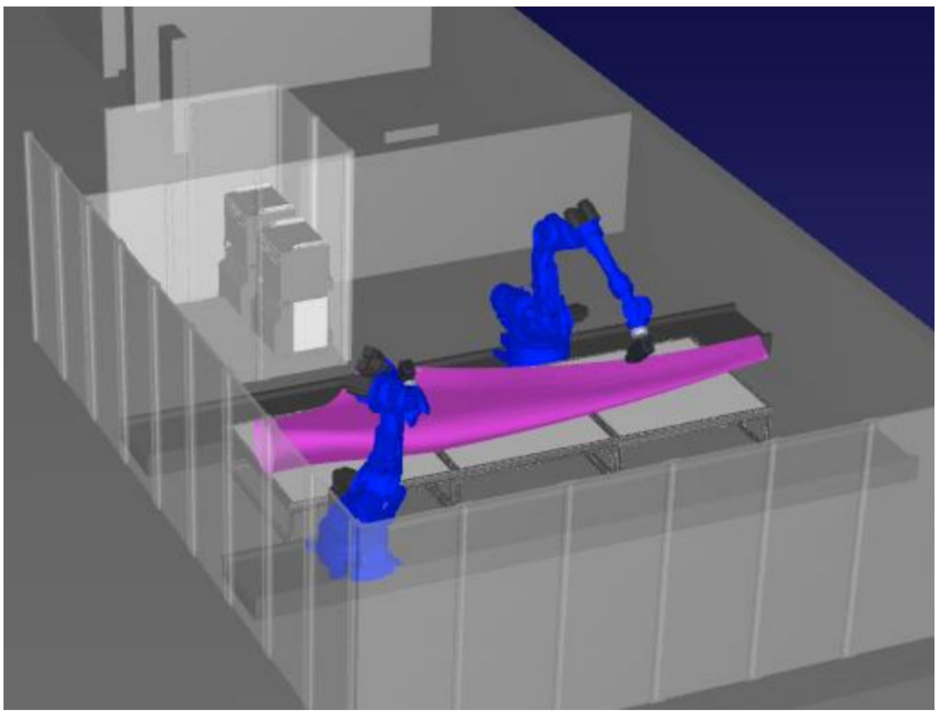
International Internship Report

Louis-Nam GROS
Isae-Supmeca / Ensta Bretagne
louis-nam.gros@ensta-bretagne.org

Supervised by
Prof. Donald RADFORD
Colorado State University - Composite Materials, Manufacture and Structures Laboratory
dradford@rams.colostate.edu



Development of a Two 7-DOF (Degrees of Freedom) Articulated Robotic Arm Cell for Additive Manufacturing Process

05/15/2023 - 09/08/2023

Acknowledgments

I would like to express my deep gratitude to several individuals who have contributed significantly to the success of this internship and the enrichment of my experience in robotics.

First and foremost, I extend warm thanks to Professor Jochum for his support and invaluable assistance throughout my internship search. His expertise and network of contacts enabled me to find this exceptional opportunity. I am grateful for his insightful advice, availability, and constant encouragement.

I also wish to convey my appreciation to Professor Radford, my internship supervisor, for welcoming me into his laboratory. His confidence in my abilities and dedication to my learning have been of immeasurable value. I sincerely thank him for his attentive guidance, insightful advice, and essential contribution to my professional development.

I would also like to thank the laboratory team for their warm welcome and support throughout my internship. Their expertise, innovative ideas, and collaboration greatly enriched my work experience. Their guidance and encouragement enabled me to explore new perspectives and deepen my knowledge in robotics. In summary, I am deeply grateful to Professor Jochum for his contribution to securing this internship, and to Professor Radford for his guidance and support throughout my tenure in the laboratory. I also want to express my gratitude to the laboratory team and my loved ones for their invaluable collaboration and unwavering support.

Abstract

This report details the process of installing and restoring Motoman robotic arms and rails in a laboratory setting. The robotic components, retrieved from industrial usage and outdoor storage, required restoration due to damage caused by time, environmental factors, and improper disassembly. The restoration process involved comprehensive cleaning, repair, and replacement of components, including controllers, cables, encoders, and safety systems.

The restoration was crucial to prepare the robotic elements for integration into the laboratory's research projects, particularly in the realm of additive manufacturing. The report outlines the steps taken for the restoration of controllers, rail systems, and Motoman robotic arms, accompanied by detailed images of the restoration process. Challenges encountered, such as communication errors and security alarms, are described, along with their solutions.

The report further explains the installation plan for the robots and rails, including the necessary electrical and safety considerations. The installation process involves precise manipulation, alignment of rails, cable routing, and safety device placement. The report also details the setup of a virtual machine with ROS (Robot Operating System), ROS-Industrial, MoveIt, and RViz to facilitate simulation and testing.

Simulations were developed to calibrate the virtual robots, simulate emergency shutdown and restart procedures, and demonstrate trajectory planning using ROS tools. The report highlights achievements and discusses potential future developments, including advanced applications, sensor integration, and continuous optimization of the robotic systems.

In conclusion, this report documents the restoration, installation, and simulation of Motoman robotic systems for research purposes. It showcases the technical challenges faced and the solutions implemented, as well as the accomplishments achieved during the internship. The experience provided valuable insights into robotics, software integration, and project management, setting the foundation for future projects in the field.

Contents

1 Composite Materials, Manufacture and Structures Laboratory (CMMS)

1.1 Presentation

The lab is directed by Dr. Don Radford, Professor of Mechanical Engineering at the University. CMMS is located at The Factory in Fort Collins, Colorado. CMMS was initiated in the fall of 1989 to address research and educational issues related to advanced fiber reinforced composite materials. The lab moved from the Engineering Research Center to The Factory in the summer of 2004. CMMS houses traditional and automated equipment to develop, perform and model composite manufacturing processes as well as to perform mechanical testing and microstructural analysis. The focus of the research efforts can be broken into broad categories including process development, shape and warpage control, and materials response simulation. In addition, the lab has provided testing services for industrial groups both regionally and nationally. These varied R&D activities have allowed both graduate and undergraduate students the opportunity to pursue education and research in the field of composite materials. By maintaining research projects in various areas of the composites field, students in the Composite Materials, Manufacture and Structures Laboratory have gained a greater appreciation for this broad field of engineering. Sure, here's the translated version of the provided section, followed by the LaTeX format you provided:

Figure 1: Aerial View of the Laboratory

Figure 2: Colorado State University Logo

1.2 Team

The CMMS laboratory consists of three main teams:

1. Dr. Radford's Team:
   - Clément Ratshedule Fémi and Summaya, PhD candidates
   - Jaren and Hari, master's students
2. Dr. Yourkeni's Team (with whom we interacted regularly, but did not directly collaborate)
3. Dr. Ma's Team (with whom we interacted regularly, but did not directly collaborate):

These teams bring together researchers, PhD candidates, and master's students working together to conduct research, develop processes, and study composite materials. The collaboration among these teams enables a multidisciplinary and enriching approach in the field of composite materials within the CMMS laboratory.

2 Mission

2.1 Context of the Internship

Within the framework of a contract with the United States Department of Energy, the laboratory is tasked with developing an additive manufacturing process for 8-meter-long wind turbine blades. To fulfill this contract, the lab acquired a robotic arm mounted on 9-meter-long linear rails through a donation from a major company.

Several aspects of the process needed to be developed, tested, and approved:

- Development and installation of the additive manufacturing robotic cell
- Development of the additive manufacturing nozzle
- Customized non-planar slicing for use with 6-axis robotic arms (rather than 3-axis printers)
- Quality and control of additive manufacturing using sensors
- Development of a customized path planner

In this context, Dr. Radford was seeking robotics and mechanical students to assist in the development of the manufacturing process. Initially, the rails and robots were to be ground-mounted. I was tasked with calibrating the system and developing simple calibration and startup programs using the Motoman programming and control software. I was also supposed to assist in developing/implementing the safety locking system, designing the printing nozzle mounting system and auxiliary power sources, measuring temperature, and securing cooling lines. Additionally, hardware configuration was part of the scope. However, upon my arrival, the robots and rails were not installed. The laboratory only had one robotic arm, a controller, and a 3-meter rail.

2.2 Internship Objectives

The initial internship objectives, which included calibrating the robots and various sensors, were modified due to the robots not being installed. Therefore, new objectives were defined for the internship. The main objective of the internship thus became the installation and commissioning of the robots. This objective can be broken down as follows:

1. Design and sizing of the installation
2. Restoration of the robots
3. Physical installation of the robots
4. Installation and testing of safety systems
5. Simulation programming for Motoman robot control: Establish a simulation program for robot control in a virtual environment.
6. Installation of power supplies and communications for the printing process (in this case, the printing nozzle)
7. Alignment and calibration of the system
8. Documentation of the system

These objectives served as a guide throughout the internship, and the different steps and achievements were oriented towards their completion.

2.3 Description of the Robots Used

The robots used are Yaskawa-Motoman ES200D robotic arms capable of carrying up to 200kg with a maximum arm length of 2.65m [6]. They are controlled by Yaskawa-Motoman DX100 controllers [2]. Normally, it is possible to flash firmware on these controllers to enable ROS communication.

2.4 Description of the Virtual Environment

2.4.1 QEMU Virtual Machine

QEMU is free software for virtual machines that allows emulation of a processor and, in general, a different architecture if needed [9]. It enables the execution of one or multiple operating systems through KVM and Xen hypervisors, or simply running binaries within the environment of an already installed operating system on the machine.

QEMU allows virtualization without emulation when the guest system uses the same processor as the host system. It can also emulate processor architectures (x86, ARM, AVR, etc.). It is capable of simulating PCI, audio, and USB devices as well.

QEMU enables the execution of one or multiple operating systems, along with their applications, in an isolated manner on a single physical machine. This facilitates sharing the physical machine's resources between guest systems relatively seamlessly. In many cases, it's not necessary to modify the system to run on QEMU.

In the context of my internship, I used a QEMU virtual machine configured with a version of Linux containing a graphical interface, allowing me to benefit from a user experience with a complete desktop environment while maintaining optimal performance.

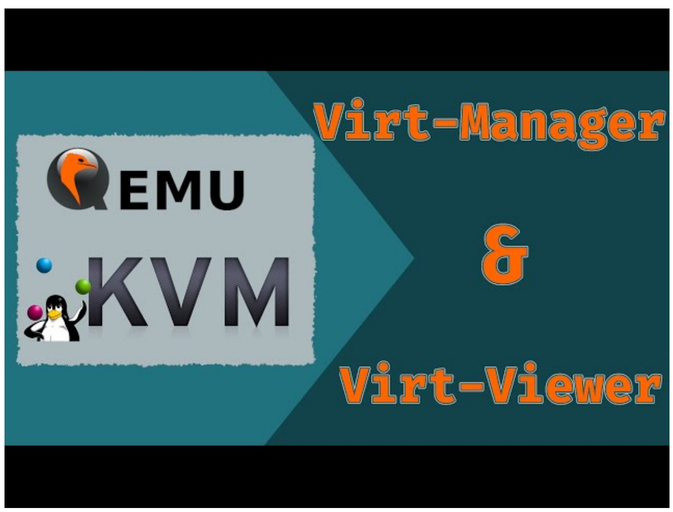


Figure 3: QEMU Logo

2.4.2 ROS - Robot Operating System

The Robot Operating System (ROS) is a set of software libraries and tools that help you develop robotic applications [10]. From managing drivers to cutting-edge algorithms, and powerful development tools, ROS has everything you need for your next robotics project. And it's all open source.

I used the ROS1 environment to simulate and control the robots. ROS1 is a widely used version of the Robot Operating System, offering a flexible and modular architecture for robot software development. It provides communication infrastructure between various components of a robotic system, allowing coordination between sensors, actuators, and control algorithms.

Through ROS1, I could use packages such as MoveIt and RViz for motion simulation and 3D visualization of the robots. MoveIt is a motion planning library that generates trajectories for robots, while RViz is a visualization tool that graphically represents robot models and sensory data.

Using the ROS1 environment facilitated the development and control of robots by providing a solid foundation for interfacing with the Motoman robots and setting up realistic simulations. This allowed me to explore different motion configurations, test planning strategies, and fine-tune the performance of virtual robots.

Figure 4: ROS Logo

Figure 5: ROS Noetic Logo

2.4.3 MoveIt Package

MoveIt is an open-source software for trajectory planning of industrial robotic arms [8]. This software is widely used in many articulated robotics applications, including:

- Motion Planning: MoveIt offers advanced trajectory planning features, enabling robots to generate smooth and safe movements.
- Manipulation: The MoveIt package provides tools for object manipulation by robots, facilitating tasks such as grasping, moving, and precise positioning of objects.
- Inverse Kinematics Calculation: MoveIt efficiently solves the inverse kinematics problem, determining joint positions of the robot based on a desired end-effector position.
- Control: MoveIt provides interfaces for robot control, allowing defining trajectories to follow and interaction with sensors and actuators.
- 3D Perception: The package also offers 3D perception capabilities for robots to understand their environment, including object detection and 3D scene reconstruction.
- Collision Checking: MoveIt integrates collision checking tools to ensure robot movement safety and obstacle avoidance.
- Synchronized Planning between multiple robots with trajectory calculation avoiding collisions between multiple robotic arms.

The MoveIt package already supports over 150 different robotic arms, offering compatibility with many popular robots in the market. Additionally, it allows configuring custom robotic arms from a CAD file.

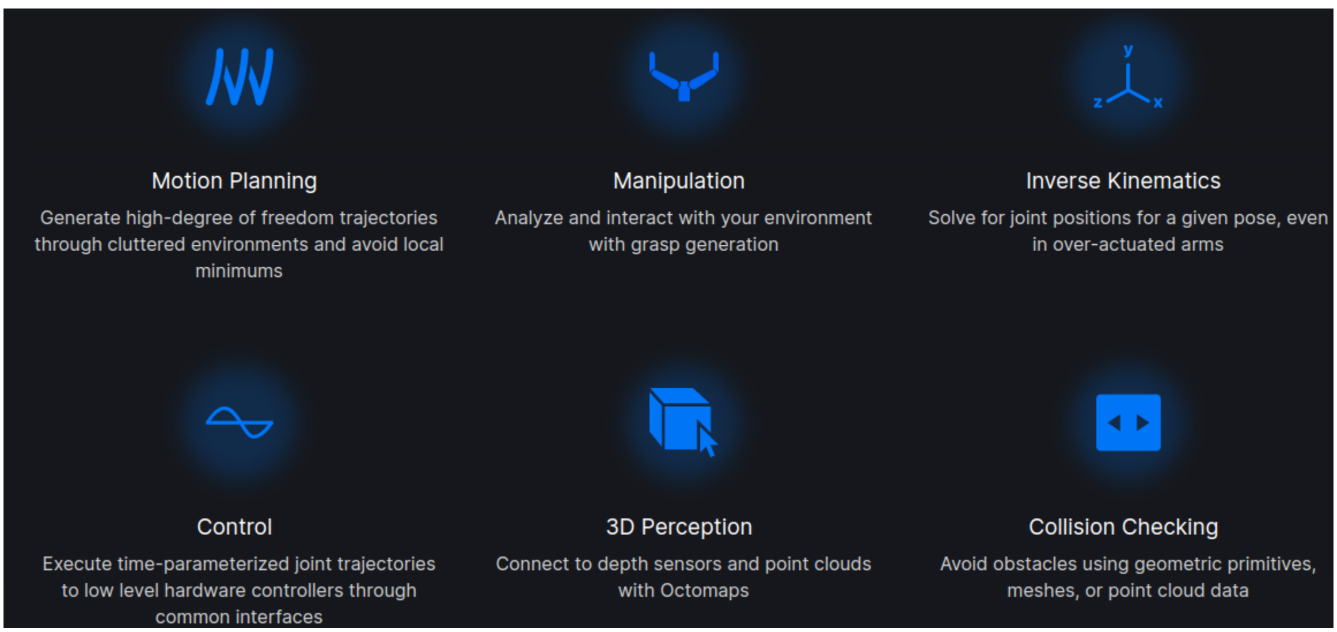


Figure 6: MoveIt Description

I used the MoveIt tool in my internship to simulate the robots and calculate their inverse kinematics. Thanks to MoveIt, I could test and validate different trajectories and motion configurations, ensuring that the Motoman robots were able to reach desired positions accurately and reliably.

2.4.4 Additional Packages: ROS-Industrial, abb, motoman

ROS-Industrial is an open-source project that extends the advanced capabilities of ROS software to industrial hardware and applications [11]. The ROS-Industrial package contains additional information about robots, including their CAD files, making it easier to integrate and use them in industrial robotics environments.

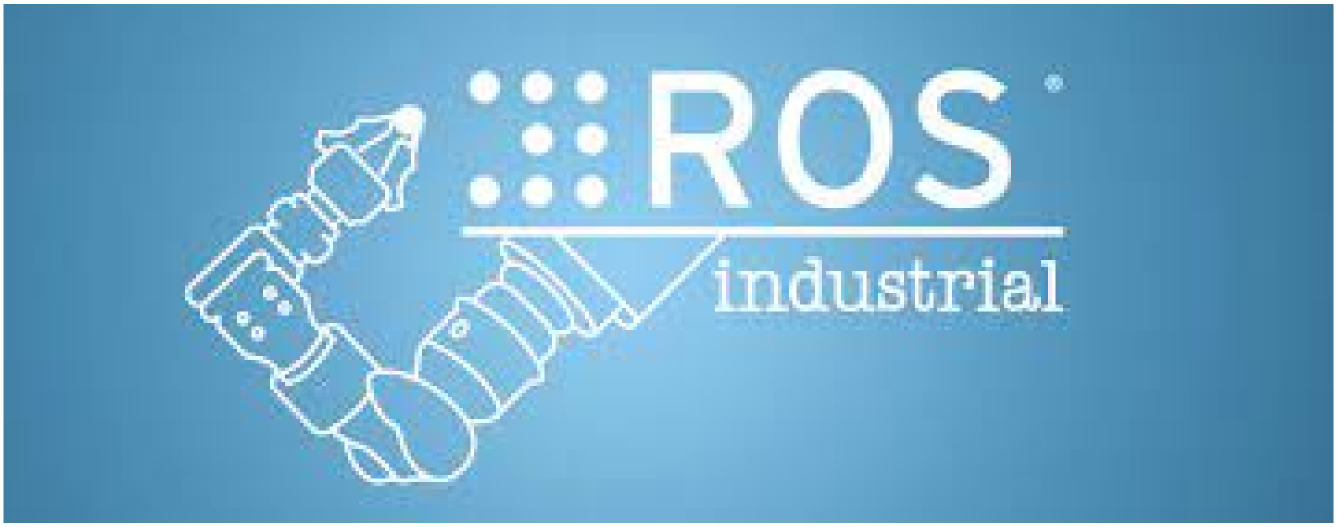


Figure 7: ROS Industrial Logo

- The motoman package [12][7]:

  The driver controller interface for Motoman was created in collaboration with Yaskawa Motoman to provide a more efficient interface for controlling Motoman robots. The Motoman driver communicates with ROS through a simple message interface, with some message types specific to Motoman. Trajectories are transmitted to the controller using a message format that captures all ROS JointTrajectoryPoint data: joint positions, speeds, accelerations, and timing synchronization. The controller stores these points and interpolates them to send commands to the controller when needed.

- The abb package:

  The ROS abb packages aim to facilitate interaction between ABB robot controllers and ROS-based systems by providing ready-to-use ROS nodes.

They enable the connection of ABB and Motoman robots, people, and ROS systems together, simplifying integration and communication between these different entities. Finally, the ROS-Industrial packages are used to convert ROS messages into the controller language, enabling smooth communication between the ROS system and the robot's controller.

2.5 Tasks to be Accomplished

Now that the objective is clear and the tools are defined, the following tasks have been identified:

- Draw installation plans and perform computer-aided design (CAD) to ensure optimal robot installation.
- Dimension the installation from electrical, mechanical, and safety perspectives, considering power needs, wiring, ventilation, etc.
- Restore the Motoman robot arms, their controllers, and the rails on which they will be installed (cleaning, greasing, wiring, etc.).
- Define an installation procedure and a list of required materials.
- Perform the mechanical installation of the robots following appropriate procedures and technical specifications.
- Define a safety protocol, including emergency stop devices and protective measures to ensure the safety of operators and work environments.
- Install, configure, and test door sensors and emergency stop buttons within the safety enclosure.
- Install, configure, and test auxiliary systems required for the robot tool (power supply, compressed air, cooling, monitoring, etc.).
- Ensure precise alignment and calibration of the system, ensuring robot movements adhere to tolerances of less than 0.5 mm.
- Develop a virtual environment to pre-test specific methods and programs for robot users and facilitate robot use while optimizing performance.
- Develop simple calibration methods and programs for robot users to ease their use.
- Thoroughly document the entire installation system, including plans, electrical and mechanical diagrams, as well as usage and maintenance procedures.

3 Installation of Motoman Robots

The main objective of this part of the internship was to install the Motoman robots in the laboratory, enabling the doctoral students to conduct 3D printing tests and studies. A dedicated area was reserved to accommodate the robots, with various possible configurations. I proposed several ideas to Professor Radford, and we selected the most suitable configuration. Subsequently, I revised the plans to optimize the workspace and account for the new configurations.

3.1 CAD and Detailed Plan

To achieve as accurate an installation as possible, I performed Computer-Aided Design (CAD) modeling of the laboratory and the surrounding areas of the robot zone. I used these models to obtain an overview and ensure proper integration of the robots in the space. Once the CAD modeling was complete and the elements were placed in their respective positions, I edited the installation plans.

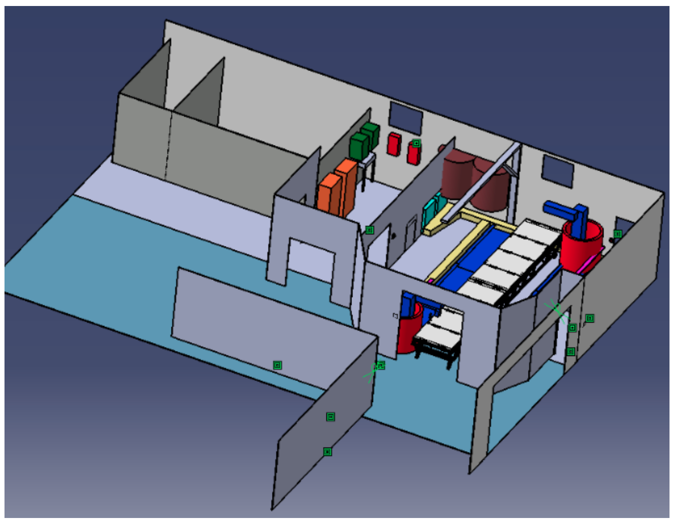


Figure 8: CAD laboratory + Motoman area

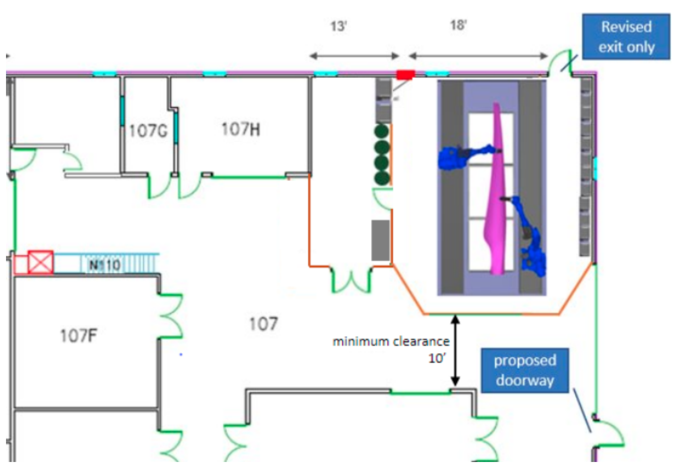


Figure 9: 2D lab plan (imperial units)

3.2 Safety

To ensure user and equipment safety, safety protocols were established. Emergency stop buttons were installed at the doors and various locations around the printing area. We then defined a safety protocol tailored to the installation and integrated all elements (emergency stop buttons, sensors, impassable barriers, rotating beacons, buzzers, etc.) into the CAD model.

3.2.1 Selection of Safety Elements

There are numerous safety elements and safety equipment available for industrial environments and robotic cells.

Figure 10: Emergency stop button

Figure 11: Door sensor

3.2.2 Safety Protocol and Emergency Stop

A precise protocol was defined for initiating a safe printing process. This protocol includes the following steps[6]:

1. Preparation of the printing area
2. Operator exits the printing area
3. Closure of the printing enclosure
4. Operator reaches the control area
5. Operator presses the arming button
6. Launching the printing program

3.2.3 Integration of These Elements into the System

I then integrated all elements (emergency stop buttons, sensors, impassable barriers, rotating beacons, buzzers, etc.) into the CAD model. I also created a wiring diagram for the safety elements to facilitate their installation and configuration.

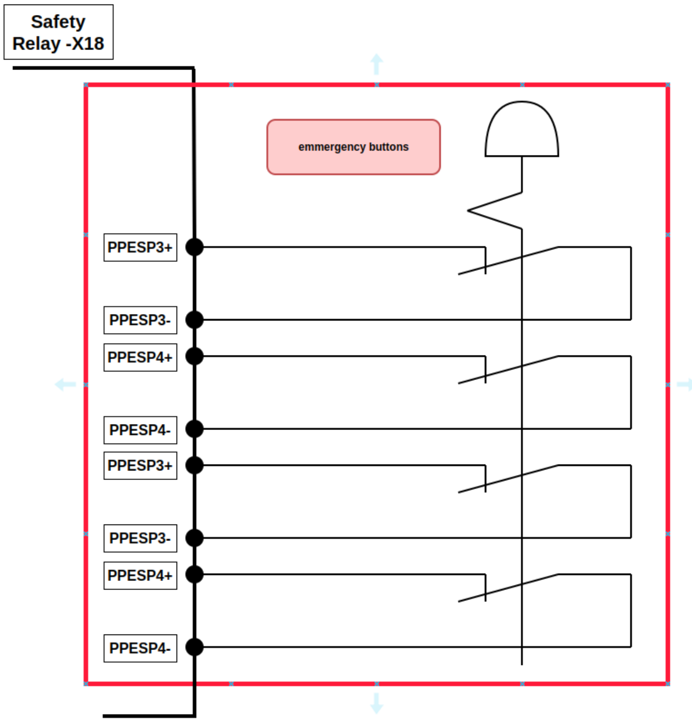


Figure 12: emmergency button wiring diagram

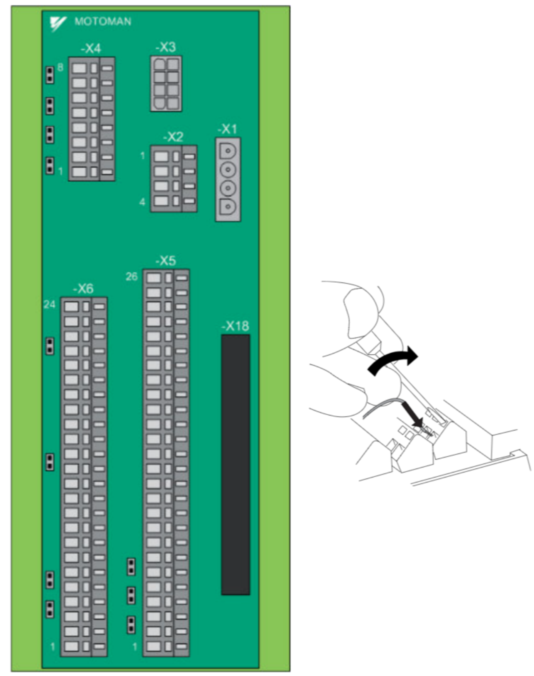


Figure 13: Safety relay

3.3 Sizing

To ensure correct installation and commissioning, it is essential to size the floor fastenings and electrical installation.

3.3.1 Mechanical Fastening

The robot arms move linearly, so it is crucial to properly anchor the rails to the floor. The precision of 3D printing is particularly sensitive to even minor unwanted movements. We examined two possible solutions[5]:

- Encase the fixed part of the rails in concrete to secure them to the floor
- Use chemical anchors

For cost and speed reasons, the chosen solution was to use mechanical anchors (threaded rods bonded with epoxy). A calculation of the mechanical anchor strength was performed to select the appropriate diameter and rod length.

3.3.2 Electrical Power Supply

The controllers and robot arms require proper electrical power supply. Additionally, certain tools and control processes for manufacturing also require specific electrical power supply. The controllers and robot arms are designed to operate on a 480V three-phase power supply (in accordance with American standards), while the current tools are powered by 110V. In line with the current electrical installation and Motoman specifications[1], we decided to install a three-phase circuit breaker (20A per phase) for each controller.

3.3.3 Dynamic Cable Carrier

The robot arms move linearly over a distance of 9 meters along the rails using a rack-and-pinion system, and they are electrically powered by cables. Due to this linear movement, it is necessary to move the electrical cables to power the robots. A fixed installation is not feasible in this case, so a dynamic cable carrier is required. We selected the most suitable dynamic cable carrier for our application. Dimensioning steps included[4]:

- Creating a list of required cables
- Calculating required dimensions
- Researching available products

We then verified the integration of the dynamic cable carrier in the installation plan.

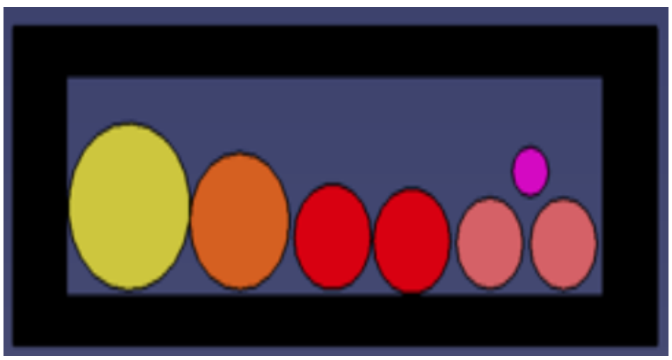


Figure 14: Vertical cut of dynamic cable carrier

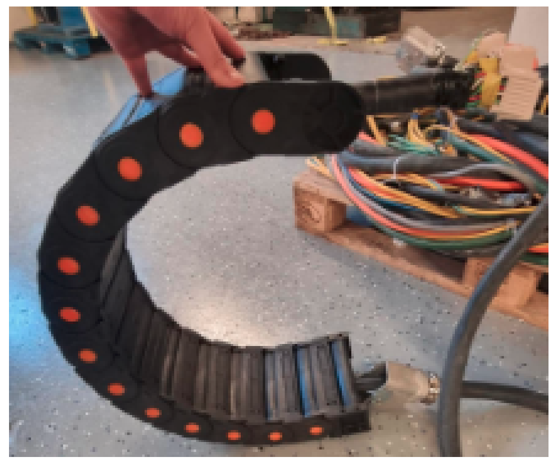


Figure 15: Dynamic cable carrier test

3.4 Restoration of the Robots

The arms, rails, and controllers obtained by the laboratory are ten years old. They have been used in industry, assembled and disassembled multiple times. Subsequently, they were stored outdoors for several years. The effects of time, humidity, UV radiation, and apparently abrupt disassembly have inevitably caused damage to the arms, rails, and controllers. Therefore, restoration was necessary. The initial step for each component was thorough cleaning.

3.4.1 Restoration of the Controllers

Cables were cut in the cabinet, and some modules and fuses were missing. Surprisingly, corrosion had not affected the electrical circuits. After initial cleaning, the following actions were taken[3]:

1. Identification and verification of all modules, followed by replacement of broken or missing ones
2. Identification and verification of all cables, with replacement of broken or missing ones
3. Identification and verification of all fuses, with replacement of broken or missing ones
4. Identification and verification of all safety circuits and emergency stop systems

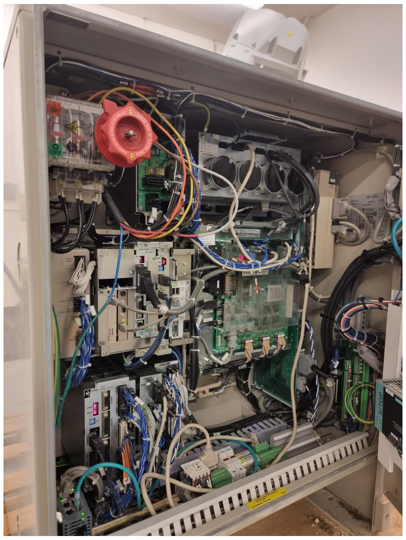


Figure 16: Controller in restoration

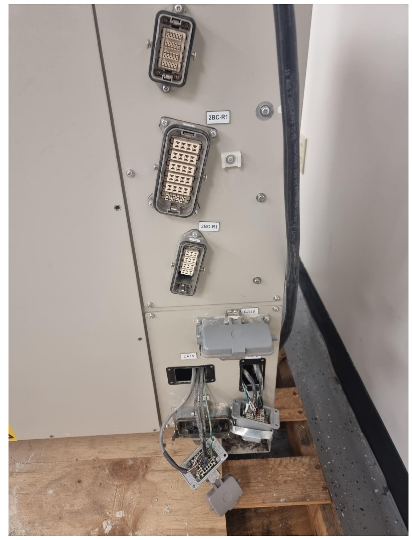


Figure 17: Harting connector connection

3.4.2 Restoration of the Rail System (7th Axis)

After cleaning, the rails required the following actions[5]:

1. Verification of all connectors (motors, encoders), with replacement of broken or missing ones.
2. Verification of the lubrication system.
3. Verification of the rack-and-pinion systems and their teeth.

3.4.3 Restoration of Motoman Arms

For the Motoman arms, the following actions were taken[5]:

1. Verification of all connectors (motors, encoders), with replacement or reconnection of broken or missing ones.

These restoration efforts were crucial to refurbishing the acquired components and ensuring their functionality within the additive manufacturing application.

3.4.4 Connection Test of the Assembly and Alarm Deactivation

Given the condition of the various parts, it was evident that we needed to test the startup of the controllers to see if they would start and what errors or alarms the controllers would generate. Surprisingly, the controller started on the first attempt, and we immediately backed up the old configuration onto a hard drive[1].

Subsequently, I encountered the following issue:

1. Switching to Maintenance Mode (Finding the Associated Password): Fortunately, the initial password provided by Motoman had not been changed.
2. Alarm 1325: Encoder Communication Problem: This alarm was caused by multiple issues:
   - Initially, encoders for all axes were affected, stemming from poor connections at the Harting connector.
   - Then, only the 7th axis was affected, resulting from a discharged battery powering the 7th axis.
   - The 7th axis remained affected, due to the encoder no longer functioning. We replaced it.
3. Alarm 1325: Security System Problem: The cables of the old security system needed to be identified (pending final installation, we temporarily bypassed the system using a jumper).

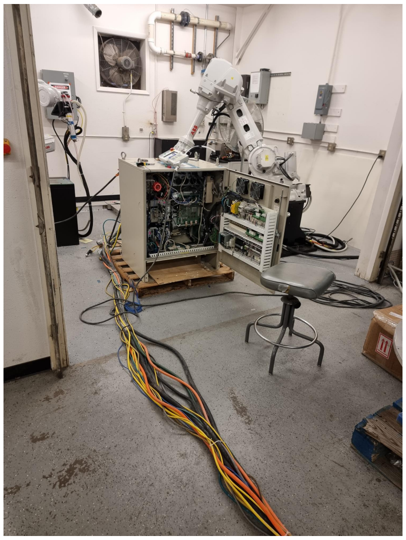


Figure 18: Articulated Motoman arm

Figure 19: Controller in restoration

4. Alarm 1325: Emergency Stop System Problem: Similarly, the cables of the old emergency stop system needed to be identified (pending final installation, we temporarily bypassed the system using a jumper).
5. Alarm 4151: Overrun Calibration Problem: The encoders needed to be recalibrated, and the overruns needed readjustment to avoid them being overly sensitive.
6. Alarm 9095: User-Defined Alarm set by previous owners to indicate tool not locked: We were unable to resolve this alarm, but two potential solutions may be considered:
   - Determine how to define and modify user alarms (likely done through Motoman software).
   - Perform a controller flash, as we already have a backup, which could potentially clear all user alarms defined for the previous owner's application.

   In any case, new user alarms will need to be defined for the additive manufacturing application.

It's important to note that the severity of Motoman alarms[1] is determined by the first digit:

- Alarms from 0 to 3 are major alarms. To clear them, the process involves:
  1. Turning OFF the main power supply
  2. Correcting the cause of the alarm
  3. Turning ON the main power supply again
- Alarms from 4 to 8 are minor alarms. To clear them, the process involves:
  1. Correcting the cause
  2. Pushing the "RESET" button under the ALARM window or using the system input signal
- Alarms with the digit 9 are user alarms. To clear them, the process involves:
  1. Correcting the cause for which the system input signal for the system or user alarm request turns ON
  2. Pushing the "RESET" button under the ALARM window or using the system input signal

"

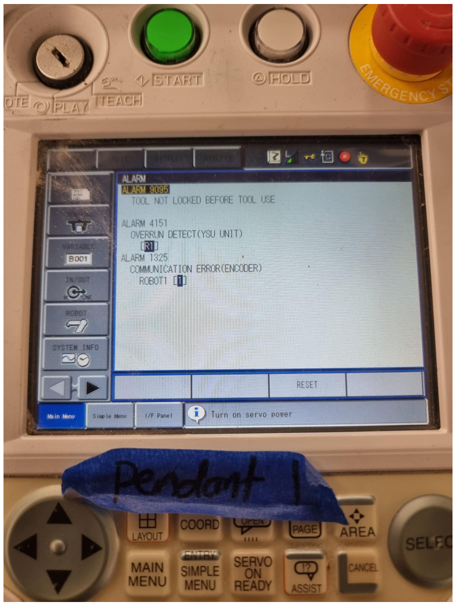


Figure 20: Alarm window

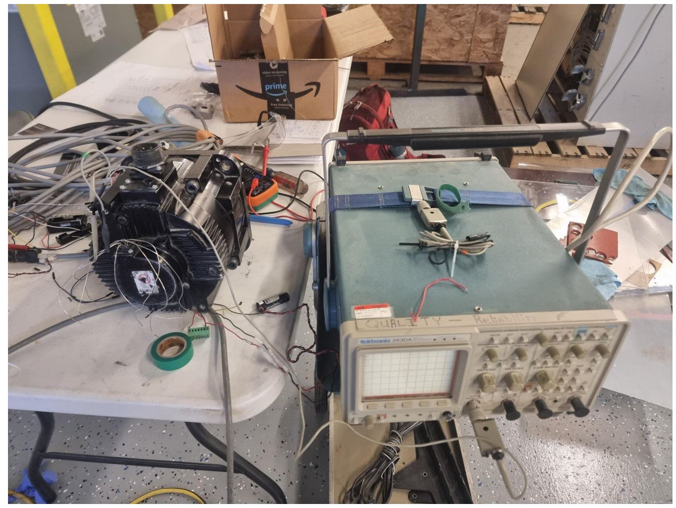


Figure 21: Encoder testing

3.5 Installation of Robots and Rails

3.5.1 Installation Planning

The installation of robots and rails requires the initial setup of the electrical power (in case it's done by the laboratory team and not by a university subcontractor):

1. Pull cables to the cabinet location.
2. Mount cabinets on the wall and install circuit breakers (20A or 30A based on availability).

The installation of robots and rails will follow the following steps:

1. Precise handling, assembly, and alignment of rails on the floor.
2. Marking and drilling holes for threaded rods that will hold the fixtures through rail holes.
3. Disassembly and movement of rails out of the Motoman robot zone.
4. Insertion of rods into holes and securing with epoxy resin for a strong fixation.
5. Waiting for two days for the resin to dry and ensure stable fixation.
6. Precise handling, assembly, and alignment of rails on the floor by passing rods through holes.
7. Adjustment of rail level using leveling bolts for optimal position.
8. Pull cables through cable carriers and install cable carriers.
9. Pull cables from emergency stop buttons and install emergency stop buttons.

3.6 Connection and Network Configuration

Once the robots and rails are installed, it will be necessary to connect and guide the controller cables to the robot arms along the rails. Additionally, compressed air, electrical, and communication networks will need to be pulled to the end effector for the process.

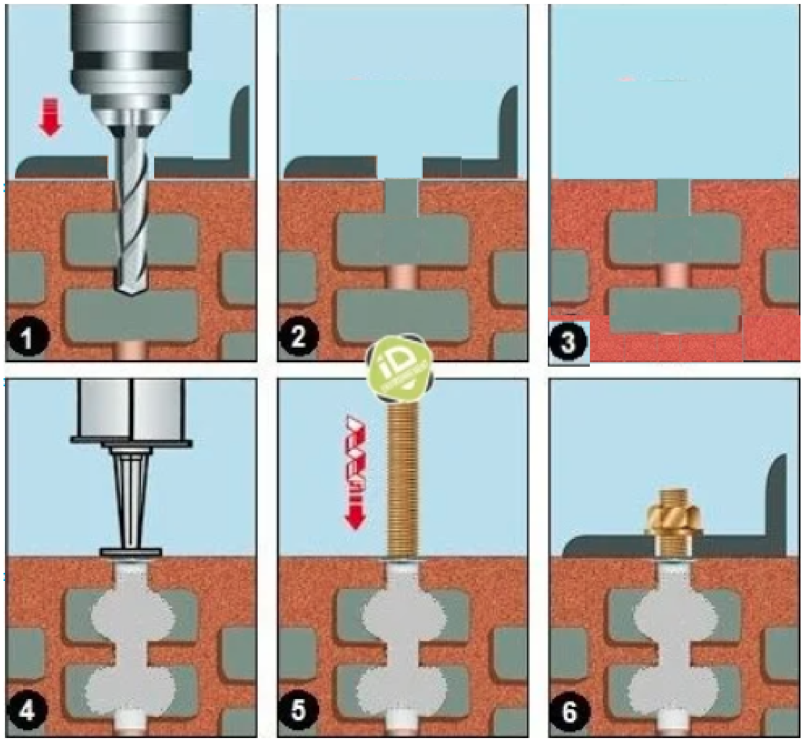


Figure 22: Installation Steps

3.7 Safety Protocol and Emergency Stop

After the hardware installation, software configuration and program development will be necessary to facilitate the startup, operation, and maintenance of the robots. Communication tests will also need to be conducted to ensure the proper functioning of the safety protocol and communication between the robots and the utilized software.

4 Virtual Machine and Simulations

In the context of simulating path planning and communication with the robots, the use of specific tools and software is crucial. Several options are available, ranging from proprietary software like Motoman and ABB to open-source alternatives like RobotDK. For this project, I opted for a combination of open-source tools, including ROS (Robot Operating System), ROS-Industrial, MoveIt, and RViz.

4.1 Virtual Machine Configuration

To simplify the use of simulation tools for laboratory students, I decided to install these software within a virtual machine. This approach offers several advantages. Firstly, it allows students, even without robotics expertise, to have a preconfigured system, avoiding the need for manual installations.

Moreover, I can create an ISO image of my virtual machine, which I can share with students. This allows them to easily install the tools on their own machines, regardless of their operating system. Lastly, as I already have ROS2 installed on my computer, this separate virtual machine avoids any potential conflicts with ROS1.

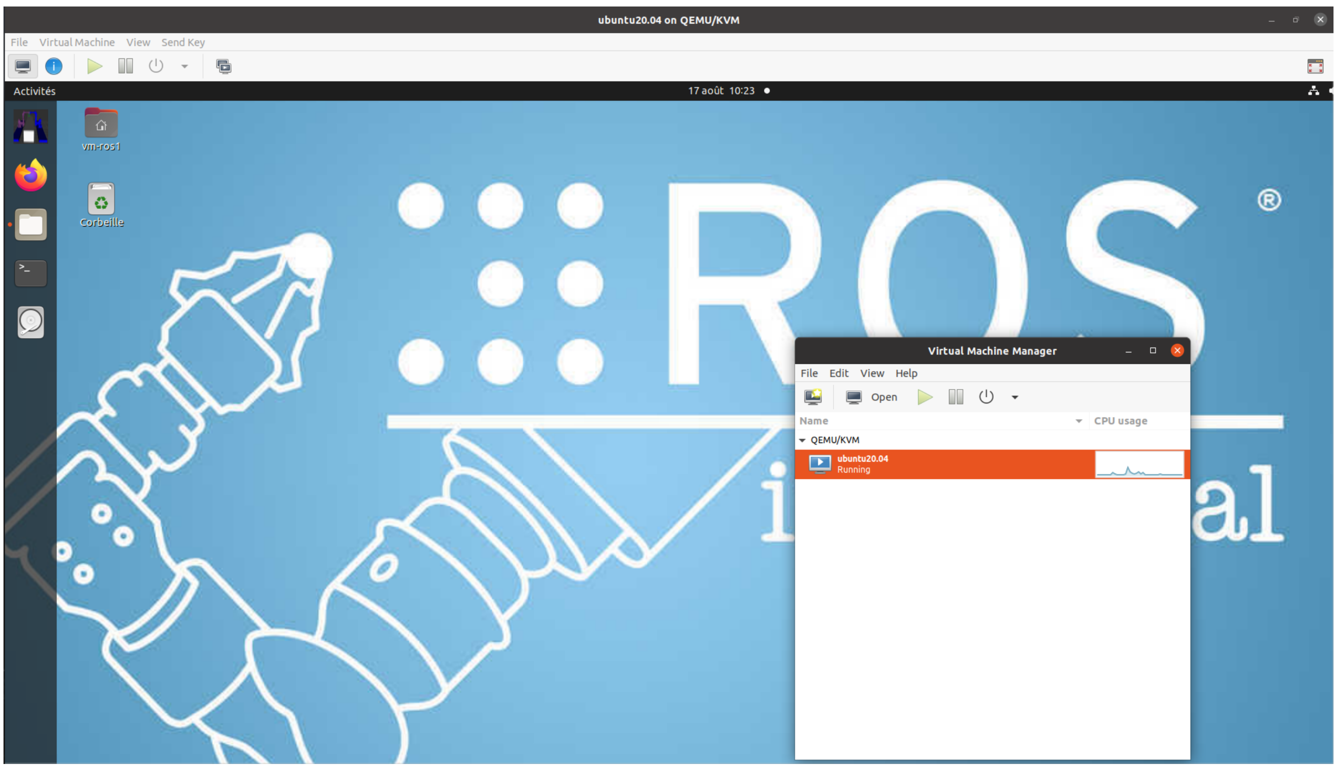


Figure 23: Virtual Machine Interface

4.1.1 Installation of Software and Packages (ROS, ROS-Industrial, MoveIt, RViz, ...)

For my virtual machine, I chose to install the ros-desktop-full version. This required downloading and installing additional packages such as Motoman, ABB, and ROS-Industrial. Some of these packages had to be recompiled, especially the drivers, to ensure compatibility and functionality with this version of ROS (Noetic).

4.2 Definition and Creation of Virtual Environment

Simulation of the virtual environment requires the establishment of a virtual model using a (.xacro or .urdf) file. Within the Motoman package of ROS-Industrial[7], the .xacro file for the ES200D robot arm was available, however, the external axis (the prismatic rail attached to the robot base) was missing. Thus, I had to complete this by adding the rail and its linkage in a .xacro file. This file acts as a kind of macro for the overall environment.

Subsequently, I designed the global environment configuration file, including the printing table and the two robots. I positioned the robots and the printing table by defining their appropriate coordinates and rotations, while loading the macro file. To create virtual models of Motoman robot arms in the simulation environment, I used the MoveIt package. This process allowed verification of kinematic and dynamic characteristics as well as motion constraints, especially for the 7th axis. As a result, virtual models of the robots were adjusted to faithfully match their physical counterparts.

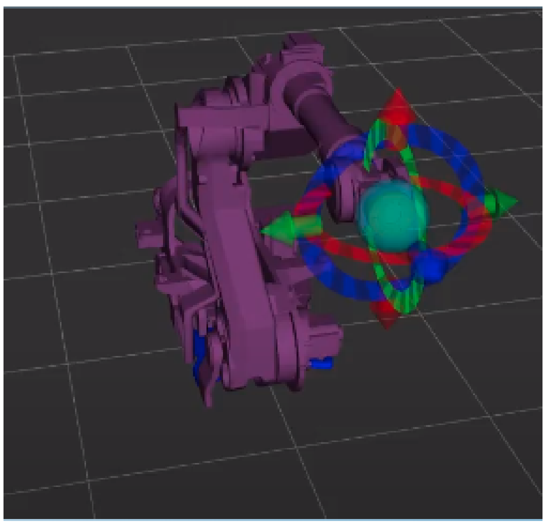


Figure 24: Motoman Package

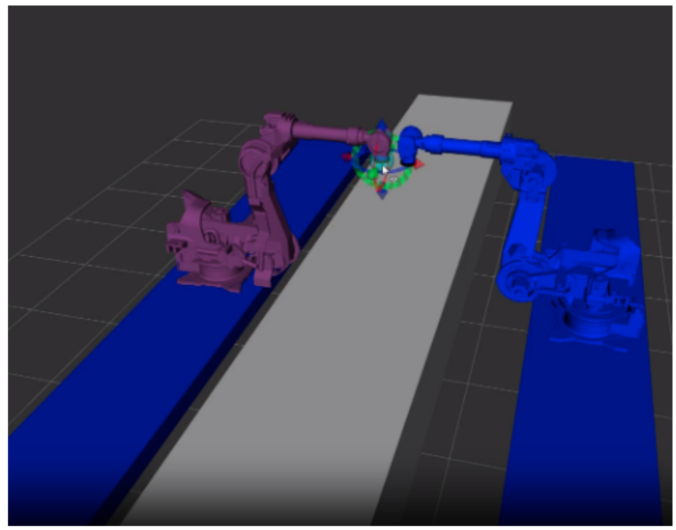


Figure 25: Simulation Environment

4.3 Simulation

Simulation of the Motoman robots was conducted using the previously configured software tools and environment[8][9]. Several simulation programs and scenarios were developed to cover various applications.

4.3.1 Calibration, Standby, and Emergency Restart Programs

A calibration program will need to be developed to accurately determine and adjust the positioning parameters of virtual robots relative to their physical real-world configuration. This process will ensure an exact correspondence between simulated movements and actual robot movements. Additionally, a specific program will be created to safely position the robots in standby mode at the end of each use. This involves removing tools, folding arm positions, and securing safety devices. Moreover, a restart program will be developed to reset the robots and bring them back to operation following an emergency situation or unexpected stoppage. This program includes system checks, position reset procedures, and resumption of operations.

5 Conclusion

5.1 Summary of Internship Achievements

This internship allowed me to accomplish several significant steps in the installation of Motoman robots in the laboratory. I planned and executed the physical installation of the robots, taking into account mechanical, electrical, and safety aspects. I also configured a virtual machine to enable simulations and communication tests with the robots. By utilizing tools and software such as ROS, ROS-Industrial, MoveIt, and RViz, I could simulate robot movements and test trajectory planning functionalities.

5.2 Assessment of Achieved Objectives

Overall, the main objectives of the internship have been achieved. The installation of Motoman robots in the laboratory was studied and planned, considering space constraints and ensuring secure rail fixation. I also set up the necessary software environment for simulations and communication testing. The virtual machine on which all the required tools are installed and configured allows easy and rapid sharing and installation for doctoral students. Safety and emergency stop protocols have been established, thus ensuring the safety of users and equipment. Furthermore, I conducted detailed CAD of the laboratory and surrounding components, which facilitated the planning and installation of the robots. I also sized mechanical fixtures and electrical installations according to the robots' requirements. The implementation of a dynamic cable carrier resolved the issue of cable movement along the rails. All documents (installation protocols, safety measures, maintenance guidelines, etc.), the virtual machine, codes, dimensioning sheets, weekly reports, and presentations have been stored on the laboratory's NAS to facilitate the transmission of my work to successors and laboratory staff.

5.3 Perspectives for Improvement and Future Development

Despite the accomplished achievements, there are prospects for improvement and future development for this Motoman robot installation. The first step would be to physically install the robots and rails. Further exploration of advanced features within ROS, ROS-Industrial, and MoveIt would be beneficial to optimize robot movements and enhance trajectory planning. Moreover, broadening the applications of Motoman robots in the laboratory by collaborating with doctoral students on specific research projects would be interesting. Furthermore, the addition of additional sensors and implementation of monitoring systems could enhance quality, accuracy, and enable better operations management.

5.4 International Mobility Assessment

This internship provided us with the opportunity to explore and visit the Midwest region of the United States, from the majestic Grand Canyon to Yellowstone National Park, passing through the Western-themed towns of Wyoming and the towering skyscrapers of Las Vegas. We had the chance to meet warm-hearted Americans who welcomed us with open arms and shared their lives, habits, and culture. This experience improved our English language skills and deepened our understanding of various aspects of American society.

In conclusion, this internship has been both enriching and productive, both professionally and personally. It allowed me to put my robotics knowledge into practice and acquire new technical skills. I had the opportunity to improve my English proficiency and explore a part of this vast country, along with its rich culture. The achievements during this internship form a solid foundation for future projects in the fields of robotics and 3D printing."

Annexes

Diagrams and CAD drawings of the Motoman robot installation

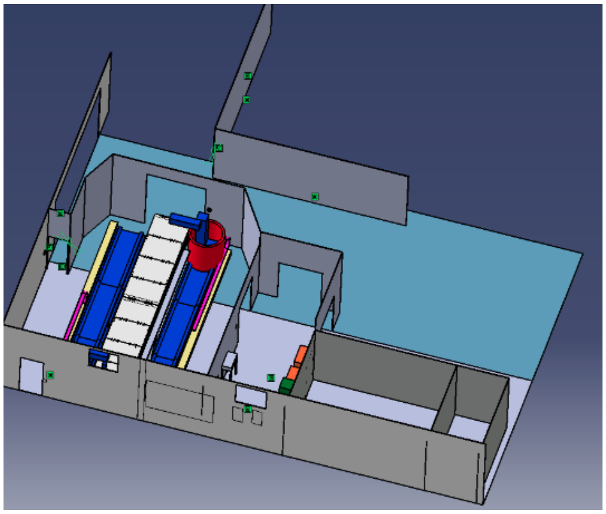


Figure 26: Motoman room CAD 1

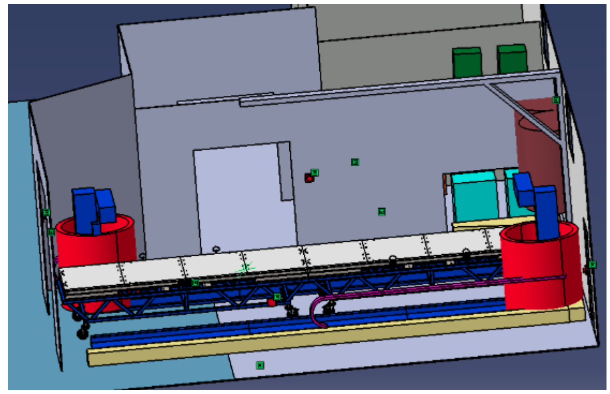


Figure 27: Motoman room CAD 2

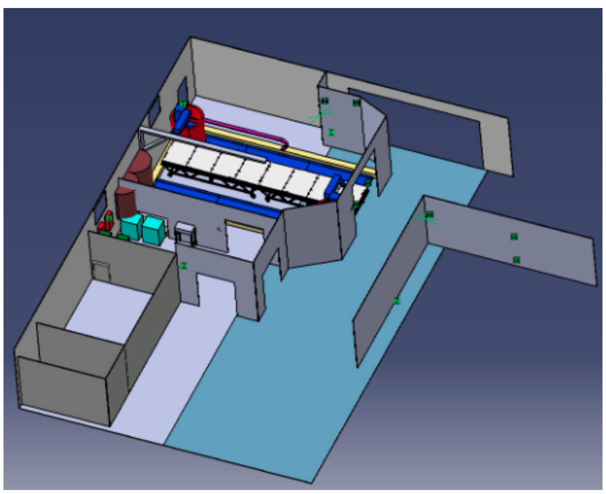


Figure 28: Motoman room CAD 3

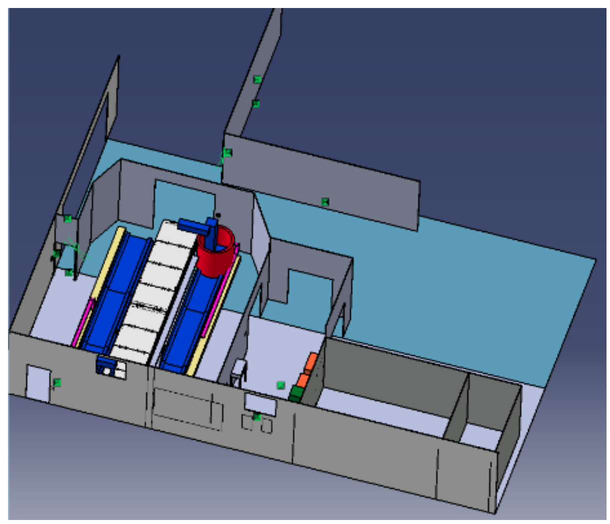
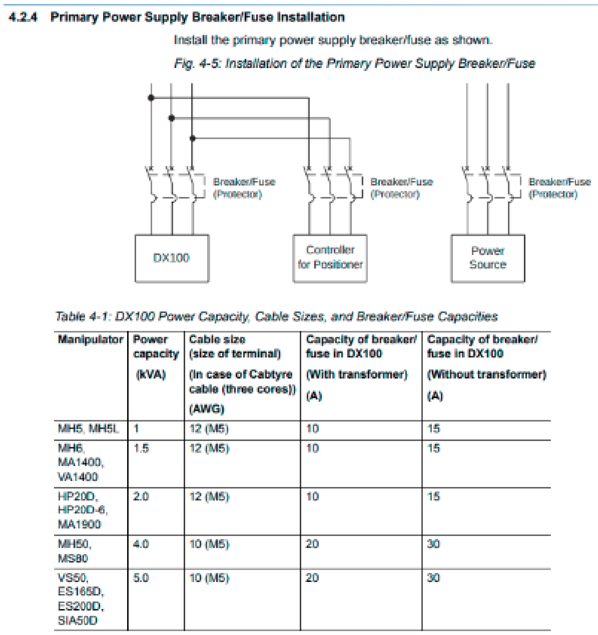


Figure 29: Motoman room CAD 4

Electrical sizing


4.2.4 Primary Power Supply Breaker/Fuse Installation

Install the primary power supply breaker/fuse as shown.

Fig. 4-5: Installation of the Primary Power Supply Breaker/Fuse

Table 4-1: DX100 Power Capacity, Cable Sizes, and Breaker/Fuse Capacities

Manipulator	Power capacity (kVA)	Cable size (size of terminal) (In case of Cabtyre cable (three cores)) (AWG)	Capacity of breaker/ fuse in DX100 (With transformer) (A)	Capacity of breaker/ fuse in DX100 (Without transformer) (A)
MH5, MH5L	1	12 (M5)	10	15
MH6, MA1400, VA1400	1.5	12 (M5)	10	15
HP20D, HP20D-6, MA1900	2.0	12 (M5)	10	15
MH50, MS80	4.0	10 (M5)	20	30
VS50, ES165D, ES200D, SIA50D	5.0	10 (M5)	20	30

Figure 30: Power supply diagram

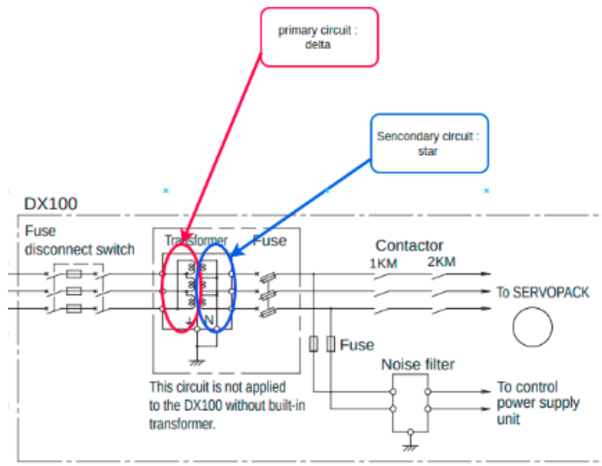


Figure 31: Transformer diagram

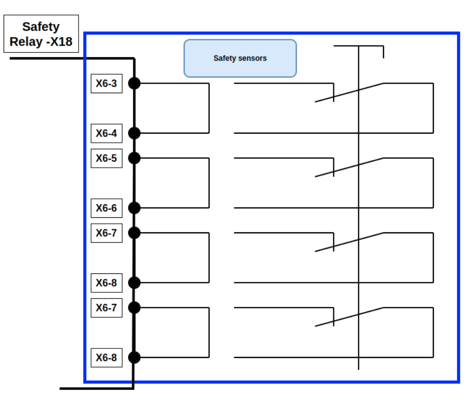


Figure 32: External safety circuit connection diagram

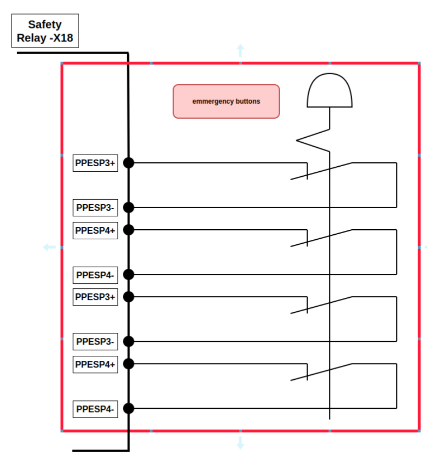


Figure 33: Emergency Stop Button Connection Diagram

Dynamic cable carrier sizing[4]

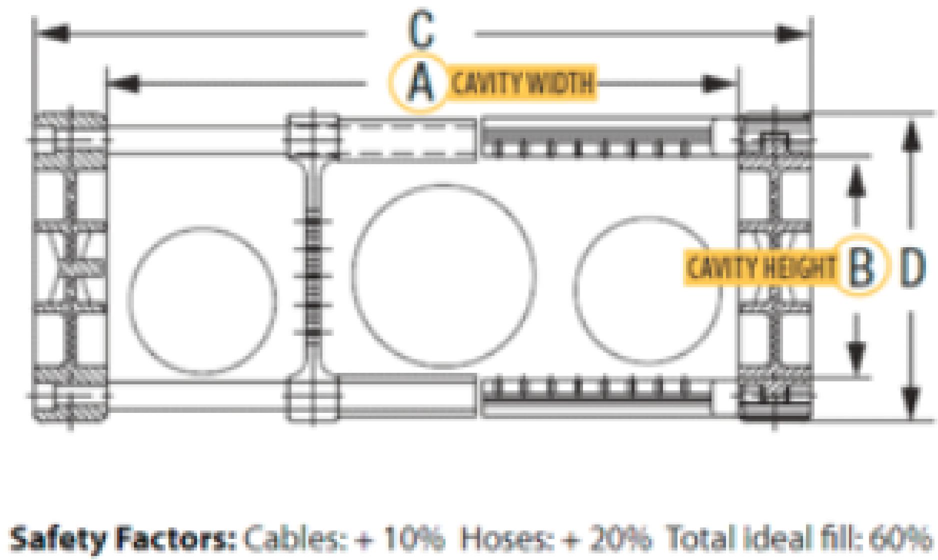


Figure 34: Cable carrier sizing diagram 1

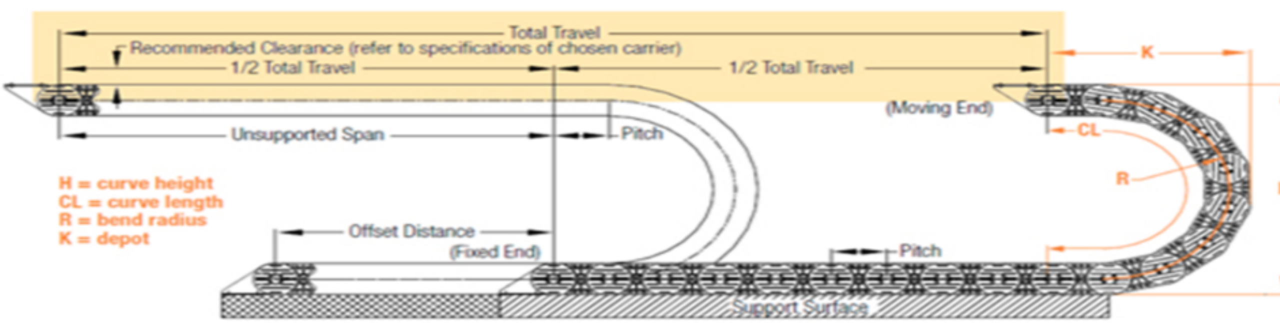
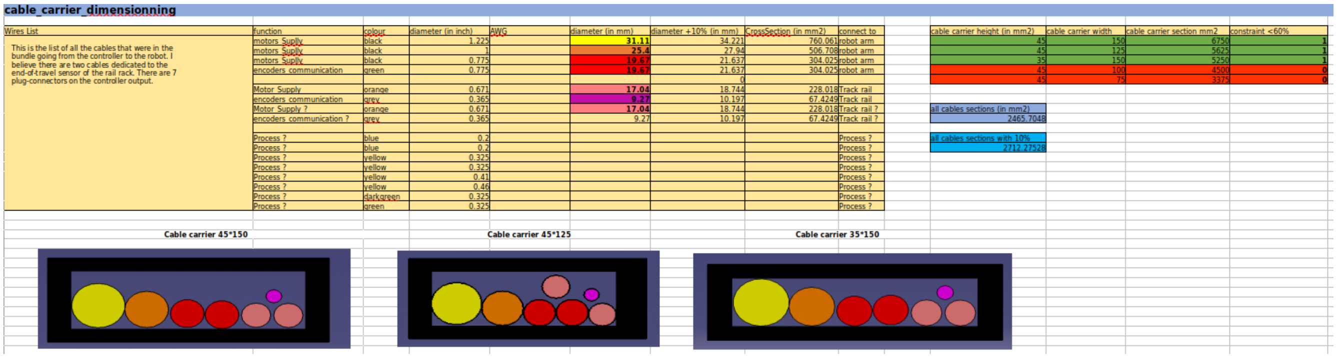


Figure 35: Cable carrier sizing diagram 2



cable_carrier_dimensionning

Wires List

This is the list of all the cables that were in the bundle going from the controller to the robot. I believe there are two cables dedicated to the end-of-travel sensor of the rail rack. There are 7 plug-connectors on the controller output.

function	colour	diameter (in inch)	AWG	diameter (in mm)	diameter +10% (in mm)	CrossSection (in mm2)	connect to
motors Suplly	black	1.225		31.11	34.221	760.061	robot arm
motors Suplly	black	1		25.4	27.94	506.708	robot arm
motors Suplly	black	0.775		19.67	21.637	304.025	robot arm
encoders communication	green	0.775		19.67	21.637	304.025	robot arm
					0		
Motor Supply	orange	0.671		17.04	18.744	228.018	Track rail
encoders communication	grey	0.365		9.27	10.197	67.4249	Track rail
Motor Supply ?	orange	0.671		17.04	18.744	228.018	Track rail ?
encoders communication ?	grey	0.365		9.27	10.197	67.4249	Track rail ?
Process ?	blue	0.2					Process ?
Process ?	blue	0.2					Process ?
Process ?	yellow	0.325					Process ?
Process ?	yellow	0.325					Process ?
Process ?	yellow	0.41					Process ?
Process ?	yellow	0.46					Process ?
Process ?	darkgreen	0.325					Process ?
Process ?	green	0.325					Process ?

cable carrier height (in mm2)	cable carrier width	cable carrier section mm2	constraint <60%
45	150	6750	1
45	125	5625	1
35	150	5250	1
45	100	4500	0
45	75	3375	0

all cables sections (in mm2)
2465.7048

all cables sections with 10%
2712.27528

Figure 36: Cable carrier sizing sheet

Figure 37: cable carrier with the cable

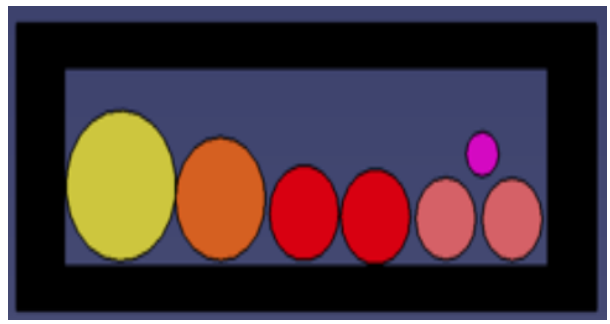


Figure 38: Cross-sectional view of the cable carrier

Robot Moving and Loading

Figure 39: Robot loading 1

Figure 40: Robot loading 2

Figure 41: Robot Unloading 1

Figure 42: Robot Unloading 2

Restoration of controllers, rails, and robots

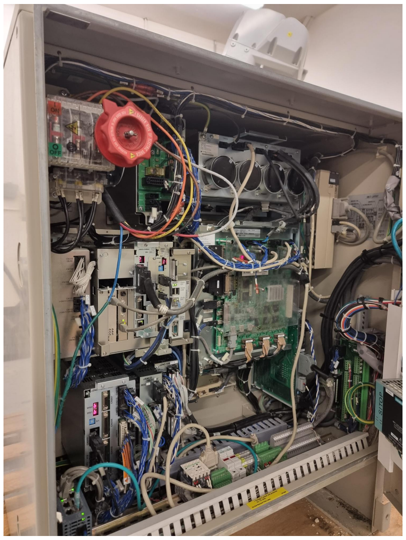


Figure 43: DX100 controller

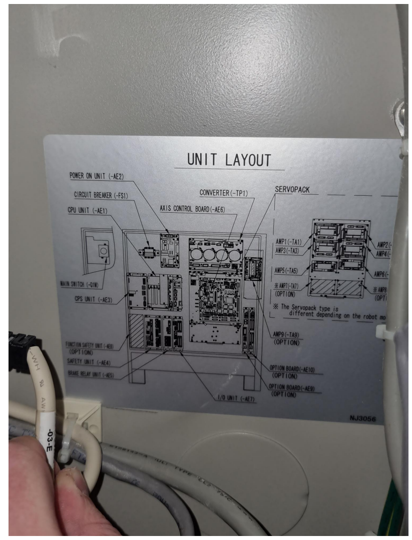


Figure 44: Controller diagram

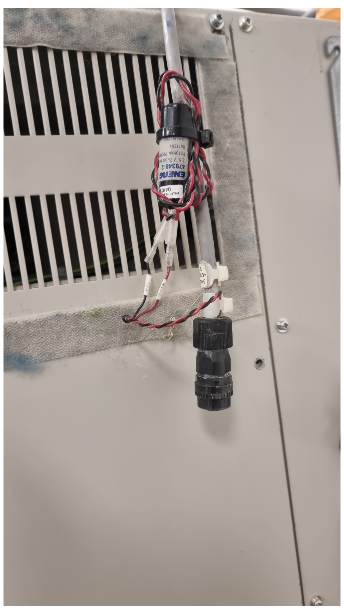


Figure 45: Battery and connector

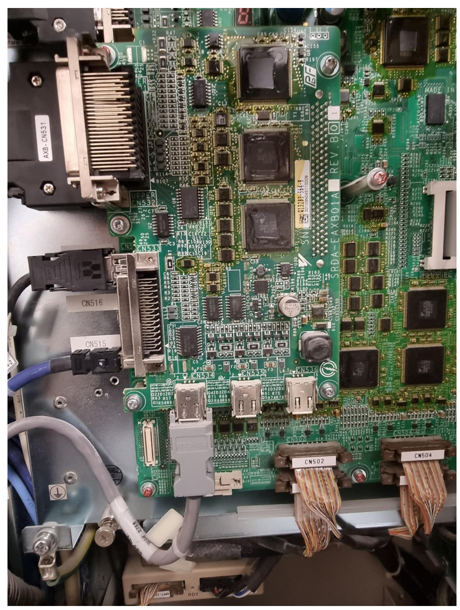


Figure 46: DX100 control board

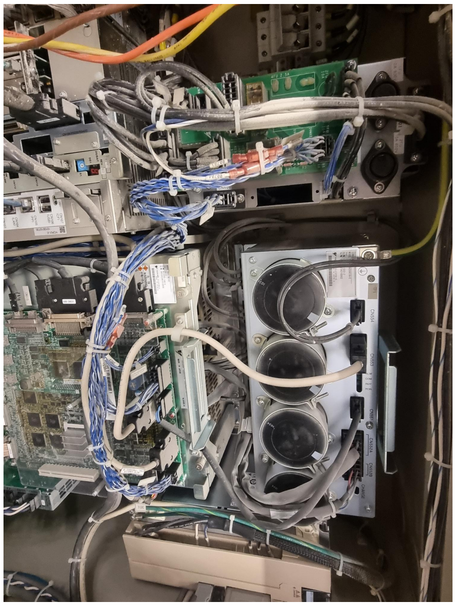


Figure 47: DX100 Power Module

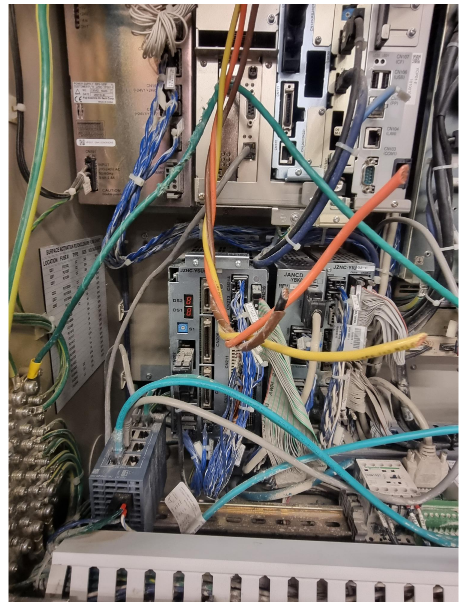


Figure 48: Safety and Emergency Stop Module

Figure 49: Rails

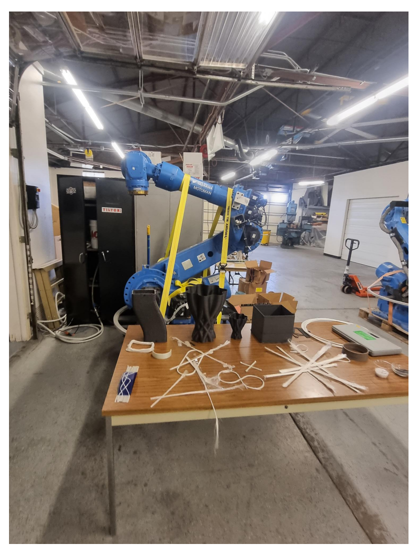


Figure 50: ES200D robot arm

List of Figures

References

[1] Dx100 maintenance manual. Technical report, Yaskawa-Motoman.

[2] Dx100 user manual. Technical report, Yaskawa-Motoman.

[3] Dx100 wiring manual. Technical report, Yaskawa-Motoman.

[4] Dynatec cable carrier sizing : https://dynatect.com/resources/how-to-size-your-cable-carrier.

[5] Es200d maintenance manual. Technical report, Yaskawa-Motoman.

[6] Es200d user manual. Technical report, Yaskawa-Motoman.

[7] Github - ros-industrial motoman driver : https://github.com/ros-industrial/motoman-driver.

[8] Moveit - ros motion planning framework : https://moveit.ros.org/.

[9] Qemu - wikipedia : https://fr.wikipedia.org/wiki/qemu.

[10] Ros - robot operating system : https://www.ros.org/.

[11] Ros-industrial : https://rosindustrial.org/.

[12] Ros wiki - motoman driver : http://wiki.ros.org/motoman-driver.