

# ABB IRB 120-30.6 Build Procedure in RoboDK

Sudip Chakraborty<sup>1</sup> & P. S. Aithal<sup>2</sup>

<sup>1</sup>Post-Doctoral Researcher, College of Computer science and Information science, Srinivas University, Mangalore-575 001, India

OrcidID: 0000-0002-1088-663X; E-mail: [sudip.pdf@srinivasuniversity.edu.in](mailto:sudip.pdf@srinivasuniversity.edu.in)

<sup>2</sup>ViceChancellor, Srinivas University, Mangalore, India

OrcidID: 0000-0002-4691-8736; E-Mail: [psaithal@gmail.com](mailto:psaithal@gmail.com)

**Area/Section:** Computer Science.

**Type of the Paper:** Research Simulation.

**Type of Review:** Peer Reviewed as per [C|O|P|E|](#) guidance.

**Indexed in:** OpenAIRE.

**DOI:** <https://doi.org/10.5281/zenodo.5782759>

**Google Scholar Citation:** [IJMTS](#)

## How to Cite this Paper:

Chakraborty, Sudip, & Aithal, P. S., (2021). ABB IRB 120-30.6 Build Procedure in RoboDK. *International Journal of Management, Technology, and Social Sciences (IJMTS)*, 6(2), 256-264. DOI: <https://doi.org/10.5281/zenodo.5782759>

**International Journal of Management, Technology, and Social Sciences (IJMTS)**

A Refereed International Journal of Srinivas University, India.

CrossRef DOI: <https://doi.org/10.47992/IJMTS.2581.6012.0169>

© With Author.



This work is licensed under a [Creative Commons Attribution-Non-Commercial 4.0 International License](#) subject to proper citation to the publication source of the work.

**Disclaimer:** The scholarly papers as reviewed and published by the Srinivas Publications (S.P.), India are the views and opinions of their respective authors and are not the views or opinions of the SP. The SP disclaims of any harm or loss caused due to the published content to any party.

## ABB IRB 120-30.6 Build Procedure in RoboDK

Sudip Chakraborty<sup>1</sup> & P. S. Aithal<sup>2</sup>

<sup>1</sup>Post-Doctoral Researcher, College of Computer science and Information science, Srinivas University, Mangalore-575 001, India

OrcidID: 0000-0002-1088-663X; E-mail: [sudip.pdf@srinivasuniversity.edu.in](mailto:sudip.pdf@srinivasuniversity.edu.in)

<sup>2</sup>ViceChancellor, Srinivas University, Mangalore, India

OrcidID: 0000-0002-4691-8736; E-Mail: [psaithal@gmail.com](mailto:psaithal@gmail.com)

### ABSTRACT

**Purpose:** *Research on robotics needs a robot to experiment on it. The actual industrial robot is costly. So, the only resort is to use a Robot simulator. The RoboDK is one of the best robot simulators now. It has covered most of the popular industrial robots. Its interface is straightforward. Just open the software, download the robot as we need, and start experiments. Up to that, no issue was found anywhere. However, the problem begins when we want to build the simulated robot by own. Lots of complexity arises like coordinate assignment, rotation not aligned, length mismatch, robot not synced with DH parameter. We begin to find some documents for making the robots. A few bits of the document are present. That is why we research it. After doing that, we prepared this paper for the researcher who wants to develop the simulated robot independently. This paper can be referenced for them. To minimize the complexity of our research, we study an industrial robot, ABB IRB 120-30.6. It is a good and popular robot. It is six degrees of freedom robot. We will use the specification and STEP file from their respective website and build a simulated robot from the STEP file for our research purpose.*

**Design/Methodology/Approach:** *We will create a simulated robot from ABB IRB 120-30.6 STEP file. To create a robot by own, we took the help of the IRB 120 robot model. To demonstrate as simple as possible, we start with that robot whose default design is already present. We match and tune the joint coordinate based on robot parameters through this experiment.*

**Findings/results:** *Here, we see how to create a custom robot. Using the IRB 120 robot model, we will create a robot model step by step. Furthermore, it will move it around its axis.*

**Originality/Value:** *Using this experiment, the new researcher can get valuable information to create their custom robot.*

**Paper Type:** *Simulation-based Research.*

**Keywords:** Custom Robot, RoboDK, Custom Robot in RoboDK, ABB IRB 120-30.6, Robot Simulator, 6 DoF Robot ARM in the simulator.

### 1. INTRODUCTION :

The robot simulator helps us to see the outcome of the process or algorithm in a safe environment. For this purpose, we need a good robot simulator [1-7]. The simulator should provide a user-friendly interface. So that with a few efforts, we can operate the simulator. Another important feature should be calculation perfection. The simulator should consistently perform as naturally as possible. We started searching for the best robot simulator for our research purpose. We found some simulators which are the best for mobile and some for robotic arms but take substantial computational resources. The simulator should be as lightweight as possible to run the software even in a low-resource system. After lots of studies, we decide to use The RoboDK robot simulator. It has lots of advantages and a good tutorial collection over the net. Its interface is straightforward. The forward and inverse kinematics interface is available on the user interface. The main feature of the IDE is popular industrial robots available in its online library. It provided API to integrate with it.

Usually, we work with standard industrial robots. No issue creates when we work with the built-in robot. Because the company's programmer creates the robot model. We face many problems like a coordinate not matching, not being aligned with the axis, arm rotation is not linear, DH parameter configuration is not in proper order, and many more. Then our secondary target becomes the primary

target. Instead of our actual research, we engage long times to solve these issues. Here we will see through the step-by-step procedure to build a robot model from the STEP file. We selected IRB1200 from ABB.

The robot we selected here is already available in the RoboDK in their online library. We can download and experiment on it. So, the question is why this paper is for. Because it is the foundation to build a custom robot when we need it. Using the readily available robot, we are unaware of what procedures and activities are playing behind the scene. We cannot understand how it is made using the DH parameter if we download and use it. When we need to create a custom robot and experiment on it, that may take some learning curve. Not only that, building a perfect robot model is a tricky job. That is why this paper is for. We will be familiar with the robot model through a known robot model, and later, we can create our own [1-7].

## **2. RELATED WORKS :**

N. Bezzo et al., in their article, describes the main challenges, problems, vision, and future steps on the topic of codesign and rapid fabrication of robotic systems [8]. In their paper, X. Wang et al. present the simulator and list some methods to enhance dynamic simulation efficiency, which can be used in other simulator research [9]. J. K. Archibald and R. W. Beard researched soccer robots [10]. Paul A. Braun et al. describe an overview of the original and enhanced configurations, describe the modifications performed to integrate the single board computer into the vision system, and analyze overall design goals [11]. L. Sanneman et al., in their paper, presents a multi-robot system capable of running autonomously or under user control from a simple graphical interface. Over 100 origami flowers are actuated with LEDs and printed pouch motors and are deployed in a modular array around additional swimming and crawling folded robots [12]. M. Johnson et al. have developed a framework to provide a dynamic regulatory system for supporting coordination in human-robot teamwork [13]. F. M. Grätz discussed how the FDT/DTM technology could integrate peripheral devices into a robot controller. The entire process from creating an initial FDT projection during the production of a robot controller to the adaption by robot users to their specific needs is covered [14]. Trovato G. et al. introduce skeuomorphism into robotics and present one first application: the development of the DarumaTO social robot. DarumaTO is inspired by the traditional Buddhist and Shinto doll called Daruma, and it has 2 degrees of freedom. It can communicate through visual tracking, voice, and facial expression—the design results in a familiar device to a Buddhist or Shinto older adult [15]. L. Horvath et al., in their paper, outlined robot assembly process planning. Furthermore, proposed an integrated model with extended application of the feature principle [16]. R. LeGrand et al., in their paper, demonstrate the relevancy of XBC. They describe the system, its capabilities, and some potential applications on XBC [17].

## **3. OBJECTIVES :**

The main objective of this research is to provide information on how to create a custom robot in RoboDK. We took help from ABB IRB 120-30.6 robot parameter and step file to execute this research. Over the internet, some little bit information on to build a custom robot. So, we start to research to do it so that our researcher gets some valuable information on it. In the introductory phase in robotics arm research, the researcher can get some helpful information on building a robot or what is behind the scene in a robot.

## **4. APPROACH AND METHODOLOGY :**

Here we will describe the methodology of our research work. In the RoboDK IDE, we will build the ABB IRB 120-30.6 robot to make a robot in the RoboDK simulator. At first, we will download the STEP file from ABB Website. They provide the drawing file. We can also start from a different vendor. After downloading the STEP drawing file, we will import it into the simulator. The two types of files we get from the web. One is the object in the ordered form of an actual robot structure. Another one is disordered. Using this file, we need to rearrange, and then it can be helpful for our use. we will experiment with the ordered file. It is easy to build. After importing the object file, we change the name of every object. Some object we delete which is not relevant in the simulator. All things assigned as L1, L2 like that. Then we add a reference frame and name it “world” Then, we put all objects under the reference frame in the world frame. Dragging the object name, we rearrange it into order. L1, L2, L3, L4, L5, and L6.

Using the model mechanism, we convert the piece of objects into a robot model. To process the robot model, we need the DH parameter of the robot model, which is downloaded from the robot model vendor. From the ABB website, we can download the robot specification. In the RoboDK, we have to enter the DH parameter value into the field from the mechanism. After completing it will create a robot model. Double-clicking on it, the user interface will open. Using the interface, we can operate the robot. By changing the joints slider, we can observe the result. When the robot creates all joint limit by default is -200 to 200. From the parameter button, we open the Robot parameter windows. In the joints limit field, we enter joints maximum and minimum value available from the vendor’s website. After completing the entry, the joint limit cannot go beyond the limit when moving the slider.

Now we will see the DH parameter of ABB IRB 120-30.6 industrial robot. We need to open RoboDK. It will open with a blank station, named “New Station(1). At the top portion of the window, under the menu bar, then the globe icon toolbar is “Open online library.” Which is depicted in figure 1.

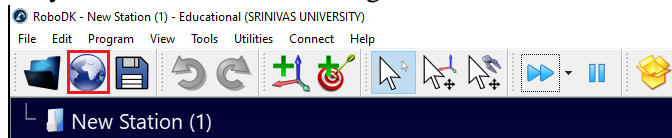


Fig. 1: Ordered STEP File

If we click the button, our window will split into two-part. Right side windows take place the station windows, and left side filled Robot Library, and we see the lots of robots are visible, depicted in figure 2. The first robot is ABB IRB 120-3/0.6. Under this robot, we need to press the Download button. If we press the download button, we can see the selected robot is placed into the right-side windows. Now we can close the Robot Library using the top-right close button.

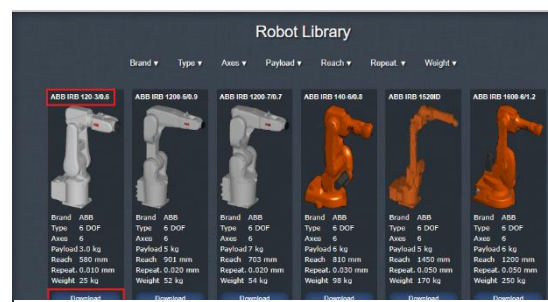


Fig. 2: Robot to download

If we double-click on the robot displayed on the workstation, a panel will open on the right side. From the panel top right, there is the “Parameter” button. We click on it. Then one Robot parameter window will open. In this window, we press “Export Table.” One small window will appear. That looks like figure 3. Now we will build the robot from scratch, and this parameter will help us to build it. This is our reference data to be used.

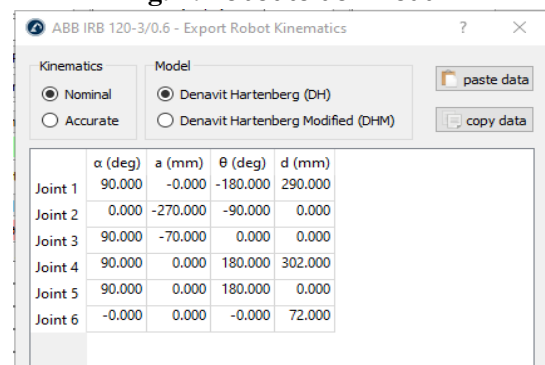


Fig. 3: DH Parameter

## 5. EXPERIMENT :



Fig. 4: File to download

Now, let us start to build our robot. We need to open the link <https://new.abb.com/products/robotics/industrial-robots/irb-120/irb-120-cad>. Now we click on the red marked file “STEP, IRB 120,3-58, IRC5”. One zip file will download into the download folder. After unzip the file, we navigate the folder “C:\Downloads\ IRB120\_3-58\_irc5\_rev01\_step\_j\ IRB120\_3-58\_irc5\_rel1\_step\_j\ STEP Files\ IRB120\_3-58\_IRB5\_rev1\_STEP\_j”. the download folder may vary from system to system. Inside the folder, we get the files depicted in figure 5.

IRB120\_3-58\_irc5\_rev01\_step\_j > IRB120\_3-58\_irc5\_rel1\_step\_j > STEP Files > IRB120\_3-58\_IRB5\_rev1\_STEP\_j

Name	Date modified	Type	Size
IRB120_3-58_IRC5_Base_rev1	01-09-2016 10:52	STEP 3D file	4,384 KB
IRB120_3-58_IRC5_Link1_rev1	01-09-2016 10:53	STEP 3D file	1,802 KB
IRB120_3-58_IRC5_Link2_rev1	01-09-2016 10:53	STEP 3D file	1,140 KB
IRB120_3-58_IRC5_Link3_rev1	01-09-2016 10:54	STEP 3D file	3,164 KB
IRB120_3-58_IRC5_Link4_rev1	01-09-2016 10:56	STEP 3D file	2,282 KB
IRB120_3-58_IRC5_Link5_rev1	01-09-2016 10:56	STEP 3D file	1,254 KB
IRB120_3-58_IRC5_Link6_rev1	01-09-2016 10:57	STEP 3D file	80 KB

Fig. 5: List of STEP Files

We need to download the product specification file depicted in figure 6

Downloads for IRB 120

Available documents: [Advanced search](#) [Documents in all languages](#)

- Show all (36)
- Popular documents (5)
- Brochure (2)
- Data sheet (2)
- Movie (26)
- Presentation (2)
- Press release (3)
- Technical specification (1)

	<b>IRB 120, M2004, Product specification</b> Summary: IRB 120, M2004, Product specification Technical specification - English - 2021-06-01 - 3,21 MB	PDF
	<b>IRB 120, data sheet, PDF</b> Summary: IRB 120 data sheet, PDF-file, A4-format Data sheet - English - 2019-11-11 - 0,39 MB	PDF
	<b>DecoRobot - Robotic Decoration Cell</b> Summary: Robotics Modular solutions for artistic artefacts Data sheet - English - 2019-09-17 - 2,19 MB	PDF
	<b>ABB Robotics Product Range Brochure 2019, 4-pages, PDF</b> Summary: No summary available Brochure - English - 2019-07-10 - 1,33 MB	PDF

Fig. 6: Robot specification file

Open RoboDK. From the File menu, click Open Menu. Select all STEP files and click open. It will take around 40 seconds(depending on the system). Figure 7 depicts the STEP files loaded into the RoboDK.

Now from the top file list, we need to do the following operation. To rename any file, select the file first, then right-click and Rename it.

IRB120\_3-58\_IRC5\_Base\_rev1>base,  
 IRB120\_3-58\_IRC5\_Link1\_rev1>L1,  
 IRB120\_3-58\_IRC5\_Link2\_rev1>L2,  
 IRB120\_3-58\_IRC5\_Link3\_rev1>L3,  
 IRB120\_3-58\_IRC5\_Link4\_rev1>L4,  
 IRB120\_3-58\_IRC5\_Link5\_rev1>L5,  
 IRB120\_3-58\_IRC5\_Link6\_rev1> L6

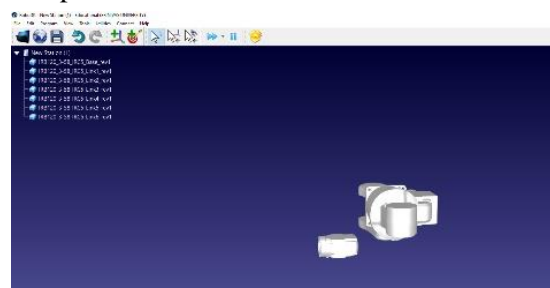


Fig. 7: After Loading STEP File

Now we need to add a reference frame. Press the “Add a Reference Frame” button—Right-click on the newly created frame and select rename from the drop-down list. We can set any name. Here we suggest the title “World.” Under the “World” frame, drag all the objects and arrange them in the order depicted in figure 8.



For the robotic arm to define the arm arrangement, we use two types of the convention. DH and modified DH. DH convention, which was introduced by Jacques Denavit and Richard S. Hartenberg. According to their tradition, coordinate frames are attached to the joints between two links. One transformation is associated with the joint, [Z], and the second is linked [X]. The coordinate transformations along a serial robot consisting of  $n$  links form the kinematics equations of the robot. The modified DH convention is a little bit different. The modified DH parameters are the locations of the coordinates system attached to the links and the order of the performed transformations. Figure 9 depicts the modified DH parameter.



Fig. 8: Object Hierarchy

Now we will create a robot model from the ordered object. To set the DH parameter, we need to extract the information which is depicted in figure 10.

ABB IRB 1200-7/0.7 - Export Robot Kinematics

Kinematics Model:  Nominal  Denavit Hartenberg (DH)  Denavit Hartenberg Modified (DHM)

	$\alpha$ (deg)	$a$ (mm)	$\theta$ (deg)	$d$ (mm)
Joint 1	0.000	0.000	0.000	399.000
Joint 2	-90.000	0.000	-90.000	0.000
Joint 3	0.000	350.000	0.000	0.000
Joint 4	-90.000	42.000	0.000	351.000
Joint 5	90.000	0.000	0.000	0.000
Joint 6	-90.000	0.000	180.000	82.000

Fig. 9: DHM Parameter

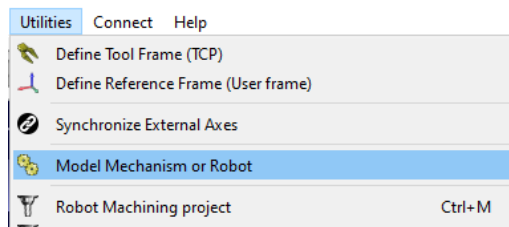


Fig. 11: Menu for Robot Creation

It is a six-axis industrial robotic arm. The orientation and the arm length will be put into the model robot field. In the menu bar under the “utilities” menu, one option, “Model Mechanism or Robot,” is available, " is depicted in figure 11.

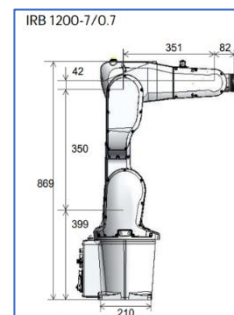


Fig. 10: Robot Configuration

When we click, one window will open, as depicted in figure 12, where we can put the value seen in Figure 10. we need to fill up  $d_1$ ,  $a_2$ ,  $d_2$ ,  $a_3$ ,  $a_4$ ,  $d_4$ , and  $d_6$  fields. After filling the field, click the OK button.

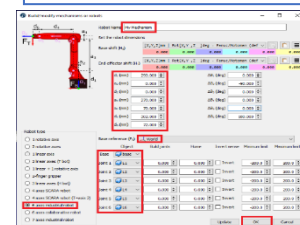


Fig. 12: Robot Configuration Entry

Figure 13 depicts our newly created robot. The left side robot is just collecting the objects, and the correct side mode is a complete robot model. By default, its assigned name is My Mechanism. To change the name or save it as a robot model, Right-click on the robotic arm from the drop-down list and click on save. Then put a proper name for it. Now to test our robot. Double click on the robot. On the right-hand side of the RoboDK Mechanism, the panel will appear. To see the forward kinematics change the joint slider and observe, the joint rotates according to our joint value.

We are observing that our robot’s all joints are rotating from -200 to 200. It is the default assigned angle. Now we have to comply with the IRB1200 model. We can extract the joints limit from the robot specification datasheet. Figure 14 depicts the joint limits value of the robot.

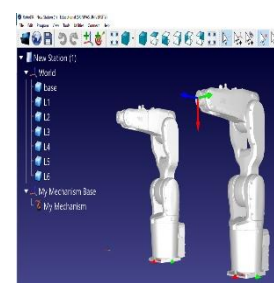


Fig. 13: Created Robot Model

The “Robot Parameter” Window will open when we click the parameter button, depicted in figure 15. seeing figure 14, the Joints value needs to fill up into J1, J2, J3, J4, J5, J6 field in figure 16.

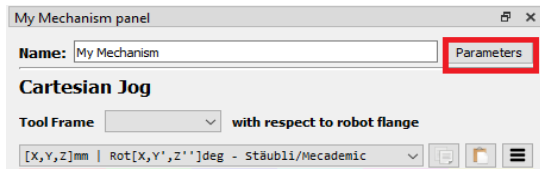


Fig. 15: Parameter Button

Location of motion	Type of motion	Range of movement
Axis 1	Rotation motion	+165° to -165°
Axis 2	Arm motion	+110° to -110°
Axis 3	Arm motion	+70° to -110°
Axis 4	Wrist motion	+160° to -160°
Axis 5	Bend motion	+120° to -120°
Axis 6	Turn motion	+400° to -400° (default) +242 revolutions to -242 revolutions maximum

Fig. 14: Axis Rotation Configuration

After all joints value entry, we need to press the OK button. Now we observe that the joints slider limit has changed. Directly changing the joints slider, we can see the result.

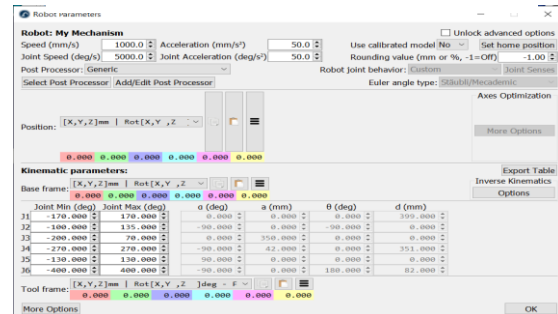


Fig. 16: Joints Limit Entry

## 6. RECOMMENDATIONS :

We provide some information here to execute the research.

- To know more details about IRB 1200, <https://new.abb.com/products/robotics/industrial-robots/irb-1200>
- For the step file, we can get from “<https://new.abb.com/products/robotics/industrial-robots/irb-1200/irb-1200-cad-models>”. Navigating the link, we need to select tab 7kg-0,70m, click on the “STEP, IRB 1200,7-70, IRC5, Joint” file. In figure 17 depicts the model and name of the download file.
- Download link for RoboDK: <https://robodk.com/download>
- After building the model successfully, we can experiment on different robot models. uArm, Dobot Magician. Niryo One etc. These robots are used for teaching and learning purposes. Compared to others, the cost is not too high. These models are also available in the RoboDK online library. Before purchase, we can get confidence using the simulated model.
- By changing the step file, we can customize it based on our requirements.

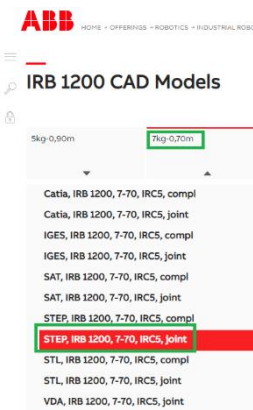


Fig. 17: STEP file

## 7. CONCLUSION :

For robotic arm research, we need a robot to experiment on it. Nevertheless, an industrial robot is too costly. Generally, individual researchers, most of the time, cannot afford it. So, they resort to the Robot simulator. RoboDK is one of the best robot simulators now. Its interface is user-friendly, and most of the standard robots are available in their online library. Forward and inverse kinematics operation is pretty simple. We experimented with the IRB1200 Robot model in this research, and step by step, and we built the model and made it workable. Following the procedure, we can quickly construct our custom robot for our research purpose.

## REFERENCES :

- 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](#)
- Chakraborty, S., & 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](#)

- [3] 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](#)
- [4] Chakraborty, S., & Aithal, P. S. (2021). Demonstration of Modbus Protocol for Robot Communication Using C#. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(2), 119-131.  
[Google Scholar](#)
- [5] Chakraborty, S., & Aithal, P. S. (2021). Demonstration of Drawing by Robotic Arm using RoboDK and C. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(1), 153-158.  
[Google Scholar](#)
- [6] Chakraborty, S., & Aithal, P. S. (2021). Forward and Inverse Kinematics Demonstration using RoboDK and C#. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(1), 97-105.  
[Google Scholar](#)
- [7] Chakraborty, S. & Aithal, P. S. (2021). Terminal++ for Robot Researcher Using C#. *International Journal of Applied Engineering and Management Letters (IJAEML)*, 5(2), 175-182. DOI: <https://doi.org/10.5281/zenodo.5773848>.  
[Google Scholar](#)
- [8] Bezzo, N., Mehta, A. Onal C. D. and Tolley, M. T. (2015). Robot Makers: The Future of Digital Rapid Design and Fabrication of Robots. *IEEE Robotics & Automation Magazine*, 22(4), 27-36. DOI: 10.1109/MRA.2015.2482838.  
[Google Scholar](#)
- [9] Wang, X, Zhu, Y. and Zhao, J. (2013). A dynamic simulation and virtual evolution platform for modular self-reconfigurable robots. *IEEE International Conference on Information and Automation (ICIA)*, 2013, pp. 457-462, DOI: 10.1109/ICInfA.2013.6720342.  
[Google Scholar](#)
- [10] Archibald, J. K. and Beard, R. W. (2002). Competitive robot soccer: a design experience for undergraduate students. *32nd Annual Frontiers in Education*, 2002, pp. F3D-F3D, DOI: 10.1109/FIE.2002.1158192.  
[Google Scholar](#)
- [11] Paul A. Braun, Randy H. Moss, Paul D. Stigall, (1986). The single-board computer: A practical alternative to a custom-designed microsystem. *Computers & Electrical Engineering*, 12(3-4), 109-118. DOI: [https://doi.org/10.1016/0045-7906\(86\)90003-0](https://doi.org/10.1016/0045-7906(86)90003-0).  
[Google Scholar](#)
- [12] Sanneman, L. *et al.*, (2015). A Distributed Robot Garden System. *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6120-6127, DOI: 10.1109/ICRA.2015.7140058.  
[Google Scholar](#)
- [13] Johnson, M., Feltovich, P. J., Bradshaw, J. M. and Bunch, L. (2008). Demonstrating Human-Robot Coordination through Dynamic Regulation. *IEEE Workshop on Policies for Distributed Systems and Networks*, 2008, pp. 231-232. DOI: 10.1109/POLICY.2008.37.  
[Google Scholar](#)
- [14] Grätz, F. M. and Waschhauser, R. (2016). Extending a robot controller's capabilities with FDT. *55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 738-743. DOI: 10.1109/SICE.2016.7749215.  
[Google Scholar](#)
- [15] Trovato, G. *et al.*, (2019). The creation of DarumaTO: a social companion robot for Buddhist/Shinto elderlies. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 606-611. DOI: 10.1109/AIM.2019.8868736.  
[Google Scholar](#)



- [16] Horvath, L., Rudas I. J. and Bito, J. F. (2002). Form feature-based generation of robot assembly paths for product variants. IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02., pp. 181-186, Vol.1, DOI: 10.1109/ICIT.2002.1189887.  
[Google Scholar](#)
- [17] LeGrand, R., Machulis, K., Miller D. P., Sargent, R. and Wright, A. (2005). The XBC: a modern low-cost mobile robot controller. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3896-3900. DOI : 10.1109/IROS.2005.1545569.  
[Google Scholar](#)

\*\*\*\*\*