L'Institut National des Sciences Appliquées des Hauts-de-France (INSA)

Instituto Militar de Engenharia (IME)

Contribution to the development of a digital twin for the Computer Integrated Manufacturing (CIM) platform

Date of internship: from 04/09/2023 to 12/01/2024

Academic Advisor : Sebastien DELPRAT

Supervisor for the host organization : Paulo César PELLANDA

Educational institution: Institut National des Sciences Appliquées Hauts-de-France (INSA HDF)

Host Organization: Instituto Militar de Engenharia (IME) - Praça Gen. Tibúrcio, 80 Urca, Rio de Janeiro - RJ, 22290-270

NON-PLAGIARISM COMMITMENT

I, the undersigned Touri Maha, a student in the Mechatronics Engineering Specialization, declare that I am fully aware that plagiarism of a document or part of a document published on any existing form of support, including the Internet, constitutes a violation of copyright as well as characterized fraud.

Consequently, I undertake to cite explicitly, each time I use it, all the sources I have used to write this report.

Rio de Janeiro, 09/01/2024

Signature:

ACKNOWLEDGEMENT

First, I would like to express my sincere gratitude to my academic supervisor, **Mr. Sebastien DELPRAT**, for giving me this remarkable internship opportunity in Brazil. Your guidance, support, and steadfast commitment have been invaluable.

I would also like to thank **Mr. ANTONIO EDUARDO CARRILHO DA CUNHA** and **Mr. Paulo César PELLANDA** for providing me with this inspiring and enriching experience at IME, contributing significantly to my learning journey. Special thanks to **Mr. ANTONIO EDUARDO CARRILHO DA CUNHA** for your mentorship and valuable insights throughout the internship period.

I am also grateful to **Mr. Hebert Azevedo Sá** for ensuring that every aspect of my internship experience was seamless and enjoyable.

Moreover, I would like to thank **Mr. IGOR Paz** for his hospitality, his kindness, and the genuine interest he took in making sure everything was in place for me to thrive in my internship.

Also, I would like to thank **Mr. Antonio da Silva Nascimento Filho** for assisting me with the technical aspects of my internship.

My sincere appreciation goes to the students of IME for their camaraderie and collaborative spirit.

Lastly, I want to express my deepest gratitude to my parents for their love, support and encouragement.

This internship has been a transformative experience, and It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the collective efforts that made it possible.

Table of contents

Introduction…….…… 5

1. **Chapter 1:** Context: Internship in Brazil…… 6
   1-1. Presentation of Brazil
   1-2. Presentation of IME
   1-3. Presentation of MecatrIME

2. **Chapter 2:** Learning objectives and methods employed…… 10
   2-1. Task description
      2-1-1 Context and motivations of the MecatrIME laboratory
      2-1-2 Objectives of the Internship
   2-2. Methods and Means Employed
      2-2-1 Documentation and research
      2-2-2 Forward and Inverse Kinematics
      2-2-3 Denavit-Hartenberg (DH) Convention
      2-2-4 Python IKPY Library
      2-2-5 ROS (Robot Operating System)
         2-2-5a Simulation of station 2 and the robotic arm
         2-2-5b Implementation of the process Turn demo st2
   2-3. Results and Interpretations

3. **Chapter 3:** Conclusion and Perspectives……..…… 34
   3-1 Challenges encountered
   3-2 Human Factor
   3-3 Added value for the MecatrIME Laboratory study case
   3-4 Tasks and their advancements
   3-5 Conclusion

4. **References**…..…….. 40
5. **Appendices**…… 41

Introduction

During the fourth year of my engineering program at INSA HDF, I did an internship at the Instituto Militar De Engenharia (IME) in Brazil, as a part of my curriculum.

Spanning a duration of four months, from September 4, 2023, to January 12, 2024, this experience provided me with a unique and inspiring opportunity to immerse myself in the captivating environment of Brazil and to contribute to enriching projects at IME.

The Instituto Militar de Engenharia is located in the beautiful city of Rio De Janeiro. It is an educational establishment of the Department of Science and Technology (DCT) responsible, within the Brazilian Army, for higher engineering education and basic research.

My primary responsibility was to contribute to an ongoing project focused on developing a digital twin for the Computer-Integrated Manufacturing (CIM) system at the MecatrIME Laboratory of IME. My responsibilities included in-depth research on robotic kinematics, translating this knowledge into a digital framework, and integrating simulated elements into the CIM system.

This experience provided a practical application of engineering concepts that I have learned during my curriculum and deepened my programming and simulation skills as well as my understanding of advanced manufacturing technologies. Overall, my internship at IME was a transformative experience, allowing me to grow professionally and personally.

In order to give a clear and analytical overview of the time spent at IME, this report will first give a presentation of the institute and the environment where the internship took place. Then it will go through all of the many duties and assignments completed during the internship, elucidating the personal and professional skills and qualities that were developed through this experience.

Chapter 1

Context : Internship in Brazil

1-1. Presentation of Brazil:

My internship took place in one of the most culturally diverse countries with breathtaking natural landscapes; Brazil. Located in South America, it is the largest country of the continent, covering an area of around 8.5 million square kilometers and has an official resident population of around 216 million. The official language is Portuguese, a legacy of the colonial period.
Brazil is known for its cultural and ethnic richness, due to the influence of indigenous peoples, Portuguese settlers, African slaves and various waves of immigration.

Brazil has a complex history, marked by its colonial past. Its independence, in 1822, was followed by significant economic growth, placing the country among the world's leading emerging economies.

Brazil's robotic industry has also been progressing, particularly in sectors like manufacturing, agriculture, healthcare, and research. The country has integrated robotics into manufacturing processes, with a focus on industries such as automotive and electronics.

Universities and research institutions in Brazil like IME contribute to global robotics advancements, although challenges include limited investment, infrastructure deficiencies and the need for a regulatory framework.

My immersion in this environment was a significant experience for me. In fact, working in a completely new nation with distinct mindsets, thriving cultures, growing industries and a new language taught me to adapt, communicate differently, and make compromises when needed.

1-2. Presentation of IME:

The Instituto Militar de Engenharia is a public institution of higher education belonging to the Brazilian army that offers undergraduate and postgraduate courses in engineering.
It was created in 1792, making it therefore the first engineering school in the Americas and the third in the world.
It is located in Urca, a neighborhood in Rio's south zone, more precisely in Praia Vermelha , next to the Sugar Loaf cable car station.

Its main mission is to train, specialize, and elevate personnel to a higher level in the scientific and technological field and collaborate, through teaching and research, for the progress of the Brazilian military and the country.

Recognized as a prestigious engineering institution, it is often called upon to participate in studies and research in both governmental and private spheres, aimed at the development of a wide variety of projects.

IME's key departments and areas of expertise are:
Fortification & Construction / Electrical / Electronics / Communication / Mechanical & Automotive / Mechanical & Armament / Chemicals /Cartography / Materials / Computer Science /Production (CFrm) / Nuclear (CFrm)

1-3. Presentation of MecatrIME:

The Mechatronics Laboratory (MecatrIME) of the Military Institute of Engineering (IME) arose from the need to concentrate existing skills in Electronics, Mechanics, Computing, Automation and Control, in order to carry out teaching, research, development and innovation in Mechatronic Systems with dual applications. The provision of the physical space and the acquisition of equipment was made possible by three projects covered by the FINEP CT-INFRA calls for proposals in 2005, 2006 and 2013, as well as budget funds.

The MecatrIME platform, illustrated in Figure 1, is composed of a management station (P9) , six production stations (P2-P7) , a control station for transporting parts on the conveyor belt (P8) and a storage station (P1) , thus totaling nine stations interconnected by a conveyor belt.

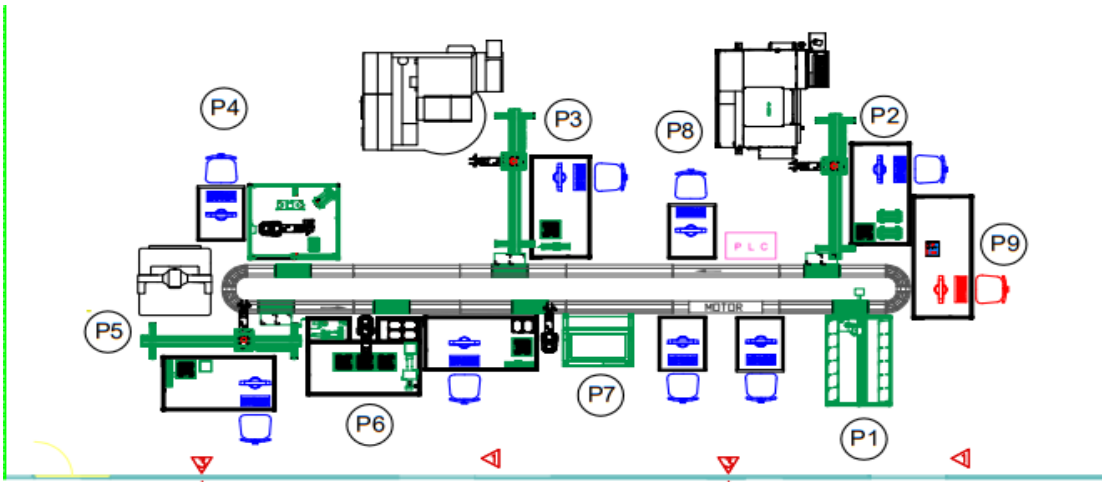


Figure 1 - MecatrIME Plant

- The description of each stations:

Station	Description
1 (Storage)	ASRS (Automated Storage and Retrieval System) - 72 shelves & reader RFID
2 (Turning)	MH5F Robot and CONCEPT Turn 250 CNC Lathe
3 (Milling)	MH5F Robot and CONCEPT Mill 250 CNC Machining Center
4 (Welding)	Automated Welding: MH5F Robot & Welding Booth
5 (QC with CMM)	Quality Control: MH5F robot mounted on 1.8m LSB rail to feed the Coordinate Measure Machine (CMM)
6 (Assembly & QC)	Assembly & Quality Control: MH5F Robot (Control, Collage & Machine Vision)
7 (Laser)	Laser Engraving: MH5F Robot & Laser Engraver
8 (Conveyor Belt)	Treadmill prepared for seven stations
9 (Management Station)	Management Station with OpenCIM Software online

The connection between them is made by cables forming a local network (Local Access Network - LAN) . Communication between computers at stations and the IME network is carried out through an Internet access point and communication between devices and computers is made using the RS232 protocol.
The stations can be used in isolation, without control from the management station. So, if there is a problem with a station, or the user only needs a single process, it can be used separately.

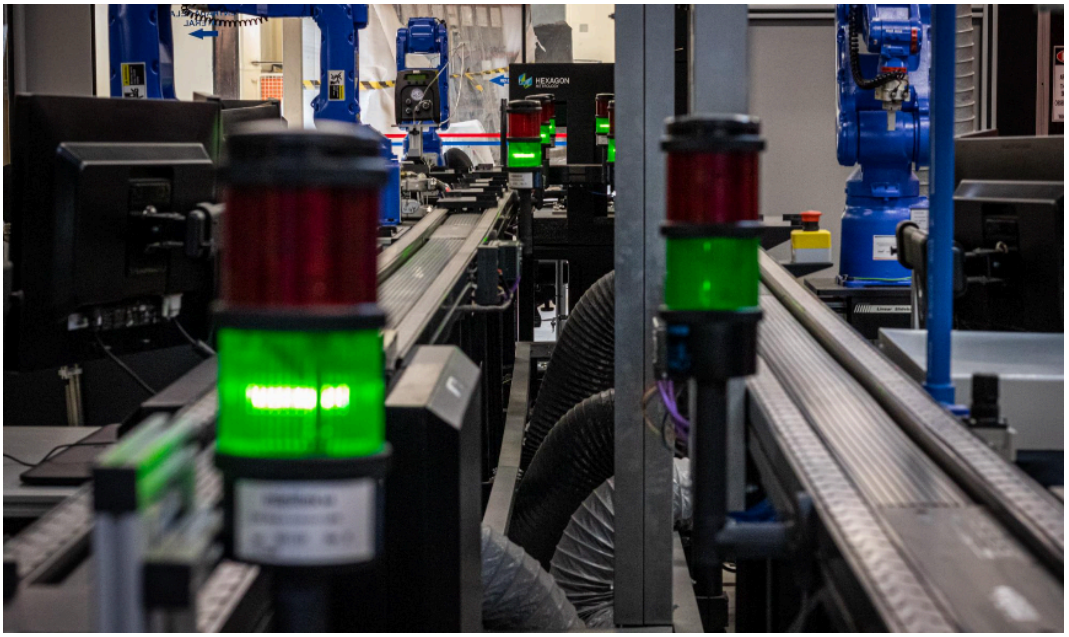


Figure 2 - MecatrIME Laboratory

Chapter 2

Learning objectives and methods employed

2-1. Task Description:

2-1-1 Context and motivations of the MecatrIME laboratory

Today, as the manufacturing sector embraces Industry 4.0, it is becoming an increasingly appealing target for cyber attackers. Without strong protections in place, bad actors can take advantage of systems for industrial espionage, intellectual property theft, IP leakage, or even production sabotage.

Acknowledging this thriving threat, the MecatrIME Laboratory, with its CIM Computer Integrated Manufacturing Platform, addresses this concern through cybersecurity research.

In order to analyze cyber vulnerabilities, suggest ways to recognize cyberattacks and avoid conducting tests on the real platform, the study case of the MecatrIME intends to create a digital twin that replicates both the physical and cybernetic components of the manufacturing environment.

The idea is to, on one hand, use kinematic or dynamic models of the equipment to mimic the physical part of the platform's assets in order to anticipate movements and interactions between the physical components, and on the other hand, to subject this digital twin to cyber attacks within a controlled cyber range environment to analyze the potential effects of these attacks on the physical platform. Moreover, the digital twin will also be used for offline programming of the platform and remote monitoring.

2-1-2 Objectives of the Internship:

As an intern in this Laboratory, I am tasked to actively contribute to the development of this digital twin of the Computer Integrated Manufacturing (CIM) system. Specifically, my assigned role revolves around developing the physical virtualization aspect of this cybersecurity research initiative. That is to say, my responsibility is to; **proficiently create a virtual representation of a station within the MecatrIME laboratory, encompassing the**

accurate kinematic simulation of the robotic arm and other essential components.

(we decided, my tutor and I, to only focus on simulating station 2 out of the nine available, as it was the only one functioning properly.)

To achieve this objective, my efforts are directed towards two distinct goals:

- First, the Development of a proof of concept for a virtual environment that accurately replicates the physical aspects of station 2 within the CIM platform.

- Second, the implementation of a kinematic simulation of a robotic arm within the developed virtual environment to detect potential collisions and to anticipate movements.

2-2. Methods and Means Employed:

Before delving into the main aspects of my internship, I opted to strategically plan and organize my workflow by utilizing a Trello board. This board provided me with a comprehensive view of the tasks at hand, those currently in progress, and those successfully completed. Utilizing this tool greatly streamlined the tracking of my work, enabling me to assign specific dates to each completed task.

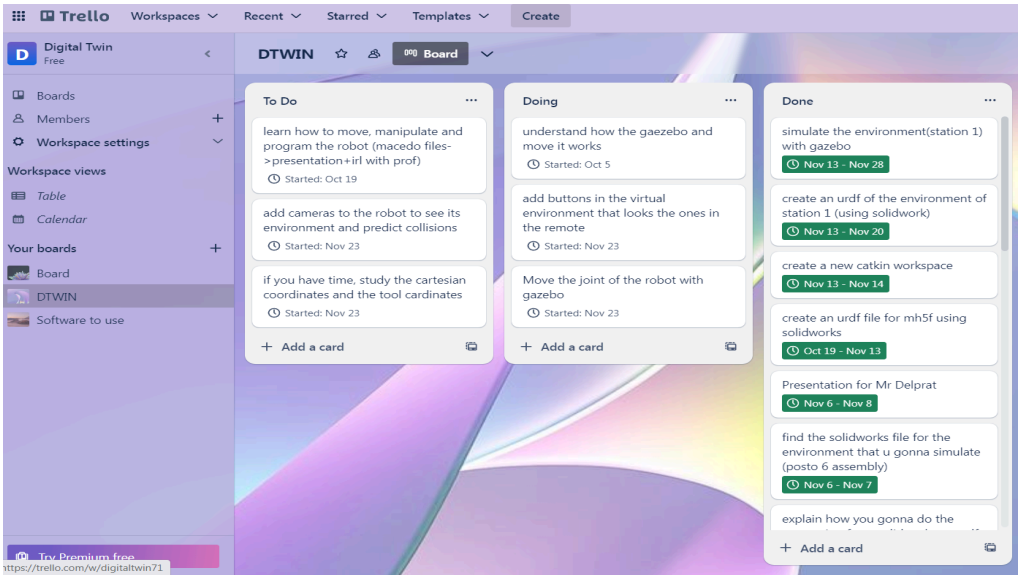
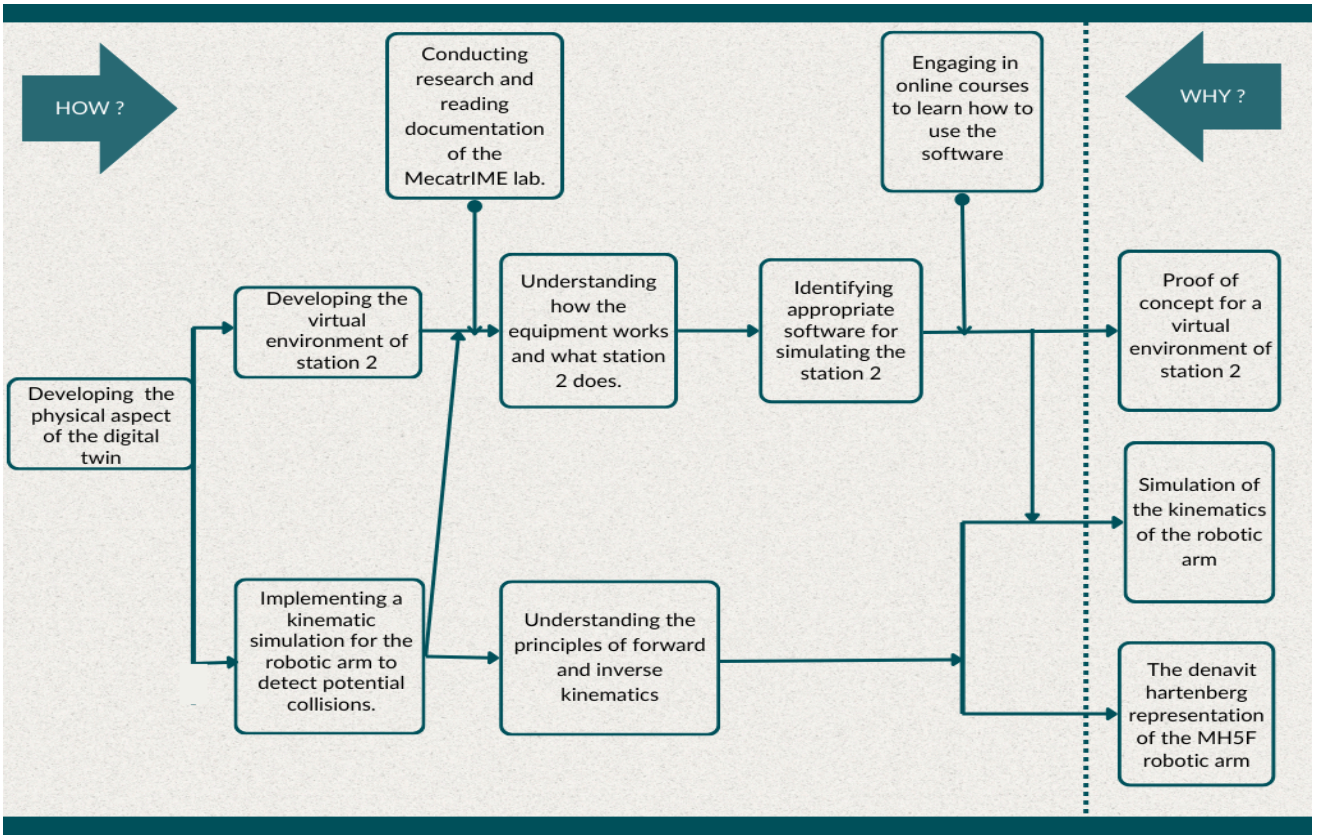


Figure 3 - Trello Board

To gain a comprehensive understanding of how to structure my work, I chose to employ the Function Analysis System Technique (FAST) in order to create a general view of the tasks at hand, facilitating a more strategic and organized approach to resolve the given problem.



2-2-1 Documentation and research:

This first step of my internship involved conducting research and immersing myself in the documentation of the MecatrIME lab. This involved a thorough examination of various documents, including previous reports and datasheets related to the diverse equipment within the lab. Special attention was given to comprehensively explore the MH5F Motoman robotic arm. This foundational step aimed to understand and know how to turn on the different equipment of the lab and understand the overall functions of each station, hardware specifications, diagrams, software used, electrical circuits and the technicalities of each equipment.

- The MH5F Motoman Robotic arm and the programming pendant :

The Motoman MH5F robot (Figure 4) boasts a 6-axis configuration, each axis providing precise rotational movements and relying on high-performance servo motors. These components work seamlessly to animate the end-effector, allowing for precise manipulation and control of tools and objects.

The programming pendant (Figure 5), is equipped with the keys and buttons used to conduct manipulator teaching operations and to edit jobs. Through hands-on experience, I discovered the programming pendant's diverse functionalities, like adjusting speed, fine-tuning movements, controlling the rotation of each joint and programming specific routines.

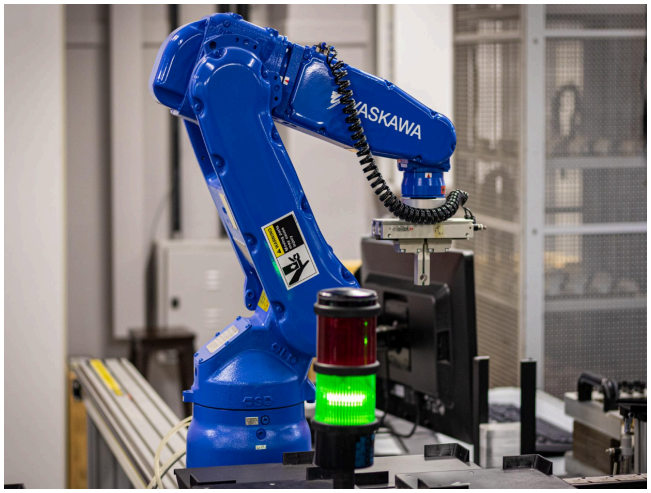


Figure 4 - Motoman MH5F Robotic Arm

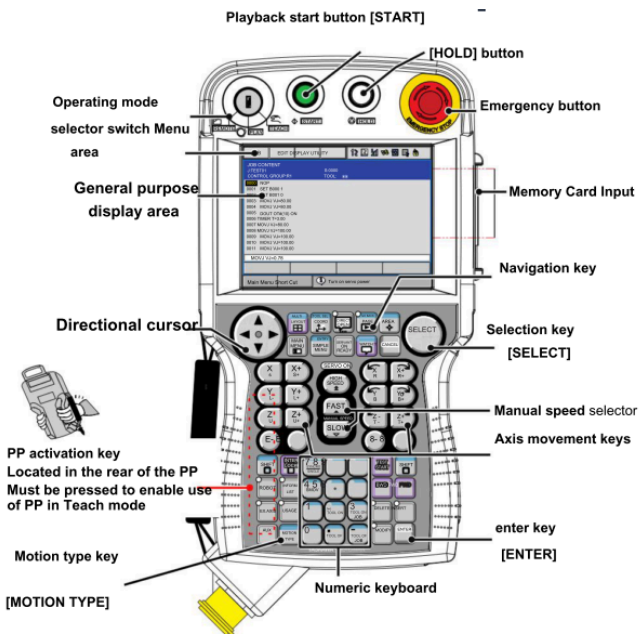


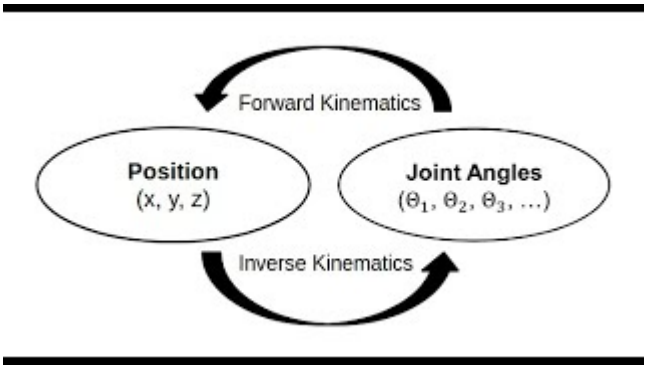
Figure 5 - Programming Pendant

2-2-2 Forward and Inverse Kinematics:

Kinematics, both forward and inverse, serve as fundamental concepts that elucidate the relationship between the joint angles and the resulting end-effector position. Understanding these principles is crucial for programming and controlling the robot accurately. Research about these concepts aimed to unravel the theoretical foundations governing the motion and positioning of the robotic arm.

- Définition :

Forward kinematics calculates the position and orientation of the end effector based on the angles and joint positions of the robotic arm. While Inverse Kinematics, on the other hand, calculates the angles and joint positions required for the end effector to reach a specified point in space.



2-2-3 Denavit-Hartenberg (DH) convention:

- Definition:

The Denavit-Hartenberg (DH) convention or notation is a commonly used convention for kinematics in robotic applications. It allows us to characterize the relative position of the end effector with just four parameters (instead of six). This comes at the price of certain restrictions in the choice of coordinate systems (reference frames). It is crucial for the kinematic simulation of robotic arms because it describes the robot properties like axis orientations, arm lengths and its position in space.

In order to find the Denavit-Hartenberg representation of the MH5F, we have to follow a bunch of steps:

- Step 1 : Assigning Coordinates Frame

The first step to facilitate the visualization of the joints and make the analysis procedure easier is to create a schematic model of the robotic arm.
To do that I started by closely examining all parameters that define the structure of the MH5F found on the datasheet.
The robot has 3 pivot connections and 3 sliding pivot connections.

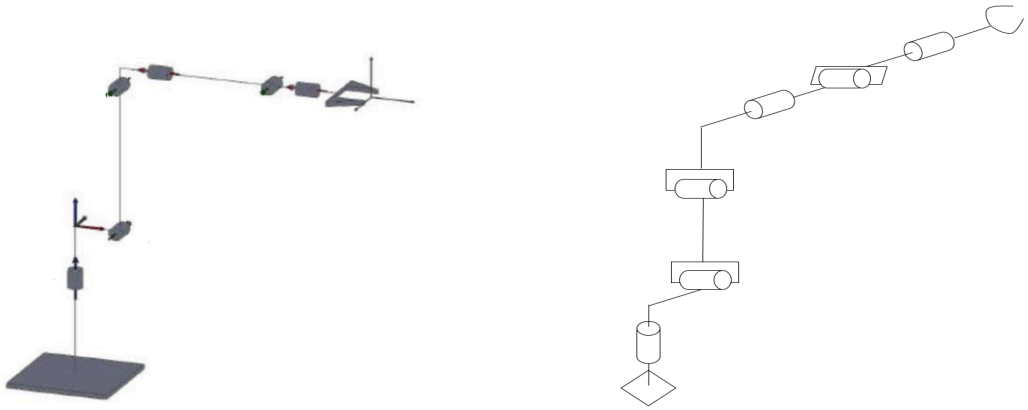


Figure 6 - Schematic model of the robotic arm

To draw the frames (i.e. x, y, and z axes) for each joint, we follow four rules that will enable us to take a shortcut for finding DH parameters for the robot:

1) The z-axis is the axis of rotation for a revolute joint.

2) The x-axis must be perpendicular to both the current z-axis and the previous z-axis.

3) The y-axis is determined from the x-axis and z-axis by using the right-hand coordinate system.

4) The x-axis must intersect the previous z-axis. (rule doesn't apply to frame 0)

Therefore, we start by assigning the z-axis for each joint according to the direction of the rotation of the joint. Then, we assign the x-axis for each joint, following the rules number 2 and number 4.

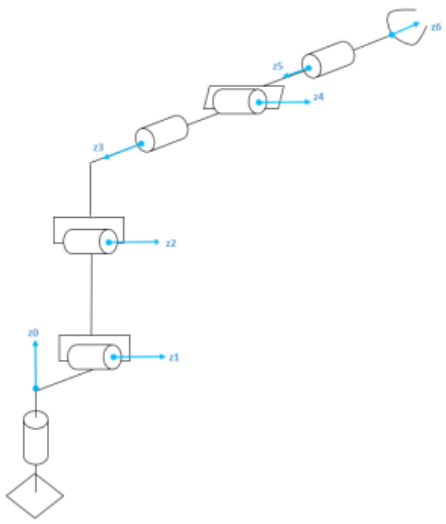


Figure 7 - Schematic model with the z axes

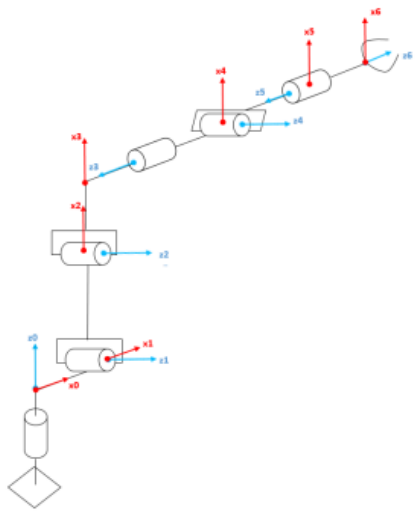


Figure 8 - Schematic model with the z and x axes

Finally, we determine the Y-axis using the right-hand coordinate system. This way we can illustrate the coordinate frames for each joint according to the DH convention in a 3D Model.

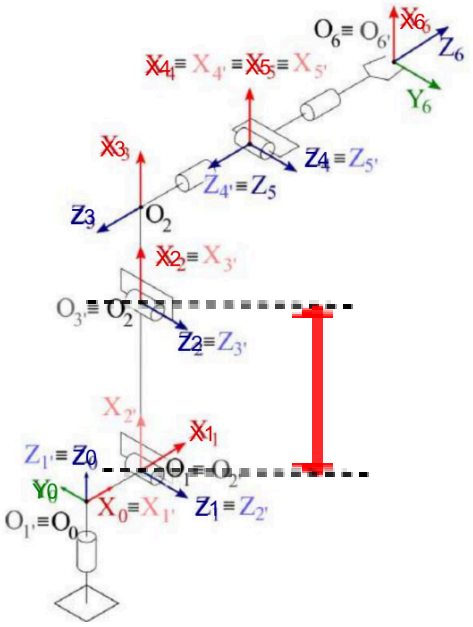


Figure 9- Schematic model and the coordinates frames for each joint

Now that we have our coordinate frames, we need to find the Denavit-Hartenberg parameters.

- Step 2 : Finding DH parameters

Since we have 7 coordinate frames here, we need to have 6 rows in our DH table (i.e. number of rows = number of frames –1).

The Denavit-Hartenberg parameter tables consist of four variables:

-The two variables used for rotation are θ and α:

- θ is the angle from xn-1 to xn around zn-1. (n is the number of a joint.)
- α is the angle from zn-1 to zn around xn.

-The two variables used for displacement are a and d:

- a is the distance between the origin of frame n-1 and the origin of frame n along the xn direction.
- d is the distance from xn-1 to xn along the zn-1 direction.

To find those variables we have to take a look at the Datasheet of the MH5F (Appendix 1) and find its dimensions.

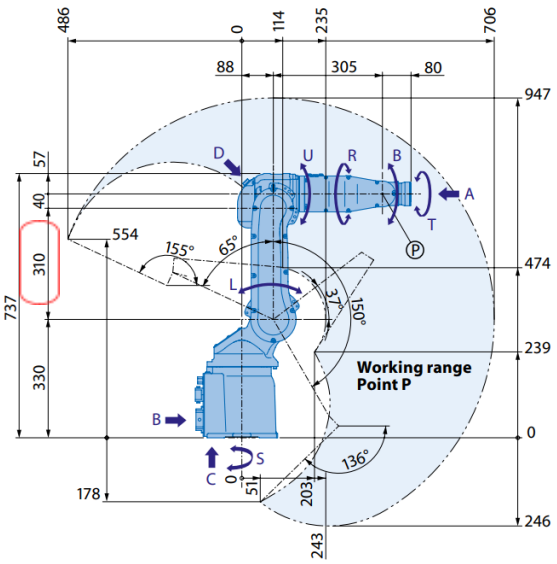


Figure 10 - Motoman MH5F measures from the datasheet

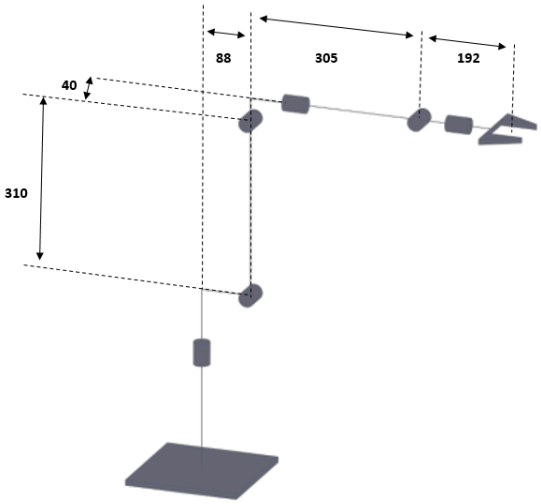


Figure 11 - Schematic model with measures

links	θ	α	a(cm)	d(cm)
1	$\theta 1$	$\pi/2$	88	0
2	$\theta 2 + \pi/2$	0	310	0
3	$\theta 3$	$-\pi/2$	40	0
4	$\theta 4$	$\pi/2$	0	-305
5	$\theta 5$	$-\pi/2$	0	0
6	$\theta 6$	π	0	-192

Figure 12 - DH parameters of the Motoman MH5F

To understand how the table was filled here's an example explained for joint number 2 for the variable a:

In Figure 9, we determine the distance between the origin of frame 1 and the origin of frame 2 along the x2 axis (the red arrow). The real distance is then retrieved from Figure 10 (in red) and finally we write it down on the table Figure 12 (in red).

- <u>Step 3</u> : Denavit-Hartenberg matrix

Once we have assigned each parameter of the DH table thanks to the geometry specifications, we have to find now the homogeneous transformation matrices (also known as the Denavit-Hartenberg matrix).

To do that we have to plug the values of the DH table into the predefined matrix down below for each one of the joints. This matrix is called the homogeneous transformation matrix for joint n:

$$\begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\ \cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\ \sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And finally, after finding the homogeneous transformation matrices for each joint (Appendix 2) , we multiply them together to get the Denavit Hartenberg representation of the robot. (Appendix 3)

Now, by successfully deriving the Denavit-Hartenberg representation for the MH5F Motoman robotic arm, I have laid a crucial foundation for further advancing my understanding of the robotic arm's motion planning, control and simulation.

<u>2-2-4 Python IKPY Library:</u>

In this part, we shifted our focus towards identifying appropriate software for simulating the MH5F Motoman robotic arm. We first decided to use the python IKPY library to compute the kinematics of our robotic arm.

- Python IKPY library :

IKPy is a fast, easy to install and use Inverse Kinematics library written in pure Python, that works on every robot. It can compute the Inverse Kinematics in position, orientation, or both and It offers the flexibility to define the kinematic chain using diverse representations such as Denavit–Hartenberg or URDF, and it facilitates the seamless importation of a kinematic chain directly from a URDF file.

```
>>> MH.forward_kinematics([0] * 7)
array([[ 0.93259108,  0.20614175, -0.29627598,  0.80962764],
       [ 0.24958096,  0.22467175,  0.94192991, -2.81465277],
       [ 0.26073592, -0.95238027,  0.15807781,  0.92504885],
       [ 0.        ,  0.        ,  0.        ,  1.        ]])
>>> import matplotlib.pyplot
>>> from mpl_toolkits.mplot3d import Axes3D
>>> ax = matplotlib.pyplot.figure().add_subplot(111, projection='3d')
>>> MH.plot(MH.inverse_kinematics([58, 300, 947]), ax)
>>> matplotlib.pyplot.show()
>>> matplotlib.pyplot.show()
```

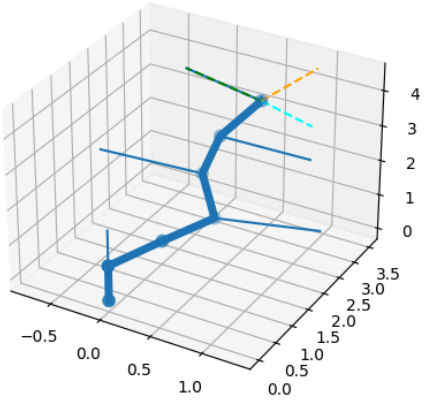


Figure 13 - The forward kinematics matrix and the representation of the robot with the IKPY library.

After installing Python and the library, writing the code and implementing the corresponding values thanks to the DH matrix; I had, as a result, the forward kinematic matrix of my robot (in red) as well as the representation of the robot in space in a randomly chosen position (without the end effector).
This outcome was helpful in enhancing the comprehension of the robot's kinematics, providing a visual and numerical understanding of its joint configurations and spatial orientation and elucidating the importance of the DH matrix.

However, after recognizing the need for a more representative simulation tool, Mr Carrilho and I decided to broaden our search for alternative software solutions. This exploration aimed to find a simulation tool that could offer a more accurate and encompassing depiction of the robotic arm's behavior, ensuring a more robust simulation environment.
That's when we encountered ROS.

2-2-5 ROS (Robot Operating System):

What is ROS ?

The Robot Operating System (ROS) is an open source platform used to create robots, both in simulation and in hardware. ROS includes a huge collection of prebuilt, modularized, customizable software packages that can be used as plugins in a robot. This allows the robot to perform virtually or in the real world various operations such as reading data from sensors, controlling the movements of actuators, mapping, localization, navigation, and more. Most of these prebuilt software packages are thoroughly tested, actively used by the robotics community all over the world, and frequently updated.

Why Use ROS?

ROS is used all over the world by researchers, industrialists, academicians, hobbyists, and so on, because it has many appealing features:

• Open-source: Free to use.

• Easy to use: There is a huge collection of prebuilt packages that provide various functionalities.

• Simulation: Provides a built-in simulator and visualizer tool. Simulation helps to get an idea of the robot's behavior in the real world. Visualization helps to view the world from the robot's perspective.

• Hardware flexibility: Can run on almost any computational device.

• Generic: Can be used to build any type of robot, not just one particular type.

• Huge community and support: Used worldwide by individuals, researchers, universities, industries, and so on. This allows users to find resources and get support from communities. Resources can be found in online courses, websites, blogs, textbooks, research papers, and so on.

Given the objectives and goals of my internship, ROS emerges as the optimal platform due to its comprehensive feature set, aligning perfectly with the requirements for a successful implementation.

Getting started with ROS:

Given the unfamiliarity with this platform, I undertook a comprehensive research initiative and engaged in free online courses, particularly on edx.org, to initiate my understanding of ROS.

The initial steps involved the installation of Ubuntu-Linux, a crucial prerequisite for leveraging the capabilities of this robotics software framework. The installation process laid the foundation for a Linux environment conducive to ROS development.

To facilitate these installations and configurations, Mr Nascimiento, a teacher at IME, provided me with a virtual machine. This virtual environment was pre-equipped with all the necessary components, including the Ubuntu-Linux operating system as well as the RAM, disk space and CPU cores needed, creating an optimal setup for ROS development.

With all the necessary tools and resources at my disposal, I proceeded to install ROS and its associated packages and started my learning journey.

How does ROS work ?

- ROS Workspace:

All of ROS commands are executed on the Command Shell (CS) via Singularity. The folder or files created from the CS are also reflected in the Linux installation in the corresponding folder.

A catkin workspace is a directory (folder) in which we can create or modify existing catkin packages. The catkin structure simplifies the build and installation process for the ROS packages.

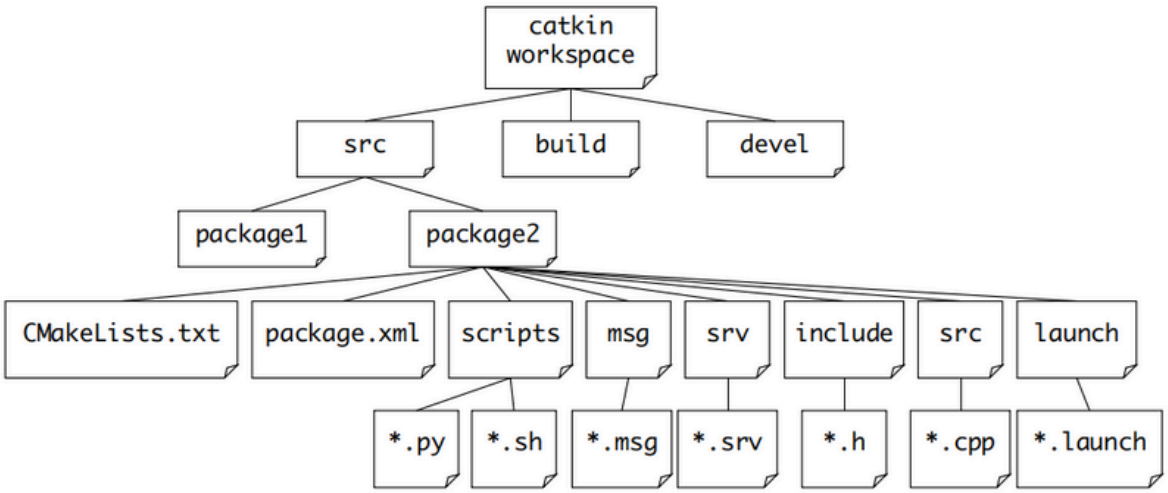


Figure 14 - The catkin workspace structure

I started by creating my catkin workspace that I named hrwros_ws . This workspace houses three files. The **src** folder contains all the packages needed to simulate my robot, the **build** folder contains generated and compiled code and generated database files and the **devel** folder is where built targets are placed prior to being installed as depicted in Figure 14.

The following commands were used for creating the workspace:

```
mkdir -p hrwros_ws/src
#verify that its been created:
ls
#go to the newly created workspace:
cd  hrwros_ws
#set up the right ROS environment:
source /opt/ros/noetic/setup.bash.
#initialize catkin:
catkin init
catkin build
```

2-2-5a. Simulation of station 2 and the robotic arm:

Now that I have created my workspace, the next step was to explore methods for visualizing the MH5F robotic arm. For this purpose, I needed an URDF file of my robot.

- URDF files:

The unified robotics description format (URDF) file is a domain specific modeling language (DSML) based on XML. It allows us to encode the kinematics, some part of the dynamics parameters and other meta-data related to the robot in a format that is readable and writable to a certain extent by both humans and machines.

ROS uses the URDF to describe the robot body layout with all of its links, joints as well as its appearance through elements like <link> (robot structure) where we describe the geometry of the link thanks to elements like <visual> and <geometry>, we also have <joints> (connections and motion constraints) where the connection between a parent link and a child link is made (code below).

Here's an example of a simple robot URDF file to understand its structure:

```
<robot name=’’example_robot”>
  <link name=”link_1” >
        <visual>
           <geometry>
                <box size="0.6 0.1 0.2"/>
            </geometry>
         </visual>
  </link>
  <link name=”link_2” >
        <visual>
          <geometry>
               <box size="0.6 0.1 0.2"/>
           </geometry>
         </visual>
 </link>
 <joint name=”joint_1” type=”revolute” >
      <parent link=”link_1” />
      <child link=”link_2” />
  </joint>
</robot>
```

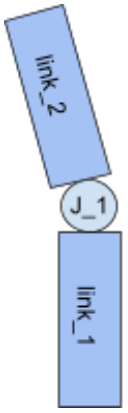


Figure 15 - Example of a simple Robot

As we can see this simple robot took only some lines given its single joint and uncomplicated geometry. However, generating the URDF for the Motoman MH5F robotic arm proves to be a complex and possibly challenging task. This is due to its eight joints, highly precise geometry, and intricate kinematics parameters.

Consequently, I sought an alternative method to create the URDF file for my robot.

- Solidworks to URDF exporter:

The SolidWorks to URDF exporter is a SolidWorks add-in that allows for the convenient export of SW Parts and Assemblies into an URDF file. The exporter will create a ROS-like package that contains a directory for meshes, textures and URDF files.

Upon downloading the add-in, I meticulously assembled all the parts of the robotic arm, ensuring adherence to the movement and placement of each joint. Additionally, I incorporated limit values. As a result, I successfully extracted a ROS package containing the config, launch, meshes, texture, URDF, CMakeLists, export and package files.

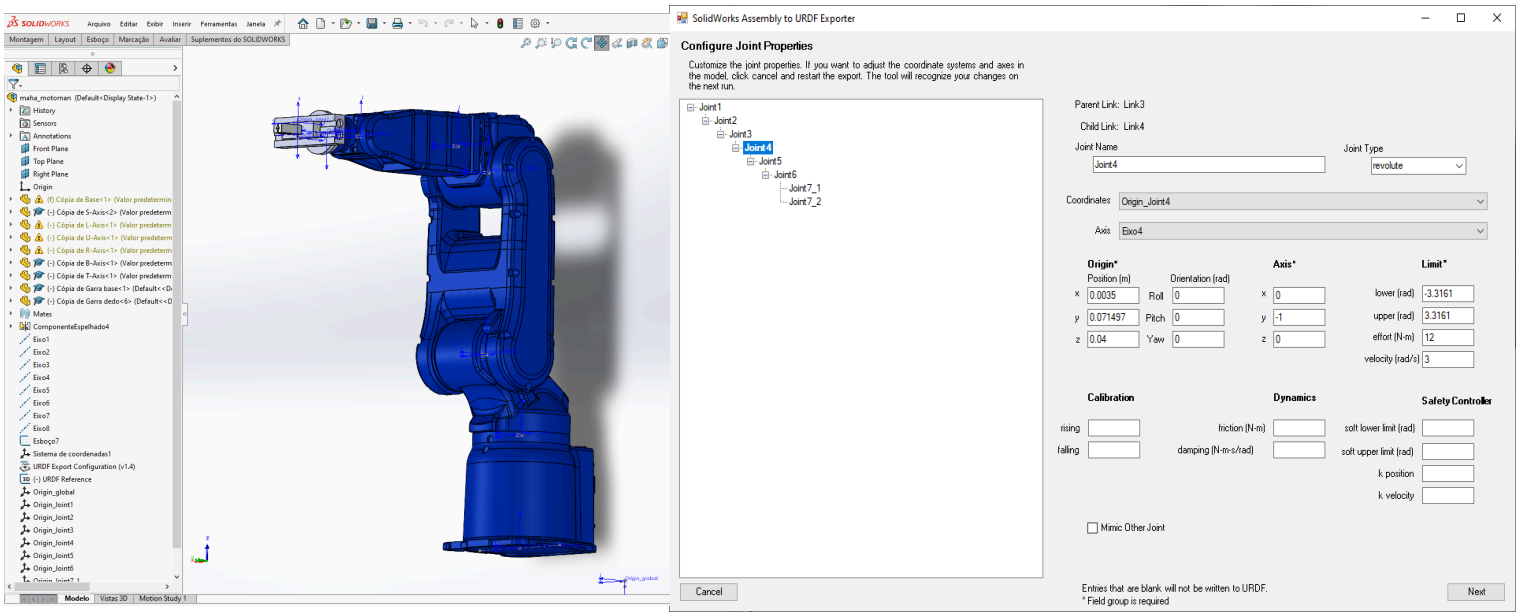


Figure 16 - Solidworks Assembly of the Motoman MH5F and the URDF add-in

The package folder generated with Solidworks was then copied in the ubuntu machine in my hrwros_ws catkin workspace.

However, the default generated package is just a robot description. Without modifying it, it can't be simulated in Gazebo (which is a set of ROS packages and an open-source 3D robotics simulator that simulates real-world physics in a high fidelity simulation). Therefore, I edited the CMakelists.txt as well as the package.xml file to add all the dependencies - external libraries and tools that must be provided by the operating system - needed. Then I modified the URDF file to make it suitable by adding the transmission tags - tags that describe the relationship between an actuator and a joint - as well as the gazebo plugin as it is necessary to be able to visualize it with Gazebo. Finally, I wrote a launch file (arm_launch.launch) that enabled me to start all the required ROS commands together in one terminal.

After writing the following commands on the first CS to start the ROS master:

```
cd hrwros_ws
source devel/setup.bash
catkin build
source devel/setup.bash
roscore
```

and on the second CS to launch the Gazebo file:

```
cd hrwros_ws
source devel/setup.bash
roslaunch urdf_maha arm_launch.launch
```

I had this output:

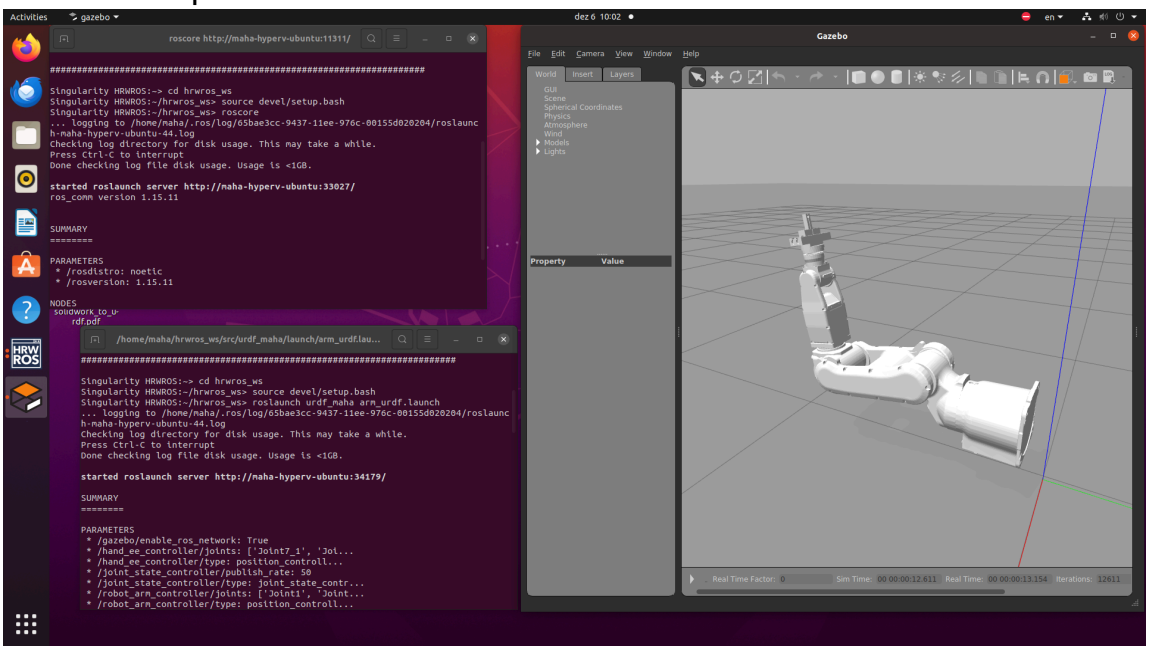


Figure 17 - ROS code and simulation of the robotic arm in Gazebo

As a result I had the 3D model in Gazebo of my robotic arm with all its joints and its links as it is in real life. However the robot is not stable and not oriented in the correct position.

Therefore, I created a new Solidworks assembly of the robot to correct the mistakes and to add the environment surrounding it.

➔ **The station 2 environment:**

The station 2 is composed of:

1. PC, its keyboard and its mouse.
2. Table
3. Rack
4. Linear slide base
5. Two buffers
6. Robots Rail
7. Two Gravitational feeders
8. CONCEPT Turn 250 CNC Lathe
9. Motoman MH5F Robotic arm

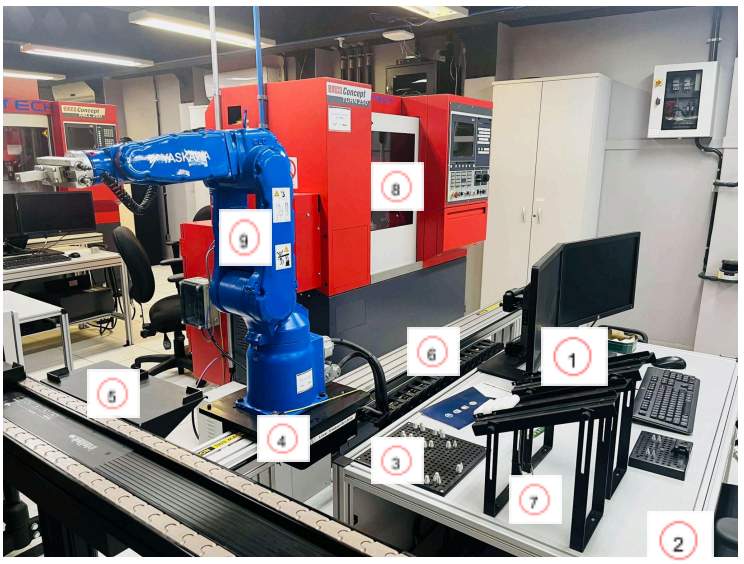


Figure 18 - The station 2 equipments

Using the same approach that was used with the robotic arm, I assembled the various components of the station in SolidWorks, ensuring that each component was properly fixed.

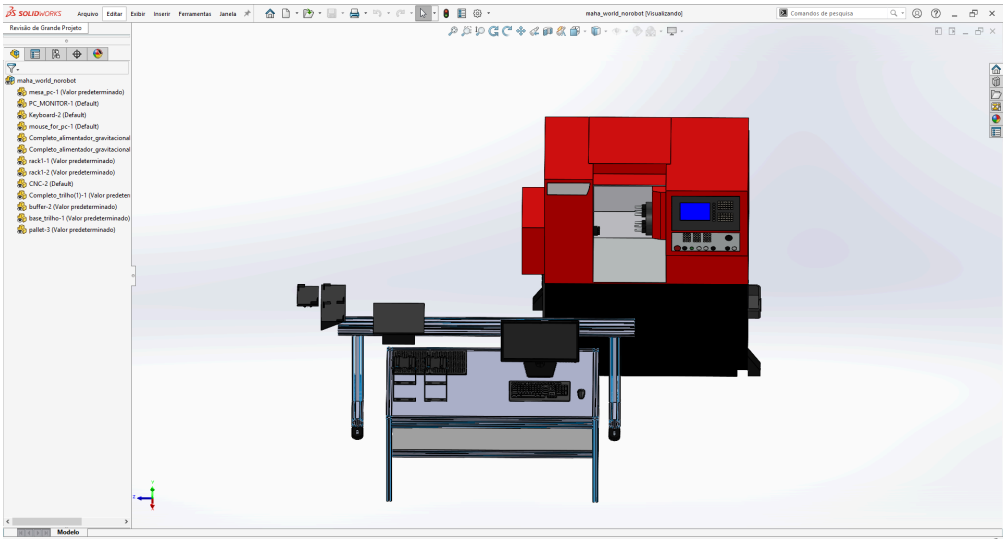


Figure 19 - SolidWorks Assembly of station 2

Subsequently, I incorporated the robot onto the station and generated the URDF file. Once the URDF was prepared and all the packages were rectified, I executed the command to visualize the station in Gazebo.

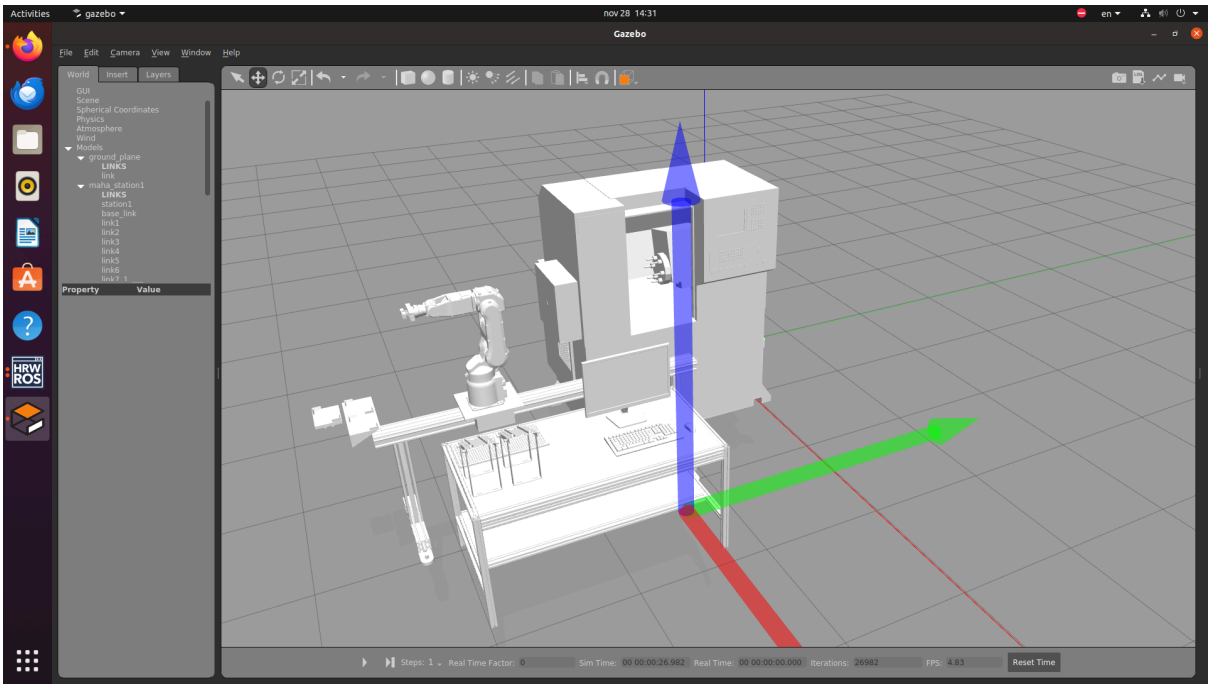


Figure 20 - The station 2 simulation in Gazebo

2-2-5b. Implementation of the process Turn demo st2:

Station 2 is used to perform a specified process on raw material parts called turn-demo-st2. The Motoman MH5F Robotic arm plays a crucial role in automating the workflow. The process device driver runs on a PC that controls the process station via the Vuniq software. This device driver receives command messages from the CIM Manager (station 9) to process the raw material part. The process device driver translates these requests into commands for the manipulator and process control to perform the following tasks:

1. **Retrieve Pallet from Conveyor**: The robot begins by picking up a pallet with its corresponding piece from the conveyor system.
2. **Place Pallet in Buffer**: The robot then places the pallet with the piece into one of the buffers, providing temporary storage.
3. **Retrieve Piece from Buffer**: The robot retrieves the piece without the pallet from the buffer.
4. **Place Piece on CNC Lathe**: The robot carefully positions the component onto the CONCEPT Turn 250 CNC Lathe.
5. **Close CNC Lathe Door**: The door of the CNC Lathe is closed, initiating the machining cycle.
6. **Wait for Cycle Completion**: The robot patiently waits for the CNC Lathe cycle to finish, ensuring precision in the manufacturing process.
7. **Retrieve Piece from CNC Lathe**: Once the CNC Lathe cycle is complete, the robot retrieves the machined piece.
8. **Place Piece on the table**: Finally, the robot positions the completed piece onto the table.

Each of these operations requires the robot to assume specific spatial positions, necessitating precise angle rotations for each of its joints. These predetermined positions and corresponding angle rotations are programmed into the robot's pendant padel in advance. The station benefits from a prior automation process, wherein the utilized software generates a comprehensive board. This board outlines each action and the associated positions required by the robot to seamlessly execute the specified tasks.

The robot navigates through positions 1 and 2 to reach position 3, and subsequently retraces its steps from position 3 back to 2 and then 1. Occasionally, when applicable, it transitions to position 4 before returning. The positions B1 and B2 correspond to the locations of the robot's linear slide

base, providing additional context for its movement and orientation within the specified operational sequence.

Operation	Position 1	Position 2	Position 3	Position 4	Position B1	Position B2
Get from conveyor	1	11	11		501	
Put to buffer	2	12	12		502	
Get from buffer	22	32	999		502	
Place on lathe	432	442	401	400	507	502
wait for cycle to finish						
Get from lathe	412	422	401	400	507	502
Put in buffer	122	132	999		502	
Get from buffer	2	12	12		502	
Put on conveyor	1	11	11		501	

To incorporate these positions into my simulation accurately, I needed to determine the joint angles in radians for each position. The pendant panel conveniently records the registration of each position along with the corresponding encoder number for each joint.

Consequently, I compiled this information into an Excel spreadsheet, detailing the various positions and their associated encoder numbers for the 6 joints (SLURBT). Subsequently, I converted these values from encoder related numbers to degrees and to radians, ensuring precision in the simulation.

	A	B	C	D	E
1	P1	S	-24636	-22,0222214	-0,38416542
2		L	48802	43,6243079	0,76100182
3		U	-852	-0,76160629	-0,0132858
4		R	-778	-0,69545739	-0,01213187
5		B	-43712	-39,074336	-0,68163008
6		T	-88399	-79,0202285	-1,37846399
7	P11	S	-24636	-22,0222214	-0,38416542
8		L	37794	33,7842116	0,5893468
9		U	8031	7,17894383	0,12523269
10		R	-639	-0,57120472	-0,00996435
11		B	-59962	-53,6002776	-0,93502707
12		T	-88562	-79,1659349	-1,38100575
13	P2	S	-36591	-32,7088449	-0,57058763
14		L	27596	24,6681775	0,43032265
15		U	-35846	-32,0428864	-0,55897035
16		R	764	0,68294273	0,01191356
17		B	-29974	-26,7938815	-0,46740438
18		T	-82911	-74,1144828	-1,29288598

Figure 21 - The positions that the end effector takes during the process on the Programming Pendant screen.

Figure 22 - The positions converted in degrees and radions.

After successfully implementing a virtual twin of Station 2, I had to enable dynamic movements. To achieve this, I utilized MoveIt and RVIZ, two powerful motion planning frameworks for robotics. MoveIt is an open-source software that facilitates path planning and motion control for robotic systems. It enables the virtual twin to simulate and execute precise movements, ensuring optimal coordination and efficiency in the station's operations.

Based on the previous position token from excel, I managed to put the exact positions on the simulation by rotating the robots joints:

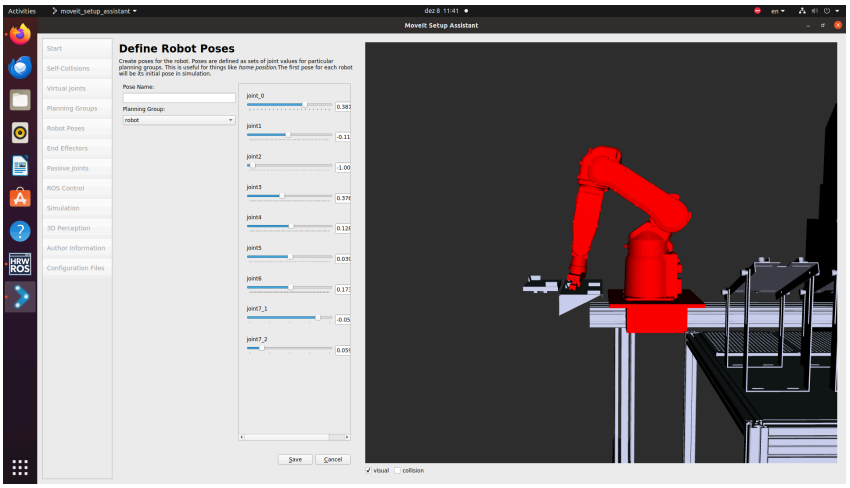


Figure 23 - Joints adjustment in MoveIt for the robot to reach position 22

Once I have created all the positions of the robot on MoveIt, I was able to implement a moving simulation of the whole process turn-demo-st2 on RVIZ:

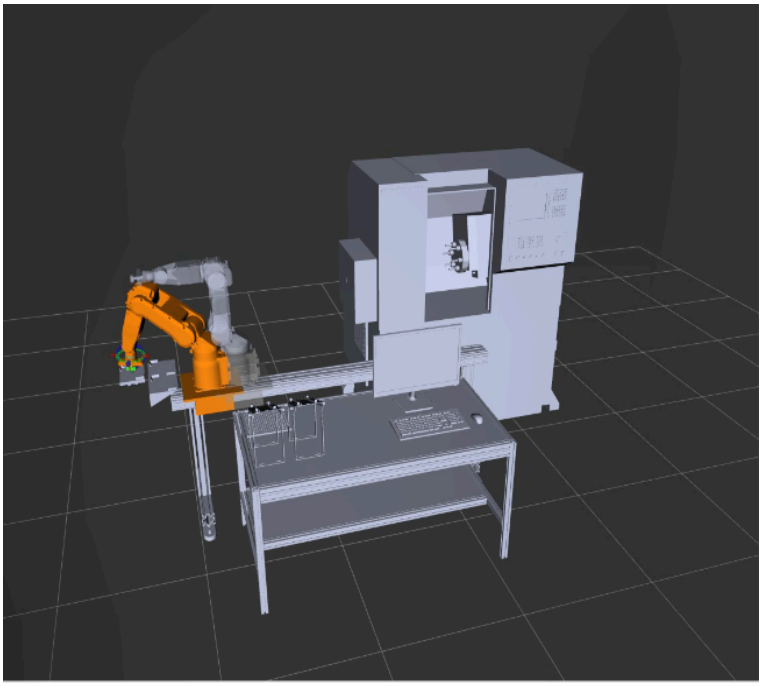


Figure 24 - Retrieve Pallet from Conveyor

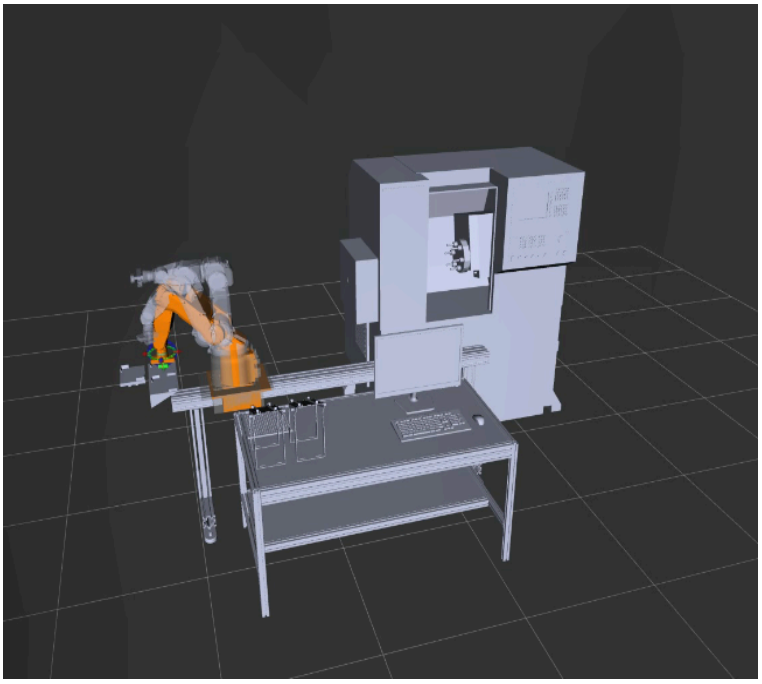


Figure 25 - Place Pallet in Buffer

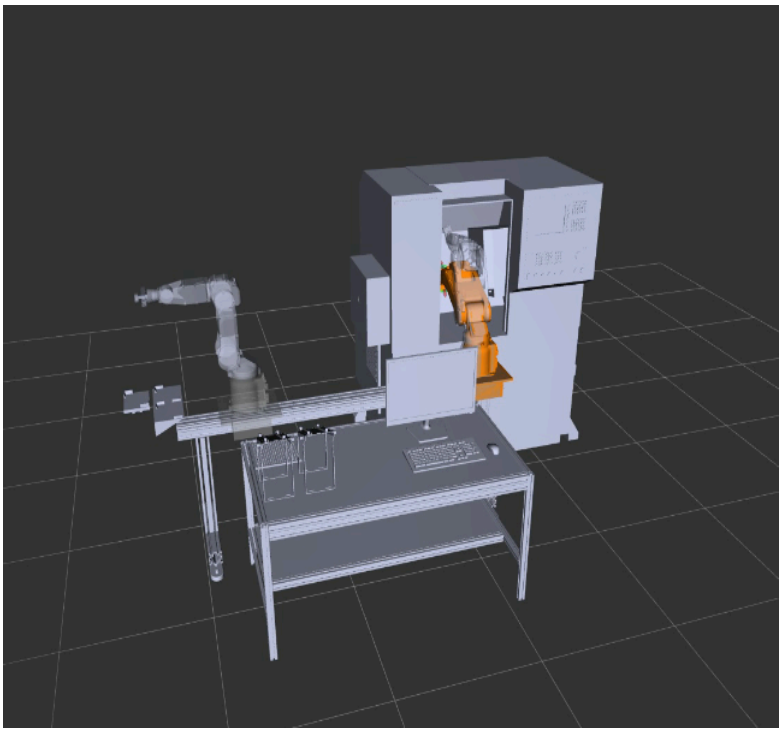


Figure 26 - Place Piece on CNC Lathe

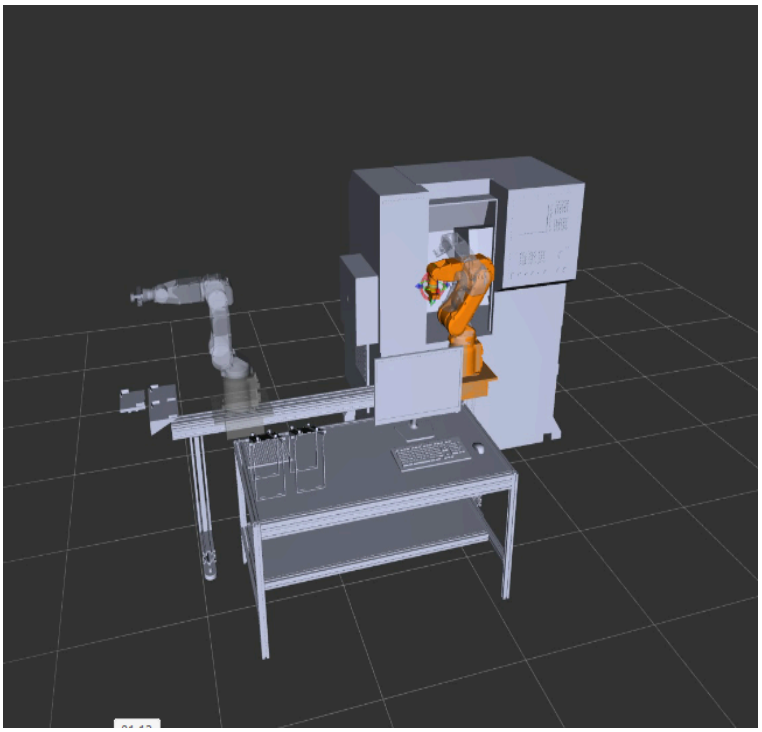
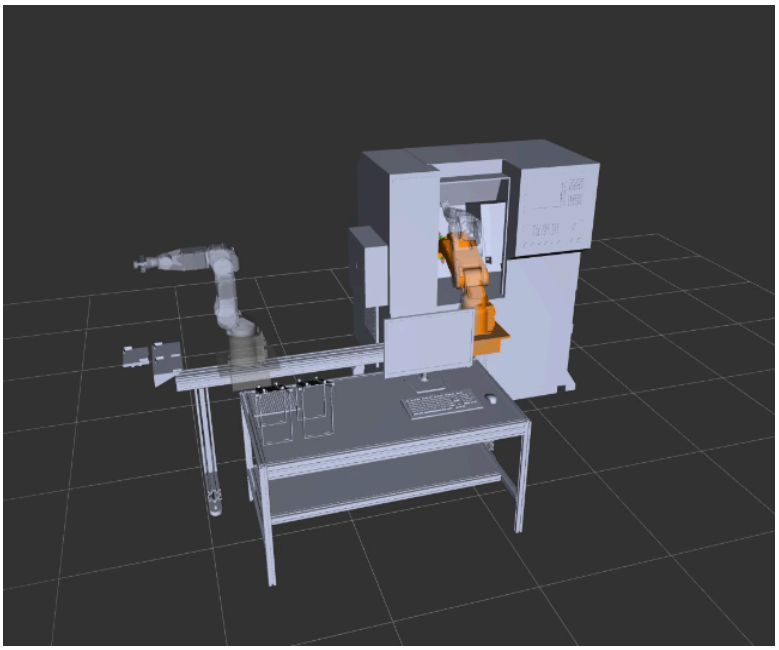


Figure 27 - Wait for Cycle Completion



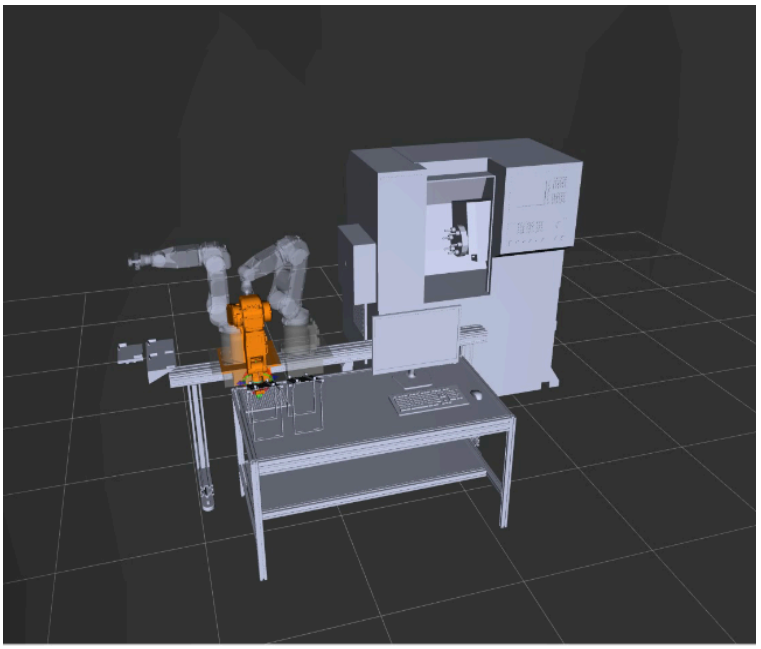


Figure 28 - Retrieve Piece from CNC Lathe

Figure 29 - Place Piece on table

2-3. Results and Interpretations:

The simulation of Station 2, with its robotic arm and associated equipment was a success. The choice of ROS as the software for this simulation was deliberate, considering its widespread adoption and reputation for reliability in the field of robotics. The simulation, executed with precision, progressed to a deeper exploration of the robotic arm's kinematics.

The integration of RVIZ (Figures 24-25-26-27-28-29) into the simulation framework was particularly illuminating. This tool not only provided a visual representation of the exact positions the robotic arm would take during its movements towards the final placement to perform the given commands but also generated insightful trajectories, illustrating how the robot would navigate to these positions while avoiding potential collisions with its environment. Which was very helpful to planning and optimizing the robotic arms kinematics.

Furthermore, MoveIt, a pivotal component of this simulation framework, proved to be an invaluable asset as well in ensuring both safety and operational efficiency. In fact, The software was able to anticipate collisions when I adjusted joints positions.

To test it, I changed the robots joints positions and made it touch, on purpose, the CNC Lathe after the robot placed the piece inside it (Figure 30) MoveIt was able to warn me that there was a collision, which is useful insight when adjusting the robots joints preventing it from entering collision-prone zones, mitigating potential security and operational risks.

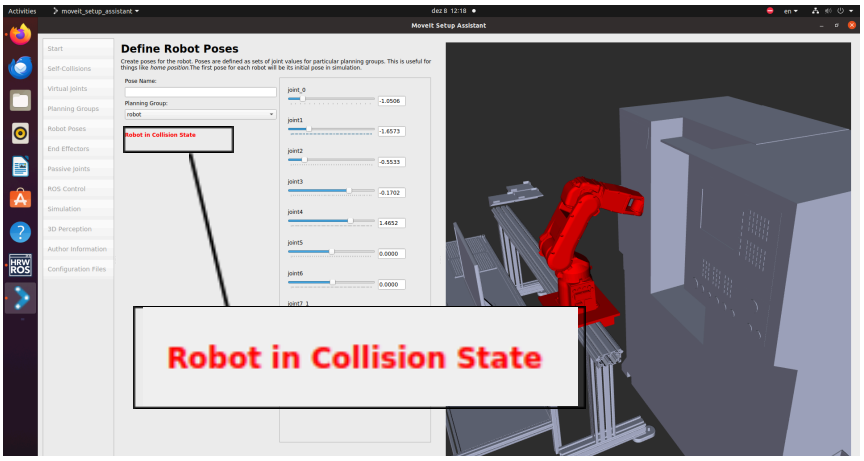


Figure 30 - Moveit detecting that the robot is in collision state

Furthermore, MoveIt's capability to detect self-collisions, as exemplified in Figure 31, contributed significantly to the overall risk assessment. This functionality not only serves as a preventive measure but also enhances the understanding of the robotic arm's operational constraints.

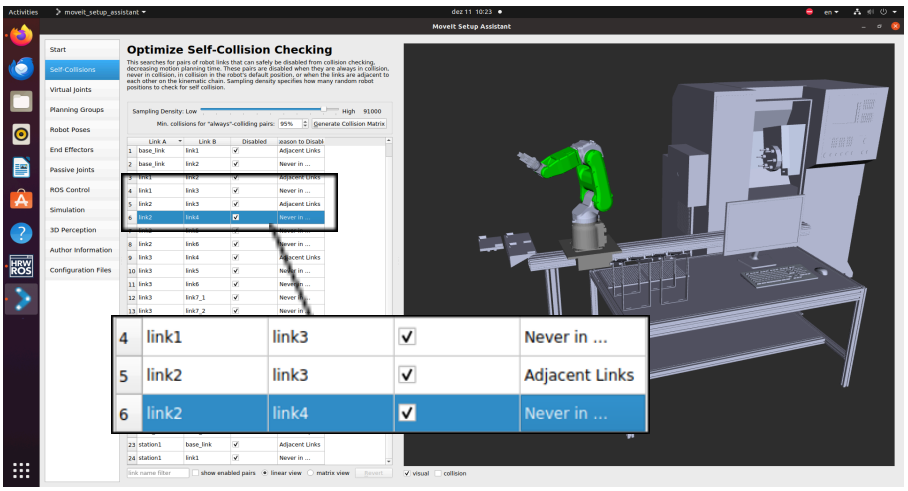


Figure 30 - Moveit checking self-collisions between the robot's links

Chapter 3

Conclusion and Perspectives

3-1. Challenges encountered:

During my internship, I encountered several challenges that significantly impacted the workflow and progress of my tasks. One major hurdle involved errors in the ROS (Robot Operating System) code, necessitating extensive research and exploration of forums and online resources to identify and implement effective solutions.

Additionally, the initial phase of the internship posed difficulties as our teacher was absent due to COVID-19, delaying the clarification of tasks and objectives.

Technical issues further compounded our challenges, including intermittent problems with the wifi connection, virtual machine glitches, and disturbances caused by ongoing construction near the lab. Another critical setback was the malfunction of the CIM (Computer Integrated Manufacturing) system, preventing us from observing the processing of raw materials and understanding the underlying mechanisms due to the presence of faulty batteries.

These combined issues demanded adaptability, resourcefulness and collaboration to overcome, ultimately contributing to a valuable learning experience, despite the setbacks.

3-2. Human Factor:

This internship has been made possible through the unwavering support and collaboration of numerous individuals at IME, who played instrumental roles in ensuring that I had all the necessary resources to fulfill my objectives and succeed.

Maintaining open communication with my professor, Mr. Carrilho, was a fundamental aspect of my organizational strategy. We established a weekly meeting schedule to discuss progress, address concerns, share intermediate results, and provide updates on the ongoing tasks. These regular check-ins

allowed for real-time feedback, ensuring that any potential issues were identified and addressed.
He not only provided guidance when needed but also facilitated access to all essential resources. In fact, Mr. Carrilho created a Google Drive repository where he organized and shared documents, reports, and information pertaining to the lab. He also introduced me to his colleagues when needed. Additionally, he suggested the use of ROS, a pivotal recommendation that significantly contributed to the success of my project.

Mr. Pellanda, my tutor, demonstrated, as well, consistent support by regularly visiting the lab to ensure everything was proceeding smoothly.

Mr Carrilho tasked me in the beginning of the internship with giving a presentation on forward and inverse kinematics, as well as a demo of the Denavit-Hartenberg representation associated with the Motoman MH5F robotic arm. I was able to present these fundamental concepts to Mr Pellanda, Mr Carrilho and other people present in the lab.

Another key figure, Mr. Paz, who played a vital role in orienting us (Me and my two other INSA colleagues present in the laboratory) during our internship. He was the one who welcomed us and gave us a tour at IME, introducing us to various classes, rooms, and fellow colleagues. Also, his meticulous attention to our comfort, material needs, and access to resources was greatly appreciated.

The presence of Mr. Azevedo, a dedicated teacher at IME, was invaluable. He consistently ensured our well-being, both within and outside the laboratory, offering guidance in Brazil.

Additionally, Mr. Nascimento, a professor of defense engineering and electrical engineering, played a crucial role in creating the virtual machine and providing support whenever I encountered challenges executing code or faced issues with the virtual machine.

Moreover I had the privilege of attending a course conducted by Mr. Carrilho and collaborated with other IME students in group projects. Their insights and involvement in student activities within IME further enriched my internship experience.

Lastly, Mr. Braz's assistance was pivotal, providing access to the classroom with SolidWorks and offering guidance whenever complications arose with the software.

The collective effort of these individuals significantly contributed to the success and enriching experience of my internship at IME.

3-3. Added value for the MecatrIME Laboratory study case:

In conclusion, I've successfully crafted a proof of concept of station 2 within the CIM. This involved implementing the kinematics of the robotic arm and conducting a thorough study on potential collision issues while anticipating movements and interactions between the various components of station 2.

Furthermore, I've found the Denavit-Hartenberg (DH) representation of the robotic arm; a valuable find that can facilitate the implementation of kinematics for the robotic arms in other stations within the CIM.

To enhance accessibility, I've compiled the joint rotation data for each position of the robotic arm (for the turn-st2-process) in an Excel sheet, aiding others (who will be taking on the digital twin project) in easily determining joint angles in both radians and degrees. This resource is expected to streamline the implementation of kinematics in simulations for other stations.

The collaborative essence of the project shines through, as my work will be passed to fellow students and interns who will continue and expand upon it. This succession plan not only ensures the project's ongoing success but also nurtures a dynamic learning environment, enabling multiple contributors to give their insights and advancements.

In totality, this simulation project significantly elevates the value of the lab's pursuit toward creating a digital twin, promising extensive implications for the realms of robotics, cybersecurity, and manufacturing.

3-4. Tasks and their advancements:

Documentation and research	
Tasks	**Percentage**
Read the documents provided in the Google Drive to understand how the CIM works	80%
Read Mr. Macedo's thesis on the Motoman MH5F, which he completed during his internship at the MecatrIME lab.	90%
Research on forward and inverse kinematics, particularly focusing on the Denavit-Hartenberg (DH) notation.	90%
Present and explain those kinematics concepts to my tutor and Mr. Carilho.	100%
Mathematically derive the DH representation of the Motoman MH5F and elucidate it using LaTeX.	100%
Search for software to simulate the robotic arm and the workstation.	80%
Python IKPY library	
Tasks	**Percentage**
Install python, pip and the IKPY library.	100%
Understand how the library works and explore all its features.	70%
Simulate the robotic arm by implementing the DH parameters in the code.	90%
ROS (Robotic Operating System)	
Tasks	**Percentage**
Take the ROS course on edx.com to understand how it works.	80%
Install ROS and all the packages needed.	100%
Understand how Gazebo and MoveIt work, and how they integrate with ROS.	85%
Create a URDF file of the MH5F using Solidworks.	100%
Create a URDF file of the station 2.	100%
Simulate Station 2 and the robot in Gazebo.	100%
Simulate the process "turn-demo-st2" using MoveIt and RViz.	100%
Move the robotic arm in Gazebo using the keyboard.	30%
Give a presentation of your final work and the results to the colleagues present in the lab.	100%

3-5. Conclusion:

To conclude I would say that this internship was a very enriching experience. It gave me an insight into the professional world. As this was my first extended experience in the workplace, I was able to discover the notions of time management, problem-solving, adaptability, research skills and some facets of teamwork.

I learned various aspects related to robotic applications, delving deeply into the intricate realms of engineering and mechatronics. This hands-on experience not only expanded my knowledge base but also provided me with an understanding of the connection between theoretical concepts and their real-world applications.

It was a great opportunity to be able to use programs such as ROS, Gazebo, Moveit and RVIZ and learn how to master them.

I was able to discover different robotic applications and ongoing research across different laboratories of the Instituto Militar de engenharia , alongside engaging with engineers, teachers and experienced professionals. This exposure significantly contributed to the refinement of my career objectives.

Moreover, I was immersed in a laboratory equipped with cutting-edge Industry 4.0 tools and equipment for the first time, which gave me an insight into the workings of a well-structured and industrially-focused manufacturing platform. In fact, I was able to learn how to turn on the different equipment of the CIM and understand the use of the paddle to move the robotic arm.

After more than four months of hands-on experience, I am eager to build on this experience to further cultivate my skills and my knowledge of the robotics domain.

Additionally, participating in an ongoing project at one of Brazil's most prestigious engineering schools, collaborating with inspiring and skilled individuals, and immersing myself in the vibrant cultural and linguistic diversity of the country have collectively constituted an extraordinary human experience for me. This unique opportunity not only deeply moved me but has also played a pivotal role in my personal and professional growth.

Finally I'd like to thank you for the attention you've given to reading my internship report, and I hope it has captivated you.

References

1. https://www.ros.org/blog/why-ros/
2. https://github.com/Phylliade/ikpy
3. Hello (Real) World with ROS – Robot Operating System | My Mooc (my-mooc.com)
4. https://sites.google.com/ime.eb.br/mecatrime/produ%C3%A7%C3%A3o
5. https://inscricoes.ime.eb.br/cfg/
6. https://www.roboplan.pt/upload/flyer-robot-mh5sii-f-mh5lsii-f-e-04_56c5c05766526.pdf
7. https://automaticaddison.com/how-to-find-denavit-hartenberg-parameter-tables/
8. M. Felipe Alvares de Azevedo Macedo “Robô motoman mh5f: modelagem, simulação e programação auxiliadas por computador”

Figures references:

- *Figure 1 / Figure 5 / Figure 9 / Figure 10 / Figure 11:*

M. Felipe Alvares de Azevedo Macedo “Robô motoman mh5f: modelagem, simulação e programação auxiliadas por computador”

- *Figure 2 / Figure 4:*

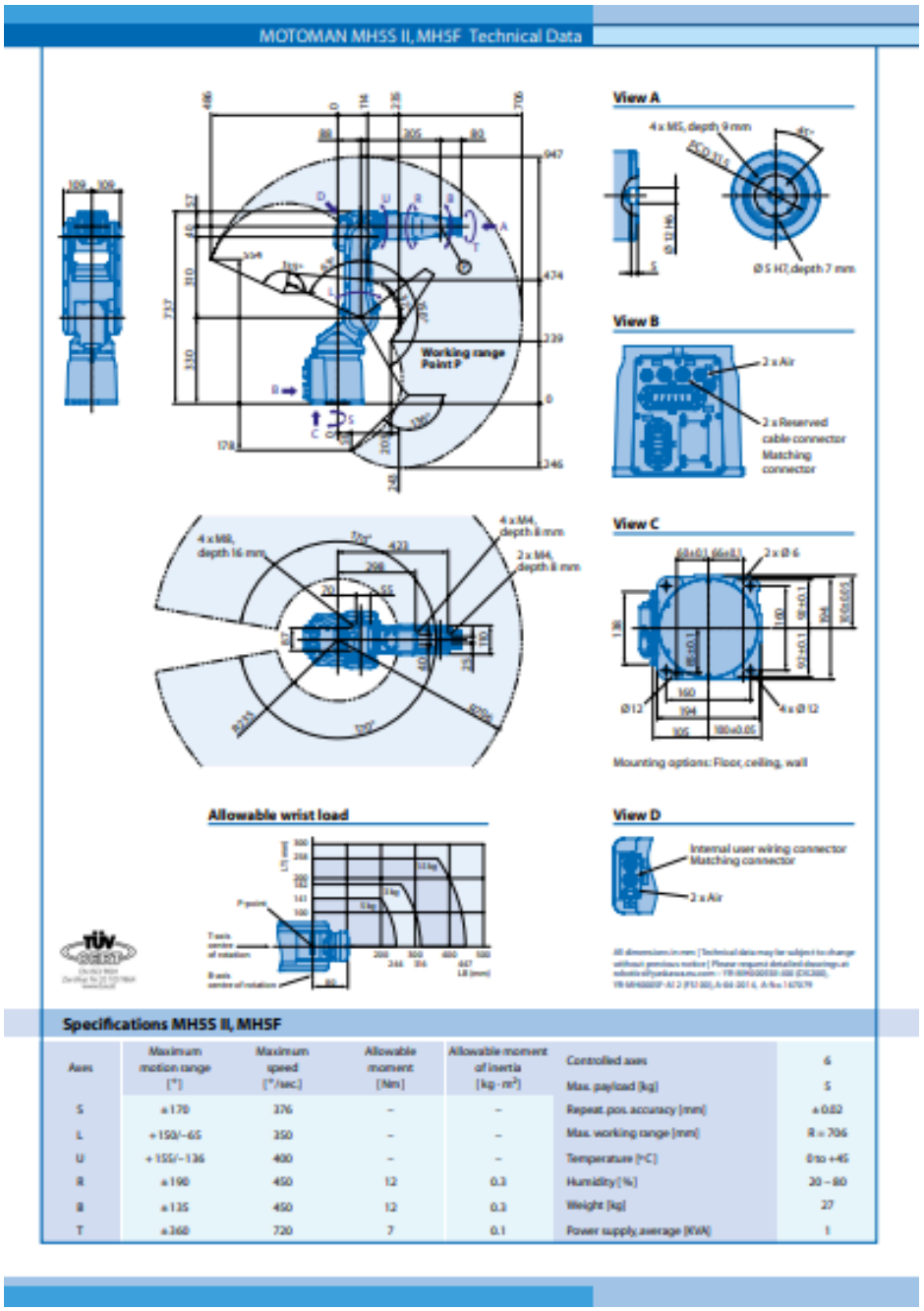
https://sites.google.com/ime.eb.br/mecatrime/produ%C3%A7%C3%A3o

- *Figure 14:*

https://medium.com/swlh/7-simple-steps-to-create-and-build-our-first-ros-package-7e3080d36faa

Appendices

Appendix 1 : Motoman MH5F Datasheet



MOTOMAN MH5S II, MH5F Technical Data

Specifications MH5S II, MH5F

Axes	Maximum motion range [°]	Maximum speed [°/sec.]	Allowable moment [Nm]	Allowable moment of inertia [kg · m²]
S	±170	376	–	–
L	+150/–65	350	–	–
U	+155/–136	400	–	–
R	±190	450	12	0.3
B	±135	450	12	0.3
T	±360	720	7	0.1

Controlled axes	6
Max. payload [kg]	5
Repeat. pos. accuracy [mm]	±0.02
Max. working range [mm]	R = 706
Temperature [°C]	0 to +45
Humidity [%]	20 – 80
Weight [kg]	27
Power supply, average [KVA]	1

Appendix 2 : DH matrices for each joint

- DH matrix for Joint 1:

$$\begin{bmatrix} cos(\theta 1) & -sin(\theta 1).cos(\pi/2) & sin(\theta 1).sin(\pi/2) & 88.cos(\theta 1) \\ sin(\theta 1) & cos(\theta 1).cos(\pi/2) & -cos(\theta 1).sin(\pi/2) & 88.sin(\theta 1) \\ 0 & sin(\pi/2) & cos(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c1 & 0 & s1 & 88.c1 \\ s1 & 0 & -c1 & 88.s1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- DH matrix for Joint 2:

$$\begin{bmatrix} -sin(\theta 2) & -cos(\theta 2).cos(0) & cos(\theta 2).sin(0) & -310.sin(\theta 2) \\ cos(\theta 2) & -sin(\theta 2).cos(0) & sin(\theta 2).sin(0) & 310.cos(\theta 2) \\ 0 & sin(0) & cos(0) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} -s2 & -c2 & 0 & -310.s2 \\ c2 & -s2 & 0 & 310.c2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- DH matrix for Joint 3:

$$\begin{bmatrix} cos(\theta 3) & -sin(\theta 3).cos(-\pi/2) & sin(\theta 3).sin(-\pi/2) & 40.cos(\theta 3) \\ sin(\theta 3) & cos(\theta 3).cos(-\pi/2) & -cos(\theta 3).sin(-\pi/2) & 40.sin(\theta 3) \\ 0 & sin(-\pi/2) & cos(-\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c3 & 0 & -s3 & 40.c3 \\ s3 & 0 & c3 & 40.s3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- DH matrix for Joint 4:

$$\begin{bmatrix} cos(\theta 4) & -sin(\theta 4).cos(\pi/2) & sin(\theta 4).sin(\pi/2) & 0.cos(\theta 4) \\ sin(\theta 4) & cos(\theta 4).cos(\pi/2) & -cos(\theta 4).sin(\pi/2) & 0.sin(\theta 4) \\ 0 & sin(\pi/2) & cos(\pi/2) & -305 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c4 & 0 & s4 & 0 \\ s4 & 0 & -c4 & 0 \\ 0 & 1 & 0 & -305 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- DH matrix for Joint 5

$$\begin{bmatrix} cos(\theta 5) & -sin(\theta 5).cos(-\pi/2) & sin(\theta 5).sin(-\pi/2) & 0.cos(\theta 5) \\ sin(\theta 5) & cos(\theta 5).cos(-\pi/2) & -cos(\theta 5).sin(-\pi/2) & 0.sin(\theta 5) \\ 0 & sin(-\pi/2) & cos(-\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c5 & 0 & -s5 & 0 \\ s5 & 0 & c5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- DH matrix for Joint 6:

$$\begin{bmatrix} cos(\theta 6) & -sin(\theta 6).cos(\pi) & sin(\theta 6).sin(\pi) & 0.cos(\theta 6) \\ sin(\theta 6) & cos(\theta 6).cos(\pi) & -cos(\theta 6).sin(\pi) & 0.sin(\theta 6) \\ 0 & sin(\pi) & cos(\pi) & -192 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} c6 & s6 & 0 & 0 \\ s6 & -c6 & 0 & 0 \\ 0 & 0 & -1 & -192 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix 3 : DH matrix of motoman MH5F robotic arm

(−c1.c3.c4−c1.c3.s2.c4−s1.s4).c5.c6+ (c1.s3.s2−c1.c2.c3).s5.c6− (s1.c4−c1.c2.s4.s3−c1.c3.s4.s2).s6	(s1.c4−c1.c2.s4.s3−c1.c3.s4.s2).c6+ (−c1.c2.c3.c4−c1.c3.s2.c4−s1.s4).c5.s6+ (c1.s3.s2−c1.c2.c3).s5.s6	(−c1.c2.c3.c4−c1.c3.s2.c4−s1.s4).s5− (c1.s3.s2−c1.c2.c3).c5	305.c2.c3.c1−40c2.s3.c1−40.c3.s2.c1−305.s3.s2.c1−310.s2.c1+88c1−192(c1.s3.s2−c1.c2.c3)c5+192(−c1.c2.c3.c4−c1.c3.s2.c4−s1.s4).s5
(−s1.c2.c3.c4−s1.c3.s2.c4+c1.s4).c5.c6+ (s1.s3.s2−s1.c2.c3).c5.c6− (−c1.c4−s1.c2.s4.s3−s1.c3.s4.s2).s6	(c1.c4−s1.c2.s4.s3−s1.c3.s4.s2).c6+ (−s1.c2.c3.c4−s1.c3.s2.c4−c1.s4).c5.s6+ (s1.s3.s2−s1.c2.c3).s5.s6	(−s1.c2.c3.c4−s1.c3.s2.c4−c1.s4).s5− (s1.s3.s2−s1.c2.c3).c5	305.c2.c3.s1−40c2.s3.s1−40.c3.s2.s1−305.s3.s2.s1−310.s2.s1+88s1−192(s1.s3.s2−s1.c2.c3)c5+192(−s1.c2.c3.c4−s1.c3.s2.c4−c1.s4).s5
(c2.c3.c4−c3.c4.s2).c5.c6+ (−c2.s3−c3.s2).s5.c6− (c2.c3.s4−s4.s3.s2).s6	(c2.c3.s4−s4.s3.s2).c6+ (c2.c3.c4−c3.c4.s2).c5.s6+ (−c2.s3−c3.s2).s5.s6	(c2.c3.c4−c3.c4.s2).s5−(−c2.s3−c3.s2).c5	40.c3.c2+305.s3.c2−310.c2+305.c3.s2−40.s3.s2−192(−c2.s3−c3.s2).c5+192(c2.c3.c4−c3.c4.s2).s5
0	0	0	1