COBEM 2021 26th International Congress of Mechanical Engineering
ABCM Associação Brasileira de Engenharia e Ciências Mecânicas
November 22-26, 2021, Virtual Congress, Brazil

COB-2021-1147
MODELING AND SIMULATING A COMMERCIAL ROBOTIC MANIPULATOR USING COPPELIASIM

Amanda Kozlowski de Morais
Felipe Arantes Lobo
João Paulo da Silva Fonseca
Universidade Federal de Goiás Av. Ingá, Prédio B5 Campus Samambaia CEP: 74.690-900 - Goiânia – Goiás – Brasil
kmoraisamanda@discente.ufg.br
felipe.arantes.lobo@gmail.com
jpsfonseca@ufg.br

Abstract: *Industrial robots and their applications have received increasing attention in recent years, mainly due to the expansion of industry 4.0 and digital manufacturing concepts, generated a growing demand for robust virtual environments that make possible to design and test a variety of applications and new possibilities for control architectures, without relying solely on the availability of hardware. This paper aims to present the application of an accessible method for building an industrial robotic manipulator simulation model. For the development of this test model, it was opted to use CoppeliaSim, a virtual experimentation platform specifically for robotics, with free versions. The robotic manipulator for the virtual environment started at a CAD3D model made available by the manufacturer, which was need to adapt. After obtaining the manipulator model, it was proceeding to a verification step, in which welding scenarios were created with tasks of movement and manipulation of the terminal element, with a weld bead representation. During the simulation, the speed of each robotic manipulator link and joint can be monitored and controlled, since the speed of the welding torch is an extremely important parameter in the result. Finally, it is concluded that the software used is applicable to the proposal of the work, being presented as an alternative tool for industries and research centers inserting themselves in Industry 4.0, requiring only the 3D manipulator model to generate simulation environment that can be applied to different scenarios, which can be pre-developed and then adapted as needed.*

Keywords: *robotic, manufacturing, simulation, industry 4.0, coppeliasim.*

1. INTRODUCTION

Industrial robots and their applications have received increasing attention in recent years, mainly due to the expansion of Industry 4.0 and digital manufacturing concepts. High levels of adaptability, flexibility and fast design changes are more request in this new industrial age, as Frank *et al.* (2019) "findings show that Industry 4.0 is related to a systemic adoption of the front-end technologies, in which Smart Manufacturing plays a central role".

There are many factors working for the growing of the modern manufacturing industry, one of them is the autonomous productions method powered by robots that can complete tasks intelligently, ensuring flexibility, versatility and safety throughout the process in which they are apply (Bahrin *et al.*, 2016).

According to Groover (2011), process can be considered autonomous because they execute their tasks with a reduced level of human participation, when compared to the equivalent manual process. The autonomous systems can basically be classified in three types, as shown in Figure 1: (1) Fixed or rigid automation system; (2) Programmable automation system; (3) Flexible or soft automation system.

A Fixed Automation system receives this name because its equipment configuration is designed for a sequence process or assembly operation specific. When the configuration is planned for that the equipment has the capability to change the sequences of operation to accommodate different products configuration, it is called Programmable Automation system. And the last one, Flexible or Soft Automation systems are those that are capable to produce a variety of products with virtually no time lost for equipment changes. They are an extension of programmable automation (Groover, 2011). Each of these systems has a different level of computerization, present either in the product design phase, or production planning, operations control and execution stage.

The extensive use of computers in manufacturing process is called Computer Integrated Manufacturing (CIM) and it is increasingly present in the modern and big industries. Providing increasingly larger production volumes, more velocity in production and in product change, and also high quality of the products, new technologies are always released in the market. Thus, it is necessary to look for tools that help this integration process in this industry 4.0 environment, so that it is possible to continue operating in the current market.

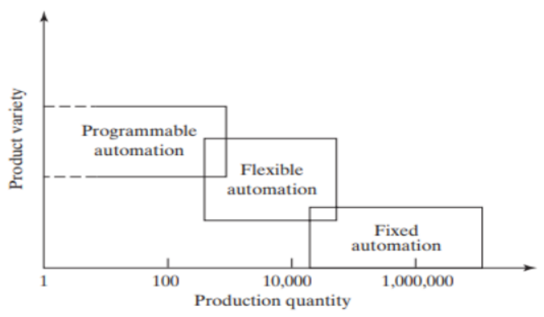


Figure 1. Autonomous systems classification by Groover (2011).

Liagkou *et al.,* (2019) discuss about how Virtual Reality is an important tool of Industry 4.0 in order to "decrease design and production costs, maintaining product quality and overcoming several technical tradeoffs like reducing rendering complexity while keeping high refresh rates or increasing resolution". This appreciation has been generating a growing demand for robust virtual environments that make possible to design and test a variety of applications and new possibilities for control architectures, without relying solely on the availability of hardware.

Simulation, other important tool that can be combined with virtual reality, is the process of representing a real or theoretical model of a physical system, executing and analyzing the model's result (Zlajpah, 2008). Simulation has become an important research tool since the beginning of the 20th century and its use was strongly boosted by the development of computers (Zlajpah, 2008). This tool helps against the disadvantages of textual robot programming that it requires the robot to be taken out of production so that a new programming can be made, when combined with the offline programming method. This method allows the program to be prepared at a distant computer terminal and then downloaded to the robot controller without interrupting production (Groover, 2011). The use of graphic simulation is relevant for validating the developed programs. In addition to the possibility of controlling virtual robots, there is the virtual environment modeling, enabling the complete graphical production line representation. This allows a study of factory layout planning, robot movement planning, equipment, devices, considering security possibilities and collision detection (Rosário, 2010).

The work "Automated Offline Programming for Robotic Welding System with High Degree of Freedoms" (Pan *et al.,* 2011) recognizes that one of the great difficulties of flexible automation-based robotics is the programming complexity. This difficulty is exemplified in the production of an armored vehicle, which is made with a robotic manipulator with 13 DOFs. The programming method used is manual and online, which takes approximately 6 months to complete the programming and 16 hours for the robotic manipulator to carry out the welding process.

As shown, there is a high technological level required in equipment and processes of in industry 4.0 which ones are not a reality for all industries and research centers, so that many still need to adapt to it and look for tools to remain competitive and active in the market. Thus, this paper will present the application of an accessible method for building an industrial robotic manipulator simulation model.

2. METHODOLOGY

According to the objective of developing a manipulator simulation model, it was opted to use CoppeliaSim, a virtual experimentation platform specifically for robotics, with free versions. Furthermore, the nearness of the research group with this software was another criterion for choice, although there are others good software such as Gazebo and Unit. CoppeliaSim is based on a distributed control architecture, in which each element can be controlled through an embedded script, a ROS (Robot Operating System) plugin, a remote API (Application Programming Interface) or a custom solution. The diverse means of control makes the CoppeliaSim ideal for applications with many robots. Scripts can be written in C/C++, Python, Java, Lua, Matlab or Octave. This work was done using the Lua language, this is the native language of CoppeliaSim.

2.1 Robot model in virtual environmet

To start development, it is necessary to build and adapt the simulation model in the CoppeliaSim virtual environment. The steps followed are shown in Figure 2 above, a flowchart of the virtual robot model adapting process for simulation in the software, which each activity is described in the next sub sections.

2.1.1. Robot model import

The robotic manipulator used in this paper, Yaskawa Motoman HP20D, was not available in CoppeliaSim, so its building in the virtual environment started at a CAD3D model made available by the manufacturer. It was accessible in extension STEP (Standard for the Exchange of Product Data) and IGES (Initial Graphics Exchange Specification) (Yaskawa, 2019), however none of these formats are compatible with the CoppeliaSim. FreeCAD (an open source parametric 3D CAD modeler) was used to convert the 3D model to OBJ (Wavefront 3D Object File) format, compatible with CoppeliaSim.

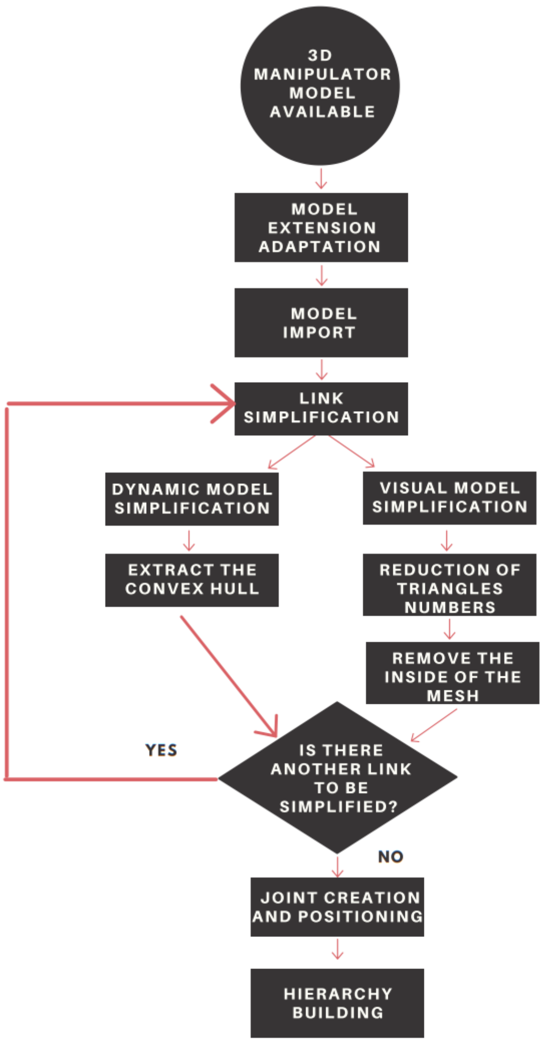

3D MANIPULATOR MODEL AVAILABLE
MODEL EXTENSION ADAPTATION
MODEL IMPORT
LINK SIMPLIFICATION
DYNAMIC MODEL SIMPLIFICATION
VISUAL MODEL SIMPLIFICATION
EXTRACT THE CONVEX HULL
REDUCTION OF TRIANGLES NUMBERS
REMOVE THE INSIDE OF THE MESH
YES
IS THERE ANOTHER LINK TO BE SIMPLIFIED?
NO
JOINT CREATION AND POSITIONING
HIERARCHY BUILDING

Figure 2. Flowchart Robot model implementation process in CoppeliaSim virtual environment.

2.1.2. Robot model simplification

After model import in CoppeliaSim is completed, it is necessary to make some modifications to the model in order to enable simulation. The more complex the scene, the greater computational effort generated. To reduce this effort, it is recommended that elements used be simplified. Elements in scenes are "seen" by CoppeliaSim as a mesh of triangles. The complexity of each element can be seen by its number of triangles. Elements with many triangles can slow the simulation, so smaller numbers of triangles are preferable. A maximum of 20000 triangles is recommended for each element, which can assume higher values in the case of a scene with few elements (Robotics, 2019).

CoppeliaSim makes available tools to simplify the model. The first tool to be used, as shown in Figure 2 steps, is "Decimate the mesh" which reduces the number of triangles when changes model geometry. Is necessary to be careful with this tool because big changes may result in an unrecognizable model. Then the "*Remove the inside of the mesh*" function is used to do what its name says: remove the inner side of the model. After these simplifications is obtained the final visual model, shown in Figure 3, which is just renderable.

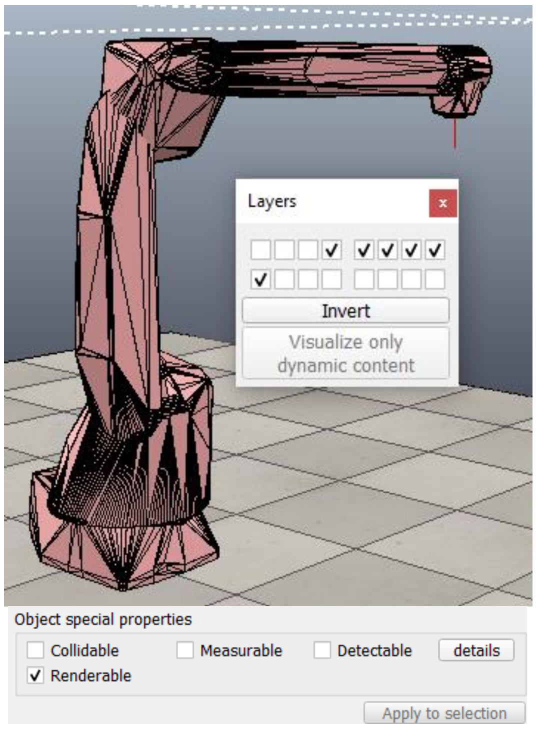
To run the simulation, it is necessary to obtain the dynamic model, shown in Figure 4. Thus the function "*Extract the convex hull"* is used to make the geometry convex, with a smaller triangles quantity in order to run a simulation with less computational effort after these simplifications. Then it is possible to enable the dynamic properties, also shown in Figure 4.

Both models, visual and dynamic, are shown according to the enabled layers as its possible to see in Figure 3 and Figure 4.

2.1.3. Joint creation and positioning

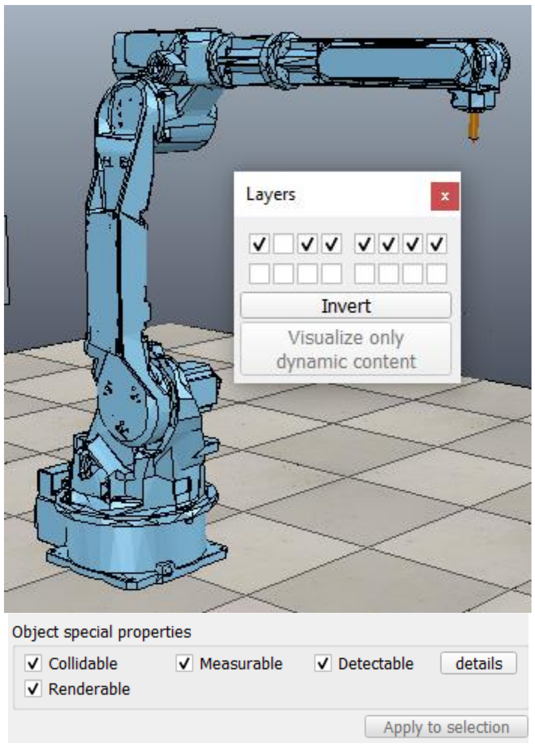
Then, if there is no other link to be simplified, it is necessary to create the joints, which will make the connection between the links and generate the right movement. For robotic manipulator of this work, the joints used are revolution joints. Three joints models are available in CoppeliaSim: revolution, prismatic and spherical.

The joints are created and positioned, as shown in Figure 5 as the red cylinders.


Layers
☐ ☐ ☐ ✓ ✓ ✓ ✓ ✓
✓ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Object special properties
☐ Collidable ☐ Measurable ☐ Detectable
✓ Renderable

Figure 3. Visual model and its properties.


Layers
✓ ☐ ✓ ✓ ✓ ✓ ✓ ✓
☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Object special properties
✓ Collidable ✓ Measurable ✓ Detectable
✓ Renderable

Figure 4. Dynamic model and its properties.

2.1.3. Hierarchy determination

For model final preparation it is necessary to create the hierarchy. Hierarchy is the relationship that dictates the order of movement, as the movement of a link in space affects the position and orientation of other links. In a practical way, when moving link number 2, links number 3 to 7 will also move, and this ordering is done through the hierarchy. The dynamic and visual model of each link must always stay together. Figure 5 shows the hierarchy of the final model.

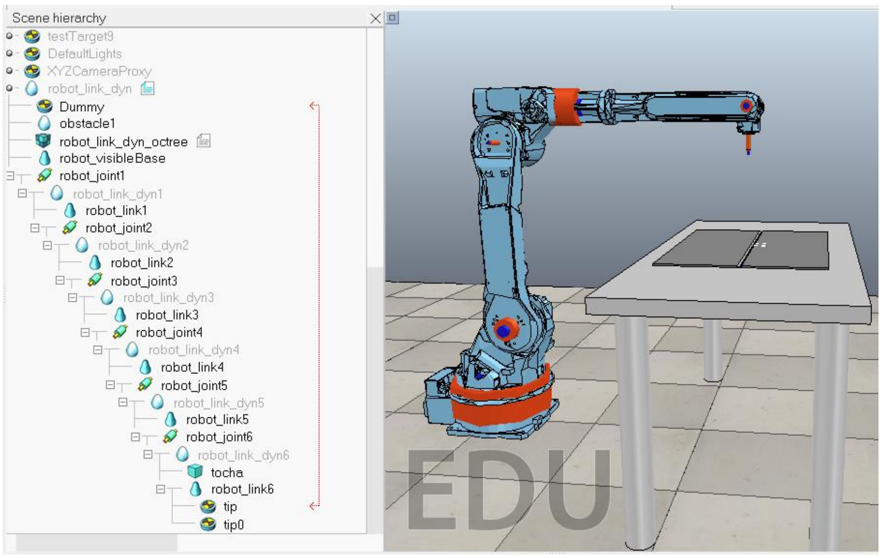


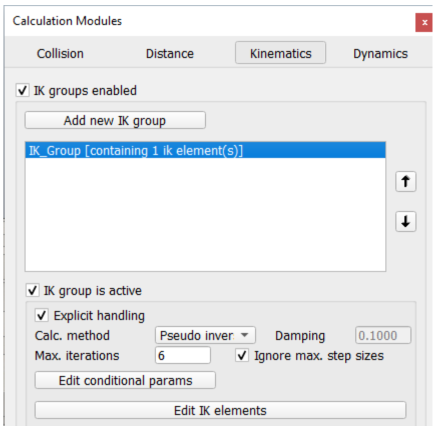
Figure 5. Joints (in red) and hierarchy (at left).

2.2 Inverse Kinematics

With the model appropriately imported and prepared the next step is design the movement of it. In order to know the position and orientation of an object in space, an object coordinate system must be related to a reference coordinate system (Craig, 2005). The mathematical relation of coordinate systems is called kinematics, which is the study of motion without considering the cause of the motion, such as forces and torques. In robotics are used two kinematics kinds: forward kinematics (FK) and inverse kinematics (IK).

CoppeliaSim has a IK function that calculate the joints orientation and position from a terminal position, as shown in its manual (Robotics, 2019). To use this function, must first create two Dummy's elements, which will be the "tip", position and orientation of the terminal element, and the "target", desired position and orientation for the terminal element. After this points determination, both are select and then use the "link selected dummies" function (Edit > Link selected dummies > IK, tip-target).

Afterwards, the inverse kinematics is created (Calculation Modules > Kinematics > Add new IK (Inverse kinematics) Group). Selecting the created IK Group displays the parameters used in calculating the inverse kinematics. Among the parameters are the calculation method (CoppeliaSim has two calculation methods for inverse kinematics, pseudo inverse and DLS), the number of iterations, and other parameters as are shown in Figure 6.



✓ IK groups enabled
IK_Group [containing 1 ik element(s)]
✓ IK group is active
✓ Explicit handling
Calc. method: Pseudo inver
Damping: 0.1000
Max. iterations: 6
✓ Ignore max. step sizes

Figure 6. Kinematics parameters.

2.3. Robot Manipulator Programming

There is a huge library of functions available, so some of them were chosen for programming the scenes needed for this article. The correct understanding of each of these functions is fundamental for understanding the program.

Despite the various functions and actions present in developed scenes, it is important to emphasize two points in particular: trajectory and movement planning and adding material.

2.3.1. Trajectory and movement planning

To start trajectory and motion planning is necessary to determine the location and orientation of a desired point as a destination for the robotic manipulator in space when programming the scenes, an object named target was used.

CoppeliaSim has trajectory and motion planning resources using OMPL (The Open Motion Planning Library). There are some functions needed to use this library (Robotics, 2019). In the software and in its manual there are some examples that shown how this resource may be used.

2.3.2. Adding material

To represent the weld beads was designed a material addition in the scene. Adding material takes place by adding voxels, the smallest element in a 3D representation. In CoppeliaSim there is the option to add octree (Add> octree), a hierarchical way of representing data in space in the form of cubes that can be subdivided into smaller cubes as needed. Thus, to add material it is necessary to create a function to enable the addition of voxels (enableOctreeCreation) in the position of the robotic manipulator terminal element and, at the end, another one to disable the addition of voxels (disableOctreeCreation).

3. RESULTS

After performing all the steps presented in the methodology and applying the simulation functions, the virtual model of the robotic manipulator Yaskawa Motoman HP20D was developed, shown in Figure 7. Then, welding scenes were developed in which the software's functionalities were applied and tested.

3.1. Scene 1: Welding Two 1/4-Inch Plates with Butt Joint and V-Chamfer in Flat 1G Position

In this scene there is the robotic manipulator, a workbench and two 1/4" plates, to be welded with butt joint and V-chamfer in the flat 1G position, as shown in Figure 7. In the scene, the welding machine, wires and the ground claw were not implemented.

A representation of the welding torch was created, the brown cylinder, visualized in Figure 7. This material deposition is done by adding voxels, using octrees.

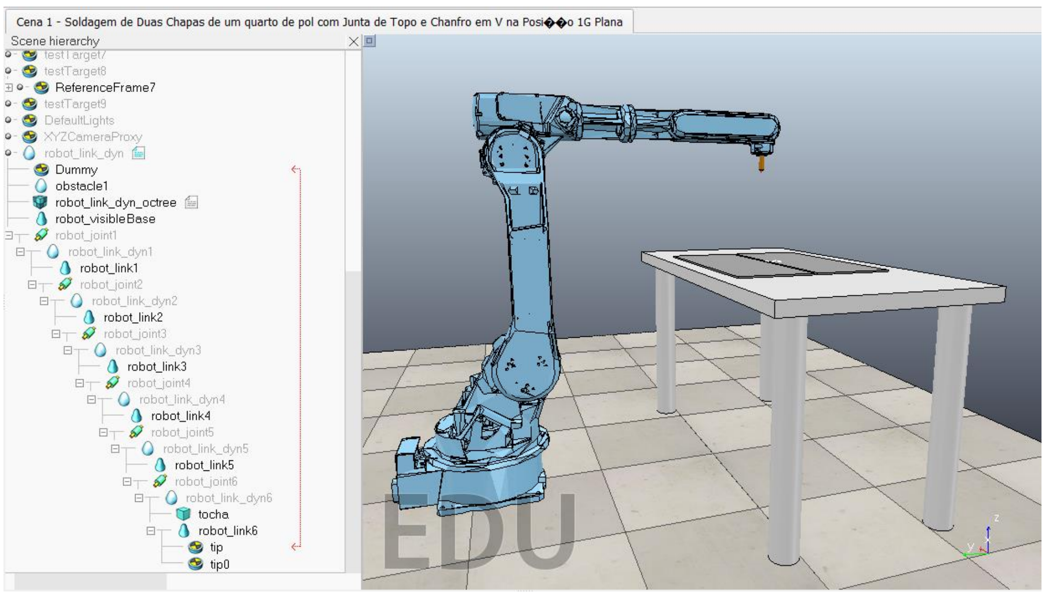


Figure 7. Scene 1 Welding Two 1/4-Inch Plates with Butt Joint and V-Chamfer in Flat 1G Position

Some techniques to minimize distortion were used. Initially, the plates were bridged with a weld point at the ends and at an intermediate point, as shown in Figure 8 (a) (b) and (c). The root pass was done in two steps. The first strand of the root pass was made from the point in the center of the plate to one end and the second from the end to the center. Then, two more passes were performed, as shown in Figure 9.

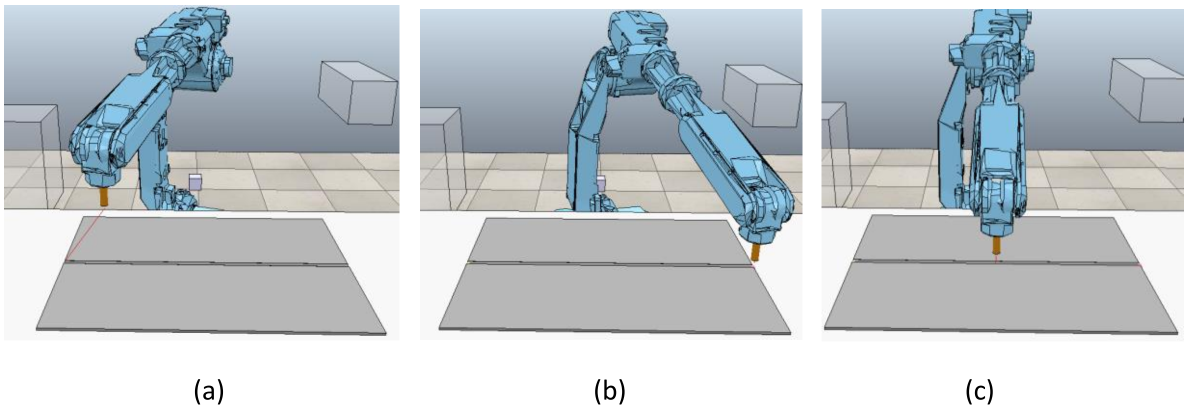


Figure 8. Weld points in the beginning of the plate (a), end of the plate (b) and in the middle (c).

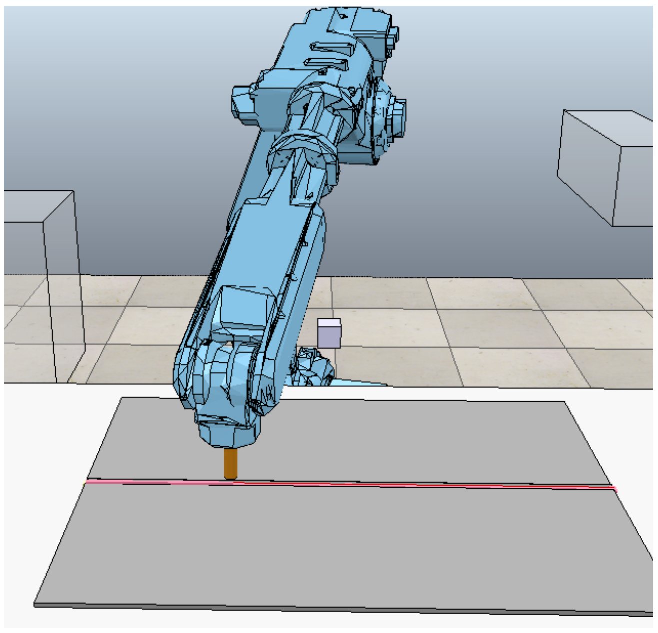


Figure 9. Welding second passes welding.

3.2. Scene 2: Welding I Profile in Position 2F and 4F

The welding process that will be simulated in this section will be the welding of an I profile in 2F (horizontal) and 4F (overhead) positions. The scene is shown in Figure 10. What makes this process interesting is the movement of the robotic manipulator without colliding with the part to be welded. The process starts with the robotic manipulator approaching the lower joint to be welded, at one end. It performs the weld, moves it away, approaches the upper joint, as shown in Figure 11, at the other end, performs the weld and then returns to its initial position and orientation.

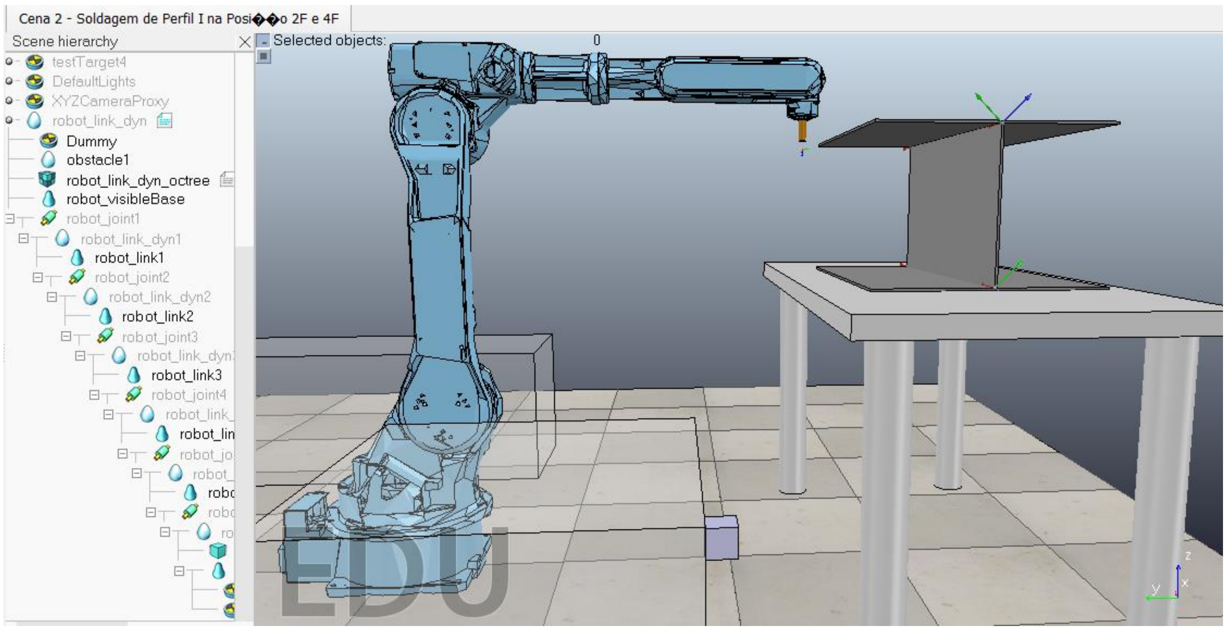


Figure 10. Scene 2 Welding I Profile in Position 2F and 4F

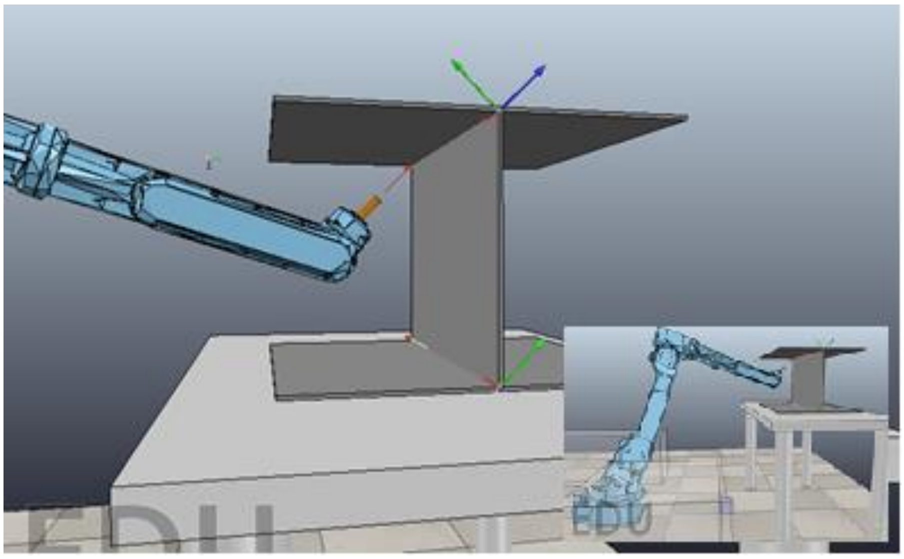


Figure 11. Robot manipulator performing the weld in upper joint.

4. CONCLUSION

This work presented a method to implement a simulation model of the Yaskawa Motoman HP20D robotic manipulator in the CoppeliaSim software. The elaborated scenes represent welding processes, one of the possible manufacturing processes capable of being performed by the robotic manipulator.

From the results presented, it is possible see some software's possibilities, such as the possibility of implementing a commercial robotic manipulator not available in its library, the trajectory planning which would be performed by the robotic manipulator and the process simulation of manufacturing.

Torch speed is a parameter that has great influence on the weld bead. It was noticed that implemented code does not have precise speed control as required in welding. This occurs because in code the velocity is given according to number of points on the trajectory, causing fluctuations in velocity.

Despite the welds being straight and with a low level of complexity, creating the trajectories required considerable time and effort. Therefore, it is concluded that it is feasible to develop processes with robotic manipulators for academic purposes or in cases of high production easily, and in cases of low production, where the process performed manually will be faster (added the time of the welding process with the time spent to implement the simulation model), the alternative to have a set of prepared environments is feasible too because once having the similar environment for the currently need, it is easy to adapt and apply to the manipulator.

Thus, with the set of prepared environments and a future integration with the robot language, different from Lua, the language used in this work, the use of this software and methodology is a good prospect for small research centers and industries inserting themselves in Industry 4.0, requiring only the 3D manipulator model to generate simulation environment that can be applied to different scenarios.

Concluding so that the software used is applicable to the proposal of the work, as an initial study of this subject.

5. REFERENCES

Frank, A. G., Ayala, N. F., and Dalenogare, L., 2019. “Industry 4.0 tecnologies: Implementation patterns in manufacturing companies”. *International Jounal of Production Economics,* Vol. 210, pp. 15–26.

Bahrin, M. A. K., Othman, M. F., Azli, N. H. N., Talib, M. F., 2016. “Industry 4.0: A review on industrial automation and robotic”. *Journal Teknologi (Sciences & Engineering),* Vol 78:6-13, pp137-143.

Groover, M. P., 2011. “Automation, production systems, computer-integrated manufacturing”. *Pearson Education, Inc*. 3nd edition.

Liagkou, V., Salmas, D., Stylios, C., 2019. “Realizing Virtual Reality Learning Environment for Industry 4.0”. *Procedia CIRP,* Vol. 79, pp 712-717.

Zlajpah, L., 2008. “Simulation in robotics”. *Mathematics and Computers in Simulation*, Vol. 79, pp. 879–897.

Rosário, J. M., 2010. “Robótica Industrial I Modelagem, Utilização e Programação”. *Editora Baraúna*, Vol. 1.

Pan, Z., Polden, J., Larkin, N., Duin, van S., Norrish, J., 2011. “Automated Oflline Programming for Robotic Welding System with High Degree of Freedoms”. *Advances in Computer, Communication, Control and Automation. Lecture Notes in Electrical Engineering,* Vol 121, pp. 685-692.

Yaskawa, HP20D/HP20F/HP20F-CR7. 2019. (web publication). http//www.motoman.lt/products/robots/product-view/?tx_catalogrobot_pi1%5Buid% 5D=300&cHash=7e1cbba13da9b964605576daa6c0beba>.Accessed 18 December 2019.

Robotics, C. 2021. “CoppeliaSim User Manual”. *https://www.coppeliarobotics.com/helpFiles/index.html.* Accessed 01 June 2021.

Craig, J. J., 2005. “Introduction to Robotics: Mechanics and Control”. 3. edition. *Editora Pearson*. Vol. 1.

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.