edited by Francesco Succurro

At this point, after correctly locating the component and picking it up, the main program would call another subprogram called SVINCOLOTRUCK.

```
SVINCOLOTRUCK  
  
1: PAYLOAD[7]  
2:J P[1] 100% CNT100  
[End]
```

Figure 4.94: SVINCOLOTRUCK program.
Reference: edited by Francesco Succurro

This subprogram contained a point registered as P[1], toward which the robot should head by moving by Joint, at its maximum speed, and approaching the target position without stopping and then continuing to the next point. This point was located midway between the cart on which the cobot is mounted and the AGV. It was used to force the robot to pass by following a well-defined trajectory, and not finding it on its own automatically.



Figure 4.95: Midpoint between the cobot cart and the AGV. Reference: edited by Francesco Succurro

After that, the program focused on depositing the component into the trays, placed on the AGV. The program invoked another subprogram called SCARICOTRUCKSDEF. Its structure is shown in the Figure 4.96.

```

SCARICOTRUCKSDEF
1:  PAYLOAD[7]
2:  UFRAME_NUM=2
3:  UTOOL_NUM=1
4:  PR[26,1]=0
5:J  P[2] 100% FINE
6:  VISION RUN_FIND 'VASSIOTRUCKS'
7:  VISION GET_OFFSET 'VASSIOTRUCKS' VR[6] JMP LBL[1]
8:  IF R[20]=0,JMP LBL[8]
9:  IF R[20]=1,JMP LBL[7]
10:
11:  LBL[8]
12:L P[3] 100mm/sec FINE VOFFSET,VR[6] Offset,PR[26]
13:L P[3] 100mm/sec FINE VOFFSET,VR[6]
14:
15:
16:  PAYLOAD[5]
17:  R[1]=86
18:  R[2]=80
19:  R[3]=0
20:  CALL COMMAND_GRIP
21:  WAIT 1.25(sec)
22:L P[3] 100mm/sec FINE VOFFSET,VR[6] Offset,PR[26]
23:  JMP LBL[5]
24:  LBL[7]
25:  PR[26,1]=76
26:L P[3] 100mm/sec FINE VOFFSET,VR[6] Offset,PR[26]
27:  PR[26,3]=0
28:L P[3] 100mm/sec FINE VOFFSET,VR[6] Offset,PR[26]
29:  PAYLOAD[5]
30:  R[1]=86
31:  R[2]=80
32:  R[3]=0
33:  CALL COMMAND_GRIP
34:  WAIT 1.25(sec)
35:  PR[26,3]=100
36:L P[3] 100mm/sec FINE VOFFSET,VR[6] Offset,PR[26]
37:
38:
39:
40:
41:  LBL[5]
[End]

```

Figure 4.96: SCARICOTRUCKSDEF program.
Reference: edited by Francesco Succurro

In this case, the frames that were used to record the positions of the robot were, the Tool Frame 1, and the User Frame 2, which corresponded to the reference system created on the portion of the AGV where the trucks trays were present. At this point the problem was to find the location of the tray, where to deposit the component. This was free to be placed at the beginning of the production cycle randomly along the channel, but it was also free to move along the channel as components were deposited. In addition, the AGV at every end of the cycle, returned to the loading location, ready to start a new cycle, with a small offset in position and orientation from its original recorded loading position.

Since the tray had a rather complex geometric shape to be recognized, it was planned to have the vision system recognize a specific marker, such as the one shown in Figure 4.97, which was applied directly to the tray. The marker was placed on the tray as in Figure 4.98 so that it would continue to be visible even after the first component was deposited, and thus the tray could be located.



Figure 4.97: Marker.
Reference: Fanuc
Italia S.r.l.



Figure 4.98: Trucks tray with the marker applied. Reference: edited by Francesco Succurro

Then a location P[2] was recorded, where the robot had to go linearly, to take a picture and look for the tray marker.



Figure 4.99: P[2] Snap Position. Reference: edited by Francesco Succurro

A vision function, called VASSOIOTRUCKS, was created, in which the camera was taught the model pattern of the marker it was to be able to recognize. Firstly, since only one picture was being taken always at the same position, a 2-D Single-View Vision Process was chosen. The 2D Single-view Vision Process can detect the workpiece on a plane with a camera and offset the robot position, depending on the parallel movement (X, Y) direction or the rotational movement (R) direction of the workpiece with respect

to the offset frame. Since the vision process was applied on the AGV, in the area where the trucks tray was present, User Frame number 2, called AGVschermo, was chosen as the offset plane. For the same reason, the calibration used for the camera was the one called CALIBRAZIONEAGVLATOSCHERMO, like shown in the Figure 4.100.

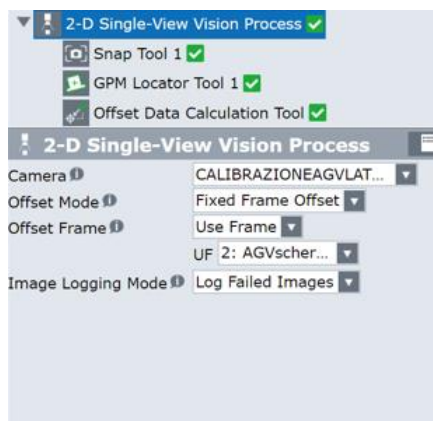


Figure 4.100: VASSIOTRUCKS vision function. Reference: edited by Francesco Succurro

At this point, a GPM Locator Tool was used to define the pattern to be recognized.

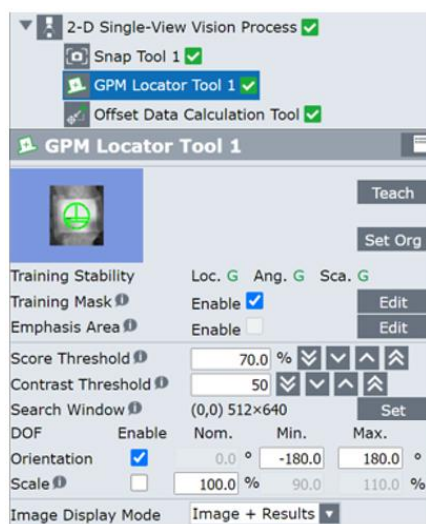


Figure 4.101: GPM Locator Tool for the VASSIOTRUCKS vision function. Reference: edited by Francesco Succurro

It is possible to see from the Figure 4.101, how the green lines that were selected represented what the camera should have been able to recognize. Thanks to a Training

Mask, it was possible to focus the search by the vision system, only on the marker. Also, in this case a Score Threshold of 70% and a Contrast Threshold of 50 were able to guarantee that the marker would be located correctly, under any lighting condition. Orientation was enabled because the tray, despite being arranged along a guide, could rotate from its original reference position, since the AGV when returning to the loading position could have a phase shift in orientation.

In the final stage of the vision function, the trucks tray was placed in a reference position, that was recorded.

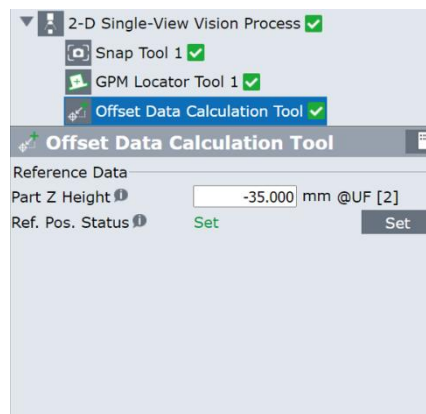


Figure 4.102: Reference position of the VASSOIOTRUCKS vision function. Reference: edited by Francesco Succurro

The height at which the marker was located with respect to the User Frame 2 was entered, and it was -35 mm. It was a negative quantity, as the reference system of User Frame 2 was located on the top of the channel banks, and not directly coincident with the plane of the AGV.

After completing the creation of the vision function, the tray was held at the newly recorded reference position. Then, while holding the truck, the place position of the component in the tray was recorded. It is possible to see along the program code, that thanks to the setting of PR[26,3] and thanks to the Offset, PR[26] instruction, it was possible to take advantage of the recorded place position P[3], both as the place itself of the component, but also as the approach position, thus avoiding defining two different

positions. The approach and place positions contained also the VOFFSET, VR[6] instruction. Applying the positional offset stored in the VR[6], the robot was able to modify and correct the place position, based on where the tray was positioned when the picture was taken. It is also possible to see that two branches called LBL[7] and LBL[8] were implemented within the code to handle the unloading of the first truck and of the second one. Again, the PR[26,1] setting and the Offset, PR[26] instruction were useful to handle the unloading of the second truck, reusing the place position of the first one, thus avoiding the need to record new positions.

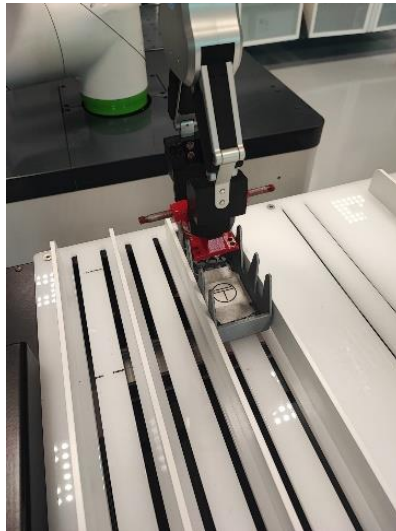


Figure 4.103: Truck place phase.
Reference: edited by Francesco Succurro

4.6.5.1.2 Wheels program

Arrived at this point, after handling the picking and placing of trucks, the focus shifted to the wheels. In fact, the main program recalled the FINDPICKPLACERUOTA subprogram, in order to locate and pick up the four wheels.

```

FINDPICKPLACERUOTA
1: PAYLOAD[5]
2: UFRAME_NUM=1
3: UTOOL_NUM=1
4: PR[1]=LPOS-LPOS
5: R[12]=0
6: R[13]=0
7: LBL[100]
8: PR[1,1]=150*R[13]
9: PR[1,2]=120*R[12]
10:J P[1:Snap position] 100% FINE Offset,PR[1]
11: R[1]=120
12: R[2]=80
13: R[3]=0
14: CALL COMMAND GRIP
15: VISION RUN_FIND 'CERCARUOTA'
16: VISION GET_OFFSET 'CERCARUOTA' VR[2] JMP LBL[1]
17:L P[3] 100mm/sec FINE VOFFSET,VR[2] Offset,PR[23]
18:L P[3] 100mm/sec FINE VOFFSET,VR[2]
19:
20: PAYLOAD[6]
21: R[1]=80.4
22: R[2]=120
23: R[3]=0
24: CALL COMMAND GRIP
25: WAIT 1.25(sec)
26:L P[3] 100mm/sec FINE VOFFSET,VR[2] Offset,PR[23]
27: JMP LBL[5]
28:
29:
30: LBL[1]
31: R[12]=R[12]+1
32: IF R[12]<3,JMP LBL[100]
33: R[12]=0
34: R[13]=R[13]+1
35: IF R[13]<3,JMP LBL[100]
36:
37:
38:
39:
40:
41: LBL[5]
[End]

```

Figure 4.104: FINDPICKPLACERUOTA program. Reference: edited by Francesco Succurro

Again, the reasoning was the same as that made for the trucks: more photos were taken on the cobot cart in order to make wheels recognition, and what has been changed is the vision function implemented, called CERCARUOTA. As with the trucks, the reference system that was used as the offset plane was the User Frame 1, and the camera was set up with the calibration called CALIBRAZIONEBANCHETTO, like shown in Figure 4.105.

Case Study

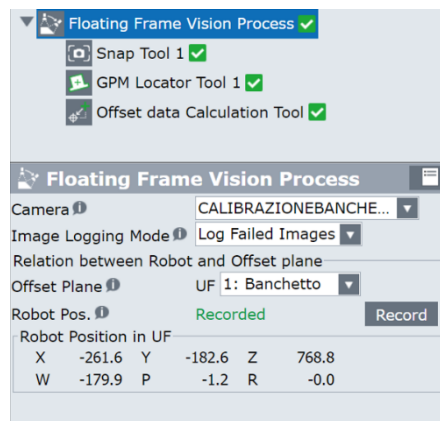


Figure 4.105: CERCARUOTA vision function. Reference: edited by Francesco Succurro

The reference model that was chosen to be recognized, was the inner circular metal shield of the wheel, as shown in Figure 4.106.



Figure 4.106: GPM Locator Tool for the CERCARUOTA vision function. Reference: edited by Francesco Succurro

By setting a Score Threshold of 70 % and a Contrast Threshold of 50, it was possible to selectively have only the inner metal ring of the wheel recognized, with no circular-shaped holes on the surface being recognized as well. In this case, the part orientation search was not enabled, since the chosen reference model was nothing more

than a circle, which had no defined orientation.

At this point, as done for the trucks, a reference position was chosen and subsequently recorded.

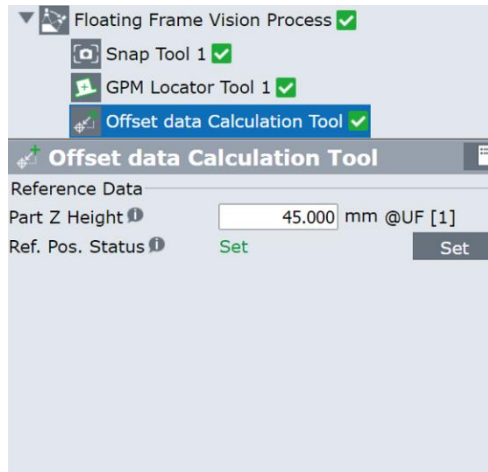


Figure 4.107: Reference position of the VASSOIOTRUCKS vision function.
Reference: edited by Francesco Succurro

The height of the wheel relative to User Frame 1 was measured, 45 mm, and entered. Again, keeping the wheel in the reference position, the pick position was recorded, so that this position could be exploited to also create the approach position through the setting of PR[23,3] and the Offset, PR[23] instruction.

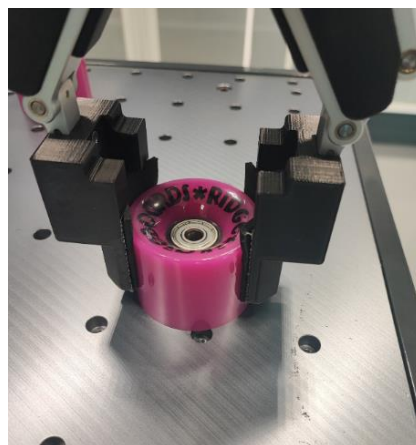


Figure 4.108: Wheel pick phase.
Reference: edited by Francesco Succurro

At this point, having located and picked up the wheel, the main program invoked the SVINCOLORUOTA subprogram to force the robot to travel a certain trajectory and pass through a certain position.

```
SVINCOLORUOTA  
  
1:  PAYLOAD [6]  
2: J  P[1] 100% CNT100  
[End]
```

Figure 4.109: SVINCOLORUOTA program. Reference: edited by Francesco Succurro

It is possible to note that it was set PAYLOAD [6], because the robot carried not only the camera plus gripper, but also the wheel on its wrist.

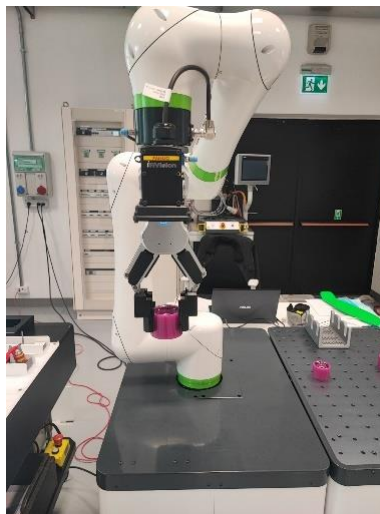


Figure 4.110: Midpoint between the cobot cart and the AGV. Reference: edited by Francesco Succurro

At this point the problem was again to locate the wheels tray and deposit the components inside it. To do this, the subprogram called SCARICORUOTE was implemented.

```

SCARICORUOTE
1: PAYLOAD[5]
2: UFRAME_NUM=2
3: UTOOL_NUM=1
4: J P[1] 100% FINE
5: VISION RUN_FIND 'RICERCAVASSOIORUOTA'
6: VISION GET_OFFSET 'RICERCAVASSOIORUOTA' VR[1] JMP LBL[1]
7: LBL[4]
8: IF R[11]<2,JMP LBL[2]
9: IF R[11]>=2,JMP LBL[3]
10: LBL[2]
11:L P[4] 100mm/sec FINE VOFFSET,VR[1] Offset,PR[23]
12:L P[4] 100mm/sec FINE VOFFSET,VR[1]
13:
14: PAYLOAD[5]
15: R[1]=100
16: R[2]=80
17: R[3]=0
18: CALL COMMAND_GRIP
19: WAIT 1.00(sec)
20:L P[4] 100mm/sec FINE VOFFSET,VR[1] Offset,PR[23]
21: R[15]=R[15]+1
22: END
23: LBL[3]
24: PR[23,1]=65.5
25:L P[2] 100mm/sec FINE VOFFSET,VR[1] Offset,PR[23]
26: PR[23,3]=0
27:L P[2] 100mm/sec FINE VOFFSET,VR[1] Offset,PR[23]
28: R[1]=100
29: R[2]=80
30: R[3]=0
31: CALL COMMAND_GRIP
32: WAIT 1.00(sec)
33: PR[23,3]=120
34:L P[2] 100mm/sec FINE VOFFSET,VR[1] Offset,PR[23]
35: R[15]=R[15]+1
36:
[End]

```

Figure 4.111: SCARICORUOTE program. Reference: edited by Francesco Succurro

Here again a marker was used, with a slightly different geometric shape this time, like shown in the Figure 4.112, so that it could be distinguished from the one applied to the trucks tray. This is because the robot, in the process of locating the wheels tray, was going to take a picture at the same location recorded to search for the trucks tray. So, if the two applied markers were identical, the camera would not know which of the two markers it really needed to recognize.

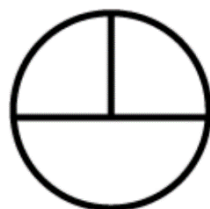


Figure 4.112: Marker. Reference: Fanuc Italia S.r.l.



Figure 4.113: Wheels tray with the marker applied. Reference: edited by Francesco Succurro

From the Figure 4.113, it can be seen that the marker was placed on the raised initial part of the tray, so that it could continue to be visible during the unloading of all four wheels.

To recognize this marker, a new vision function was created, called RICERCAVASSOIORUOTA. As shown in the Figure 4.114 the User Frame selected as the offset plane was identical to those selected for the truck tray location vision function, as well as the calibration of the selected camera.

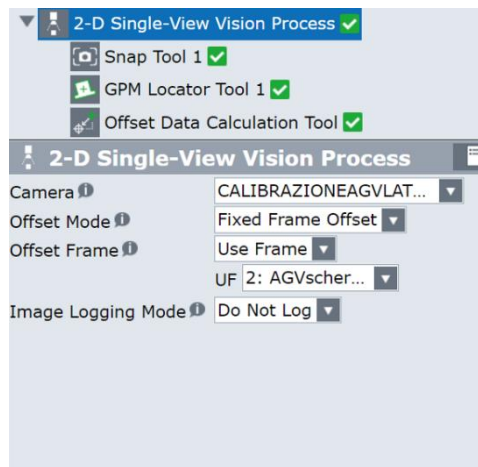


Figure 4.114: RICERCAVASSOIORUOTA vision function. Reference: edited by Francesco Succurro

The only difference was the reference model that was selected to be recognized by the vision system.

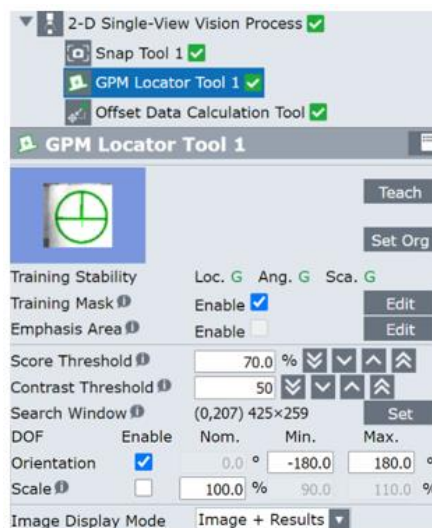


Figure 4.115: GPM Locator Tool for the RICERCAVASSOIORUOTA vision function. Reference: edited by Francesco Succurro

Again, the tray was placed at a certain reference position, which was subsequently recorded. The height at which the marker was located relative to the selected User Frame, -20 mm, was also measured and entered.

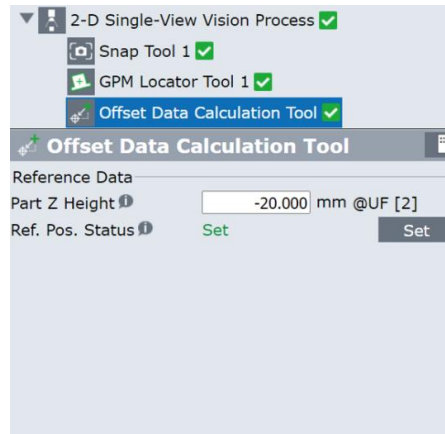


Figure 4.116: Reference position of the RICERCAVASSOIORUOTA vision function. Reference: edited by Francesco Succurro

At this point, holding the tray at the reference position just recorded, the robot, with the wheel still in its grip, was brought to the desired component place position, and this was recorded. The place position chosen was indicatively at the top of the centring rod, so that the component, once dropped from this height, would be guided to its storage. Through the use of one register R[11], used as counter, which managed two branches of the program LBL[2] and LBL[3], it was possible to manage the deposition of the wheels, of two on the first rod, and of the other two on the second one.



Figure 4.117: Wheel place phase. Reference: edited by Francesco Succurro

4.6.5.1.3 Board program

After managing the pick and place of the wheels, what was missing to load onto the MIZUBOT 4.0 was the board. After unloading the last wheel, the robot was programmed so that, during its path back to the robot cart where the board was located, it would take a picture at the board tray, to locate it and store its placement inside the vision register. The reasoning for taking the photo looking for the tray before picking up the component stemmed from the fact that the board is a bulky object and thus once squeezed between the gripper jaws, it would interfere with the camera's field of view, thus not allowing the tray to be recognized.



Figure 4.118: Photo to the boards tray.
Reference: edited by Francesco Succurro

To facilitate recognition of the tray, it was provided with a special marker, identical to that applied on the wheels tray.



Figure 4.119: Boards tray with the marker applied.
Reference: edited by Francesco Succurro

The main's subprogram, called RICERCAVASSOIOTAVOLA, handled these operations.

```

RICERCAVASSOIOTAVOLA
1: PAYLOAD [5]
2: UFRAME_NUM=3
3: UTOOL_NUM=1
4: J P [27] 100% FINE
5: VISION RUN_FIND 'RICERCAVASSOIOTAVOLE'
6: VISION GET_OFFSET 'RICERCAVASSOIOTAVOLE' VR[4] JMP LBL[1]
[End]

```

Figure 4.120: RICERCAVASSOIOTAVOLA program.
Reference: edited by Francesco Succurro

This contained the location at which the photo was taken, and the execution of a vision function called RICERCAVASSOIOTAVOLE. Since the marker applied was identical to the one applied on the wheels tray, the reference model that was taught to the camera was the same, and also the part search parameters, such as Contrast Threshold, Score Threshold and Orientation. The only difference was the offset plane and camera calibration setting: since the photo was being taken in the portion of the space where the boards tray was located, User Frame 3, called AGVfondo, and the camera calibration called CALIBRAZIONEAGVTAVOLESKATE were selected, like shown in the Figure 4.121.

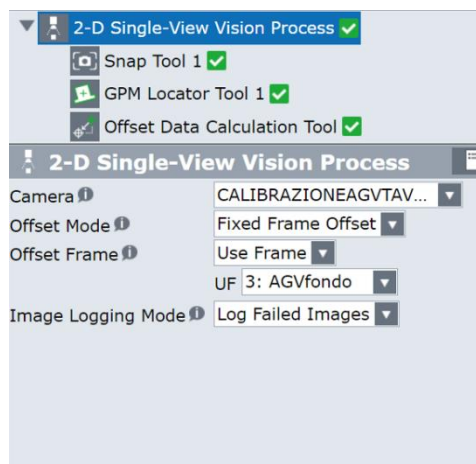


Figure 4.121: RICERCAVASSOIOTAVOLE vision function.
Reference: edited by Francesco Succurro

Finally, the reference position of the tray was set, and the height of the marker relative to the chosen offset plane was measured and entered.

Case Study

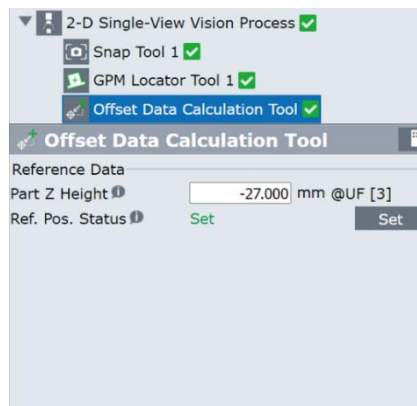


Figure 4.122: Reference position of the RICERCAVASSOIOTAVOLE vision function.
Reference: edited by Francesco Succurro

At this point, the subprogram PICKTAVOLA was recalled from the main, in which specific points of approach and pick of the board were set, without using the vision system, since the board was always placed in the same position, and therefore not free to be randomly placed on the cobot cart.

```
PICKTAVOLA
1:  PAYLOAD [5]
2:  UFRAME_NUM=1
3:  UTOOL_NUM=1
4:
5:J  P[3] 40% FINE
6:L  P[2] 100mm/sec FINE
7:  PAYLOAD [8]
8:  R[1]=45.5
9:  R[2]=120
10: R[3]=0
11: CALL COMMAND_GRIP
12: WAIT 1.50 (sec)
13:L P[3] 100mm/sec FINE
[End]
```

Figure 4.123: PICKTAVOLA program. Reference: edited by Francesco Succurro

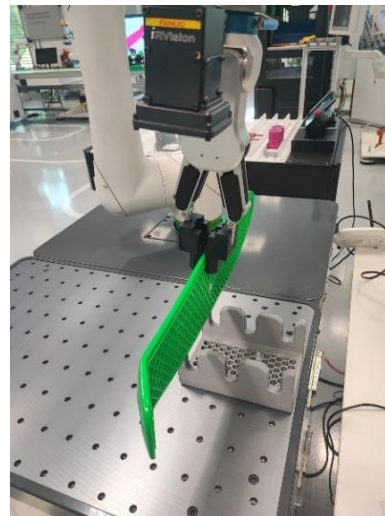


Figure 4.124: Board pick phase. Reference: edited by Francesco Succurro

After that, the robot, on its way to the AGV, was made to pass through a predetermined trajectory, the same one made for the trucks and wheels, thanks to the subprogram called SVINCOLOTAVOLA.

```

SVINCOLOTAVOLA

1: PAYLOAD[8]
2:J P[1] 100% CNT100
[End]
    
```

Figure 4.125: SVINCOLOTAVOLA program.
Reference: edited by Francesco Succurro

In the last step, through the SCARICOTAVOLA subprogram, the robot was able to unload the board into the tray, thanks to the application to the approach and place positions of the positional and orientational offset recorded during the photo in the vision register VR[4].

```

SCARICOTAVOLA

1: PAYLOAD[5]
2: UFRAME_NUM=3
3: UTOOL_NUM=1
4: PR[26,1]=0
5:J P[1] 40% FINE VOFFSET,VR[4] Offset,PR[26]
6:L P[1] 100mm/sec FINE VOFFSET,VR[4]
7: PAYLOAD[5]
8: R[1]=100
9: R[2]=80
10: R[3]=0
11: CALL COMMAND_GRIP
12: WAIT 1.25(sec)
13:L P[1] 100mm/sec FINE VOFFSET,VR[4] Offset,PR[26]
[End]
    
```

Figure 4.126: SCARICOTAVOLA program.
Reference: edited by Francesco Succurro

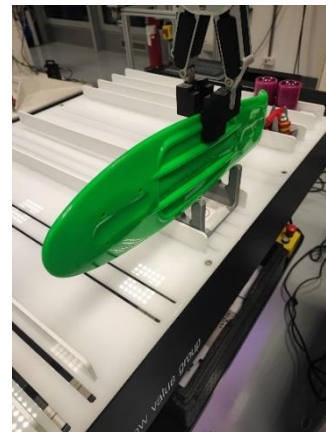


Figure 4.127: Board place phase. Reference: edited by Francesco Succurro

4.7 Mizubot 4.0 programming

4.7.1 Introduction

Mizubot 4.0 is a robotic system that automates the process of delivering materials on an assembly line, avoiding non-ergonomic, low value-added, and dangerous activities for operators. Thanks to the integrated production sequence management software, it autonomously adapts in real time to any changes in the production sequence, ensuring reliability of the line to customers and significantly reducing the possibility of making mistakes. But the core of this solution is the software part of the integration between the handling system, the line, the MES, the Pull Scheduling system and the Material Management system, which enables their communication, so that Mizubot can always be connected in real time to the operating conditions of the process and is able to replenish materials according to the actual consumption of the production lines.

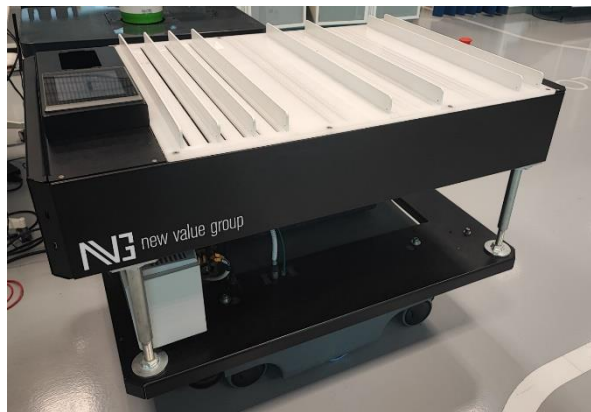


Figure 4.128: Mizubot 4.0. Reference: edited by Francesco Succurro

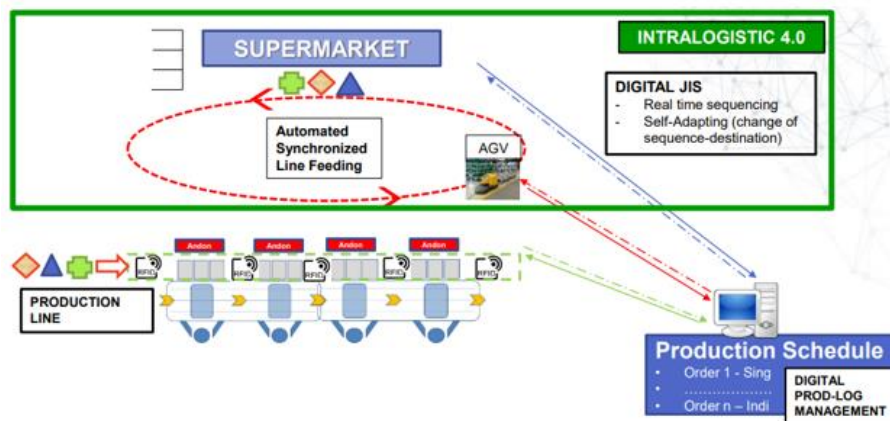


Figure 4.129: Intralogistics 4.0. Reference: New Value s.r.l.

This is an automation capable of transporting and delivering materials to assembly lines, installed on an AGV. In this case, it can be seen from Figure 4.128 that the surface of the Mizubot consists of several channels, on which there are openings of the same length. Below this surface are the push paddles, which, thanks to a rotational actuator, can rotate and then pop out of these openings and slide along the channels to unload the components onto the assembly station. In addition, the rotational actuator changes the orientation of the push paddles, during unloading channels. Specifically, in an initial phase, the orientation of the paddles is kept vertical, and once the components have been pushed to the edge of the Mizubot surface, ready to be unloaded on the assembly station, their orientation is rotated and becomes horizontal, so that a final thrust can be given to unload the components.

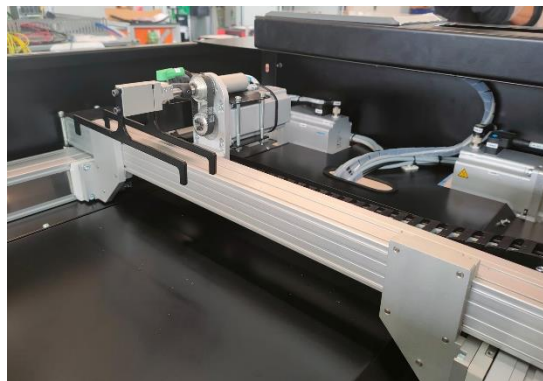


Figure 4.130: Push paddles. Reference: edited by Francesco Succurro



Figure 4.131: Push paddles with vertical orientation. Reference: edited by Francesco Succurro



Figure 4.132: Push paddles with horizontal orientation. Reference: edited by Francesco Succurro

As shown in Figure 4.133, the push paddles can be controlled and moved along the x-axis (short side) and y-axis (long side) of the Mizubot surface, and then along the different channels, thanks to 3 actuators that are:

- X-axis master;
- X-axis slave;
- Y-axis.

Programming of the different unloading channels and push paddles can be done directly on the Mizubot trough a touch interface located on its surface. It is a system that can be repurposed according to the type of materials it has to handle and the type of delivery it has to make on the workstation, using different work surfaces. So, a very fast setup and change-over can be done. The Mizubot moves autonomously around the plant based on assigned missions and can recognize and avoid obstacles and people. The system can interact with a possible automatic warehouse for material preparation. This allows the entire enslavement process to be managed autonomously without operator intervention. The on-board system is composed of a PC and a programmable logic controller (PLC), which together can control both the AGV and the devices to handle the trays/bins.

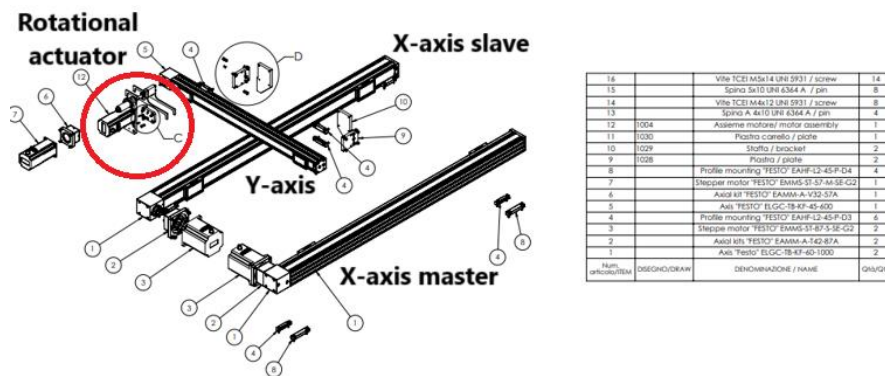


Figure 4.133: Internal structure of the Mizubot 4.0. Reference: New Value s.r.l.

Like shown in the Figure 4.134, Mizubot 4.0 is based on the concept of "Mizusumashi", a lean technique of in-line material delivery that enables synchronized replenishment of materials to production lines in compliance with Takt Time and scheduling with "pull" logic. Unlike applications with AGVs on the market today, Mizubot 4.0 reduces allows multiple deliveries of different materials to different points

on the line with a single mission. As an additional strength over competitors, Mizubot4.0 can be installed on all AGVs currently on the market. The ease of integration with the most diverse realities is an important feature of our system as well as one of the fundamental paradigms of Industry 4.0.

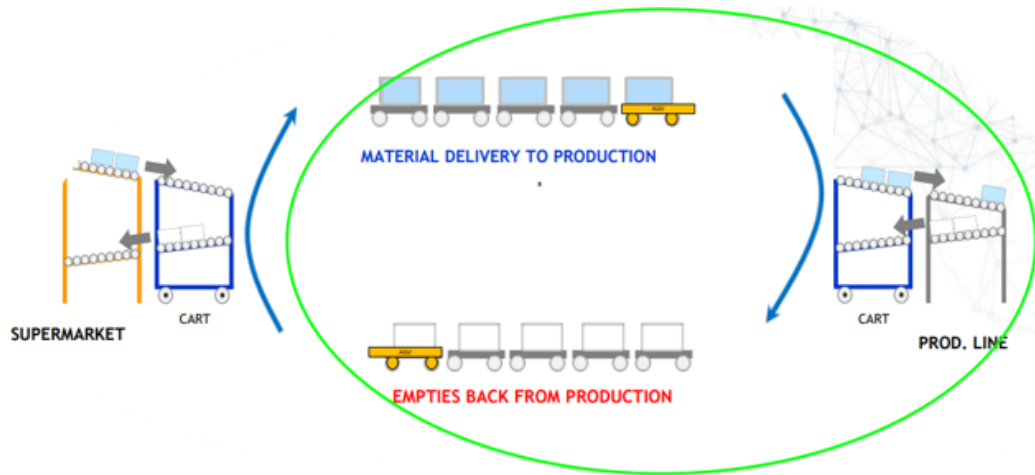


Figure 4.134: Mizusumashi roundtrip. Reference: New Value s.r.l.

4.7.2 Mission of the AGV

The AGV on which the Mizubot 4.0 is based is the MiR200. It is an autonomous mobile robot capable of carrying loads of up to 200 kg within manufacturing, warehouse and other industrial environments not open to the public. It allows the installation of customized upper modules such as containers, grids, elevators, belts and even a collaborative robotic arm, for any application. The upper modules are easily replaced to adapt the robot to different tasks [43].



Figure 4.135: MiR 200. Reference: urly.it/3yrv8

The MiR 200 is equipped with a navigation and control system that aims to lead the robot from one point to another of the map while safely avoiding people and obstacles. It has three types of sensors for obstacle detection:

- Safety laser scanners;
- The 3-D cameras;
- The ultrasonic sensors.

Two safety laser scanners placed diagonally on two corners (one in front and one back) of the robot, scan the surrounding environment. Each safety laser scanner has a 270° field of view, so it overlaps and provides full 360° visual protection around the robot, as can be seen in the Figure 4.136. In motion, the safety laser scanners constantly scan the environment surroundings to detect objects.

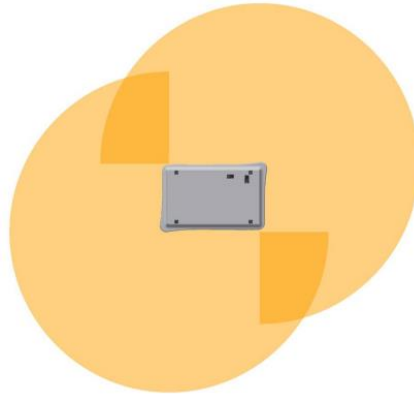


Figure 4.136: The two security laser scanners. Reference: MiR 200 Manual

Two 3D cameras at the front of the robot detect objects in front of the robot. The 3D cameras detect objects:

- Vertically up to the height of 1,800 mm at a distance of 1,950 mm in front of the robot;
- Horizontally within an angle of 118° and at 180 mm for the first detection on the ground.

The 3D cameras are for navigation purposes only. They are not part of the safety system of the robot.

There are four ultrasonic sensors on the robot: two in the front or rear and two in the front oriented to look to the sides. The ultrasonic sensors make it possible to detect objects that cannot be detected by the camera or laser scanners.



Figure 4.137: There are four ultrasonic sensors on the robot: two in the front (left) and two in the back (right). Reference: MiR 200 manual

4.7.2.1 Map creation

The map enables the robot to move safely and efficiently in its surroundings. The map illustrates the physical area in which the robot is used. The robot must have a map for each area in which it is used. It is important to create a well-maintained and reliable map so that the robot can operate effectively and safely. In fact, a new map was created for this application. "Mapping" was done, which consisted of having the robot move to the intended work environment within the Digital Line of the Competence Center. During this phase, as the robot moved, laser scanners detected all the data needed to generate the map and physical obstacles such as walls, that were recorded on the map. After that, at a later stage, it was possible to modify the map to make it clean and as true to the real environment as possible. At the end of this process the created map, shown in the Figure 4.138 was saved.

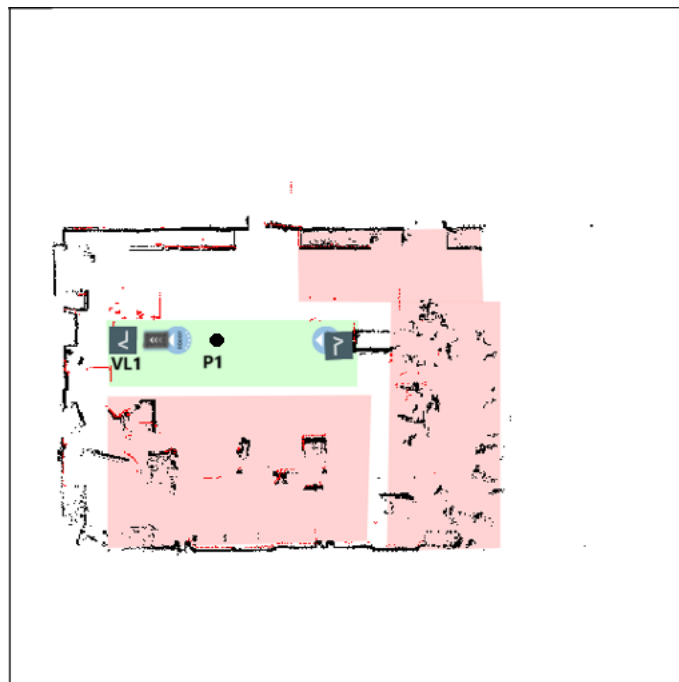


Figure 4.138: Created map. Reference: edited by Francesco Succurro

It can be seen from the map that a marker called VL1 has been placed. The markers are defined as X-Y coordinates on a map indicating the positions to be reached by the robot and allow it to precisely position itself with respect to these elements, thanks to the use of security laser scanners. A VL marker was used that allows the robot more precise

positioning than all other types. As shown in the Figure 4.139, it consisted of a V-mark with a 350-mm plate attached to the right of the V-shape. VL markers are intended to accommodate the robot so that the front or back of the robot faces the marker. In this case, the marker was placed near the collaborative robotics station because it was important for the AGV to position itself as accurately as possible relative to the cobot.

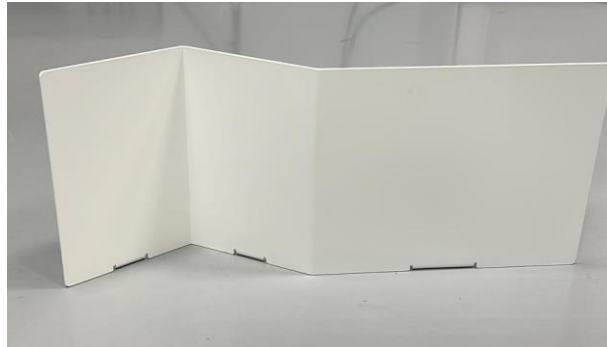


Figure 4.139: VL marker. Reference: edited by Francesco Succurro

In addition, again from the Figure 4.138 it can be seen that a position called P1 was recorded, which was necessary in the creation of the AGV's mission and will be discussed in the next section. Positions are defined as X-Y coordinates on a map indicating the positions to be reached by the robot. The positions serve as destinations or landmarks for the paths used in the missions.

4.7.2.2 Mission Creation

The MiR robot functions on the basis of missions created. A mission consists of actions, such as moving actions, logical actions, anchoring actions, and sounds, which can be brought together to make a mission containing all the necessary actions. Missions can then be incorporated into other missions.

Specifically for this application a mission was created, that was capable of transporting components from the loading station, alongside the fanuc cobot, to the assembly station, where they would be unloaded. The mission created is shown in the Figure 4.140.

Case Study



Figure 4.140: Created mission. Reference: edited by Francesco Succurro

As can be seen from the first program instruction, the AGV was programmed so that it would go, wherever it was at the start of the mission, to position P1 near the collaborative robotics station. This was because from this position the AGV was able to detect the presence of the marker. In fact, the next instruction represented the phase of docking the AGV to the marker: the robot used the safety laser scanners to detect the marker and reach the correct position relative to the detected marker. At this point once the AGV had reached the loading position alongside the cobot, it would wait to be loaded so that it could then depart to the assembly station. The waiting and restarting phase of the AGV drive was handled by register 5 of the PLC: when the AGV arrived at the loading position after recognizing the marker, register 5 was set to 1, and the AGV before restarting waited for the latter to become 0, i.e., for the loading phase to be completed. In this application once the loading was finished by the Fanuc cobot, register 5 was manually set to 0 thanks to a "continue mission" button on the Mizubot's touch interface. In this way, the AGV could continue with its mission. From this position, it was made to

move linearly, through the Relative Move instruction, by a total length of 5.95 m, so that it could reach the final assembly station. A linear move was used since both the final station and the robot cart were aligned. At this point, the AGV was programmed to begin the unloading phase of the three channels where the wheel tray, truck tray, and board tray were located, respectively. The unloading phase was managed by register 4 of the PLC: in particular, it was initially set to 1, indicating the activation of the unloading program of the first channel, and waited until it became 0, i.e., for the unloading phase to end, before it could begin the unloading of the next channel. In this case, register 4 was automatically set to 0 once the channel unloading ended. And so, by the same mechanism, register 4 was set first to 2 and then to 3 to handle the unloading of the other two channels. After that, the AGV would go into a waiting phase before it could restart and return to the loading station. This phase was also handled by register 5 of the PLC: specifically in the waiting phase register 5 was set to 1, and the AGV before restarting waited for this to become 0. Again, register 5 was manually set to 0 once unloading was finished, by pressing the "continue mission" button on the Mizubot interface. At this point the AGV was made to move a linear displacement of 1.5 m to get out of the encumbrance, and then was programmed to go to the starting P1 position, from which, through the VL1 docking instruction it would search for the marker and go to the load position again, ready for the next cycle to be reloaded by the cobot.

4.7.3 Configuration of the unloading channels

At this stage, the different unloading channels of the Mizubot 4.0 were configured using the software developed by New Value Group company. The system's touch interface was used to do this.

The Mizubot is a system designed to be able to unload components to lines or stations that can be positioned on either side, left or right. In this case, the assembly station was placed to the left of the Mizubot during the unloading phase. In fact, as can be seen from the Figure 4.141, the first selection to be made was the unloading side of the Mizubot on the station. For this line, named Line 1, the unloading on the left was chosen.



Line	Line name	Load Position	Enabled
1	Line 1	Left	Yes
2	Line 2	Left	Yes
3	Line 3	Right	Yes
4	Line 4	Left	Yes

Figure 4.141: Unload side. Reference: edited by Francesco Succurro

After that, through a second screen (Figure 4.142), it was possible to choose the order of unloading channels and the mode of delivery. For this application, it was chosen to start with unloading the fifth channel (boards), then the third (trucks), and then the first (wheels). Regarding the delivery mode, it is important to remember that the Mizubot can unload components in 4 modes: center to left, center to right, left to right, and right to left. The mode indicates the initial position from which the push paddles start and the

push direction to which you want to have the components unloaded. For example, a center-to-left delivery mode may be useful if I have components that are in the left half of the track, while a right-to-left mode may be useful for unloading trays that are in three-quarters of the rail from the left. These modes thus allow flexibility in managing the delivery of components to the different lines to be replenished.

For this application, since there was only one tray placed in the left half for each channel, a half-track delivery mode was chosen, as shown in Figure 4.142. Since the side selected to unload was the left side, the unloading mode applied was from center to left.

The screenshot shows the 'Mission Editor' window. At the top, there are fields for 'Mission ID: 2013', 'AGV Mission: e3ef4101-13fd-11ee-ad71-94c691a', 'Model: CIM test - 3 canali', and 'Line: Line 1'. A 'Worktable' dropdown is set to 'SAMSS'. Below these are sections for 'Track 1' and 'Track 2', each with dropdowns for 'NVWS1' through 'NVWS8'. The 'Delivery mode' section shows 'HALF TRA' selected for all eight NVWS slots. At the bottom, a table lists mission details:

#	Model	Line	AGV Mission	Worktable
2013	CIM test - 3 canali	Line 1	e3ef4101-13fd-11ee-ad71-94c...	SAMSS
31	D12 UVL01	Line 1	D12 UVL01	Default
37	D12 UVL01-B-TD	Line 1	D12 UVL01-B-TD	Default
33	D12 UVL02	Line 2	D12 UVL02	Default
38	D12 UVL02-B-TD	Line 2	D12 UVL02-B-TD	Default
34	D12 UVL03	Line 3	D12 UVL03	Default
39	D12 UVL03-B-TD	Line 3	D12 UVL03-B-TD	Default

Figure 4.142: Order in which channels are unloaded and delivery mode. Reference: edited by Francesco Succurro

In addition, for each mode it was possible to customize the starting position of the thrust paddles (Start X, Start Y) and the position to which thrust was to be applied by the paddles (End Y), as shown in the Figure 4.143. The end position was represented by the Y coordinate alone, since in pushing along each channel the X coordinate remained fixed, and what varied was the Y coordinate. In summary, a change in X caused the paddles to move from one channel to another, while a change in Y caused the paddles to move along the same channel.

Case Study

Tracks Editor - □ ×

Manual Mode ON
Reset Errors
Homing
Save
Close

Track name: Track 1

	Center->Left	Center->Right	Right->Left	Left->Right	Actual Pos.
Start X:	7,6 ■	7,6 ■	7,6 ■	7,6 ■	X: 0
Start Y:	232 ■	258,5 ■	39,6 ■	437,2 ■	Y: 0
End Y:	548 ■	33,5 ■	533,6 ■	33,5 ■	C: 0
	Test Center->Left	Test Center->Right	Test Right->Left	Test Left->Right	

#	Track	From X C>L	From Y C>L	To Y C>L	From X C>R	From Y C>R	To Y C>R	From X R>L	From Y R>L	To Y R>L	From X L>R	From Y L>R	To Y L>R
1	Track 1	7,6	232	548	7,6	258,5	33,5	7,6	39,6	533,6	7,6	437,2	33,5
2	Track 2	76,9	231,5	548	76,9	231,5	33,5	76,9	39,1	533,6	76,9	437,2	33,5
3	Track 3	146,2	231	548	146,2	231	33,5	146,2	40,8	533,6	146,2	437,2	33,5
4	Track 4	303,7	232,8	533,6	303,7	232,8	33,5	303,7	39,7	533,6	303,7	437,2	33,5
5	Track 5	460,9	232,9	548	460,9	232,9	33,5	460,9	41,2	533,6	460,9	437,2	33,5
6	Track 6	531	233,3	548	531	233,3	33,5	531	40	533,6	531	437,2	33,5
7	Track 7	688	233	533,6	688	233	33,5	688	40,4	533,6	688	437,2	33,5
8	Track 8	845,8	287,4	533,6	845,8	287,4	33,5	845,8	39,3	533,6	845,8	437,2	33,5

Figure 4.143: Configuration of different delivery modes. Reference: edited by Francesco Succurro

Chapter 5

Conclusions and future developments

The present study aimed to implement and test the automatic servo-system of a skateboard production line.

In particular, the work focused on the loading phase of the Mizubot 4.0 by the collaborative robot. Here indeed arose the critical aspect of the application, the deposition of the parts on the AGV: the trays on which the parts were deposited were free to slide along the AGV's channels, and thus after each deposition, following the release of the part, the trays may have moved from their original position. In addition, the AGV, at each end of cycle, after unloading the parts to the operator, would return to the loading position next to the robot, with a small offset in position and orientation given by the inaccuracy of the AGV's docking phase to the VL marker.

The technological challenge was to properly deposit the components in the trays, despite the sum of these position and orientation offsets. For this reason, the Fanuc vision system was implemented and tested. Through the creation and execution of vision functions aimed at recognizing specific markers installed on the trays, the vision system was able to locate them each time and update the positioning of the parts in them. In addition, the trays, made specifically for this application, were specially designed so that they could be loaded by the robot with pinpoint accuracy. This was done to test and understand how far the camera's ability to locate and objectify the position of a part could go. The result was, for example, accurate positioning of the wheels in the tray centering rods with each cycle, thus demonstrating that the vision system was robust and extremely accurate. In addition, it was verified and tested that the depositing of components was done in the same way and with the same accuracy, despite moving the cart on which the robot was mounted by a small amount.

These results make it possible to demonstrate that the cobot, with the vision system on board, can replicate the visual capabilities of the operator. In this way, the task that would have been performed by the operator can be completely replaced by the collaborative robotics station, without compromising the accuracy of the application. This reduces non-ergonomic, unsafe, and non-value-added activities by operators.

The implemented line can simplify production processes that are prone to human error, enabling a reduction in errors in the delivery of materials to the lines. In addition, such a solution is flexible since it can work with most commercially available robots and AGVs and can be adapted for any front-serving production line.

There may be several future developments of this line, among which is the possibility of integrating an automatic warehouse on the picking side, so that the process of delivering materials to the line could become fully automatic. In addition, communication could be implemented between the CRX-10iA and the Mizubot, so that the AGV can receive an end-of-load signal from the cobot, and then can restart autonomously without the operator's operation of the "Continue Mission" button. In order to improve instead the positioning of the AGV in the anchoring phase to the VL marker, magnetic strips could be installed on the floor.

A next step in the application could be the integration of line-side IoT systems, so that the system would be able to measure the actual amount of material present and thus limit the replenishment to the strictly necessary material. In this sense, a synchronization with the MES system and with the pull scheduling system that interfaces with the company's management systems would bring the system to be connected in real time to the operating conditions of the process and would make it possible to ensure compliance with production orders.

References

- [1] [Online]. Available: <https://www.ilsole24ore.com/art/tre-proposte-sviluppare-competenze-40-e-rilanciare-made-italy-AEXMHQ3C>.
- [2] L. Beltrametti, N. Guarnacci, N. Intini and C. La Forgia, *La fabbrica connessa. La manifattura italiana (attra)verso Industria 4.0*, Milano: Guerini e Associati, 2017.
- [3] [Online]. Available: https://blog.osservatori.net/it_it/smart-technologies-quarta-rivoluzione-industriale.
- [4] [Online]. Available: <https://www.digital4.biz/supply-chain/industria-40-politecnico-milano-sei-tecnologie-abilitanti/>.
- [5] [Online]. Available: <https://www.make-consulting.it/industria-4-0-tecnologie-abilitanti/>.
- [6] [Online]. Available: <https://www.ptc.com/it/blogs/corporate/digital-twin-for-predictive-maintenance>.
- [7] [Online]. Available: <https://www.key-4.com/lintegrazione-verticale-e-orizzontale-nellindustria-4-0/>.
- [8] [Online]. Available: <https://blog.made-cc.eu/10-vantaggi-di-una-fabbrica-smart-e-sostenibile>.
- [9] W. Wang, X. Zhu, L. Wang, Q. Qiu and Q. Cao, “Ubiquitous Robotic Technology for Smart Manufacturing System,” *Ubiquitous Robotic Technology for Smart Manufacturing System*, vol. 2016, no. 6018686, p. 14, 2016.
- [10] [Online]. Available: <https://www.ilsole24ore.com/art/industria-40-transizione-digitale-inizia-ora-AEfmMEG>.
- [11] [Online]. Available: <https://www.linkmanagement.it/progetti/sistema-aps-advanced-planning-and-scheduling/>.
- [12] [Online]. Available: <https://www.digital4.biz/marketing/guida-al-software-crm-cos-e-e-come-utilizzarlo/>.
- [13] [Online]. Available: [https://www.smeup.com/magazine/blog/cosa-significa-erp/#:~:text=L'acronimo%20sta%2C%20letteralmente%2C,risorse%20aziendali%20\(Resource%20Planning\)..](https://www.smeup.com/magazine/blog/cosa-significa-erp/#:~:text=L'acronimo%20sta%2C%20letteralmente%2C,risorse%20aziendali%20(Resource%20Planning)..)
- [14] [Online]. Available: <https://news.beta80group.it/wms-che-cos-e-e-perche-e-fondamentale-per-la-logistica>.
- [15] [Online]. Available: <https://blog.sew-eurodrive.it/smart-factory-cos-e-come-funziona-la-fabbrica-intelligente>.
- [16] Competence Industry Manufacturing 4.0, “Bilancio di Sostenibilità CIM4.0,” Competence Industry Manufacturing 4.0, Torino, 2022.
- [17] [Online]. Available: <https://rocco.faculty.polimi.it/robotica/introduzione.pdf>.
- [18] [Online]. Available: <https://www.arcoossrl.com/automazione-robotica-industriale-cos-e-quali-vantaggi-per-le-industrie/>.

-
- [19] [Online]. Available: <https://www.ilsole24ore.com/art/crescita-doppia-cifra-i-robot-industriali-2021-balzo-25percento-AEHhwYXB>.
- [20] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer London Ltd, 2010.
- [21] Politecnico di Torino, *Robotics course slides*.
- [22] [Online]. Available: <https://www.mwes.com/types-of-industrial-robots/cylindrical-robots/>.
- [23] [Online]. Available: <https://www.mwes.com/types-of-industrial-robots/polar-spherical-robots/>.
- [24] [Online]. Available: <https://www.mwes.com/types-of-industrial-robots/scara-robots/>.
- [25] [Online]. Available: <https://www.kuka.com/it-it/prodotti-servizi/sistemi-robot/good-to-know-robot-antropomorfo>.
- [26] F. Vicentini, *La robotica collaborativa*, Milano: Tecniche Nuove, 2017.
- [27] I. Gibson, D. Rosen and B. Stucker, *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, Springer, 2010.
- [28] [Online]. Available: [https://it.wikipedia.org/wiki/STL_\(formato_di_file\)](https://it.wikipedia.org/wiki/STL_(formato_di_file)).
- [29] [Online]. Available: https://schunk.com/it/it/sistemi-di-presa/pinza-parallela/c/PUB_8295.
- [30] [Online]. Available: https://schunk.com/it/it/sistemi-di-presa/pinze-autocentranti/c/PUB_8301.
- [31] [Online]. Available: https://schunk.com/it/it/sistemi-di-presa/pinza-angolare/radiale/c/PUB_8299.
- [32] [Online]. Available: <https://www.universal-robots.com/it/blog/quali-sono-i-modelli-di-gripper-e-loro-utilizzo/>.
- [33] [Online]. Available: <https://www.weerg.com/it/guide/file-formato-stl>.
- [34] [Online]. Available: <https://markforged.com/it/resources/blog/how-to-create-high-quality-stl-files-for-3d-prints>.
- [35] [Online]. Available: <https://star5a3smartsrl.com/filamento-pla-stampa-3d/>.
- [36] [Online]. Available: <https://www.fanuc.eu/it/it/robot/robot-filter-page/robot-collaborativi/crx-10ial>.
- [37] [Online]. Available: <https://crx.fanucamerica.com/wp-content/uploads/2022/10/CRX-10iA-data-sheet.pdf>.
- [38] [Online]. Available: <https://crx.fanucamerica.com/fanuc-cobot-crx-10ia/>.
- [39] Fanuc Italia S.r.l., *Training Academy. Robot Collaborativi. R014*.
- [40] FANUC CORPORATION, *iRVision 2D Camera Application - Operator's Manual*, 2017.
- [41] [Online]. Available: https://www.fanucamerica.com/docs/default-source/robotics-files/irvision/2019_fac_irvision_brochure_digital-101419.pdf?sfvrsn=67096c26_2.
- [42] FANUC CORPORATION, *R-30+B Mini Plus CONTROLLER - Operator's Manual*, 2012.

References

- [43] [Online]. Available: <https://www.directindustry.it/prod/mobile-industrial-robots-aps/product-173282-2178013.html>.
- [44] [Online]. Available: <https://cianoshapes.com/wp-content/uploads/2019/02/Scheda-tecnica-S5.pdf>.