How To Create Our Custom Model in CoppeliaSim From 3D File

Sudip Chakraborty [1] & P. S. Aithal [2]

[1] D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India,

OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in

[2] Vice Chancellor, Srinivas University, Mangalore, India,

OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

Subject Area: Computer Science.
Type of the Paper: Experimental Research.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed In: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.8117666
Google Scholar Citation: IJAEML

How to Cite this Paper:
Chakraborty, S., & Aithal, P. S. (2023). How To Create Our Custom Model in CoppeliaSim From 3D File. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *7*(2), 164-174. DOI: https://doi.org/10.5281/zenodo.8117666

International Journal of Applied Engineering and Management Letters (IJAEML)
A Refereed International Journal of Srinivas University, India.

Crossref DOI: https://doi.org/10.47992/IJAEML.2581.7000.0180

Received on: 02/06/2023
Published on: 30/06/2023

How To Create Our Custom Model in CoppeliaSim From 3D File

Sudip Chakraborty [1] & P. S. Aithal [2]

[1] D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India,
OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in

[2] Vice Chancellor, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

ABSTRACT

Purpose: This paper aims to provide a comprehensive guide to creating custom models in CoppeliaSim from 3D files. It equips users with the necessary knowledge and skills to design and simulate their unique robotic systems within the CoppeliaSim environment. By creating it, users can design robots with unique geometries and configurations that align with their specific requirements and research objectives. It enables users to represent real-world robots or prototypes in the virtual environment accurately. It provides a platform to test and validate robot designs before physical implementation. Users can simulate different scenarios, assess the feasibility of their concepts, and identify potential issues or improvements early in the design process, thereby saving time and resources. It equips students, researchers, and robotics enthusiasts with practical knowledge and hands-on experience designing and simulating custom robot models. It also encourages collaboration and knowledge sharing within the robotics community.

Design/Methodology/Approach: Download the desired model for Simulation from the website. Import 3D model into the environment. Configures various properties of the model. This includes defining physical properties, and visual appearance, for realistic rendering. The process also determines the kinematic properties of the model. It involves creating linkages, defining joint properties, and establishing parent-child relationships between objects. Add child scripts. Save the model for future use.

Findings/Result: Here, we provide the process for importing 3D files into CoppeliaSim. Users could import their necessary models without an issue, ensuring the accurate representation of 3D object geometry in the simulation environment. Users could establish linkages, define joint properties, and set parent-child relationships, allowing for accurate kinematic simulations and motion analysis.

Originality/Value: It provides a comprehensive step-by-step guide on creating custom models in CoppeliaSim from 3D files. It covers all the essential aspects and ensures that users understand the entire process clearly, making it a valuable resource for beginners and experienced users.

Paper Type: Experimental-based Research.

Keywords: CoppeliaSim custom model, Simulation using the custom model.

1. INTRODUCTION :

CoppeliaSim, a widely-used robot simulation software, provides users with a powerful platform to design and simulate robotic systems. One of the valuable features of CoppeliaSim is the ability to create custom models, allowing users to tailor their simulations to specific robot configurations and tasks. This paper presents a comprehensive guide to creating custom models in CoppeliaSim from 3D files. By leveraging the flexibility of 3D files, users can import existing designs or create 3D models using various CAD (Computer-Aided Design) software and seamlessly integrate them into CoppeliaSim. Creating custom models in CoppeliaSim opens up a world of possibilities for researchers, developers, and enthusiasts in robotics. It enables the accurate representation and Simulation of robots with unique geometries, such as novel manipulators, specialized end-effectors, or complex mobile robot designs. This level of customization empowers users to explore and test different robot configurations, kinematics, and dynamics. They are ultimately leading to improved performance and functionality. It aims to provide a step-by-step guide that walks users through creating custom models in CoppeliaSim from 3D files. By following this guide, readers will understand the techniques and tools required to develop custom models in CoppeliaSim. This knowledge will enable them to efficiently design and simulate a wide range of robotic systems, promoting innovation and advancement.

2. RELATED WORKS :

Farley, A. et. Al. in their paper compares CoppeliaSim, Gazebo, MORSE, and Webots regarding motion accuracy for mobile robot simulation. The authors evaluate the simulators based on quantitative metrics and provide insights into their strengths and weaknesses [1]. Chakraborty, S. et al. demonstrate the use of CoppeliaSim for simulating a 3D printer. They showcase the simulator's capabilities in replicating the behavior of an actual 3D printer and discuss its potential applications in additive manufacturing [2]. Momeni, M. et al. study utilizes CoppeliaSim to automate the fabrication of reinforcement cages in a robotized production cell [3]. Chakraborty, S. et al. demonstrate forward kinematics using CoppeliaSim and the C programming language [4]. Faina, A. et al. introduce HoRoSim, a holistic robot simulator, and discusses its integration with Arduino code, electronic circuits, and physics simulations [5]. Bogaerts, B. et al. explore the connection between CoppeliaSim and virtual reality (VR) [6]. Sun, Z. et al. focus on constructing an intelligent visual coal and gangue separation system using CoppeliaSim [7]. Chakraborty, S. et al., in their paper, presents an inverse kinematics demonstration of a custom robot using C# programming language and CoppeliaSim [8]. Tursynbek, I. et al. The authors of this paper focus on the modeling and Simulation of spherical parallel manipulators using CoppeliaSim [9]. In their article, Cid, A. et al. present the development and validation of an inspection robot for confined spaces using a simulated environment in CoppeliaSim. The authors discuss how they create a realistic simulation environment in CoppeliaSim to test and validate the functionality and performance of the inspection robot before real-world deployment [10]. Overall, the literature study highlights the diverse range of applications of CoppeliaSim in robotics research. The study demonstrates the simulator's capabilities in motion accuracy, 3D printing, automated fabrication, kinematics, virtual reality integration, visual systems, and parallel manipulator modeling. CoppeliaSim is a valuable tool for researchers and developers in robotics, enabling them to simulate and test various robotic systems and algorithms before real-world implementation.

3. OBJECTIVES :

The objectives of this paper are as follows:

(1) To provide a clear, comprehensive step-by-step guide on creating custom models in CoppeliaSim from 3D files.
(2) To facilitate users' understanding of creating custom models, from importing 3D files to configuring and simulating the models.
(3) To empower users with the knowledge and skills to design and simulate their unique robotic systems in CoppeliaSim.
(4) To ensure accuracy and reliability by providing guidelines for correctly orienting and scaling the imported 3D files within CoppeliaSim.
(5) To highlight the value and potential of custom model creation in CoppeliaSim for robotics researchers, developers, and enthusiasts.
(6) To encourage further research and development in custom model creation, such as advanced geometry optimization techniques or integration with CAD software.
(7) To foster collaboration and knowledge sharing within the robotics community by providing a comprehensive resource for creating custom models in CoppeliaSim.
(8) To contribute to advancing robotics simulation research and development by equipping users with the tools and techniques to push the boundaries of robot design and simulation capabilities in CoppeliaSim.

4. APPROACH AND METHODOLOGY :

Creating custom models in CoppeliaSim from 3D files follows a systematic process that ensures accuracy, efficiency, and a comprehensive understanding of the model

- **Importing the 3D File**: Demonstrate the steps to import the prepared 3D file into the CoppeliaSim environment. Explain the options and settings related to the importation process—guide users in verifying the successful importation of the model and addressing any potential issues.
- **Configuring Model Properties**: Provide instructions on configuring various properties of the custom model in CoppeliaSim. Explain how to define physical properties to ensure accurate interaction with the environment. Guide users in setting up visual appearances, materials, and textures for realistic rendering.
- **Defining Model Kinematics**: Explain the process of determining the kinematic properties of the custom model. Guide users to create linkages, define joint properties, and establish parent-child relationships between objects. Emphasize accurately representing the robot's kinematic structure for accurate simulations.
- **Conducting Simulations**: Explain the steps to set up and perform simulations with the custom model in CoppeliaSim. Guide defining simulation parameters, such as time step, duration, and solver settings. Guide users in creating simulation scenarios to test and evaluate the behavior and performance of the custom model.

5. EXPERIMENT :

Now we do some hands-on experience. We will create a ceiling fan model here. We need to follow the following steps:

1) Install CoppeliaSim from https://www.coppeliarobotics.com/downloads

2) Create an account for free in https://free3d.com/

3) Download the model https://free3d.com/3d-model/india-ceiling-fan-586433.html. Alternatively, download from the repository link: https://github.com/sudipchakraborty/How-To-Create-Our-Custom-Model-in-CoppeliaSim-From-3D-File.git.

4) Extract the zipped folder.

5) Open CoppeliaSim. from the File menu, and navigate Import > Mesh. Select the "Ceiling Fan. obj" file.

6) We observed that one fan appeared inside the IDE, depicted in Figure 1. the two imported objects are listed under the Scene hierarchy: "**Ceiling_Fan_0**" and "**Ceiling_Fan_1**". Our raw material came to the workspace. Before the next task, we need to know some tools to execute the rest task.

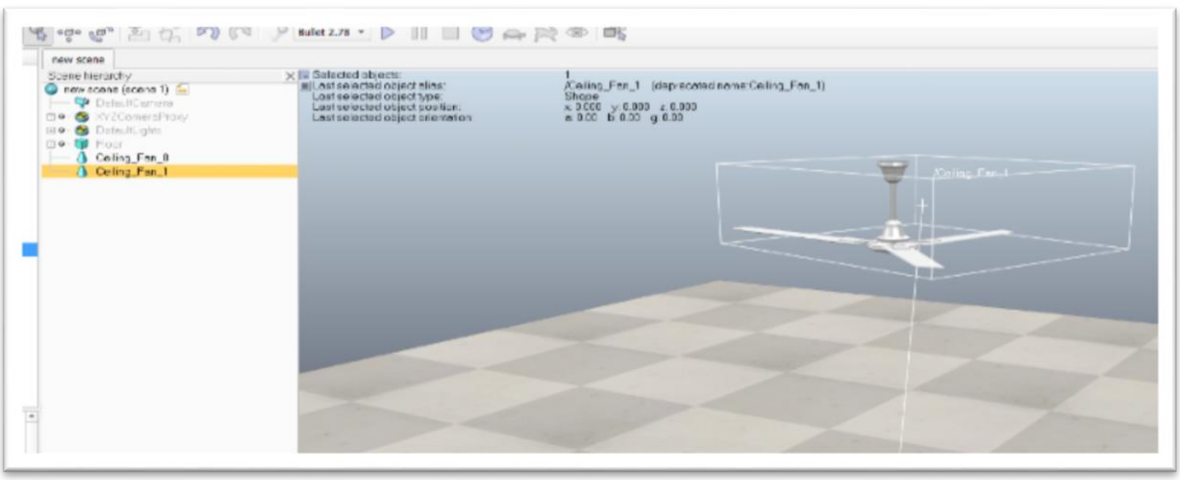

Fig. 1: the model imported into the IDE [source: Author's]

5.1 Standard operation on any object:

- **Change Object color**: to change the color of any object, follow the below steps:
    a. Click the object in the workspace or from the scene hierarchy windows.
    b. Click the "**scene object properties**" button from the left side toolbar.
    c. One window will appear. Under Visual Properties, click on the "**Color**" button.
    d. Several color options buttons are available. Change the slider position and see the preview in the workspace object in real-time.
- **Object/Item Translation/Position**: To change the position of the object, we need to follow the steps below:
    a. Select the object in the workspace.
    b. Under the top menu bar, click on the "**Object/item shift**" button.
    c. The options are available under the "Mouse Translation" tab if we want to move using the mouse.
    d. Use the "Position" or "Translation" Tab to add direct value entry.
- **Object/Item Rotation/Orientation**: To rotate the object, one tool is available. From the top menu bar, click "**Object/item rotate**."
    a. Select the object in the workspace.
    b. The three tabs are available for convenience, i.e., "Mouse Rotation," "Orientation," and "Rotation."
    c. Use the appropriate tab as we need.
- **Distance Measurement**: we can measure the distance of any object.
    a. Select the object in the workspace.
    b. Under the top menu bar, click on "**Modules**."
    c. Click on "**Geometry/Mesh**"
    d. Click on "**Measure distance/direction**."
    e. One green dot will appear. Click start and end positions.
    f. The distance and direction will be displayed at the bottom message box.
- **Group/Ungroup**: Sometimes, we must group or ungroup the selected objects. We need to follow as below.
    a. Select more than one object in the workspace or the scene hierarchy window
    b. Right-click on the selected objects.
    c. Go to the "**Edit**" menu.
    d. Click on "**Shape grouping/merging**."
    e. Under this command category, four options are available: "group," "ungroup," "merge," and "divide."
    f. These are very helpful when we work with Mesh.
- **Shape reference frame**: when we account object's kinematics properties, the reference frame is an essential parameter to care for. The world, parents, or Mesh may respect the reference frame. To execute the operation.
    a. Select object
    b. Right-click on the object. Click on edit.
    c. Under the edit menu. Click on "**Shape reference frame**." Under this menu, three options are available. According to our need, select "**relocate to world origin,**" "**relocate to parent origin,**" or "**relocate to mesh center**."
- **Object Display**:
    a. To display objects in bigger/smaller sizes, Place the mouse on the object and scroll the mouse.
    b. To move the object left/right/top/bottom, click the "Camera pan" button at the top toolbar, click on the object, and move the mouse.
    c. To view the object's different sides, click the "**Camera rotate**" button at the top menu bar. Clicking on the object moves the mouse.

Now we complete the rest task.

7) Select the object "**Ceiling_Fan_0**" and "**Ceiling_Fan_1**" from the scene hierarchy and apply the command "**divide**." It will divide into several objects depicted in Figure 2.

8) Select object Using object/item shift tools individually, identify and rename for better understanding.

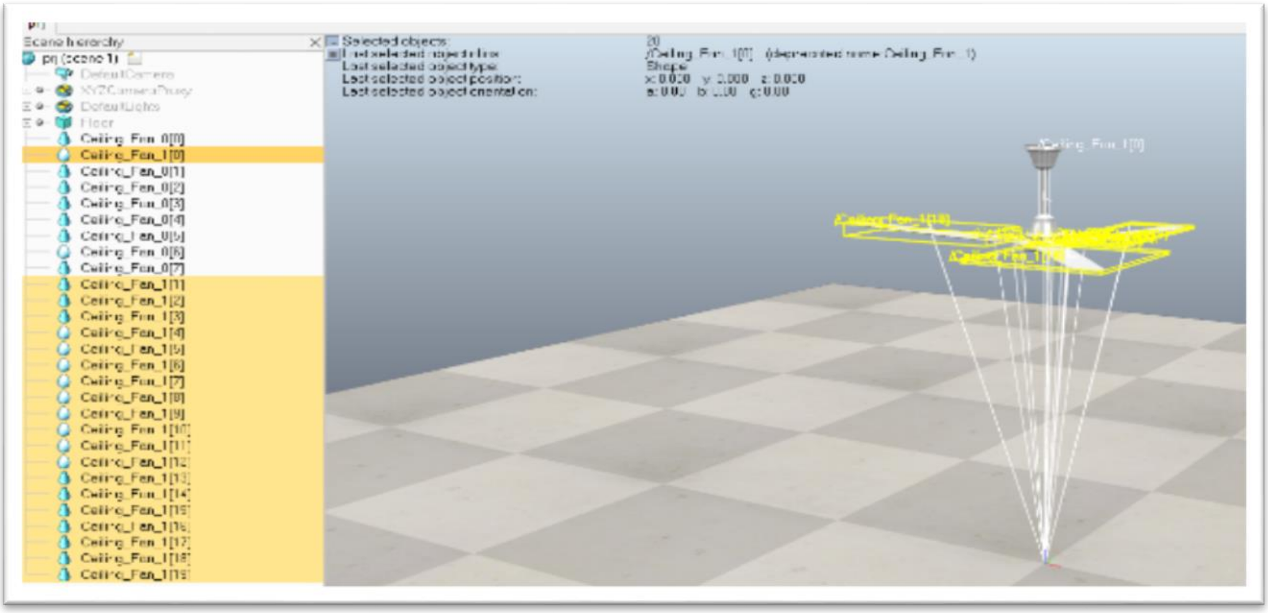

Fig. 2: the fan object is splitted into multiple objects showing inside the IDE [source: Author's]

(9) We can delete some objects which are not most relevant to our Simulation. Select the object and press the "delete" button to delete any object.

(10) We need to change the reference frame. For that, Right-click on workspace>Edit> Select all. Again, right-click> shape reference frame> **relocate to mesh center**.

(11) According to the object, drag and maintain the parent-child hierarchy depicted in Figure 3.

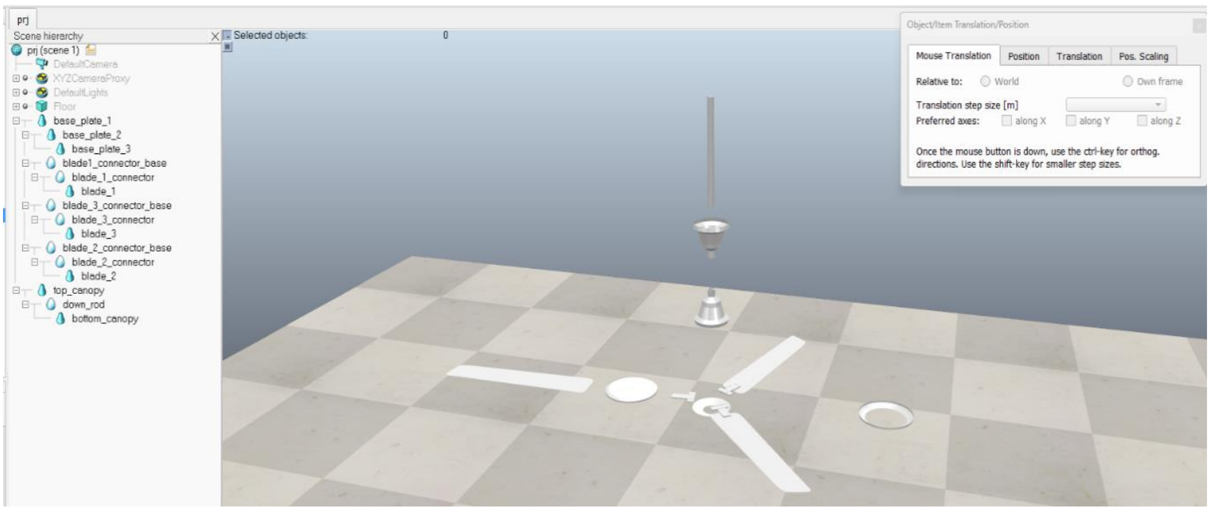

Fig. 3: the parent-child object hierarchy [source: Author's]

(12) After the top steady portion is complete, we must add a joint. At the top, Add menu > Joint > Revolute. From the left side toolbar > click the "**Scene object properties**" button. Edit visual properties, set Length [m]: 0.01, Diameter:0.02

(13) The complete list of objects, positions, and parent-child hierarchy is depicted in Figure 4.

Sl. No.	Frame	Position			Reference
		X-cord.[m]	Y-cord.[m]	Z-cord.[m]	
1	down_rod	0.00	0.00	1.00	World
2	top_canopy	0.00	0.00	0.055	down_rod
3	bottom_canopy	0.00	0.00	-0.058	down_rod
4	bearing	0.00	0.00	-0.905	down_rod
5	base_plate_2	0.00	0.00	-0.005	base_plate_1
6	base_plate_3	0.00	0.00	-0.008	base_plate_2
7	base_plate_1	0.00	0.00	0.001	bearing
8	blade1_connector_main	0.03859	0.04543	0.004	base_plate_1
9	blade_1_connector_sub	-0.005	0.026	0.002	blade1_connector_main
10	blade_1	-0.02877	0.12197	0.00756	blade_1_connector_sub
11	blade2_connector_main	-0.05566	0.03222	0.00302	base_plate_1
12	blade_2_connector_base_sub	0.01678	-0.02356	-0.00002	blade2_connector_main
13	blade_2	0.08401	-0.09382	-0.00089	blade_2_connector_base_sub
14	blade_3_connector_main	0.02548	-0.05921	0.00233	base_plate_1
15	blade_3_connector_sub	0.01601	0.02499	0.00067	blade_3_connector_main
16	blade_3	0.04714	0.11322	-0.00356	blade_3_connector_sub

Fig. 4: The newly created model [source: Author's]

(14) Change the object color as we prefer.
(15) Now we convert it into the model. Select all available fan objects. Click the "**Scene object Properties**" button from the left side. Then click on the "**Common**" tab. Under the model definition, click "**object is model**." Then click on "**Apply to selection**." We will observe that the "**Save model as**" menu is now highlighted under the file menu.
(16) Now we will export our model. In the folder path C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\models, create a folder like "MyModel."
(17) Keep selecting our model in the workspace. From the file menu>" **Save the model as**" and then click on "**CoppeliaSim model**…". One message box will appear. Click on "OK".
(18) After pressing OK, another window will appear. Click **no**.
(19) The appeared window is for the thumbnail. We can select from any file. By default, it shows as in the Workspace image. The zoom button can zoom the image. **V-Shift** for up/down and **H-Shift** for left/right for proper alignment. After adjustment, click ok.
(20) Now enter a file name and select path C:\Program Files\CoppeliaRobotics\CoppeliaSimEdu\models\MyModel. Click the "**Save**" button.
(21) Now close and open again the Coppeliasim IDE. Under the My Model menu, our model is available. Drag it into the workspace and use it. As depicted in Figure 5.

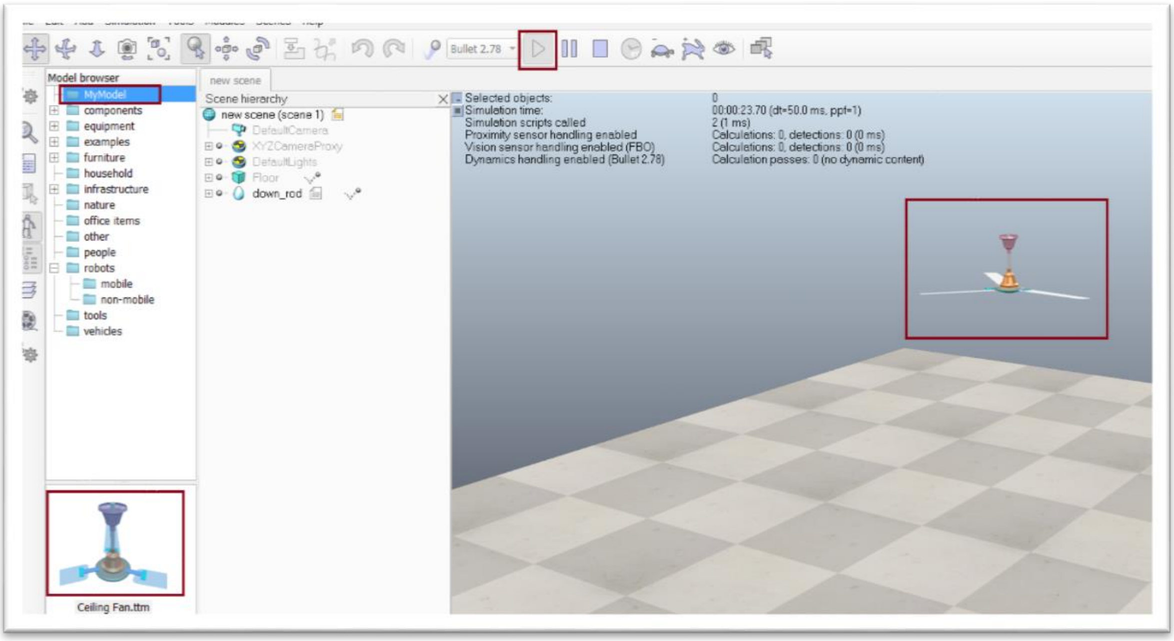

Fig. 5: the newly created model [source: Author's]

(22) Now we will rotate our model fan through code. Drag one model. Right-click on down_rod. Add> Associated child script > nonthreaded > Lua, depicted in Figure 6.

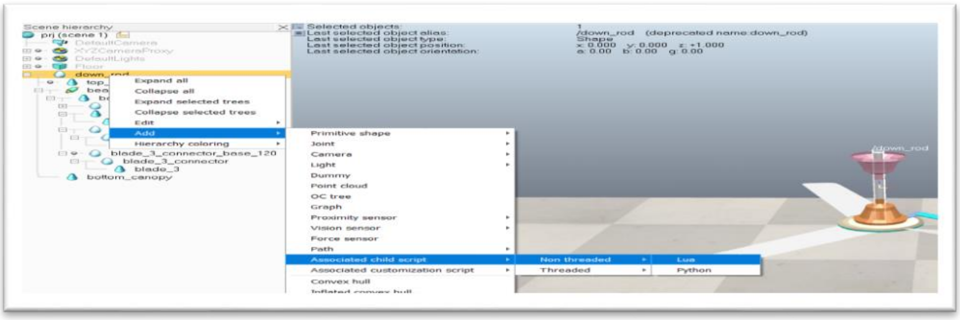

Fig. 6: the script addition to the object [source: Author's]

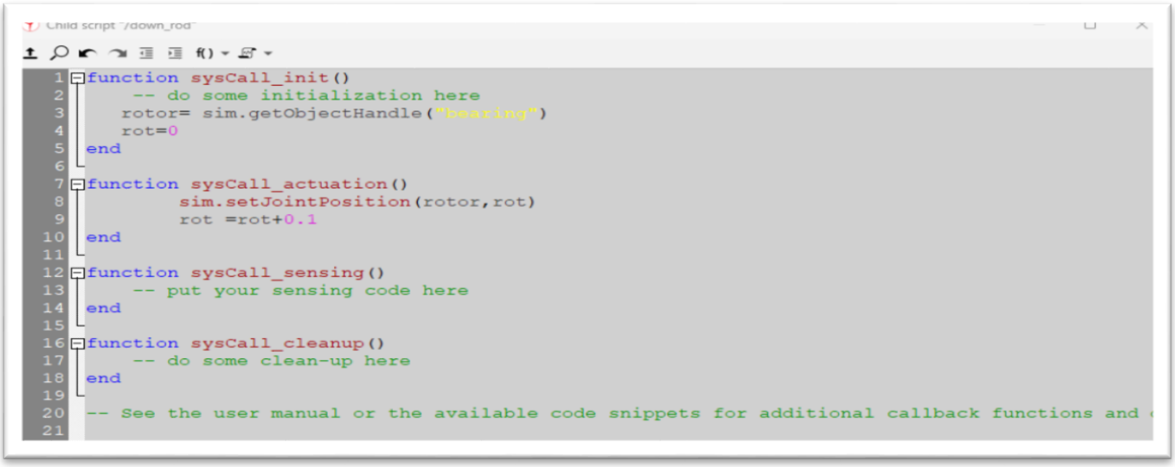

Fig. 7: the Lua script to move the fan [source: Author's]

(23) Double-click on the script icon. Add the below script. As depicted in Figure 7.

(24) We are adding the script so the fan starts moving when the simulation runs.

(25) Run the Simulation by pressing the "**start/resume simulation**" button. Observe that our model fan is rotating. As depicted in Figure 8.

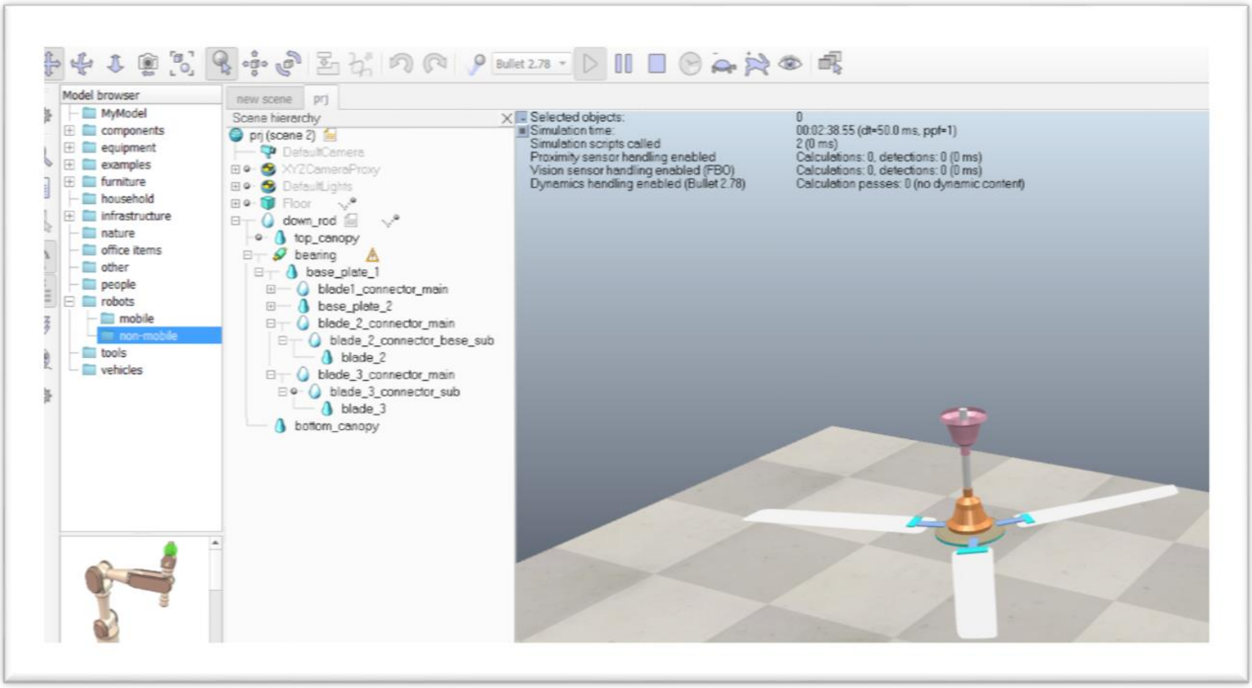

Fig. 8: the model fan is moving [source: Author's]

6. RECOMMENDATIONS :

- To be familiar with CoppeliaSim: https://www.srinivaspublication.com/journal/index.php/ijcsbe/article/view/1322/630 [11].
- The paper on Forward kinematics in CoppeliaSim: https://srinivaspublication.com/wp-content/uploads/2021/04/3.-Robot-Simulation_Fullpaper.pdf [12].
- The paper to build custom robotic arm in CoppeliaSim: https://srinivaspublication.com/wp-content/uploads/2021/04/4.-A-Custom-Robot-in_Fullpaper.pdf [13].
- The paper on Inverse kinematics in CoppeliaSim: https://srinivaspublication.com/wp-content/uploads/2021/05/7.-Inverse-5_Fullpaper.pdf [14].
- The paper to create a 3D printer in CoppeliaSim: https://www.srinivaspublication.com/journal/index.php/ijaeml/article/view/1129/574 [15].
- This described model is not as actual as accurate. This a concept. According to our project requirement, we need to make perfection.
- We tried with .obj here. CoppeliaSim can Import *.dxf, *.ply, *.stl also. Among them, .stl is the most popular.
- Several options are available when we import mesh files, and we can experiment further for better understanding.

7. CONCLUSION :

CoppeliaSim allows users to import existing designs or create models using CAD software, facilitating seamless integration. Here, we discussed the various stages of creating custom models. The presented methodology was supported by experimental findings, demonstrating the successful creation and Simulation of custom models in CoppeliaSim. Users can rely on this guide to accurately represent and test their unique robotic designs, enabling them to study kinematics, dynamics, and performance within a simulated environment. By equipping users with the knowledge and skills to create custom models, we empower them to explore new possibilities, innovate, and advance the field of robotics simulation. Looking ahead, there are several opportunities for further research and development. Enhancements to the methodology could include more advanced techniques for geometry optimization, integration with advanced CAD software, or the incorporation of machine learning algorithms for automated model generation.

REFERENCES :

[1] Farley, A., Wang, J., & Marshall, J. A. (2022). How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE, and Webots with a focus on accuracy of motion. Simulation Modelling Practice and Theory, 120, 102629. Google Scholar

[2] Chakraborty, S., & Aithal, P. S. (2022). A Simulated 3D Printer in CoppeliaSim. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *6*(1), 22-32. Google Scholar

[3] Momeni, M., Relefors, J., Khatry, A., Pettersson, L., Papadopoulos, A. V., & Nolte, T. (2022). Automated fabrication of reinforcement cages using a robotized production cell. Automation in Construction, 133, 103990. Google Scholar

[4] Chakraborty, S., & Aithal, P. S. (2021). Forward Kinematics Demonstration of 6DF Robot using CoppeliaSim and C. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *5*(1), 29-37. Google Scholar

[5] Faina, A. (2022). HoRoSim, a holistic robot simulator: Arduino code, electronic circuits, and physics. In Robotics in Education: RiE 2021 12 (pp. 256-267). Springer International Publishing. Google Scholar

[6] Bogaerts, B., Sels, S., Vanlanduit, S., & Penne, R. (2020). Connecting the CoppeliaSim robotics simulator to virtual reality. *SoftwareX*, *11*(1), 100426. Google Scholar

[7] Sun, Z., Li, D., Huang, L., Liu, B., & Jia, R. (2020, September). Construction of intelligent visual coal and gangue separation system based on CoppeliaSim. In 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE) (pp. 560-564). IEEE. Google Scholar

[8] Chakraborty, S., & Aithal, P. S. (2021). An Inverse Kinematics Demonstration of a Custom Robot using C# and CoppeliaSim. *International Journal of Case Studies in Business, IT, and Education (IJCSBE)*, *5*(1), 78-87. Google Scholar

[9] Tursynbek, I., & Shintemirov, A. (2020, December). Modeling and Simulation of spherical parallel manipulators in CoppeliaSim (V-REP) robot simulator software. In 2020 International Conference Nonlinearity, Information and Robotics (NIR) (pp. 1-6). IEEE. Google Scholar

[10] Cid, A., Nazário, M., Sathler, M., Martins, F., Domingues, J., Delunardo, M., ... & Azpúrua, H. (2020, November). A simulated environment for the development and validation of an inspection robot for confined spaces. In 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR), and 2020 Workshop on Robotics in Education (WRE) (pp. 1-6). IEEE. Google Scholar

[11] Chakraborty,Sudip, & Aithal,P. S. (2022).Conveyor Belt Speed Control Through CAN BUS in CoppeliaSim using Arduino Mega2560. *International Journal of Case Studies in Business, IT, and Education (IJCSBE)*, *6*(1), 194-201. Google Scholar

[12] Chakraborty, Sudip, & Aithal, P. S. (2021). Forward Kinematics Demonstration of 6DF Robot using CoppeliaSim and C#. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *5*(1), 29-37. Google Scholar

[13] Chakraborty, Sudip, & Aithal, P. S., (2021). A Custom Robotic ARM in CoppeliaSim. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *5*(1), 38- 50. Google Scholar

[14] Chakraborty, Sudip, & Aithal, P. S., (2021). An Inverse Kinematics Demonstration of a Custom Robot using C# and CoppeliaSim. *International Journal of Case Studies in Business, IT, and Education (IJCSBE)*, *5*(1), 78-87. Google Scholar↗

[15] Sudip Chakraborty & Aithal, P. S. (2022). A Simulated 3D Printer in CoppeliaSim. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *6*(1), 22-32. Google Scholar↗

****