Understanding the Robotic Manipulator RV-2AJ for future viability study of integration into a Legacy FMS Digital Twin application

Leonardo Henrique Camilo Patrão and Eduardo Paciência Godoy
Scientific Inititiation Program PIBIC Junior CNPQ/Reitoria UNESP
Universidade Estadual de São Paulo (UNESP) – Campus Sorocaba
Sorocaba, São Paulo, Brasil
Leonardopatrão80@gmail.com ; eduardo.godoy@unesp.br

Vitor Mendes Caldana
Professor, Electronics Department
Instituto Federal de São Paulo (IFSP) – Campus Sorocaba
Sorocaba, São Paulo, Brasil
vitor.caldana@ifsp.edu.br

Abstract
Industrial Robotic Arms and Manipulators are widely used in assembly lines and automation, making them quicker, precise and efficient. They execute highly repetitive and high-effort tasks, improving work security and reducing accidents. The MELFA (Mitsubishi Electric Factory Automation) offers advance technology in this scenario. This viability study will be done on model RV-2AJ, with focus on its main component, such as driver CR1-571 and the handheld device, as well as the MELFA BASIC IV programming language and its practical application to determine if it possible to integrate the robot to a Digital Twin scenario of a Flexible Manufacturing Line. A comprehensive study of the robot, its moving procedures and components is presented as well as the programming and files generated to reach a conclusion that the integration of the robot in a Digital Twin environment is, at this point, possible. Future study in Feedback signals is necessary.

Keywords
Industrial Robots, MELFA, RV-2AJ, CR1-571 Controller, MELFA BASIC IV Programming

1. Introduction
Before mentioning the concept of an industrial robot arm it is important to understand that the robot is not just simply a machine. It is capable of accomplishing tasks and modify them as necessary, becoming an ideal option for humans and for the industry. The term robot comes from the Czech word "robota", that means servant or forced worker, and is referred to electromechanical devices capable of executing work in an autonomous or pre-programmed way. Robots are often used in environments with little light or in tasks that are repetitive or dangerous for humans. According to the Robotics Institute[1] "a robot is a reprogrammable multifunctional manipulator, designed to move materials, parts, tools or special devices by means of programmed motions and variables, allowing the accomplishment of diverse tasks. The International Federation of Robotics (IFR)[2] differentiates industrial robots from the other types, defining them as automatic, reprogrammable and multifunctional machines with three or more axles, capable of position and orient material, tools and special devices, in both fixed and moving positions (Guadalupe and Adriana, 2009).

Robots are capable of performing their tasks through actuators (electrical, pneumatic, sound, etc) allowing actions that can produce sound, activation of displays or luminous elements, movement of the arm, movement of robotic claws and even the displacement of the robot itself. They are controlled by algorithms that connects inputs and outputs, using software and processing units from simple circuits to complex computers. Robots can also read data

[1] https://www.ri.cmu.edu/about/
[2] https://ifr.org/

from sensors and send out electrical signals, allowing them to interact with the environment. By this definition several automatic devices could be considered robots. In industry the majority of robots is designed to move parts in the environment, splitting into two major categories: moving robots and manipulative robots, each with specific applications that meet the needs of modern programming (Correa et al., 2020).

This paper is organized as follows: section 2 will bring the necessary literature and manual reviews on industrial robots while section 3 focuses on the materials of the RV-2AJ Industrial Robot and its parts. Section 4 will present 2 practical studies that were performed and the relevant discussion and finally section 5 is conclusion and future work.

1.1. Objectives

The objective of this paper is to understand how the programming of the industrial robot MELFA RV-2AJ, installed at the Automation Laboratory in UNESPs Sorocaba Campus, is performed and check the viability of integrating this robot in the current digital twin project. As a first step of the viability study, it was fundamental to understand how the robot works, its programming capabilities and the characteristics. This paper will cover the fundamentals study and the basic programming, while a future paper will cover the feedback and final results of the analysis.

This viability study is fundamental for the next steps of the research as it is by understanding how the programming is transferred we will determine if the Digital Twin platform can integrate with the RV-2AJ. Future study in the feedback from the robot to the controlling software will determine if the full integration is possible, allowing the Digital Twin project of the laboratory to program and re-program, control the robot and obtain feedback.

2. Literature Review

In order to understand how the programming and the robot itself works, this literature review will focus mainly on the manuals provided by Mitsubishi for the three main parts of the system: The Robotic Arm RV-2AJ (MITSUBISHI, 2002, 2024), the controller CR1-571 (MITSUBISHI, 2009b) and the programming software COSIMIR INDUSTRIAL 4.1 (MITSUBISHI, 2009a). A thesis by Guadalupe and Adriana (2009) and Perugachi and Calvopiña (2012) was also used, as the main goal of the thesis was a detailed study of the technology and applications of the robot since this is a current and important technology in the Industry 4.0 scenario (Korsoveczki et al., 2024).

As mentioned by Duque and Duque (2007), it is important to acquire knowledge in the handling of a industrial manipulator, since the autonomous robot technology is fundamental in the current industry. Their work is divided into 4 chapters: 1. Industrial Robots, where the basics is explained; 2. RV-2AJ Robotic arm with particularities of the specific model and its specifications; 3. Programming using COSIMIR software and 4. Practices with several applications and tests.

2.1. Robot Commands

Robot programming is done through many different commands and different types of movement. All theses movements take into consideration a coordinate system in a cartesian space to define positions or specific destinations to the robot. Each point (P1, P2 etc.) will included data that indicates the location of the robot (or the tool attached to it) in a three-dimensional space. These coordinates are expressed in a X, Y, Z format as follows:

- X: Horizontal Axel Position (Left/Right);
- Y: Depth Axel Position (Front/Back);
- Z: Vertical Axel Position (Up/Down);

To follow are the movement commands and how they interact and interfere with the robot. These are just the most basic commands that can be used and are necessary to be able to program the robot.

2.1.1. Angular Movement (MOV)

In this movement mode, the arm will generate an interpolar trajectory between the origin and destination points of movement. The robot moves with an angular interpolation of the axis and not a linear one, so it is not 100%

predictable which route it will take. It is the fastest movement a robot can perform. The details of the possible commands are showed in Table 1 below.

Table 1: MOV Commands

Command	Specification
MOV P1	Moves from the origin point to P1 (as defined in the code)
MOV P1+P2	Moves from the origin point to the result of the addition of coordinates P1 and P2
MOV P1, -50	Moves from the origin point to P1 minus 50mm backwards in reference to the claw position (Z axe)

2.1.2. Linear Movement (MVS)

With this mode, the robot will describe a linear trajectory in the three-dimensional space between its origin point and the destination, going through a known path when compared with the MOV command. It is typically a slow movement and it is used mainly in critical moments when the robot is near the final point of the task such as delivering or retrieving a part. The details of the possible commands are showed in Table 2 below.

Table 2: MOV Commands

Command	Specification
MVS P1	Moves from the origin point to P1 (as defined in the code)
MVS P1+P2	Moves from the origin point to the result of the addition of coordinates P1 and P2
MVS P1, -50	Moves from the origin point to P1 minus 50mm backwards in reference to the claw position (Z axe)

2.1.3. Circular Movement (MVR and MVC)

In this type of movement, the robot will move in a circular way between three reference points. If the robot is not at the starting position when the command is given it will slowly move to it before executing the circular movements command. To illustrate better the command from Table 3 below, Figure 1 shows the four types of movement as well as the results from each of them. The reference points are detailed in each line of the table.

Table 3: MVR and MVC Commands

Command	Specification
MVR	Defines three points for circular move: Beginning, Middle and End. The arm will pass by the three reference points (as displayed by point P1, P2 and P3 of Figure 1)
MVR2	Defines three points for circular move: Beginning, End and Reference. The arm will move between the Beginning and End points via the shortest path. The reference point (P5) is necessary to determine the size of the circle. (as displayed by point P3, P4 and P5 of Figure 1)
MVR3	Defines three points for circular move: Beginning, End and Center. The arm will move between the Beginning and End points via the shortest path. The center point (P7) is necessary to determine the size of the circle. (as displayed by point P4, P6 and P7 of Figure 1)
MVC	Defines three points for circular move: Beginning, Passage1 and Passage 2. The arm will move completing a full circle starting at the Beginning, then Passage1 and the Passage2 ending at the Beginning point. (as displayed by point P6, P8 and P9 of Figure 1)

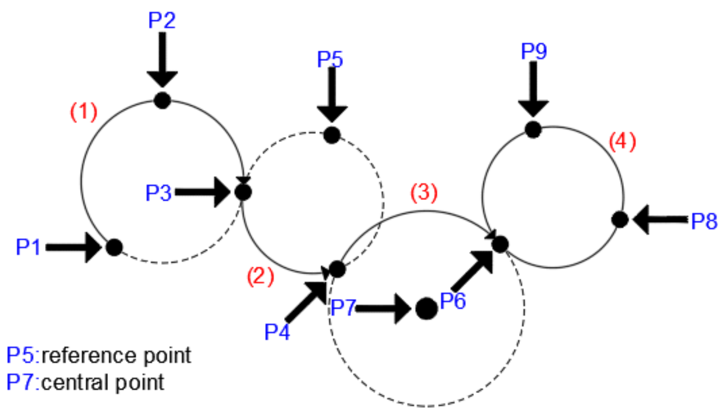

Figure 1: Example of MVR and MVC Commands (The authors)

2.1.4. Continuous Movement (CNT)
This set of instructions allows a continuous movement between the points defined by instructions such as MOV, MVS, MVR and MVC, without accelerations or decelerations and displayed at Table 4 below.

Table 4: CNT Commands

Command	Specification
CNT 1	Enables continuous movement
CNT 1,100,200	Defines the separation distance with the initial point (100mm for example) and the final point (200 mm for example). The command configures the movement from the initial point keeping fluidity and trajectory and adjusts the distances for a smooth transition from beginning to end.
CNT 0	Disables continuous movement

2.1.5. Speed Related Commands
To control the speed in which the arm moves there are a set of four commands as displayed in Table 5.

Table 5: Speed Related Commands

Command	Specification
ACCEL	Defines the percentage of acceleration and deceleration in reference to the maximum allowed
OVRD	Defines the robot´s working speed in percentage to the maximum speed
JOVERD	Defines the axis interpolation speeds in percentage to the maximum speed
SPD	Defines the speed (in mm/s) of the linear and circular movement of the claw or final effect.

2.1.6. Claw Control
To perform the open and close commands of the claw fitted in a robot arm the commands displayed in Table 6 below are used

Table 6: Claw Commands

Command	Specification
HOPEN 1	Opens the pneumatic claw
HCLOSE	Closes the pneumatic claw

2.1.7. Timer / Delay
The Timer, or Delay, is an instruction that causes the program to wait for a period of time. After the specified time is elapsed the program will resume in the next line of code. The command is displayed in Table 7.

Table 7: Delay Commands

Command	Specification
DLY 5	Waits for 5 seconds (for example) to resume program execution

2.1.8. Additional Moving Modes
There are also five additional moving methos that can be accessed via the handheld control device of the controller CR1-571. These commands are listed below in Table 8.

Table 8: Handheld movement Commands

Command	Specification
JOINT	This movement command allows for independent individual control of each joint of the robot, moving it on a singular axis.
TOOL	This mode moves the robot according to the position and angulation of the claw. By using this command all other joints will move keeping the claw in the same "level" as before.
XYZ	Similar to the TOOL command, but with cartesian coordinates, this movement also keeps the claw "leveled" as it moves towards the X, Y, Z coordinate.
3-AXE XYZ	This movement type limits the move to the coordinate system, in either of the coordinates as described in section 2.1
CYLINDER	This mode stablishes a cylinder along the Z axis that the robot can move inside of. It will move in the axis defined by the limitations.

2.2. Programming

Programming of the robot can be made by the Handheld device installed in the controller box or via a specific software. Since the goal of this research is to determine the viability of accessing the robot remotely, this review will focus on the software responsible for programming the RV-2AJ robot: COSIMIR INDUSTRIAL V4.1 as seen in Figure 2 below (Mihalikova et al., 2008). In their thesis, Guadalupe and Adriana (2009) shows in chapter 3 the proper use the software by creating new files, setting up the system so it is possible to monitor digital inputs and outputs and also how to use the MELFA BASIC IV programming language.

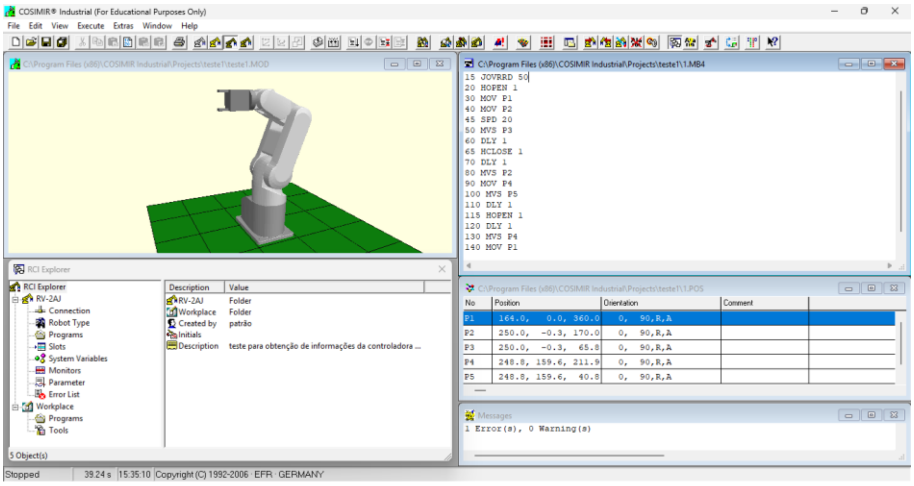

Figure 2: COSIMIR INDUSTRIAL 4.1 Interface (The authors)

COSIMIR generates a series of files for each project that is created, however only two files are essential to program the controller. These files have the extension ".POS" and ".MB4". The ".POS" file is a text description of a position created by Trimble Navigation GPS or receivers from the global GPS. The file will contain and ID followed by longitude and latitude values of each ID. It can also locate the points by describing the origin point. The file needs specific software to be able to read it, such as the "My Topo POS2MXF" or "Trimble Navigation GPD Pathfinder Office". (Ficheiros, 2025; ReviverSoft, 2025). In the particular case of the RV-2AJ, the file contains the position in the cartesian plain of the necessary points, storing the coordinates. The ".MB4" file are related to the common "Model Builder" as they are categorized as a primary type of file. (Solvusoft, 2025). In the particular case of the RV-2AJ they store the program itself

3. Method and Materials

As mentioned before, this paper is part of a bigger research project that is focused on creating a Digital Twin of the FMS Legacy Systems. Many of the platforms and languages have been selected before this paper, and for the purpose of this paper the most significant choices can be found in (Sousa et al., 2024), in which the Node-RED environment is used and the FESTO FMS system is detailed. This paper will focus on the Robot Station of the FMS that is equipped with a MELFA RV-2AJ robot by Mitsubishi and its necessary parts.

The COSIMIR INDUSTIAL 4.1 software is a tool for 3D Simulation and Offline programing of robot-based work-cells. It is vastly used to develop solutions in the area of robotics and automated systems. The version installed in the Automation Laboratory is provided by FESTO (IRR, 2000). The use of COSIMIR is fundamental to better understand the essential files needed to program the RV-2AJ robot arm (.POS and .MB4) so it is possible to recreate this files in another platform or simply use the files created offline in a repository do the system can choose for itself. Therefore, the method implement by this research was an extensive review of the manual and related texts and the use of practical applications to verify the file formats as well as testing the programming skills of the authors.

It is important to acknowledge here the participation, int the second semester of 2024, in Prof. Prof. Dr. Mauricio Becerra Vargas discipline of Robotics at UNESP. The main author of this paper was allowed to attend the discipline as a listener creating a great environment to learn and develop the necessary knowledges and fundamentals or robot programming.

3.1. MELFA RV-2AJ

The MELFA RV-2AJ is a complete solution for robot arm automation. It has the robot arm (RV-2AJ), a controller box (CR1-571) and a Handheld device to allow programming and troubleshooting. The Robotic Arm has five degrees of freedom (five joints) and is capable of carrying up to 2 kg. Its operating area is around 410 mm and can it can achieve speeds of 2.200 mm/s. In Figure 3 we can see on the left-hand side a picture of the arm and on the right-hand side the mechanical drawing of it.

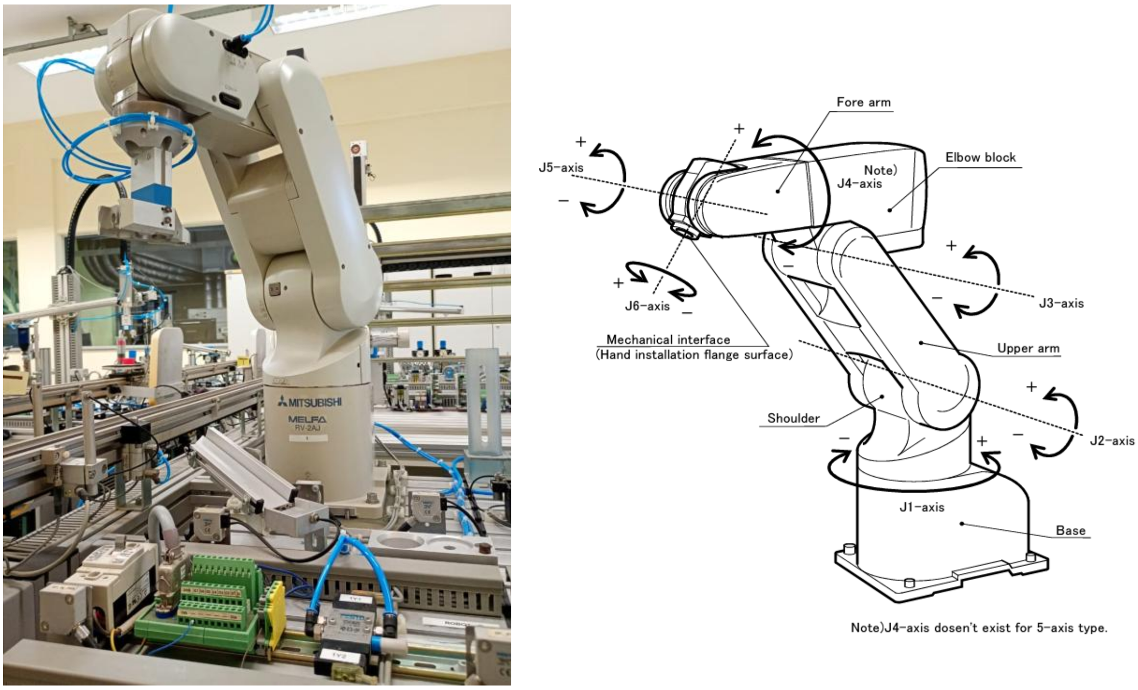

Figure 3: MELFA RV-2AJ Robot Arm (The authors)

The Controller box CR1-571, as showed in Figure 4, is the control system of the robot. It handles the calculations, cinematics and dynamics of robot movement. It can store up to 88 different programs and allows the execution of up to 32 in parallel in multi-task mode. It contains a RS-232 serial port for communication with COSIMIR and 16 digital inputs/outputs.

Figure 4: MEFLA CR1-571 Control Box (The authors)

The Handheld, also known as "learning box" and displayed in Figure 5, is used to move the robotic arm and determine the positions required for each task. The movement capabilities are detailed in section 2.1.8.

The pneumatic claw that is installed at J6 axis is also called final effect is display in Figure 6. It is a pneumatic clamp with three cylindrical retention capabilities and an infrared optic sensor to determine if a part is being transported/held. It can handle pressures up to 5 bar.

Figure 5: MEFLA Handheld (The authors)

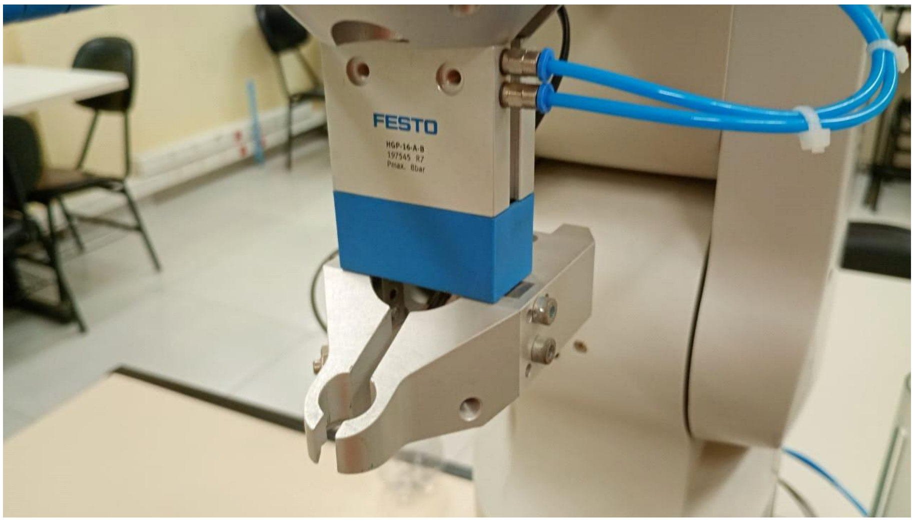

Figure 6: Pneumatic Clamp (The authors)

4. Results and Discussions

Understanding how COSIMIR INDUSTRIAL 4.1 operates is essential to be able to create an interface layer that can provide the same codes and read the same signals, allowing other instances to communicate with the robot. By verifying its viability, new features can be implemented by the Digital Twin and its uses for the robot arm. The program for production can be remotely selected by the RFID tags from the main Digital Twin application as it is providing real-time data on the sensors and actuators of the robot allowing for a 3D view outside the proprietary software.

For that purpose, 2 practice scenarios were created so we could understand how COSIMIR INDUSTRIAL 4.1 and the controller CR1-571 exchange information. The first scenario is a simple part transfer from one surface to another. The second and more complex scenario involves the movement of multiple pieces. Through these practical applications it was possible to verify both .POS and .MB4 files created and determine how COSIMIR transfers the information to the control module. The file formats are described in detail in section 2.2.

4.1. Example 1 – Single Piece

This first example was performed so the commands could be better understood. Also, the relative simplicity of the program allowed for a deeper analysis of the files generated by COSIMIR. The program is detailed in Table ***9*** below with each line described for its function. An illustrative representation is provided in Figure 7.

This example consists in programming of the robotic arm parameters (lines 1 and 2), moving it to pick up the part by stopping in intermediary positions (P2) to reach it´s final position at P3. After it picks up the part it moves back to P1 and then towards it´s final position at P5 by stopping in another intermediary position (P4) After the part is released in P5 the arm moves back to it´s original position P1.

Table 9: Example 1

Command	Detailed Function
10 OVRD 50	Defines base speed at 50%
15 JOVRD 50	Defines joint speed at 50%
20 MOV P1 ;	Moves the arm to P1
30 MOV P2 ;	Moves the arm to P2
40 SPD 20 ;	Defines movement speed at 20%
50 MVS P3 ;	Linear move to P3
60 DLY 1 ;	1 second delay
70 HCLOSE 1 ;	Closes Claw
80 DLY 1 ;	1 second delay
90 SPD 50 ;	Defines movement speed at 50%
100 MVS P2 ;	Linear move to P2
110 MOV P1 ;	Moves the arm to P1
120 MOV P4 ;	Moves the arm to P4
130 SPD 20 ;	Defines movement speed at 20%
140 MVS P5 ;	Linear move to P5
150 DLY 1 ;	1 second delay
160 HOPEN 1 ;	Open Claw
170 DLY 1 ;	1 second delay
180 SPD 50 ;	Defines movement speed at 50%
190 MVS P4 ;	Linear move to P4
200 MOV P1 ;	Moves the arm to P1
210 END ;	End program

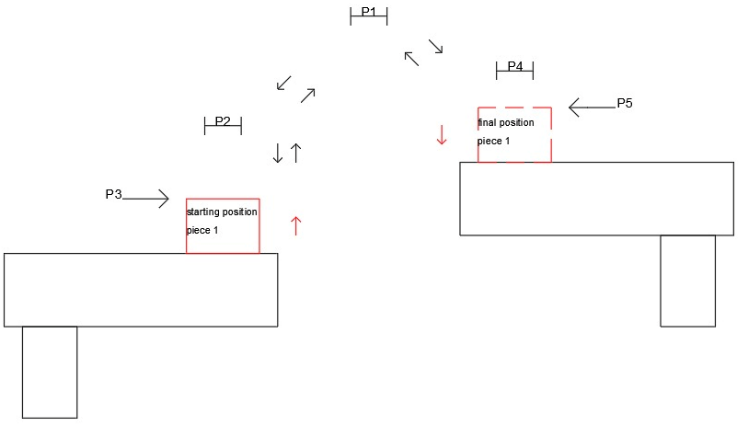

Figure 7: Example 1 illustrative process (The authors)

4.2. Example 2 – Multiple parts displacement

This second example was performed so complexity was added and the commands and programming were better explored. In this scenario a pile of three parts will be moved. Stacking of the parts is an important procedure and requires fine tuning of the coordinates so the pile won´t collapse on itself. The program is detailed in Table 10 **Table *9*** below with each line described for its function. An illustrative representation is provided in Figure 8.**Figure 7**

This example consists in programming of the robotic arm parameters (lines 1 and 2), moving it to pick up the part by stopping in intermediary positions (P2) to reach its final position at P3. After it picks up the blue part it moves back to P1 and then towards its final position at P9 by stopping in another intermediary position (P6) After the blue part is released in P9 the arm moves back to its original position P1. Here, the process is started once more, now moving to P4 to pick up the green part with to deliver it at P8. Finally, we do the process once more for the red part from P5 to P7. Once all parts are moved, the robot moves back to P1. It is important to notice that, due the nature of the exercise, the order that the parts are stacked is changed from Blue; Green; Red to Red; Green; Blue.

Table 10: Example 2

Command	Detailed Function
10 OVRD 50 ;	Defines base speed at 50%
15 JOVRD 50 ;	Defines joint speed at 50%
20 MOV P1 ;	Moves the arm to P1
30 MOV P2 ;	Moves the arm to P2
40 SPD 20 ;	Defines movement speed at 20%
50 MVS P3 ;	Linear move to P3
60 DLY 1 ;	1 second delay
70 HCLOSE 1 ;	Closes Claw
80 DLY 1 ;	1 second delay
90 MVS P2 ;	Linear move the arm to P2
100 MOV P1 ;	Moves the arm to P1
110 MOV P6 ;	Moves the arm to P6
120 MVS P9 ;	Linear move the arm to P9
130 DLY 1 ;	1 second delay
140 HOPEN 1 ;	Open Claw
150 DLY 1 ;	1 second delay
160 MVS P6 ;	Linear move the arm to P6
170 MOV P1 ;	Moves the arm to P1
180 MOV P2 ;	Moves the arm to P2
190 MVS P4 ;	Linear move to P4
200 DLY 1 ;	1 second delay
210 HCLOSE 1 ;	Closes Claw
220 DLY 1 ;	1 second delay
230 MVS P2 ;	Linear move the arm to P2
240 MOV P1 ;	Moves the arm to P1
250 MOV P6 ;	Moves the arm to P6
260 MVS P8 ;	Linear move the arm to P8
270 DLY 1 ;	1 second delay
280 HOPEN 1 ;	Open Claw
290 DLY 1 ;	1 second delay
300 MVS P6 ;	Linear move the arm to P6
310 MOV P1 ;	Moves the arm to P1
320 MOV P2 ;	Moves the arm to P2
330 MVS P5 ;	Linear move the arm to P5
340 DLY 1 ;	1 second delay
350 HCLOSE 1 ;	Closes Claw
360 DLY 1 ;	1 second delay
370 MVS P2 ;	Linear move the arm to P2
380 MOV P1 ;	Moves the arm to P1
390 MOV P6 ;	Moves the arm to P6
400 MVS P7 ;	Linear move the arm to P7
410 DLY 1 ;	1 second delay
420 HOPEN 1 ;	Open Claw
430 DLY 1 ;	1 second delay
440 MVS P6 ;	Linear move the arm to P6
450 MOV P1 ;	Moves the arm to P1
460 END ;	End

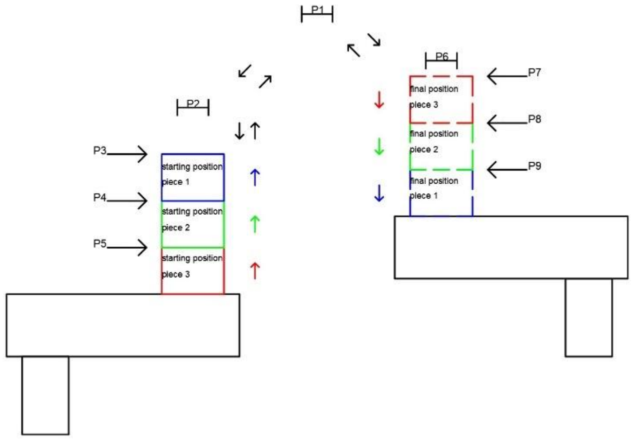

Figure 8: Example 2 illustrative process (The authors)

5. Conclusion and future work

Based on the studies performed by reading the Mitsubishi manuals, the referred literature and by the class taken in the second semester of 2024 it was possible to achieve considerable advances in the understanding the programming procedures and details of the MELFA RV-2AJ Robotic Arm. This preliminary knowledge allowed for the execution of not only the two examples displayed in this paper but several other exercises to enhance the understanding of the robotic arm, its capabilities and functionalities.

After analyzing the .POD and .MB4 files created by the COSIMIR INDUSTRIAL 4.1 software it was possible to completely understand them and it is possible to replicate them easily outside the COSIMIR environment. The files are simple to understand and do not require the software to be translated or implemented. One file contains the positions and the other on the set of instructions. It is possible to conclude that the "writing" part of the viability study is concluded and deemed possible as the files used can be replicated.

The ability of the MELFA CR1-571 control box to store several programs is also very positive, as this can be downloaded before-hand and the Digital Twin / Node-RED Platform can simply select which one to run by sending the correct signal to the robot arm. This possibility will be investigated in future research.

Since COSIMIR has the 3D view of the current status of the Robots actual position, there is feedback information coming from the control box to the software. The next part of this research is understanding the format in which that is reported, since it was not possible to see any references in the literature review.

Acknowledgements

This paper was developed in part by researches of the GPI4 Research Group[3]

Prof. Dr. Mauricio Becerra Vargas for allowing the participation in his class of the student Leonardo.

References

Correa, A. P. A., Vanegas, E. F. L., and Abad, J. C. Z. (2020). *Desarrollo de un manual de guías de prácticas orientado al aprendizaje de la Robótica Industrial* [bachelorThesis]. http://dspace.ups.edu.ec/handle/123456789/18296

Duque, J. F., and Duque, V. A. V. (2007). *Documentación y Programación de prácticas del brazo robotico de la celda de manufactura flexible de la UTP*. https://hdl.handle.net/11059/1109

Ficheiros. (2025). *POS Extensão de Arquivo | O que é um arquivo .POS e como abri-lo*. https://ficheiros.com.br/extensao/pos/

Guadalupe, H. F. L., and Adriana, L. C. (2009). *DOCUMENTACIÓN Y PROGRAMACIÓN DE PRÁCTICAS CON EL ROBOT INDUSTRIAL RV-2AJ ESTACIÓN 6 DEL LABORATORIO MÓVIL DE AUTOMATIZACIÓN Y ROBÓTICA DE LA UNIVERSIDAD VERACRUZANA* [Tese, Universidad Veracruzana]. https://www.uv.mx/instru/files/2021/01/TesinaBrazoRobotico-RV2-AJ.pdf

[3] http://dgp.cnpq.br/dgp/espelhogrupo/350090

IRR. (2000). *COSIMIR: Getting Started*. Institute of Robotics Research. http://www.int76.ru/upload/iblock/74c/74caae11f575cafdc754991bc9f43d67.pdf

Korsoveczki, G., Vasváry, T., Szarka, A. V., and Sarvajcz, K. (2024). Digital twin design of a 5 degrees of freedom industrial robotic arm for engineering education purposes. *Measurement: Sensors*, 101323. https://doi.org/10.1016/j.measen.2024.101323

Mihalikova, J., Rovnakova, S., Liska, O., and Zivcak, J. (2008). Design of the Modular Production System by simulation software's Cosimir®. *2008 IEEE International Conference on Computational Cybernetics*, 223–226. https://doi.org/10.1109/ICCCYB.2008.4721410

MITSUBISHI. (2002, março 18). *MELFA Instruction Manual*. https://www.unisgroup.nl/pdf/EN/Mitsubishi-CR1-Manual.pdf

MITSUBISHI. (2009a, setembro). *RV-1A/2AJ Series: INSTRUCTION MANUAL ROBOT ARM SETUP and MAINTENANCE*. https://suport.siriustrading.ro/02.DocArh/07.RI/03.Seria%20RV%20(Vertical)/03.RV-A/02.Manuale/RV-1A,2AJ%20-%20Instruction%20Manual%20(Arm%20Setup,%20Maintenance)%20BFP-A8052-E%20(09.09).pdf

MITSUBISHI. (2009b, outubro). *RV-1A/RV-2AJ Series: Standard Specifications Manual (CR1-571 Controller)*. http://suport.siriustrading.ro/02.DocArh/07.RI/03.Seria%20RV%20(Vertical)/03.RV-A/02.Manuale/RV-1A,2AJ%20-%20Standard%20Specifications%20Manual%20BFP-A8050-K%20(09.09).pdf

MITSUBISHI. (2024). *Robôs industriais MELFA*. https://www.mitsubishielectric.com/fa/br_pt/products/rbt/robot/items/vertical/index.html

Perugachi, X. J. A., and Calvopiña, P. O. (2012). *Desarrollo de una guía de prácticas de laboratorio para el MPS 516-FMS de Festo de la carrera de Electrónica de la Universidad Politécnica Salesiana* [bachelorThesis]. http://dspace.ups.edu.ec/handle/123456789/3590

ReviverSoft. (2025). *POS Extensão de arquivo—O que é .pos e como abrir? - ReviverSoft*. https://www.reviversoft.com/pt/file-extensions/POS

Solvusoft. (2025). *Extensão de Arquivo MB4: O Que É and Como Abrir?* https://www.solvusoft.com/pt-br/file-extensions/file-extension-mb4/

Sousa, D. R. de, Caldana, V., and Godoy, E. P. (2024, maio 7). *Upgrading legacy systems for Industry 4.0 with Node-RED and OPC-UA*. 5th South American Industrial Engineering and Operations Management Conference. https://doi.org/10.46254/SA05.20240068

Biographies

Leonardo Henrique Camilo Patrão is currently a student of the technical course of Electronics integrated to High Scholl at the IFSP Campus Sorocaba, started in 2023 to be concluded at the end of 2025. Since 2024 holds a scholarship of the Scientific Initiation program PIBIC Junior CNPQ/Reitoria Unesp at the Sorocaba Campus of UNESP researching Robot Manipulating Arms

Vitor Mendes Caldana began his career with a technician course in Electronics from Liceu de Artes e Ofícios in 1999, followed by an undergraduate degree in Electronic Engineering from Universidade Presbiteriana Mackenzie in 2004. In 2016, finished his M.Sc. in Industrial Engineering. Since 2023 is a student at UNESP to obtain his Ph.D. in Electronic Engineering in the Industry 4.0 field. In 2014 began his teaching career in FIEB as a substitute teacher, followed by an associate professor position for the Technical Course of Electronics. In 2016 became a full-time professor by joining IFSP, moving to the Sorocaba Campus to implement the Electronics High-School Technical Course. In 2018 started the Research Group in Industry 4.0 at IFSP and has been its leader since. Between 2019 and 2020, along with his colleagues, designed and implemented the first Post-Graduate Program in Industry 4.0 of IFSP at the Sorocaba Campus. He is currently involved in research projects in Industry 4.0. https://lattes.cnpq.br/8361188962318020.

Eduardo Paciencia Godoy received the B.Eng. degree in Control and Automation Engineering at Itajubá Federal University (MG-Brazil) in 2003 and the M.Sc. and Ph.D. degrees in Mechanical Engineering at University of São Paulo (SP-Brazil) in 2007 and 2011. Currently he is an Associate Professor of Unesp (SP-Brazil) with research interests in IoT and Industry 4.0. http://lattes.cnpq.br/0072632067545698.