DIGITAL TWIN MODEL OF TWO-ARM COLLABORATIVE ROBOT FOR HUMAN ARMS MOTION SIMULATION USING REVERSE ENGINEERING

By

Mohammad Aman Ullah Al Amin

Seong Dae Kim
Associate Professor of Engineering Management
University of Tennessee at Chattanooga
(Chair)

Hyunsoo Lee
Professor, Industrial Engineering
Kumoh National Institute of Technology
South Korea
(Committee Member)

Erkan Kaplanoglu
Associate Professor of Mechatronics
University of Tennessee at Chattanooga
(Committee Member)

Serkan Varol
Assistant Professor of Engineering Management
University of Tennessee at Chattanooga
(Committee Member)

DIGITAL TWIN MODEL OF TWO-ARM COLLABORATIVE ROBOT FOR HUMAN ARMS MOTION SIMULATION USING REVERSE ENGINEERING

By

Mohammad Aman Ullah Al Amin

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial
Fulfillment of the Requirements of the Degree
of Master of Engineering Management

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

August 2021

ABSTRACT

This study proposes a digital twin (DT) model for a two-arm collaborative robot that can be deployed to simulate human arm motions using the reverse engineering process. A collaborative robot named ABB Yumi – IRB14000 was considered for this study. The purpose of the experiment was to find the best version of the digital twin model by applying translation and rotation constraints in every part of the CAD model of the robot. After adding features to the robot part files, Virtual Reality Modeling Language (VRML) format files were being created to assemble it in 3D world Editor for DT formation and a grid layout was created that contained the control panel of the collaborative model digital twin to connect it with the real world. Finally, a cyber-physical system (CPS) interface was built to replicate human motion. Deep reinforcement learning will be implemented using these two models for human motion simulation.

DEDICATION

This research work is dedicated to my parents.

ACKNOWLEDGEMENTS

First and foremost, I thank the Almighty Allah for providing me an absolute opportunity and peaceful life amidst the covid-19 situation to pursue a master's degree in Digital Twin (DT) Modeling using CAD software and programming platform at the University of Tennessee at Chattanooga (UTC). My deepest appreciation and humbleness is to Him. I am also grateful to my parents and my family members for supporting me in pursuing my master's degree without any difficulties. Their mental and financial support let me go through this difficult path without much anxiety.

I am also grateful to Allah for having Dr. Seong Dae Kim as my academic advisor and coordinator. Dr. Seong Dae Kim is a nice and humble person with deep knowledge of advanced technologies in engineering management, and I am thankful to him for his supervision, support, guidance, constructive criticism, and encouragement. Without his proper guidance, I do not think my graduate study would be so meaningful and smooth. Also, my gratitude reaches out to Dr. Hyunsoo Lee, who has helped me from the beginning of my research. With his insightful direction, I was energized to walk on the road that I did not think I could conclude. He helped me in every way, from the concept building to the finishing point, to materialize my work along with my advisor. Special thanks to Dr. Erkan Kaplanoglu for his enormous support and for teaching me details about the robot operation, & Dr. Serkan Varol for their intellectual support and constructive criticism of my thesis manuscript. Moreover, my gratitude goes to Dr. Lani Gao for her advice and excellent recommendation to pursue my Ph.D. study.

I want to thank Professor Ahad Nasab, Head of Engineering Management and Technology department of the University of Tennessee at Chattanooga (UTC), for considering me to have funding support and lab resources. These were used for the Digital Twin Model formulation and for providing me additional facilities.

I would like to thank Ms. Shanae Anderson, Ms. Lora Cook, and Ms. Bonny Clarke, and other staff of the Engineering Management & Technology Department and Graduate Office for their enormous support to solve the administrative issues at UTC.

I am grateful to my fellow roommate, Mr. Syed Mohammad Tarek, for the encouragement and constructive conversations made during almost two years of a long journey. Moreover, I give special thanks to Babatunde Atolagbe for his support on-campus as well as off-campus. Last but not least, I would be thankful to the late Dr. Richard Wilson and other Bengali Community members who made my stay in the U.S. memorable and enjoyable. Their initial support helped me settle down here in Chattanooga.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DT, Digital Twin

VRML, Virtual Reality Modeling Format

AI, Artificial Intelligence

CAD, Computer-Aided Design

IoT, Internet of Things

STEP/STP, Standard for the Exchange of Product Data

CNN, Convolution Neural Network

IIOT, Industrial Internet of Things

GPU, Graphical Processing Unit

ML, Machine Learning

DRL, Deep Reinforcement Learning

PLC, Programmable Logic Controller

HIL, Hardware-in-the-loop

SIL, Software-in-the-loop

DTWN, Digital Twin Wireless Network

RCNN, Region-Based Convolutional Neural Network

CHAPTER 1

INTRODUCTION

Digital Twin (DT) is the latest slogan that accelerated the industry 4.0 revolution. Industry 4.0 concept came to the public attention in a presentation back in 2011 by Professor Wolfang Wahlster, Director & CEO of the German Research Center for Artificial Intelligence. Industry 4.0 refers to the automation of processes and machines by intelligent networking, cyber-physical system, Internet of Things, Cloud Computing enabling smart manufacturing, process optimization, and getting more responsive to customer demand and orientation (Alcácer et al., 2019). The IoT devices are empowering the smart industries that create a new way of thinking about the automation process despite its initial high cost. The next era is about implementing machine learning algorithms in mass industries to automate the process.

Digital Twin is a compact package that is linked with the Internet of Things, Cloud Computing, Artificial Intelligence, Edge Computing, etc. that resulting to process, machine, and system automation. DT is combining operation Technology and Information Technology through augmented reality. By analyzing the data for optimal decision making of the motion as well as the interconnectivity of the machines and digital twin, an efficient system can be developed which can take quality and lean concept to a whole new level (Pech et al., 2021). With the advanced technology, Big data Analytics, IoT, the industrial revolution 4.0 is being initiated by transforming process and value creation into data.

At present, the world is about connecting the dots. The recent development of cheaper computation technology makes the Digital Twin one of the feasible iterative innovations happening around the globe. As the I4.0 revolution is about connecting components, Digital Twin is in the leading position to do it for several fields. It helps to store the data from several components, to homogenize the entire data collection and to understand the pattern and make an insightful analysis. The data can be modified and then can be tested in a virtual machine creating a unique situation.

The DT model is a digital replica of a physical model which considers a dynamic computerized unified system model that can be interacted by real-time synchronization with the basic run time of the physical model allowing simultaneous data transmission (Batty, 2018). In short, DT is the digitalized clone of the physical model. DT is the outcome of an infrastructure that combines the Internet of Things (IoT), Cyber-Physical System (CPS), and materialized Industry 4.0 concept (i-SCOOP, 2017). Many models represent the oversimplified physical model as well as the complete replication of the real model, but there are a few types of research on the two-arm collaborative robot model for process automation purposes. Aliaj worked on skin marker motion capture datasets. They worked on human motion replication considering the inertial forces, moments of high speed, and other biomechanical and orthopedic factors (Aliaj et al., 2020). Amir used Solidworks and Simulink to replicate human arm motion to the virtual model developed in Solidworks (Amir et al., 2017). They considered the Simulink model for accurate motion capture to rely on functionality control. Jones et al. (2018) used the IoT to have control over robotic arms from any place of the world. They used human arm motion data with the help of a position sensor and upload it into a cloud platform and then use it to configure the position of the robotic arm motion remotely (Jones et al., 2018). Only a few types of researches have been done on collaborative robots and their automation process. Some system used Field Programmable Gate Array to do a specific task which is customized for specific work and it takes a lot of modification to make it work for other tasks. Most of them used the same degrees of freedom for input/output.

Digital Twins are 4 types in terms of applicability which are component Twins, asset twins, system/unit twins, and process twins (Plank, Dec 16, 2019). Component Twins or part twins are considered as a single component or part of the entire system. Ex: motor, rotor, etc. Asset Twins are the combination of part twins. It contains several components and a certain task to accomplish by working together. Ex: water pump, combustion engine, etc. System twins are the next higher-level twin that contains several asset twins to make a combination of different types of work-oriented asset or all production units and to execute different actions related to them. Ex: Airplane, Spaceship, etc. The process twin combines all system twins where it represents the whole production floor or facility. Process twin represents all the other twins' functionalities, and the entire process is effective when they all work together and complete the requirements. Ex: Complete Manufacturing process (Plank, Dec 16, 2019).

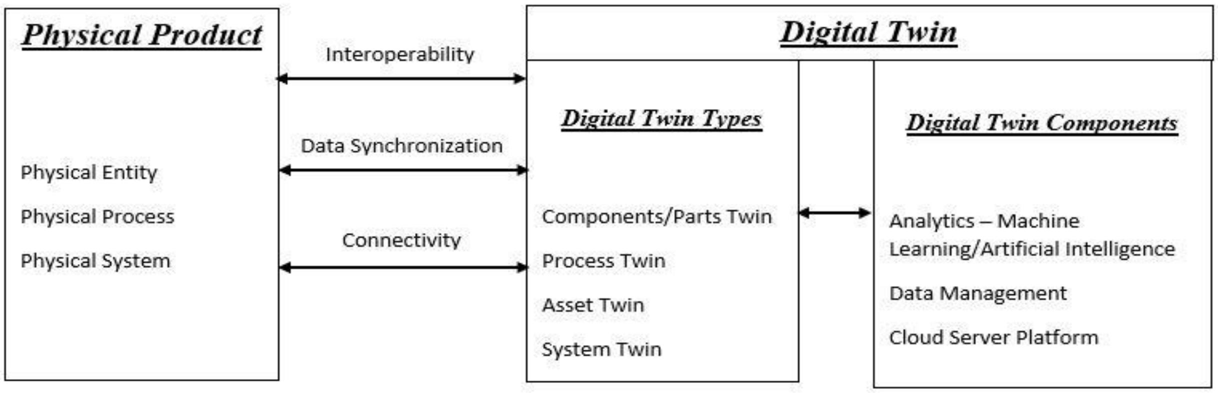


Figure 1.1 Conceptual Representation of Digital Twin

Reverse engineering means going back through the original design process. It is also called back engineering. The process involves breaking down the whole assembly into parts. It is mainly used to extract the right information from the original product to recreate it with further development and accuracy. It is used to determine the process of the machine or components recreation part by part or layer by layer. To enable reverse engineering, a product is taken under consideration to disassemble and investigate its internal mechanisms to gather the original information of the components. After gathering all necessary information, the data can be used to create the required model with the necessary modification of the 3D model as well as functionalities.

1.1 Problem Statement

The industrial sector is inclining to the automation concept day by day. The revolution of Industry 4.0 is the result of the uprising of the digital movement. Digital Twin is an old concept used for various purposes and now it is being used for developing the manufacturing system to spacecraft administration. The robots are taking over the repetitive work and makes people's life smooth. The automation process of the digital world is one of the biggest challenges in the present world. There are so many factors wherein people are trying to automate the system process as well as the components working for it.

Some of the researchers are working to find out the minimum process downtime while some others are working on data analysis of the system. Few works have been done that are related to the manufacturing process automation considering a human motion replication to improve the process over time with a collaborative robot. This research is addressing the research scope of the formation of the digital model of a collaborative robot and the implementation of deep reinforcement learning for human motion recognition and use it to automate the digital twin movement simultaneously.

1.2 Objectives of the Study

The objectives of this study are as follows:

i. Overview of Digital Twin Model and Human-Cyber Physical System (CPS) Interface.

ii. Study of features/constraint, parts orientation, and data required for creating Digital Twin.

iii. Creating Human-CPS for transferring Human motion to Digital Twin model

iv. Building a real-time applicable Digital Twin model

1.3 Scope of the Study

This research focuses on Digital Twin construction. It includes developing a Digital Twin model from the CAD file of the collaborative robot using the CAD software to convert it to a VRML file to create the Digital Twin Model in the MATLAB_3D world editor. This research is also focused on the building of the Human-Cyber Physical System interface.

1.4 Thesis Overview

This research work is divided into five chapters. In Chapter I, the digital twin concept is introduced. In Chapter II, The Literature Review is discussed. In Chapter III, the methodology of the collaborative robot formation and the 3D World Editor model assembly is discussed. In Chapter IV, the detailed procedures of VRML file creation, the rotation and translation data collection, and implementation in 3D world Editor are discussed in detail. The Application interface created by the App Designer module is also discussed here. The challenges and the future work are discussed here too. Chapter V concludes the study and provides recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

In model representation, any physical process/system is to be simplified for easier presentation. Generally, a model does not replicate the whole system. There could be several models of a single physical system but there could be only one digital twin for the entire system and that is based on its total features. So, a digital twin is used for familiarizing a system with the same functionalities as the real system and can interact with the physical system's every feature consistently, parallelly, and virtually. Models are simple/part representations of physical systems. If a model can be created in such a way that it can represent all the characteristics/features of a system can only be said, digital twin.

2.1 History of Digital Twin

Dr. David Hillel Gelernter, an American scientist-Professor of Yale University (Computer Science) gave the concept in 1991 in a book named "Mirror Worlds". This concept was first used in 2002 by Professor Michael Grieves from the Florida Institute of Technology and then publicly introduced in the "Society of Manufacturing Engineers" conference later in 2002. He talked about the origin of the digital twin concept and how we can relate the physical world with the digital world. He talked about the types of Digital Twin and the life cycle of the digital twin model (Grieves et al., 2017). The majority (55 percent) of the research that had been done on the digital twin is still in concept. 26 percent of the publications focused on case studies.

35 percent of the publications discussed digital shadows while 28 percent discussed digital twin. The digital twin with bidirectional data transfer capability is discussed in only 18 percentage of all the research which means the number of research that had been done on the digital twin application is less than 20 percent (Kritzinger et al., 2018). So, the present scenario of the research for digital is still in the conceptual area and it is turning into a more applied area. Taking these factors under consideration, I selected the research scope for my research with my current interest as well as resources.

2.2 Types of Digital Twin

The digital twin is categorized into three parts. 1. Digital Twin Prototype (DTP), 2. Digital Twin Instance (DTI), 3. Digital Twin Aggregate (DTA) (Grieves et al, 2017). DTP is related to physical system/process/entity design and analysis. It can be created before physical product existence. DTI is the digital twin of an individual part of a physical system/body. DTA is the complete addition of all DTIs' of a physical machine/system that is used for learning and predictive/consequence analysis to find the unknown pattern of data and other useful insights from DTI's aggregation (Council, 2019, Dec 2; TWI, 2021). Fuller et al. discussed the definition and misconceptions about digital twins and the difference between the digital twin, digital shadows, and digital model. They discussed the primary field for digital twin which are smart cities, manufacturing, healthcare, and the industrial implementation of it. They gathered the challenges that the digital twin concept is going to face which are aligned with data analytics challenges, IoT challenges such as IT infrastructure, useful data, privacy and security, standardized modeling, etc. They researched on history of IoT and data analytics, deep learning.

They concluded that the relationship of the IoT, data analytics, and digital twin is mutual, and the growth of digital twin depends on them (Fuller et al., 2020).

2.3 Digital Twin in Automation

Some researchers discussed the service-oriented architecture of the digital twin and the prospect in the service industry. By connecting data between the physical objects and virtual models, they proposed that it can improve the efficiency of production planning, improve production time and reduce cost (Qi et al., 2018). Rosen et al. described the autonomous system and how it is going to be changed into a digital twin system with certain conditions. They discussed the development of digital twin which are individual application (1960+) to simulation tools (1965+), then simulation-based System Design (2000+) to DT concept. They discussed autonomy and autonomous nature in detail as well as the cyber-physical production system with the conclusion is that the combination of real-life data and simulation model leads to a good prediction of the future (Vachálek et al., 2017). Some researchers constructed digital twins for Macro Manufacturing Unit (MMU) and described the basic function. It was concluded that the digital twin requires an accurate 3D model of the actual machine to build up a feasible digital twin (Lohtander et al., 2018). Tao et al. worked on a digital twin of a shop floor to create a virtual shop floor. He created a process twin for the shop floor and showed interaction and interconnection between them. They discussed the challenges they faced such as real-time interaction, variability, uncertainty and unpredictability of physical space, and continuous integration between physical and virtual spaces despite scaling discrepancies (Tao et al., 2017). Some researchers used sensory data and machine information to create digital twins. They chose it because the digital twin model can represent the physical model's characteristics such as vibration power consumption etc. They created a schematic to describe their concept (Cai et al., 2017).

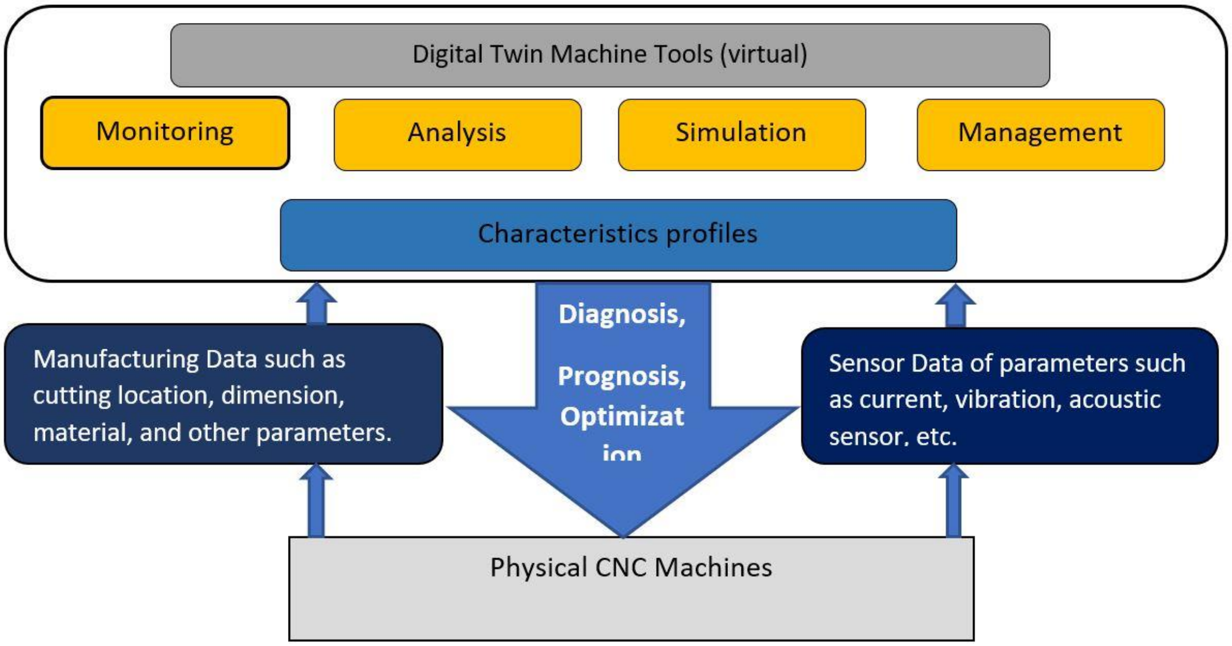


Figure 2.1 Schematic Diagram of digital twin integrated with sensory data and information (adapted from (Cai et al., 2017))

Botkina et al. created a digital twin for a cutting tool with the help of a line information system architecture which is an event-based service-oriented architecture. They also used a tweeting machine to get real-time data from CNC machine and LISA architecture for sequential structured information transfer (Botkina et al., 2018). Uhlemann et al. introduced a learning concept that showed the potential advantages of data acquisition and simulation-based data processing. He showed the comparison between digital twin and value stream mapping and concluded with a statement that indicates the digital twin models' advantages (Uhlemann et al., 2017). Schroeder et al. used AutomationML for information exchange and high-level physical object model. The combined customer information system with augmented reality was discussed (Schroeder et al., 2016). Qi et al. discussed the big data analysis and digital twin in manufacturing. They discussed how IoT improvement over the years is maximizing the possibilities of digital twin implementation. Big data analysis for smart manufacturing is getting important because of the availability of equipment data, product data, environment data, data from manufacturing information systems, etc. They made a comparison between big data and digital twin and showed a fusion concept because of the advantage of big data in terms of efficiency and the advantage of the digital twin in terms of application (Qi et al., 2018). Some researchers worked on the cloud-based cyber-physical system where the physical model is represented by a digital twin connected through the cloud (K. M. Alam et al., 2017).

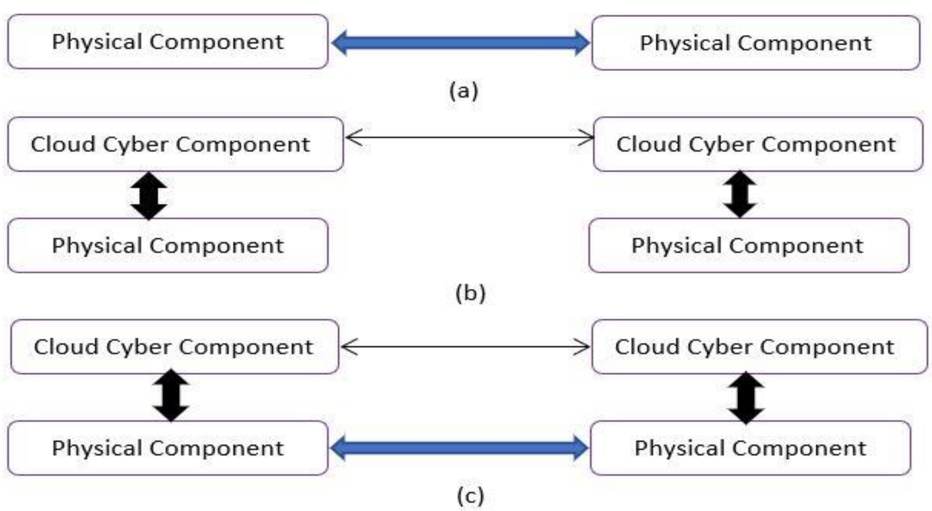


Figure 2.2 Three types of interaction between C2PS things. a) Physical-Physical b) Cyber-Cyber and c) Hybrid Cyber-Physical (adapted from (K. M. E. S. Alam et al., 2017))

Zheng et al. Discussed an application framework of a digital twin as well as the concept, characteristics, and application. They discussed the information processing layer that is combined with data mapping, data preprocessing, data analysis and mapping, and data fusion. They studied a case study on the welding production line and processed data to validate the research with virtual manufacturing concepts (Zheng et al., 2019).

2.4 Collaborative Robot Digital Twin

The collaborative robot with automation capability can work side by side for repetitive/different work with safety. It has flexible hands, parts-feeding the system, camera-based location, and motion control. The focus is on the current market demand for automotive robots which are safe by design, small footprint, fenceless installation suitability, easy to move, deploy, and redeploy, etc. It also has some other features such as flexible feeding parts management, vision-guided assembly, high accuracy, and sped effective assembly, etc. It has several customer benefits such as padded arms, portability, Lower maintenance, integrated vision, and hands, etc. It can be used for small part Assembly, collaborative assembly, accurate and fast assembly, material handling, testing, and packaging, processing. It also helps to verify the future system by enabling simulation considering new features (Schroeder et al., 2016).

2.5 Virtual Commissioning

Virtual commissioning is the process of using 3D technology to test the ability, performance, or other parameters of the physical model within a virtual environment so that it can be implemented later with the physical model. Dzinic et al. discussed the virtual commissioning approach to verify the PLC program by four steps which are 3D model development, PLC programming, simulation model, and communication between physical and digital model through SIL and HIL (Dzinic et al., 2014). They used a simulation-based verification tool for model development and a predefined model from the library which was responsible for the inconsistency among data.

2.6 Deep Reinforcement Learning

Machine learning is a branch of artificial intelligence that analyzes data automatically after being trained up with a certain number of training sets. By pattern recognition and data analysis method, it can make decisions without or less human intervention. Digital twin, machine learning, and artificial intelligence enable to development of speed, efficiency, quality, flexibility, and accuracy, etc. A digital twin is mainly used to optimize the operation procedure and provide chances to try other methods. It permits testing different methods of process and verify the options before going to actual processing. It also helps to verify the future system by considering new factors for simulation. There are three kinds of machine learning nowadays. Supervised learning, Unsupervised Learning, and Reinforcement Learning. Reinforcement learning is being used to find out the unforeseen pattern of data. We plan to use many quality data sets to train our reinforcement learning as well as Convolutional Neural Network (CNN) and then let the Digital twin construct the production optimization framework based on IoT and Machine learning considering time-series data, schedule time, etc. Klaus et al. described a variable workspace and movements of humans to address the problem related to safety. They introduced a machine learning strategy to control the robot movement to avoid collision through path planning, cluster analysis, and Artificial Neural networks (Dröder et al., 2018). Min et al. proposed a framework that used big data to train and optimize the digital twin model. The purpose of this model is to support different kinds of petrochemical manufacturing processes. They discussed the process twin and predict the controlling command by big data analysis, data correlation, and time series analysis (Min et al., 2019). Natalia researched human motion analysis with deep learning. She worked on gesture recognition hand-pose estimation, motion capture, and user validation. She used Spatio-temporal descriptors for recognizing the actions or gestures and used supervised and unsupervised methods to train the model and showed the accuracy of the methods (Neverova, 2016). Cronrath et al. discussed the implementation of deep reinforcement learning for digital twin up-gradation. They created the algorithm for digital twin and then used it on sheet metal for an experiment. They did the test with ten thousand iterations and came up with an improvement and advised on doing more iteration for more improvement depending on the number of dimensional actions (Cronrath et al., 2019). Alexopoulos et al. discussed the machine learning implementation in digital twin as the data is not going to be a problem anymore. They used machine learning and transfer learning with digital twin for adapting the system and pre-trained model from the ImageNet database and combined it with their dataset (Alexopoulos et al., 2020). They worked on the dataset of images and this paper helped me to understand the direction of my research study. Liu et al. researched a dual-arm deterministic policy gradient algorithm, a control strategy, and the effectiveness of the algorithm to make the robot certain activities such as reach, push, pick up by using Kinect camera and Mask RCNN algorithm (Liu et al., 2021).

2.7 Collaborative Robot with Deep Reinforcement Learning

Lu et al. discussed the use of blockchain technology empowered by the Federated Learning Model, DTWN model, and reinforcement learning algorithm to develop an optimized solution (Lu et al., 2020). Dai et al. research on similar kinds of research we did. They used DTN for building a network and then used stochastic computation and resource allocation and DRL for IoT. The model showed that the DRL agent took the reward and gave the feedback as an action to the digital twin (Dai et al., 2020). Ghadirzadeh et al. used a motion-capture suit to analyze the motion and to pick an object from the box. They improved robot action decision-making, enabling robots to take the optimized decision and let robots learn from the sensor data (Ghadirzadeh et al., 2020). Most of the research had been done with the deep reinforcement learning for process twin. Some of the researchers did work on the collaborative robot with a motion sensor. Convolutional neural network implementation had been done on image recognition and the creation of digital twin based on these approaches recently. There was no work been done which is combining all these concepts. So, I came with the idea of using the deep reinforcement learning algorithm to let the collaborative robot learn from the human motion and implement it to execute activities done by the human.

2.8 Reverse Engineering

Brian Hess Described reverse engineering, its process, and the purpose of reverse engineering. He discussed the 3D drawing creation of the older components which were created without 3D CAD files or digital file storage. To create a modern archive of the older product, a company can use a reverse engineering concept for this purpose. He discussed the purpose of reverse engineering which are legacy part replacement, parts service or repair, failure analysis, parts improvement, and diagnostics and problem-solving (Hess, 2019). M. Ayani et al. discussed the machine reconditioning and retrofitting, and the complication related to technical documentation. Outdated documentation or absence of it demands the reverse engineering process for executing the reconditioning project. They discussed the importance of Digital Twin and virtual commissioning to automate the system for retrofitting and physical reconditioning (Ayani et al., 2018)

The existing most similar framework was about using Deep Reinforcement Learning (DRL) to improve time-efficiency and effective decision making under uncertainties. They used graph convolutional networks (GCNs) and recurrent Q-learning (Ghadirzadeh et al., 2020). They used a motion-capture suit for capturing motion data. The difference between our work was that I was not planning to use any motion capture suit and that is why I developed the model in 3D World Editor. The motion data can be transferred from human to Human-CPS through motion mapping and then convert into robot motion. Even though there were similarities in using deep reinforcement learning to train the robot for certain activities, but the learning method is different and that is why the collaborative model digital version was created to replicate the motion before applying it to the physical model.

CHAPTER 3

METHODOLOGY

A reverse engineering process had been done on the CAD file downloaded from the ABB official website. The CAD file did not contain the features required for creating the 3D model assembly. The features had been added to the drawing file after knowing the requirement for creating the Digital Twin Model through the reverse engineering process. As I had worked on the CAD file of the robot and it consists of part files and it worked on different types of the task as a unit, it was supposed to be the asset twin. The first step to develop the digital twin for a two-arm collaborative robot was to work on the CAD files. At first, the CAD files were tired to create by me, but it was time-consuming and nearly impossible to create them from the actual product within this short time. The CAD file of the collaborative robot is collected from the ABB Yumi official website. The CAD file was an assembly file and could not be used for Digital Twin Model due to its lack of features which were vital for Digital Twin Modeling. The CAD file was uploaded as Standard for the Exchange of Product Data (STEP) file in the Autodesk Inventor. The assembly file was segregated into the part file for each component. The part files had added the features required for Digital Twin construction. Then, the part files were assembled in The CAD software. After the assembly, the locations of the part files in the assembly file were kept fixed and the part files were saved as STEP files with that location. The part (STEP) files were converted into VRML files by CAD exchanger software for uploading it into MATLAB 3D world Editor.

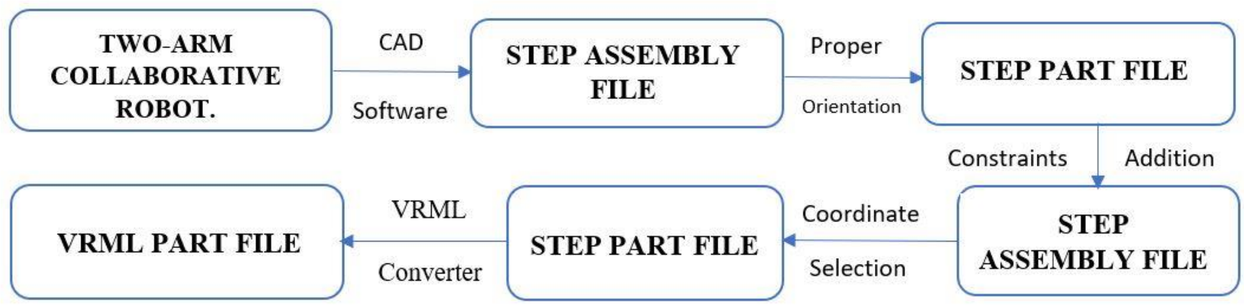


Figure 3.1 An Outline of CAD file conversion to VRML file

The VRML files were added sequentially in the 3D world editor. A 3D world editor virtual-template had been used to create an app layout to control the robot by implementing the APP Designer platform in MATLAB.

Figure 3.2 Tools for Digital Twin Modeling

A Cyber-Physical System Model is also created similarly. The App Designer layout is the bridge between the digital twin and the cyber-physical system. A deep learning algorithm named Convolutional Neural Network and motion mapping is under development which will be used to transfer the motion data from the image to CPS. The CPS interface will be used to transfer the data to the Digital Twin model.

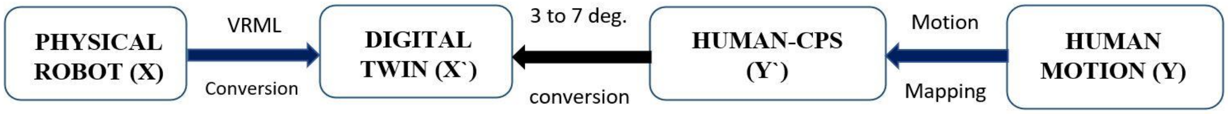


Figure 3.3 An outline of Digital Twin Model

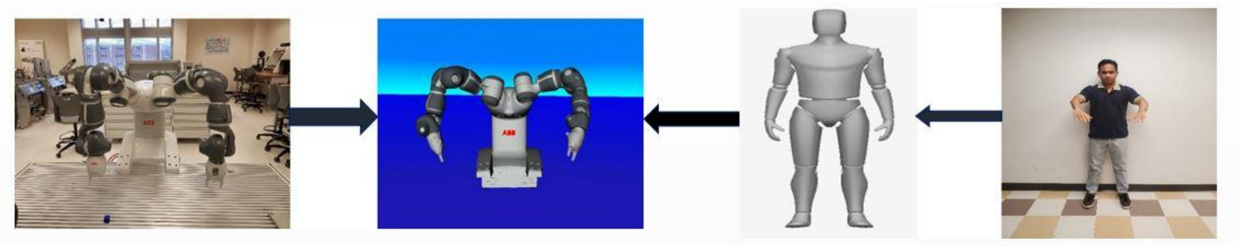


Figure 3.4 An illustration of Digital Twin Model Assembly

3.1 VRML file Creation

Two-Arm Collaborative Robot Model

The CAD assembly file of the collaborative robot model is being deployed in The CAD software and the assembly file is then segregated part by part as step files. (The part files were centered. Then an assembly file was opened where the base file was imported and centered to the origin. The link1_ L file was imported and linked to the base then the link file is featured with rotational from joint function. By selecting an axis centered to the part file the translation feature was added. Similarly, other links were added and provided with rotation and translation features). The part files are featured with Rotation and translation constraints as it is to be defined in the 3D world editor. After it is featured, the part files are assembled as a complete model. Then, the parts are kept in that position for taking coordinates and saved as part files to maintain the position and orientation when converted into VRML files. The coordinates of the part files in the assembly file are noted in the excel file for putting in the 3D world Editor. In the 3D world Editor, the VRML files are attached sequentially with their coordinates, rotation, and translation values.

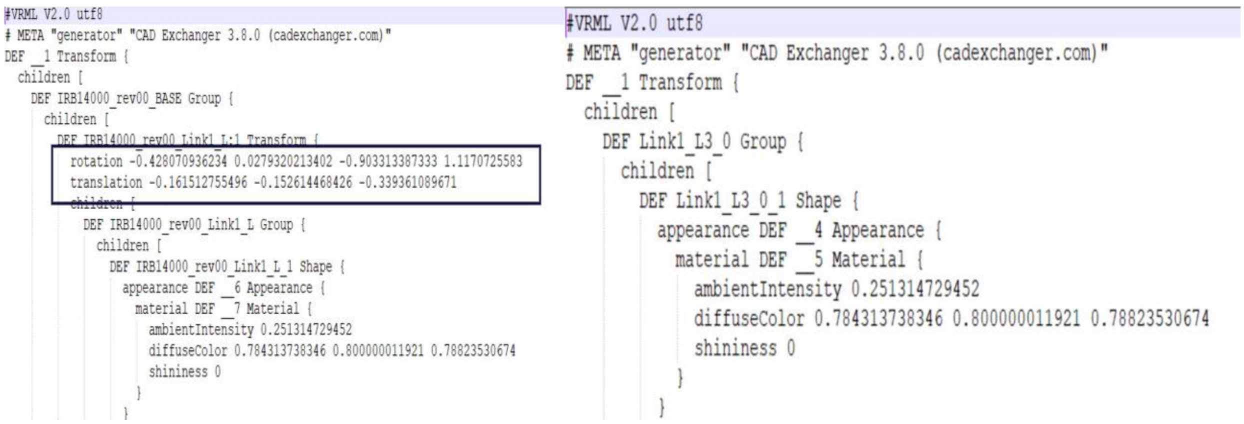


Figure 3.5 VRML files (text) of link1_l file with & without features

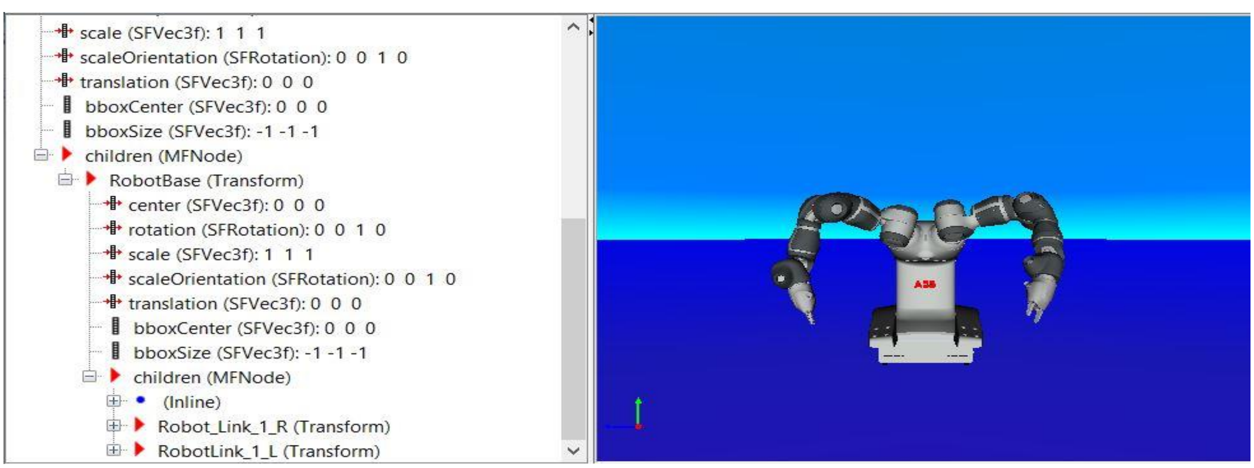


Figure 3.6 Digital Twin Model in 3D World Editor

Human-Cyber Physical System

The Cyber-Physical System is the collaborating computational entity connected with the real world. This model (Human-CPS) is used to transfer the motion data from Human pose to digital twin. The model is developed in Fusion 360 with all necessary features like rotation, translation. Then the assembly file is converted into STEP files and then reoriented in The CAD software to figure out the coordinate data of the part file like a robot model. Then the CPS model was converted into a VRML file for 3D world Editor.

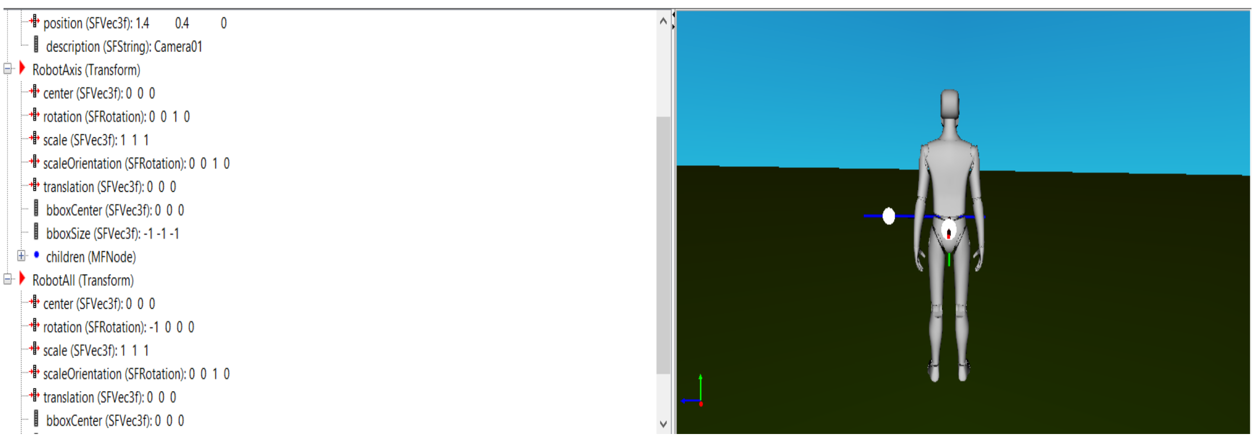


Figure 3.7 Human CPS in 3D World Editor

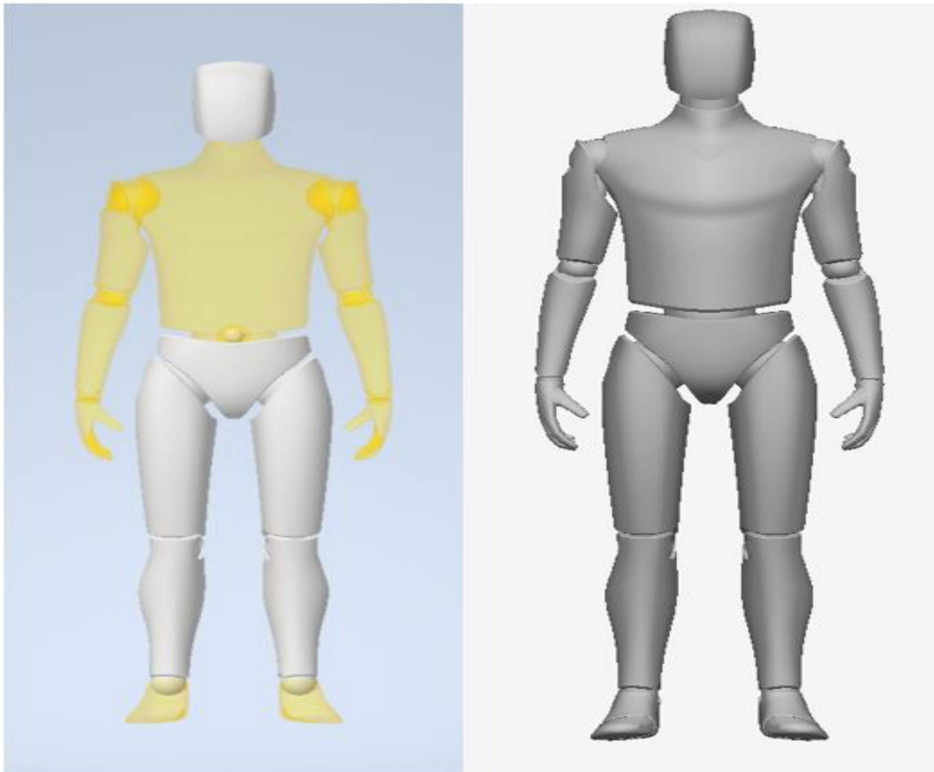


Figure 3.8 Human CPS Model as CAD file and VRML file

3.2 Digital Twin Model

The study builds and analyzes a digital model of the two-arm collaborative robot that will further be used for imitating human arm motion through deep reinforcement learning. The CAD model of the robot was used to create the 3D (STEP) files for each robot part and the part files were assembled with proper rotation and translation features in The CAD software. Then, the parts are converted to Virtual Reality Modeling Format (VRML) and assembled in MATLAB 3D World Editor. A human CPS model was developed similarly to transfer the human arm motion to the robot's digital twin. After the creation of the Digital model of a two-arm collaborative robot in 3D world Editor, the app designer platform was used for regulating the model. The app designer is used for creating an application that offered a grid layout and is used for organizing the 3D model interface. The Application provided us the user interface and the 3D world editor provided us with a digital model, rotational data, translational data, and position of the part file in the model. So, the model can be regularized manually with the application and can also be used for regularization through deep reinforcement learning. Currently, the author is working on the deep reinforcement learning model with Convolutional Neural Network to transfer human motion to the Cyber-Physical System. Once, it is successfully be done then the transfer of the motion data is the next step which will be transferred from CPS to the digital model. To do this procedure complete, the author must convert the three degrees of freedom data from human motion to the seven degrees of freedom data for the digital twin model.

CHAPTER 4

RESULTS AND DISCUSSION

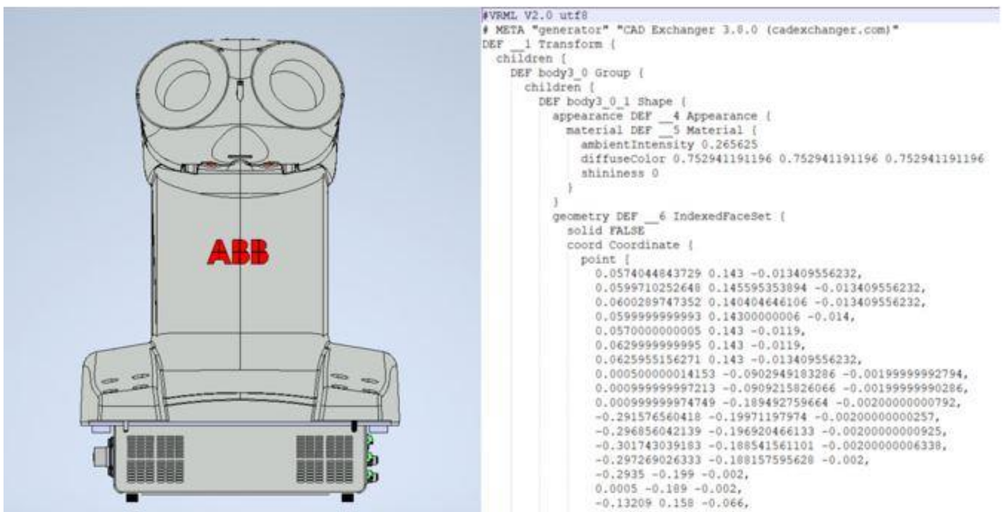
This section discusses the construction of the Digital Twin model and its future implementation for Human motion imitation. The accuracy of this model depends on the rotation constraint, translation constraint, and position co-ordinate of every part. Because of these features, the original CAD file from the ABB website could not be used. It also discusses the progress of the deep reinforcement learning concept related to the future work of this project. Moreover, this section discusses the challenges were faced during the formation of this model with RobotStudio®.

4.1 ABB Yumi Robot

ABB Yumi is an updated version of Human-Robot Collaboration. In the lab, the department has one of the collaborative robot models which is Yumi® - IRB 14000. It is a high-functioning dual-arm robot made for mass factory assembly automation. It is a robot with 7 degrees of freedom. Because of its prospect, the author chose this robot to study and to automate the operation done by it. ABB robot is a versatile collaborative robot with huge potentialities, especially for industrial use. The core concept behind this robot to build-up is to use it for mass production as well as for custom manufacturing.

4.2 Digital Model Formation

For digital twin formation, the CAD files of the collaborative robot were collected from the designated official website of ABB Robot. The files were in part files. These files were assembled in The CAD software and then converted into VRML files for digital twin model formation. But the file was not suitable for the conversion as the file did not have any rotation and translation constraints/features which were determining factors for creating the model.



Robot Base CAD File

Robot Base VRML File

Figure 4.1 CAD & VRML files of base (without features addition)

There were 7 parts in every gripper which were base, two slides, two fingers, and two cups. The gripper parts also did not have any translation or rotation features. The CAD software was used for recreating the orientation as well as features that were required. There were 29 parts in total to construct the digital model. Each gripper has seven parts individually. Since the gripper parts had nothing to do with the motion individually, the gripper was made without adding rotation and translation to individual parts. After The assembly of the gripper parts, it acts like a single part and the number of the parts file became 19 from 29.

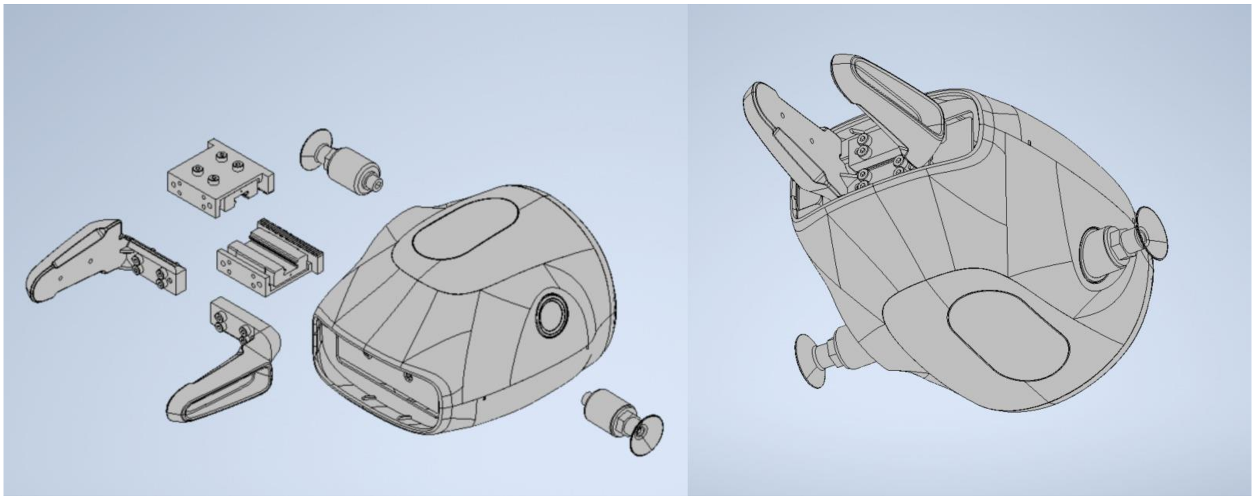
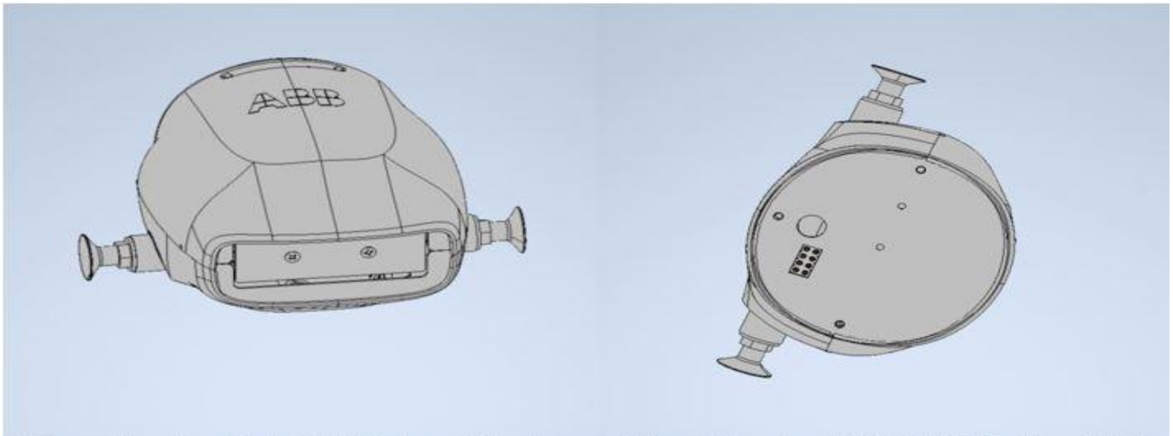


Figure 4.2 Gripper layout before and after assembly

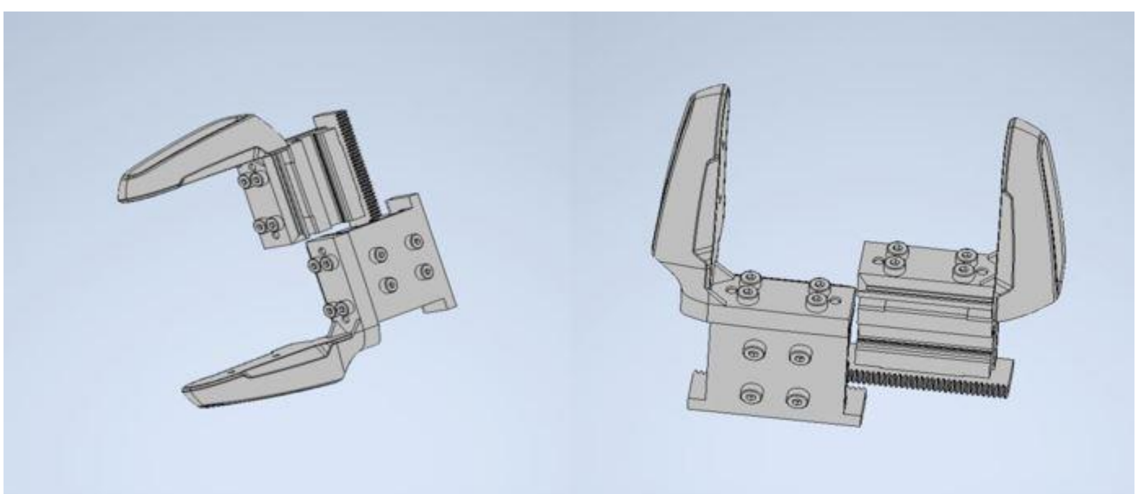
The gripper had two kinds of the part. First, some parts did not have to move individually which were attached with the gripper main body such as the cups tool.



Gripper Fingerless (Left) CAD File (Front View) Gripper Fingerless (Right) CAD File (Rear View)

Figure 4.3 Gripper CAD file without slides & fingers

The other parts are moving parts of the gripper that includes the slides and the fingers. The slides were added to the finger as in the real gripper. The slides are connected to a motor and eventually, it regulates the motor. So, to follow the same strategy, the fingers were connected to the slides and they move together.



Gripper Fingers (Left) CAD File Gripper Fingers (Right) CAD File

Figure 4.4 Gripper fingers & slides CAD file

As these parts need to move, rotation and translation feature added to these part files individually. These features allowed the parts to move concerning certain points and it could be controlled through regularization.

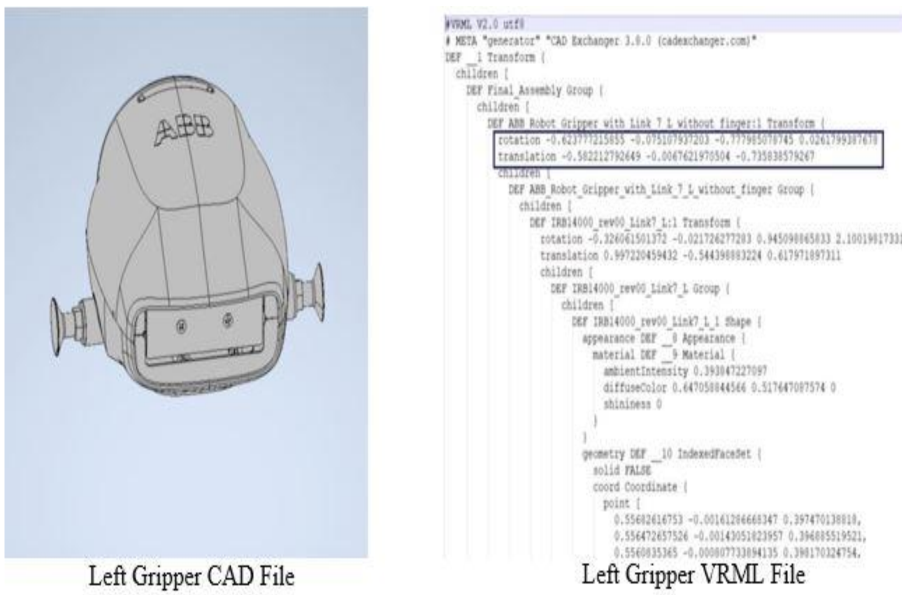
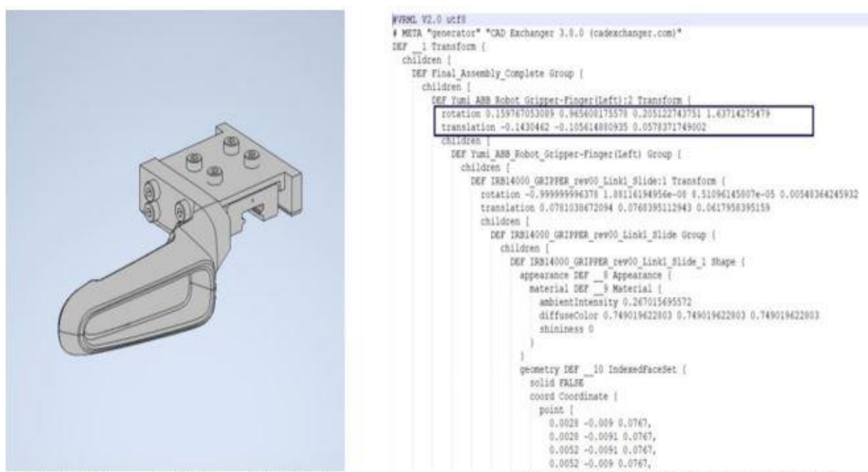


Figure 4.5 Left gripper CAD & VRML file (features added)

Left Gripper Left Finger CAD File   Left Gripper Left Finger VRML File

Figure 4.6 Left-gripper left fingers CAD & VRML file (features added)



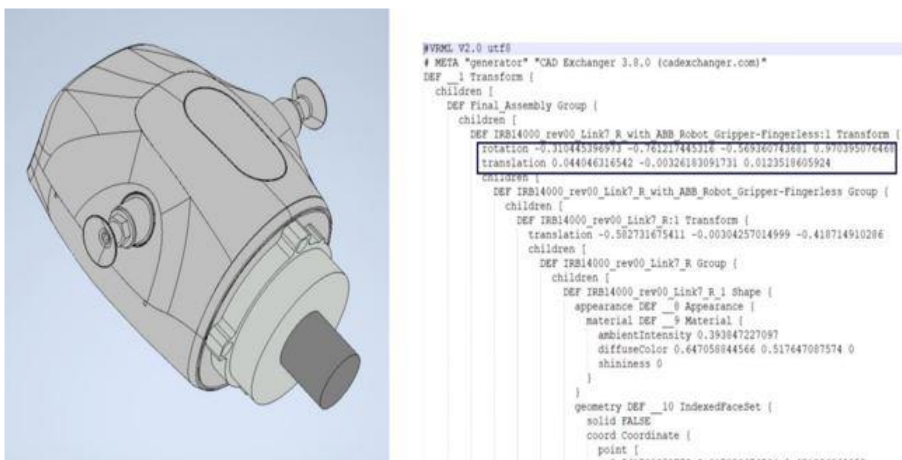
Right Gripper Left Finger CAD File   Right Gripper Left Finger VRML File

Figure 4.7 Right-gripper left finger CAD & VRML file (features added)

After completing the Gripper configuration with features added, the link parts other than the gripper were reconfigured with features that were rotation and translation.

Robot Link_1 Left CAD File Robot Link_1 Left VRML File

Figure 4.8 CAD & VRML files of link_1 left (with features)



Link_7 Right with Gripper CAD File Link_7 Right with Gripper VRML File

Figure 4.9 CAD & VRML files of link_7 right with gripper(fingerless) (with features)

The base was set for the origin point of the final assembly file. The additional features did not have to be added to this part as it was built to be fixed in its position at the origin of the assembly file.

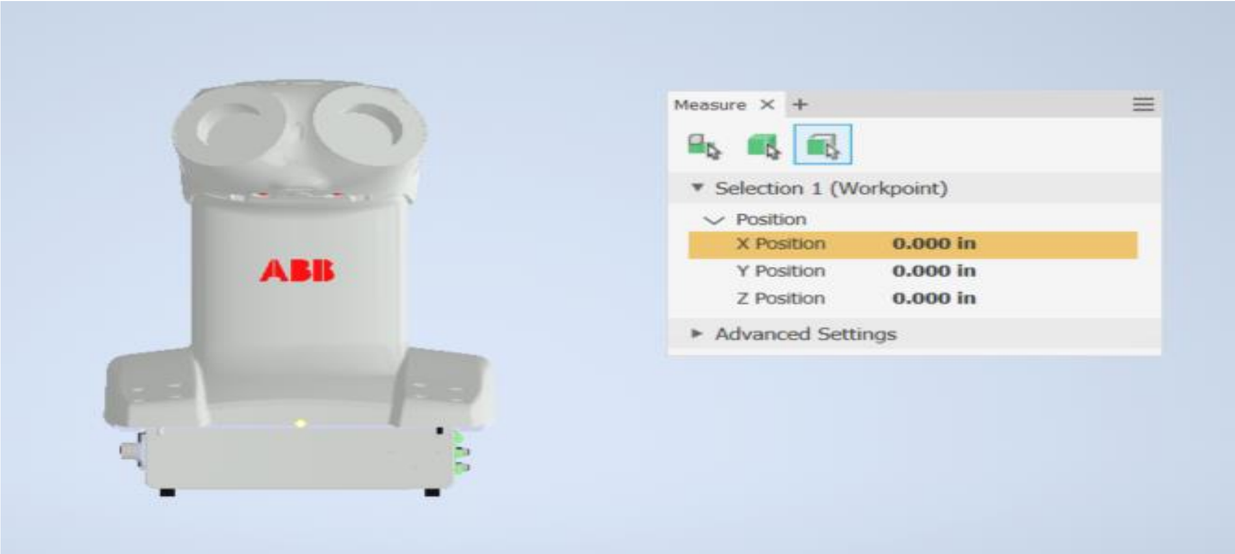
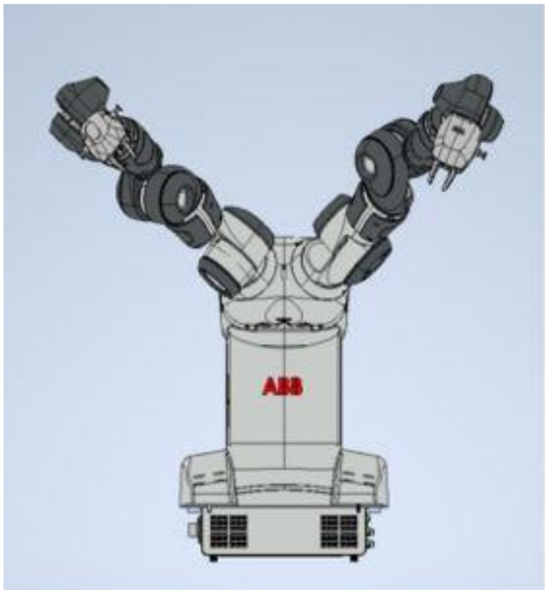


Figure 4.10 Base of the ABB Yumi - IRB 14000 Robot

Then the link files were added to this base sequentially. First, the Link_1Left was added to the base. As the Link was supposed to move, additional features like rotation, translation were added to this link_1Left. The other Links were added in the same procedure. After Adding all links together, we got the complete assembly with rotation and translation feature.



Complete Robot Assembly CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.9.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Final_Assembly_Complete Group {
      children [
        DEF IRB14000_rev00_BASE:1 Transform {
          rotation 5.55111512313e-17 1 3.18023172558e-15 0.0349065850239
          children [
            DEF IRB14000_rev00_BASE Group {
              children [
                DEF IRB14000_rev00_BASE_1 Shape {
                  appearance DEF __6 Appearance {
                    material DEF __7 Material {
                      ambientIntensity 0.265625
                      diffuseColor 0.752941191196 0.752941191196 0.752941191196
                      shininess 0
                    }
                  }
                  geometry DEF __8 IndexedFaceSet {
                    solid FALSE
                    coord Coordinate {
                      point [
                        0.0625955156271 0.143 -0.013409556232,
                        0.0574044843729 0.143 -0.013409556232,
                        0.06 0.143 -0.014,
                        0.0629999999995 0.143 -0.0119,
                        0.0570000000005 0.143 -0.0119,
                        0.000499999998491 -0.0902949183796 -0.0019999998307,
                        0.00100000000193 -0.0909215825819 -0.0020000000075,
                        0.00100000000018 -0.189492759634 -0.0020000000025,
                        -0.291576560417 -0.199711979756 -0.0020000000075,
                        -0.296856042133 -0.196920466129 -0.0020000000001,
                        -0.301743036024 -0.188541561804 -0.0019999999855,
                        -0.297269026287 -0.188157595613 -0.002,
```

Complete Robot Assembly VRML File

Figure 4.11 ABB Yumi - IRB 14000 robot assembly with features

Here, the rotation is present, but we fixed the base file and that is why the initial VRML code showed only rotation, but other parts were featured with both as we mentioned earlier.

4.3 Rotation & Translation data collection

The most important thing about assembling the robot is the point of rotation and axis of rotation. Every point of rotation and axis of rotation were precisely chosen so that the robot can rotation by the exact axis from its perfect point of reference. To allow the exact point of rotation and the axis of rotation to be perfect, the points are taken from the CAD file and then be used in the 3D world Editor.

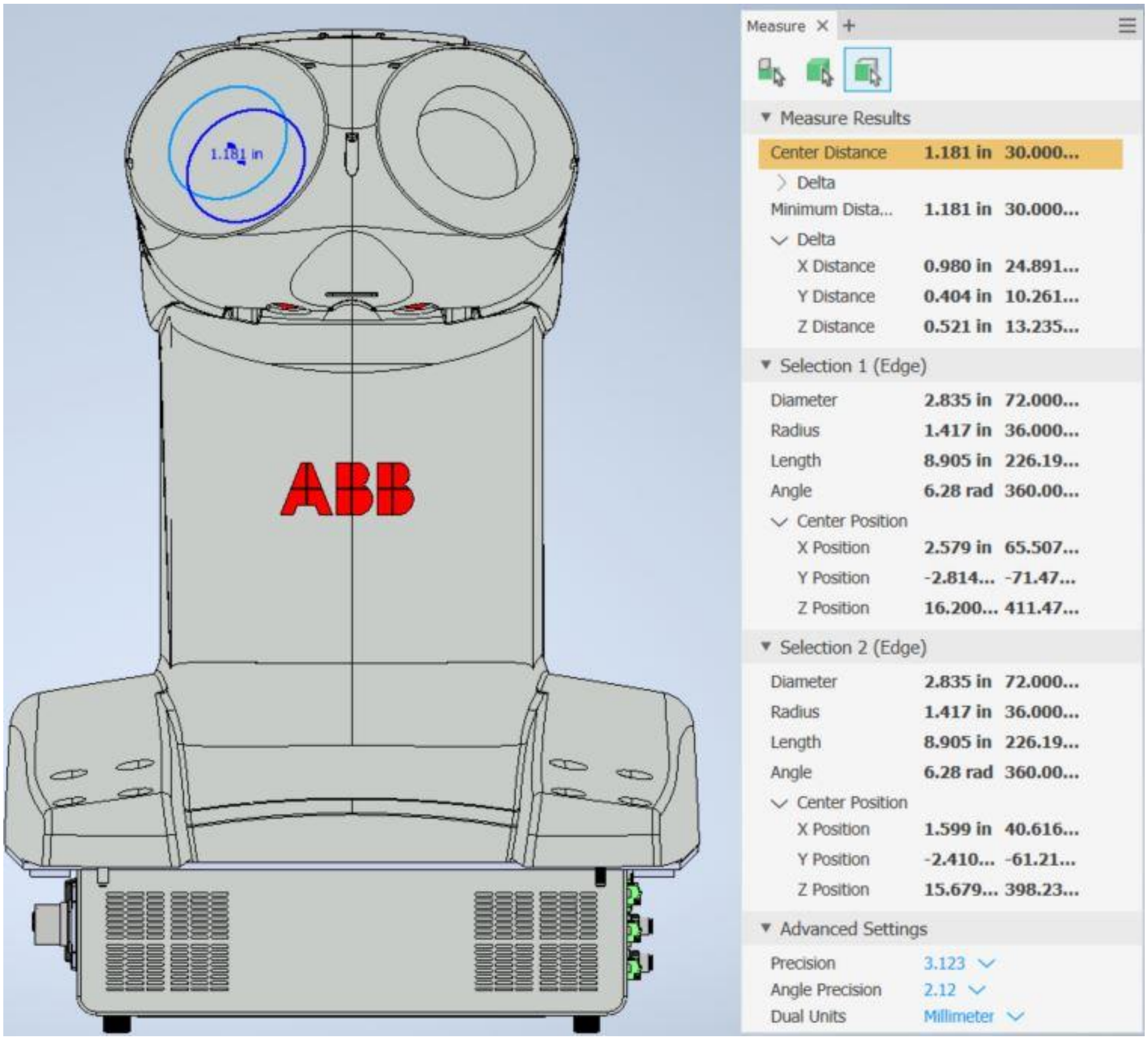


Figure 4.12 Base (left hole) coordinate data for rotation & translation

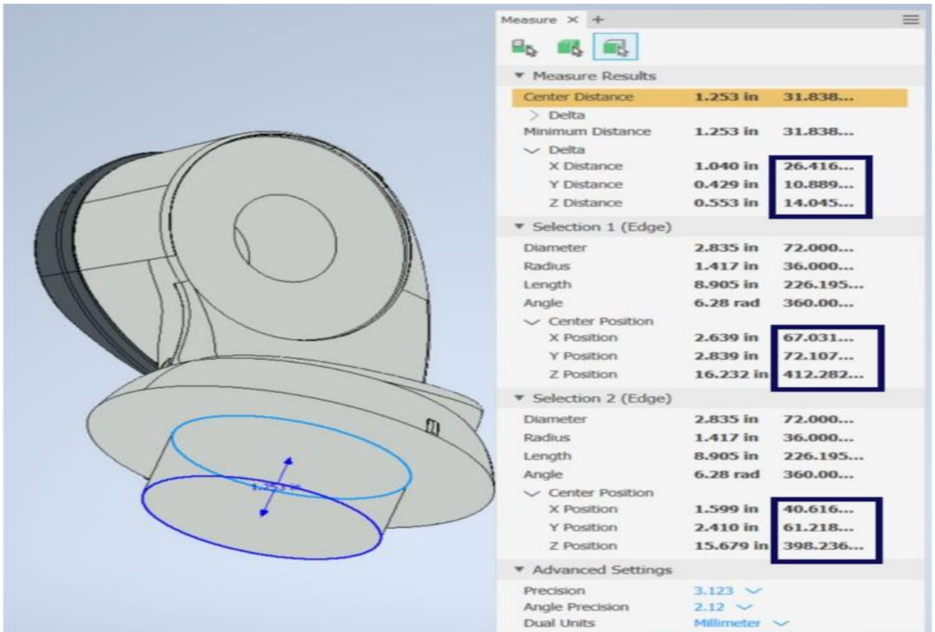


Figure 4.13 Link_1 right coordinate data for rotation & translation

As the Link_7 and the Fingerless Gripper are joined together and in the physical model they worked together, they were attached and recognized as one part file.

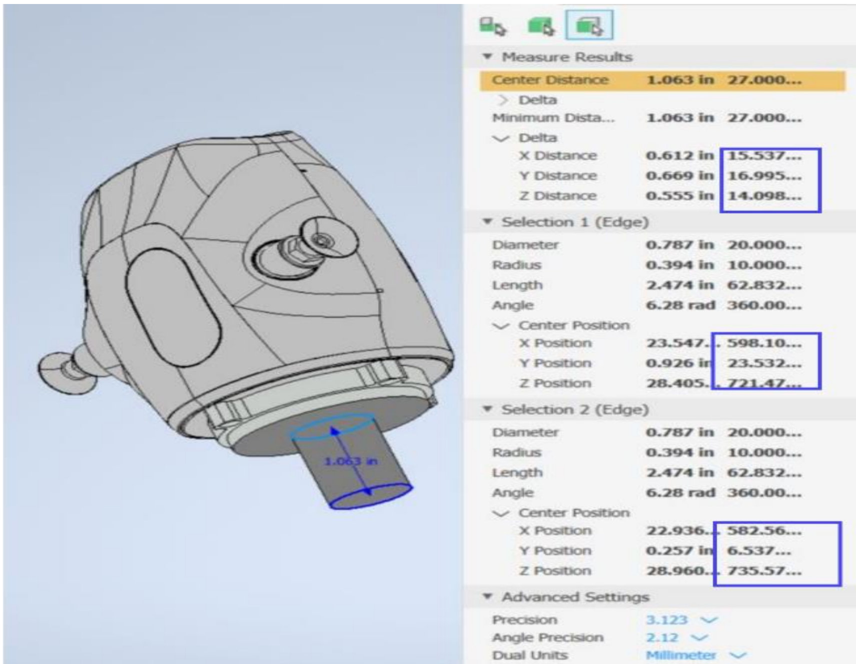


Figure 4.14 Link_7 left & fingerless gripper coordinate data for rotation & translation

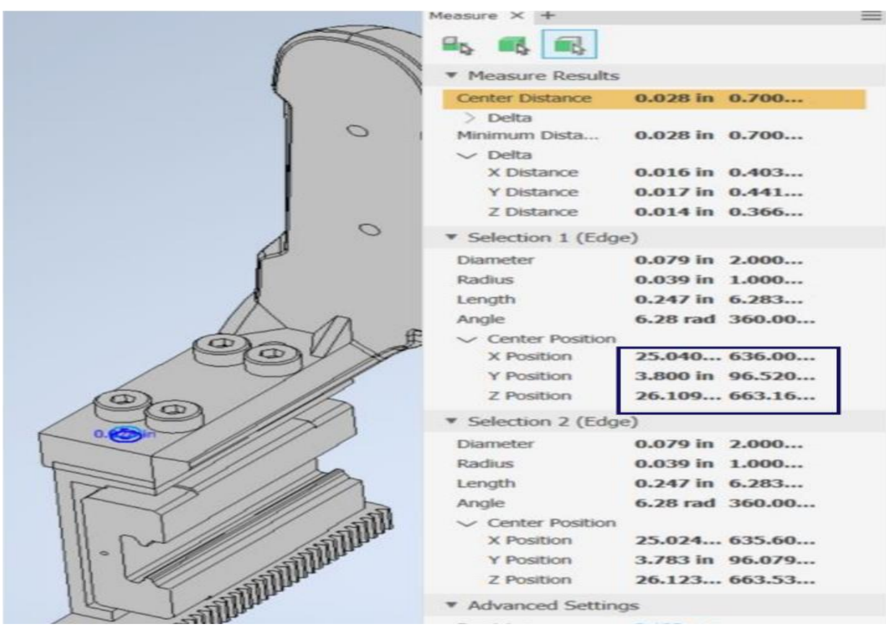


Figure 4.15 Left gripper left finger (close) coordinate data for rotation & translation

All these data are taken into consideration. The selection 1 data are the original plane of the part considered. Selection 2 indicates the plane where the next link files were going to be attached with the present part file. The translation data I used in the 3D world editor was taken from here and every selection of 2 data was the translation data as the next part moves from that point of reference. The translation data from the excel file was used after divided 1000 times due to resolve the scaling problem. The rotation features in 3D world Editor had two-part. First three value was about rotation coordinates (x, y, z) value and the second part was the rotation angle value (4^{th} value). The subtractions of selections 1 & 2 were the rotational coordinates value for the 3D world Editor for that specific robot part. The rotation angle value was adjusted with the robot link configuration. The Gripper data were taken for only selection 1 data and three data were considered to recognize the close, half-open, and close state of the individual finger of the grippers.

Table 4.1 Robot left arm co-ordinate data

ROBOT LEFT ARM			
Robot Base Hole Coordinate			
	X	Y	Z
PART ORIGIN (SELECTION 1) (mm)	65.507	-71.479	411.471
NEXT LINK JOINT (SELECTION 2) (mm)	40.616	-61.218	398.236
LINK_1 LEFT			
PART ORIGIN (SELECTION 1) (mm)	67.031	-72.107	412.282
NEXT LINK JOINT (SELECTION 2) (mm)	40.616	-61.218	398.236
LINK_2 LEFT			
PART ORIGIN (SELECTION 1) (mm)	173.343	-79.282	433.819
NEXT LINK JOINT (SELECTION 2) (mm)	182.491	-80.012	416.049
LINK_3 LEFT			
PART ORIGIN (SELECTION 1) (mm)	318.48	-75.485	520.755
NEXT LINK JOINT (SELECTION 2) (mm)	280.552	-84.037	501.582
LINK_4 LEFT			
PART ORIGIN (SELECTION 1) (mm)	401.292	-40.141	518.998
NEXT LINK JOINT (SELECTION 2) (mm)	395.493	-13.626	518.642
LINK_5 LEFT			
PART ORIGIN (SELECTION 1) (mm)	486.426	-30.908	658.615
NEXT LINK JOINT (SELECTION 2) (mm)	462.903	-36.368	635.094
LINK_6 LEFT			
PART ORIGIN (SELECTION 1) (mm)	553.125	25.877	722.611
NEXT LINK JOINT (SELECTION 2) (mm)	544.662	38.251	728.201
LINK_7 LEFT (GRIPPER BASE)			
PART ORIGIN (SELECTION 1) (mm)	598.105	23.532	721.477
NEXT LINK JOINT (SELECTION 2) (mm)	582.568	6.537	735.576
LEFT GRIPPER_LEFT FINGER			
CLOSED (SELECTION 1) (mm)	636.007	96.52	663.168
HALF OPEN (SELECTION 1) (mm)	645.6315	88.617	664.2475
OPEN (SELECTION 1) (mm)	655.256	80.714	665.327
LEFT GRIPPER_RIGHT FINGER			
CLOSED (SELECTION 1) (mm)	673.222	74.168	677.235
HALF OPEN (SELECTION 1) (mm)	663.5975	82.071	676.155
OPEN (SELECTION 1) (mm)	653.973	89.974	675.075

Table 4.2 Robot right arm co-ordinate data

ROBOT RIGHT ARM			
Robot Base Hole Coordinate			
	X	Y	Z
PART ORIGIN (SELECTION 1) (mm)	65.507	71.479	411.471
NEXT LINK JOINT (SELECTION 2) (mm)	40.616	61.218	398.236
LINK_1 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	67.031	72.107	412.282
NEXT LINK JOINT (SELECTION 2) (mm)	40.616	61.218	398.236
LINK_2 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	132.012	144.236	461.195
NEXT LINK JOINT (SELECTION 2) (mm)	131.043	160.865	450.125
LINK_3 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	231.407	218.277	583.592
NEXT LINK JOINT (SELECTION 2) (mm)	211.152	193.103	547.55
LINK_4 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	303.666	255.698	623.071
NEXT LINK JOINT (SELECTION 2) (mm)	308.789	276.114	605.933
LINK_5 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	280.972	354.656	751.594
NEXT LINK JOINT (SELECTION 2) (mm)	276.956	333.733	725.47
LINK_6 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	318.222	436.532	809.838
NEXT LINK JOINT (SELECTION 2) (mm)	317.032	452.986	796.844
LINK_7 RIGHT (GRIPPER BASE)			
PART ORIGIN (SELECTION 1) (mm)	344.167	433.506	847.264
NEXT LINK JOINT (SELECTION 2) (mm)	324.203	432.369	847.652
RIGHT GRIPPER_LEFT FINGER			
CLOSED (SELECTION 1) (mm)	448.636	441.117	822.2
HALF OPEN (SELECTION 1) (mm)	449.1245	436.5025	833.8065
OPEN (SELECTION 1) (mm)	449.613	431.888	845.413
RIGHT GRIPPER_RIGHT FINGER			
CLOSED (SELECTION 1) (mm)	450.259	437.007	867.883
HALF OPEN (SELECTION 1) (mm)	449.7705	441.6215	856.2765
OPEN (SELECTION 1) (mm)	449.282	446.236	844.67

The Left-arm coordinate data and the right arm coordinate data were used to calculate the rotation axis vector for every link as well as the translation value for every link of the robot. For the rotation axis vector, the co-ordinate value of the earlier part of the joint (Selection 1) is taken as the first value and the co-ordinate of the later part of the joint (selection 2) is taken as the second value. Then, the second value is subtracted from the first value (selection 1 – selection2) to get the rotation axis value of the earlier link.

For translation value, the next link joint value (selection 2) for that link is taken and then the earlier link's next link joint value (selection 2 of earlier joint) is subtracted from it. For example, link 4 translation value = (selection 2_Link 4 – Selection_3)/1000. Here, I divided the result by 1000 times so that it would not create any scaling problem for the 3D world editor. For, Link_1 for both left and right the translational value is going to be selection 2 value as they started from the base file.

Table 4.3 Robot left arm rotation & translation data

ROBOT LEFT ARM			
ROTATION & TRANSLATION	X	Y	Z
LINK_1 LEFT			
PART ORIGIN (SELECTION 1) (mm)	67.031	-72.107	412.282
NEXT LINK JOINT (SELECTION 2) (mm)	40.616	-61.218	398.236
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	26.415	-10.889	14.046
TRANSLATION (SELECTION 2)/1000 (m)	0.040616	-0.061218	0.398236
LINK_2 LEFT			
PART ORIGIN (SELECTION 1) (mm)	173.343	-79.282	433.819
NEXT LINK JOINT (SELECTION 2) (mm)	182.491	-80.012	416.049
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	-9.148	0.73	17.77
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.141875	-0.018794	0.017813
LINK_3 LEFT			
PART ORIGIN (SELECTION 1) (mm)	318.48	-75.485	520.755
NEXT LINK JOINT (SELECTION 2) (mm)	280.552	-84.037	501.582
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	37.928	8.552	19.173
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.098061	-0.004025	0.085533
LINK_4 LEFT			
PART ORIGIN (SELECTION 1) (mm)	401.292	-40.141	518.998
NEXT LINK JOINT (SELECTION 2) (mm)	395.493	-13.626	518.642
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	5.799	-26.515	0.356
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.114941	0.070411	0.01706
LINK_5 LEFT			
PART ORIGIN (SELECTION 1) (mm)	486.426	-30.908	658.615
NEXT LINK JOINT (SELECTION 2) (mm)	462.903	-36.368	635.094
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	23.523	5.46	23.521
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.06741	-0.022742	0.116452
LINK_6 LEFT			
PART ORIGIN (SELECTION 1) (mm)	553.125	25.877	722.611
NEXT LINK JOINT (SELECTION 2) (mm)	544.662	38.251	728.201
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	8.463	-12.374	-5.59
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.081759	0.074619	0.093107
LINK_7 LEFT (GRIPPER BASE)			
PART ORIGIN (SELECTION 1) (mm)	598.105	23.532	721.477
NEXT LINK JOINT (SELECTION 2) (mm)	582.568	6.537	735.576
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	15.537	16.995	-14.099
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.037906	-0.031714	0.007375
LEFT GRIPPER_LEFT FINGER			
CLOSED (SELECTION 1) (mm)	636.007	96.52	663.168
HALF OPEN (SELECTION 1) (mm)	645.6315	88.617	664.2475
OPEN (SELECTION 1) (mm)	655.256	80.714	665.327
LEFT GRIPPER_RIGHT FINGER			
CLOSED (SELECTION 1) (mm)	673.222	74.168	677.235
HALF OPEN (SELECTION 1) (mm)	663.5975	82.071	676.155
OPEN (SELECTION 1) (mm)	653.973	89.974	675.075

Table 4.4 Robot right arm rotation & translation data

ROBOT RIGHT ARM			
ROTATION & TRANSLATION	X	Y	Z
LINK_1 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	67.031	72.107	412.282
NEXT LINK JOINT (SELECTION 2) (mm)	40.616	61.218	398.236
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	26.415	10.889	14.046
TRANSLATION (SELECTION 2)/1000 (m)	40.616	61.218	398.236
LINK_2 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	132.012	144.236	461.195
NEXT LINK JOINT (SELECTION 2) (mm)	131.043	160.865	450.125
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	0.969	-16.629	11.07
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.090427	0.099647	0.051889
LINK_3 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	231.407	218.277	583.592
NEXT LINK JOINT (SELECTION 2) (mm)	211.152	193.103	547.55
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	20.255	25.174	36.042
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.080109	0.032238	0.097425
LINK_4 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	303.666	255.698	623.071
NEXT LINK JOINT (SELECTION 2) (mm)	308.789	276.114	605.933
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	-5.123	-20.416	17.138
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.097637	0.083011	0.058383
LINK_5 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	280.972	354.656	751.594
NEXT LINK JOINT (SELECTION 2) (mm)	276.956	333.733	725.47
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	4.016	20.923	26.124
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	-0.031833	0.057619	0.119537
LINK_6 RIGHT			
PART ORIGIN (SELECTION 1) (mm)	318.222	436.532	809.838
NEXT LINK JOINT (SELECTION 2) (mm)	317.032	452.986	796.844
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	1.19	-16.454	12.994
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.040076	0.119253	0.071374
LINK_7 RIGHT (GRIPPER BASE)			
PART ORIGIN (SELECTION 1) (mm)	344.167	433.506	847.264
NEXT LINK JOINT (SELECTION 2) (mm)	324.203	432.369	847.652
ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	19.964	1.137	-0.388
TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.007171	-0.020617	0.050808
RIGHT GRIPPER_LEFT FINGER			
CLOSED (SELECTION 1) (mm)	448.636	441.117	822.2
HALF OPEN (SELECTION 1) (mm)	449.1245	436.5025	833.8065
OPEN (SELECTION 1) (mm)	449.613	431.888	845.413
RIGHT GRIPPER_RIGHT FINGER			
CLOSED (SELECTION 1) (mm)	450.259	437.007	867.883
HALF OPEN (SELECTION 1) (mm)	449.7705	441.6215	856.2765
OPEN (SELECTION 1) (mm)	449.282	446.236	844.67

4.4 Digital Twin Model Assembly in 3D World Editor

These Rotation and Translation data were used in MATLAB® 3D world Editor. We assembled the links as children of previous links and used these rotation and translational data to be linked up together.

Figure 4.16 ABB robot assembly in 3D World Editor using rotation & translation data

The Left Links are sequentially added to the earlier left links and the Left Link_1 is added to the Left Base Hole of the ABB Robot. The Left Link_1 Contains from Left Link_2 to Left Gripper and Fingers.

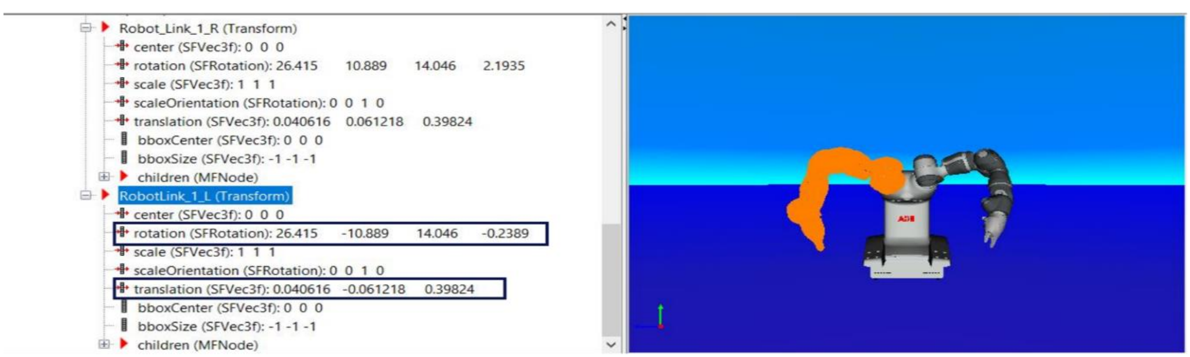


Figure 4.17 Link_1 left adjusted in 3D World Editor using rotation & translation data

The Right Links were added to the right hole of the base and the Right Link_1 contained the other links from Right Link_2 to Right Link_8 Gripper with Fingers.

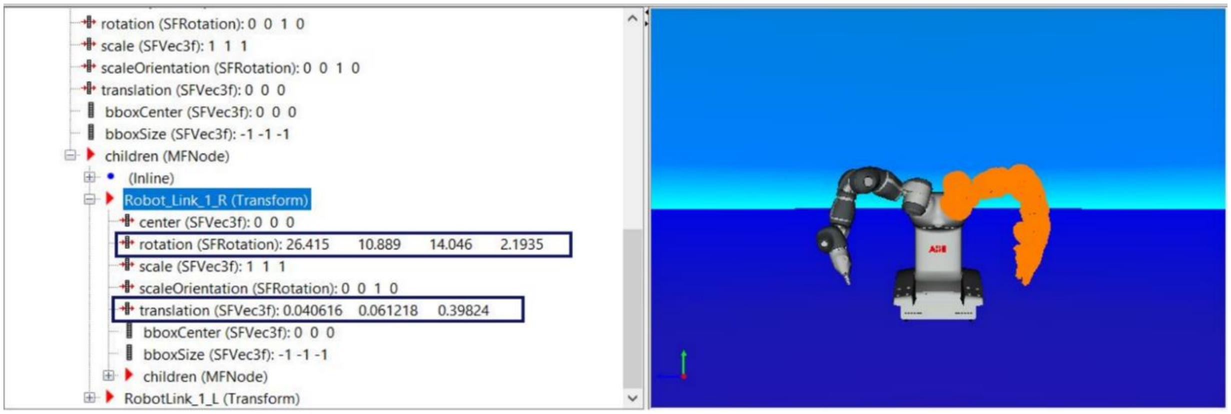


Figure 4.18 Link_1 right adjusted in 3D World Editor using rotation & translation data

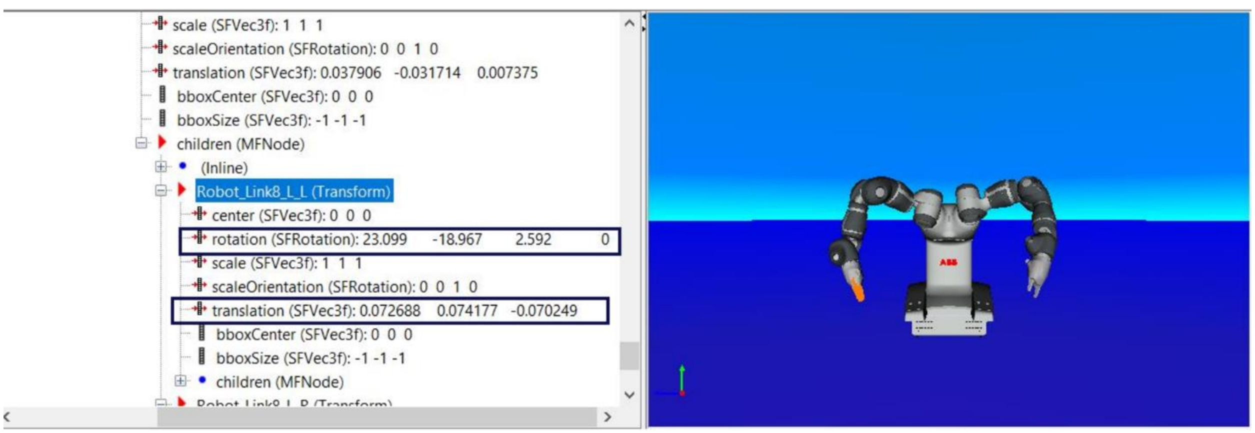


Figure 4.19 Link_8 left (left finger) adjusted using rotation & translation data

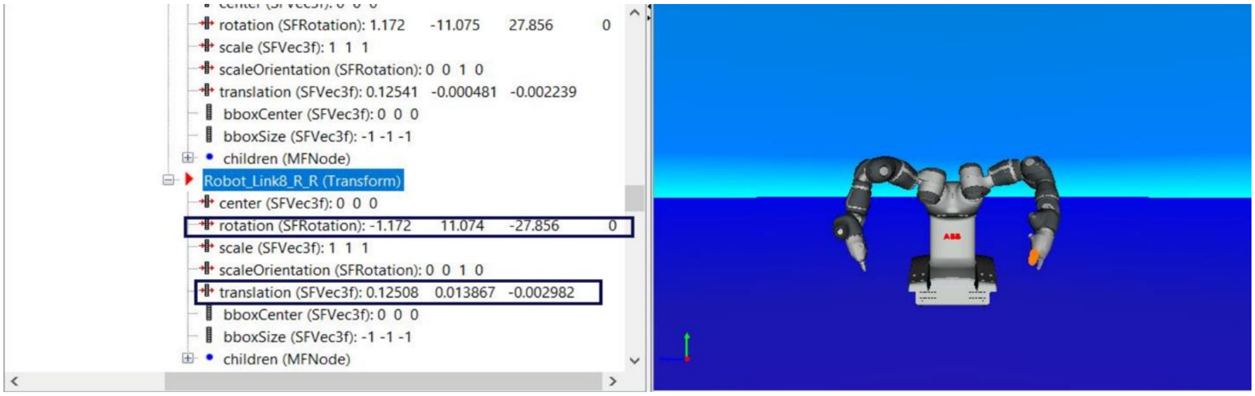


Figure 4.20 Link_8 right (right finger) adjusted using rotation & translation data

4.5 Application for Digital Model Regularization

The robot was set in 3D world editor, but it needed an interface to be controlled by humans or Deep Reinforcement Learning. App Designer Application had been used to develop the interface or control panel to regularize the movement of the robot. It consists of several panels. The manual operation panel uses knobs to control the links of the robot. Each robot link is linked up with knobs individually.

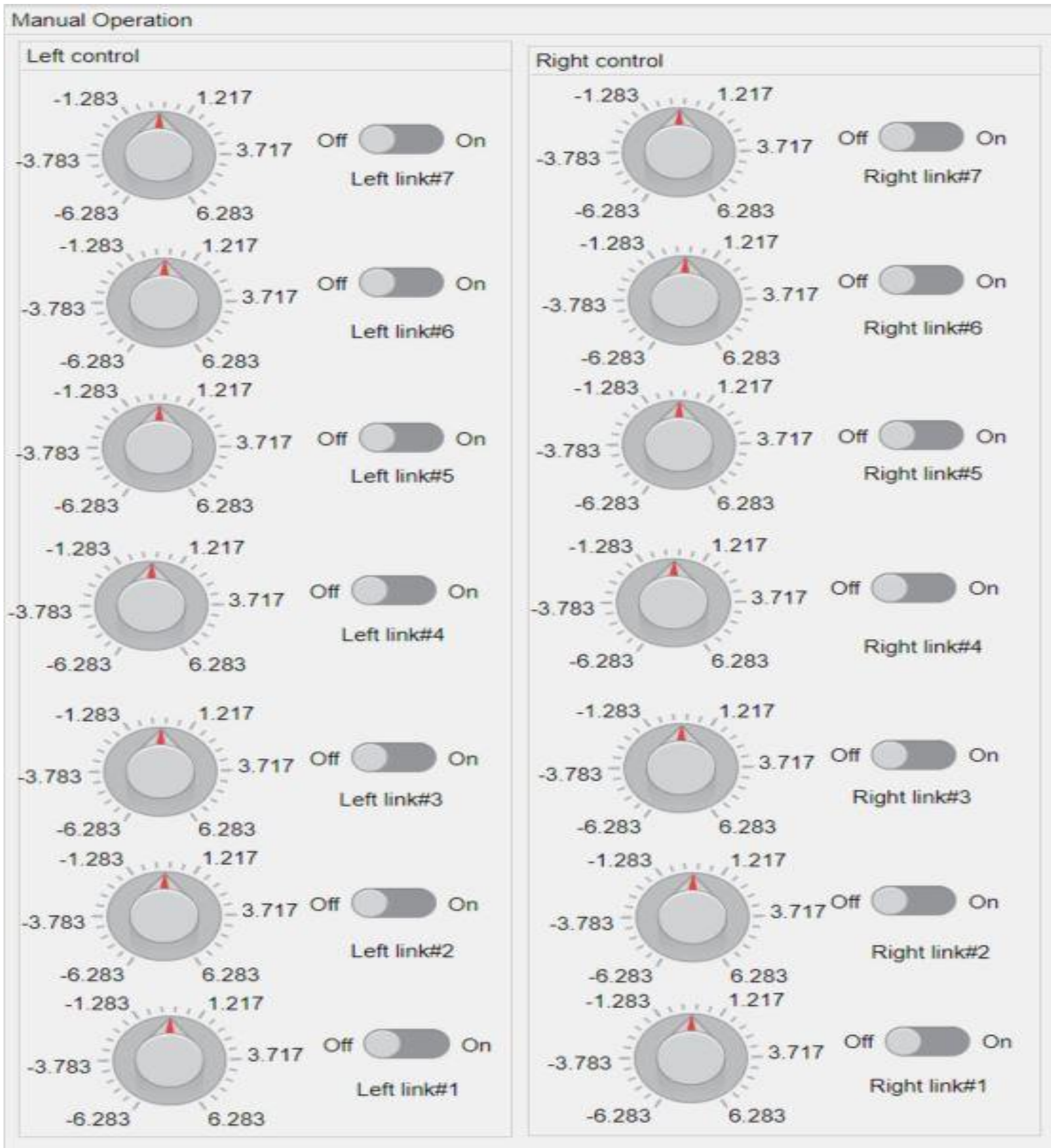


Figure 4.21 Link_8 right (right finger) adjusted using rotation & translation data

The Robot Configuration View gives us the present rotation and angle of the links which will be used to manipulate the robot arm with precise co-ordinate.

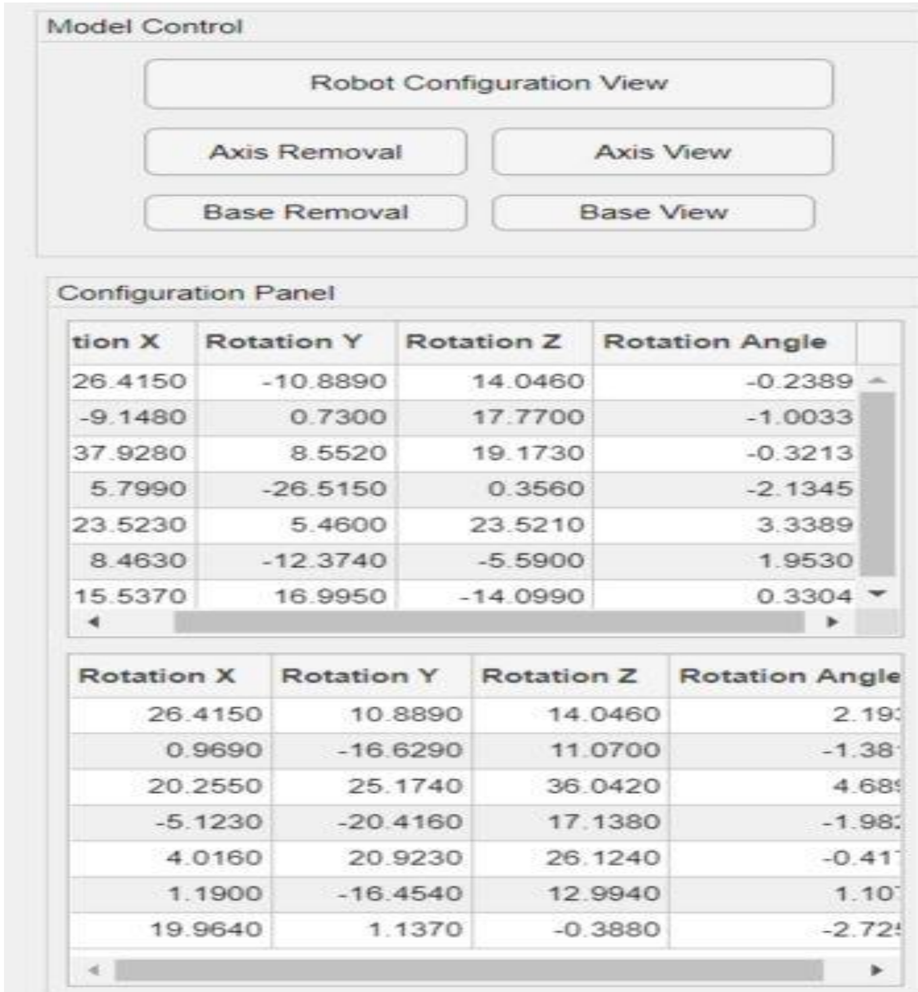


Figure 4.22 Robot configuration view using rotation & translation data

The Gripper Control controls the gripper by opening and closing it and making it on and off during the process.

Figure 4.23 Gripper configuration view in app designer

The rotation and the translation data were being used in the App Designer application to initialize the link position.

```
function viewerButtonPushed(app, event)

    global vr01;
    global leftData;
    global rightData;

    leftData = [26.415 -10.889 14.046 -0.2389;
                -9.148 0.73 17.77 -1.0033;
                37.928 8.552 19.173 -0.3213;
                5.799 -26.515 0.356 -2.1345;
                23.523 5.46 23.521 3.3389;
                8.463 -12.374 -5.59 1.9530;
                15.537 16.995 -14.099 0.3304;
        ];

    rightData = [26.415 10.889 14.046 2.1935;
                 0.969 -16.629 11.07 -1.3812;
                 20.255 25.174 36.042 4.6897;
                 -5.123 -20.416 17.138 -1.9826;
                 4.016 20.923 26.124 -0.4177;
                 1.19 -16.454 12.994 1.1072;
                 19.964 1.137 -0.388 -2.7250;
                ];

    vr01 = vrworld('robotTest_1003.wrl');

    open(vr01);
    view(vr01);


    vr01.Robot_Link1_L.scale = [0 0 0];
    vr01.Robot_Link2_L.scale = [0 0 0];
    vr01.Robot_Link3_L.scale = [0 0 0];
    vr01.Robot_Link4_L.scale = [0 0 0];
    vr01.Robot_Link5_L.scale = [0 0 0];
    vr01.Robot_Link6_L.scale = [0 0 0];
    vr01.Robot_Link7_L.scale = [0 0 0];

    vr01.Robot_Link1_R.scale = [0 0 0];
    vr01.Robot_Link2_R.scale = [0 0 0];
    vr01.Robot_Link3_R.scale = [0 0 0];
    vr01.Robot_Link4_R.scale = [0 0 0];
    vr01.Robot_Link5_R.scale = [0 0 0];
    vr01.Robot_Link6_R.scale = [0 0 0];
    vr01.Robot_Link7_R.scale = [0 0 0];
```

Figure 4.24 Rotation & translation data used in the application

```
% Button pushed function: initialButton
function initialButtonPushed(app, event)
    global leftData;
    global rightData;
    global vr01;

     app.initialzationLamp.Color = [0 1 0];

    vr01.Robot_Link7_L.rotation = [leftData(7,1), leftData(7,2), leftData(7,3),0.3304];
    pause(1);
    vr01.Robot_Link6_L.rotation = [leftData(6,1), leftData(6,2), leftData(6,3),1.9530];
    pause(1);
    vr01.Robot_Link5_L.rotation = [leftData(5,1), leftData(5,2), leftData(5,3),3.3389];
    pause(1);
    vr01.Robot_Link4_L.rotation = [leftData(4,1), leftData(4,2), leftData(4,3),-2.1345];
    pause(1);
    vr01.Robot_Link3_L.rotation = [leftData(3,1), leftData(3,2), leftData(3,3),-0.3213];
    pause(1);
    vr01.Robot_Link2_L.rotation = [leftData(2,1), leftData(2,2), leftData(2,3),-1.0033];
    pause(1);
    vr01.Robot_Link1_L.rotation = [leftData(1,1), leftData(1,2), leftData(1,3),-0.2389];

    pause(1);
     vr01.Robot_Link7_R.rotation = [rightData(7,1), rightData(7,2), rightData(7,3),-2.7250];
    pause(1);
    vr01.Robot_Link6_R.rotation = [rightData(6,1), rightData(6,2), rightData(6,3),1.1072];
    pause(1);
    vr01.Robot_Link5_R.rotation = [rightData(5,1), rightData(5,2), rightData(5,3),-0.4177];
    pause(1);
    vr01.Robot_Link4_R.rotation = [rightData(4,1), rightData(4,2), rightData(4,3),-1.9826];
    pause(1);
    vr01.Robot_Link3_R.rotation = [rightData(3,1), rightData(3,2), rightData(3,3),4.6897];
    pause(1);
    vr01.Robot_Link2_R.rotation = [rightData(2,1), rightData(2,2), rightData(2,3),-1.3812];
    pause(1);
    vr01.Robot_Link1_R.rotation = [rightData(1,1), rightData(1,2), rightData(1,3),2.1935];

    leftData(:,4) = [-0.2389 -1.0033 -0.3213 -2.1345 3.3389 1.9530 0.3304]';
    rightData(:,4) = [2.1935 -1.3812 4.6897 -1.9826 -0.4177 1.1072 -2.7250]';

    app.initialzationLamp.Color = [1 0 0];

end
```

Figure 4.25 Sample code view of the application

This Application has a building capacity so that I did not have to create the code in this environment. The application requires certain data such as the number of links, rotation data, gripper data, etc. The Interface is self-generated.

4.6 Testing the application

The application can be to regularize the movement of the digital model and it was designed in a way to be controlled by other adequate programs for imitating human motion through deep reinforcement learning. Right now, the application can control the robot arm through knobs manually.

4.7 Challenges & Future Work

In the beginning, the concept was implemented in MATLAB® without the features. But without the features, there was no way to figure out the value of the position coordinates of the parts. Then, the challenges were with adding features. Initially, it was not known how to make sure that the VRML file would show the rotation and translation features. After, figuring out the option, it was difficult to find out the way to keep the robot arm in the proper orientation. The next challenge was with the rotation and translation data. The features were added but it was not known how to know exact rotation and translation data from the features. Using the initial position of the link parts and the next part linked up position made me know the exact number to put in rotation and translation bar. After putting the features and data with proper orientation of the individual link part in the 3D World Editor, the digital model of the robot is introduced. In the App Designer, the interface was created to control the robot. Connecting the digital model and the application was the hardest part as it required the correlation between the model and interface. The CPS model was also introduced, and it would be in connection with the digital model. I already used a Human CPS model to create the digital model that contained the features like the robot digital model. The next hard challenge is going to be the conversion of the 3 degrees of freedom Arm motion (input data) to 7 degrees of freedom (Output data) of the collaborative robot. If the robot had 3 or fewer degrees of freedom it would be much easier to replicate the motion. To replicate human motion, it required the conversion of 3 degrees of freedom to 7 degrees of freedom and we planned to use human motion mapping and train our robot for capturing the arm motion accurately. Another major challenge is going to be the virtual commissioning of the digital twin. Currently, The ABB robot is being operated by software named RobotStudio. The RobotStudio is developed with a primary focus for industrial use. It does not allow any code for deep reinforcement learning so that is why it is nearly impossible to execute the plan on the physical robot for now. The software is not well equipped for the coding or interface to build up a digital twin model. I discussed the matter with the support team of the ABB Robot, and they were unable to give any suggestion that might help. Once, I develop the code for deep reinforcement learning for the transition of human motion mapping, I will try to use the coordinates of the arms position to update the physical robot arm's location simultaneously.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

The digital twin model is very useful for the automation of the industry as well as in other sectors like the medical sector, space, etc. The collaborative Yumi robot is very flexible because of its seven degrees of freedom movement. With precision and self-automated capacity, there are many opportunities for the robot. The digital twin model is developed with the CAD file. The CAD file was added with rotation and translation constraints to make it work for making it suitable for the 3D world Editor. After the addition of features, the robot was properly oriented for the assembly and the rotation and translational data were collected for further use. The part files were then added sequentially in the 3D world Editor with the rotational and translational data. The App Designed was used to create the interface to control the digital twin. The Human CPS model was also built in the same procedure. The digital model can be regularized by the application manually. The future work is to create the mapping transition of human motion to transfer the motion to the robot model by the interaction of the Human CPS and the digital model using deep reinforcement learning.

The study recommends the following:

1. The features should be added to the parts file considering the proper orientation to avoid displacement of the robot parts.
2. The factors of the future model implementation should be known very well otherwise it may cause a waste of time due to repetition of the process.
3. The rotation and translation data should be carefully collected from the parts by selecting the correct plane sequentially. Otherwise, it may cause misalignment during the development of the model in 3D world Editor.
4. The application in the App Designer should be developed carefully with proper knowledge of implementation otherwise, it may cause the program to crash.
5. Future research should be done with the robot that can be run by a programming language with full capacity to use the output of the research with full potential.
6. Future researchers should have a passion to learn about robotics to determine the output.

REFERENCES

Alam, K. M. E. S., Abdulmotaleb. (2017). C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE Access, 5*, 2050-2062.

Alcácer, V., & Cruz-Machado, V. (2019). Scanning the industry 4.0: A literature review on technologies for manufacturing systems. *Engineering science and technology, an international journal, 22*(3), 899-919.

Alexopoulos, K., Nikolakis, N., & Chryssolouris, G. (2020). Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. *International Journal of Computer Integrated Manufacturing, 33*(5), 429-439.

Aliaj, K., Feeney, G. M., Sundaralingam, B., Hermans, T., Foreman, K. B., Bachus, K. N., & Henninger, H. B. (2020). Replicating dynamic humerus motion using an industrial robot. *PloS one, 15*(11), e0242005.

Amir, S., Kamal, M. S., Salam, K. A., & Bari, M. (2017). *Sensor based human arm motion and gesture replication system using MATLAB.* Paper presented at the TENCON 2017-2017 IEEE Region 10 Conference.

Ayani, M., Ganebäck, M., & Ng, A. H. (2018). Digital Twin: Applying emulation for machine reconditioning. *Procedia Cirp, 72*, 243-248.

Batty, M. (2018). Digital twins. In: SAGE Publications Sage UK: London, England.

Botkina, D., Hedlind, M., Olsson, B., Henser, J., & Lundholm, T. (2018). Digital twin of a cutting tool. *Procedia Cirp, 72*, 215-218.

Cai, Y., Starly, B., Cohen, P., & Lee, Y.-S. (2017). Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing. *Procedia manufacturing, 10*, 1031-1042.

Council, M. L. (2019, Dec 2). Digital Twins. Retrieved from https://www.manufacturingleadershipcouncil.com/2019/12/02/digital-twins/

Cronrath, C., Aderiani, A. R., & Lennartson, B. (2019). *Enhancing digital twins through reinforcement learning.* Paper presented at the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE).

Dai, Y., Zhang, K., Maharjan, S., & Zhang, Y. (2020). Deep Reinforcement Learning for Stochastic Computation Offloading in Digital Twin Networks. *IEEE Transactions on Industrial Informatics*.

Dröder, K., Bobka, P., Germann, T., Gabriel, F., & Dietrich, F. (2018). A machine learning-enhanced digital twin approach for human-robot-collaboration. *Procedia Cirp, 76*, 187-192.

Dzinic, J., & Yao, C. (2014). *Simulation-based verification of PLC programs.*

Fuller, A., Fan, Z., Day, C., & Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *Ieee Access, 8*, 108952-108971.

Ghadirzadeh, A., Chen, X., Yin, W., Yi, Z., Bjorkman, M., & Kragic, D. (2020). Human-centered collaborative robots with deep reinforcement learning. *IEEE Robotics and Automation Letters*.

Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary perspectives on complex systems* (pp. 85-113): Springer.

Hess, B. (2019). What Is Reverse Engineering and How Does It Work? Retrieved from https://astromachineworks.com/what-is-reverse-engineering/

i-SCOOP. (2017). Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0. Retrieved from https://www.i-scoop.eu/industry-4-0/

Jones, K. S., & Kumar, R. U. (2018). *Robotic Arm With Real Time Human Arm Motion Replication Using IoT.* Paper presented at the IOP Conference Series: Materials Science and Engineering.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine, 51*(11), 1016-1022.

Liu, L., Liu, Q., Song, Y., Pang, B., Yuan, X., & Xu, Q. (2021). A Collaborative Control Method of Dual-Arm Robots Based on Deep Reinforcement Learning. *Applied Sciences, 11*(4), 1816.

Lohtander, M., Ahonen, N., Lanz, M., Ratava, J., & Kaakkunen, J. (2018). Micro manufacturing unit and the corresponding 3D-model for the digital twin. *Procedia manufacturing, 25*, 55-61.

Lu, Y., Huang, X., Zhang, K., Maharjan, S., & Zhang, Y. (2020). Low-latency Federated Learning and Blockchain for Edge Association in Digital Twin empowered 6G Networks. *IEEE Transactions on Industrial Informatics*.

Min, Q., Lu, Y., Liu, Z., Su, C., & Wang, B. (2019). Machine learning based digital twin framework for production optimization in petrochemical industry. *International Journal of Information Management, 49*, 502-519.

Neverova, N. (2016). *Deep learning for human motion analysis.* Université de Lyon,

Pech, M., Vrchota, J., & Bednář, J. (2021). Predictive Maintenance and Intelligent Sensors in Smart Factory. *Sensors, 21*(4), 1470.

Plank, T. (Dec 16, 2019). Digital Twins: The 4 types and their characteristics. Retrieved from https://www.tributech.io/the-4-types-of-digital-twins/

Qi, Q., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *Ieee Access, 6*, 3585-3593.

Qi, Q., Tao, F., Zuo, Y., & Zhao, D. (2018). Digital twin service towards smart manufacturing. *Procedia Cirp, 72*, 237-242.

Schroeder, G. N., Steinmetz, C., Pereira, C. E., & Espindola, D. B. (2016). Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine, 49*(30), 12-17.

Tao, F., & Zhang, M. (2017). Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. *Ieee Access, 5*, 20418-20427.

TWI. (2021). WHAT IS DIGITAL TWIN TECHNOLOGY AND HOW DOES IT WORK? Retrieved from https://www.twi-global.com/technical-knowledge/faqs/what-is-digital-twin

Uhlemann, T. H.-J., Schock, C., Lehmann, C., Freiberger, S., & Steinhilper, R. (2017). The digital twin: Demonstrating the potential of real time data acquisition in production systems. *Procedia manufacturing, 9*, 113-120.
Vachálek, J., Bartalský, L., Rovný, O., Šišmišová, D., Morháč, M., & Lokšík, M. (2017). *The digital twin of an industrial production line within the industry 4.0 concept.* Paper presented at the 2017 21st international conference on process control (PC).
Zheng, Y., Yang, S., & Cheng, H. (2019). An application framework of digital twin and its case study. *Journal of Ambient Intelligence and Humanized Computing, 10*(3), 1141-1153.

APPENDIX A

ABB ROBOT CAD FILES ROTATION AND TRANSLATION FEATURES

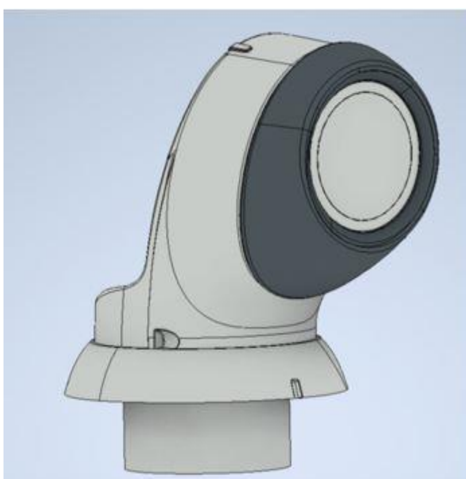
ABB Robot CAD Files without Rotation and Translation Features

Robot Link_1Left CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link1_L3_0 Group {
      children [
        DEF Link1_L3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.251314729452
              diffuseColor 0.784313738346 0.800000011921 0.78823530674
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.123264965233 0.146446228437 0.429142246007,
                0.119234108035 0.119767660511 0.455544304709,
                0.119234108035 0.147904210712 0.483975529494,
                0.123264965233 0.174582778638 0.457573470793,
                0.0406454451884 0.106054162966 0.412921631866,
                0.0665114359983 0.0389813694528 0.416945422332,
                0.0406021165705 0.0280922881157 0.401986669785,
                0.0147361257607 0.0951650816293 0.39796287932,
                0.167615824873 0.143960953458 0.473713534911,
                0.167500534737 0.143545763945 0.474751733535,
                0.167248721392 0.143931554769 0.473627780392,
                0.167142350039 0.143517078593 0.474668063014,
                0.0723526027968 0.137336141505 0.468497587149,
                0.0725565573889 0.139278855971 0.468848388922,
                0.074006978688 0.145312713431 0.475440433757,
                0.119159034843 0.166399512601 0.503658342725,
```

Robot Link_1Left VRML File

Figure A 1 STEP & VRML files of Link_1 Left (without features addition)

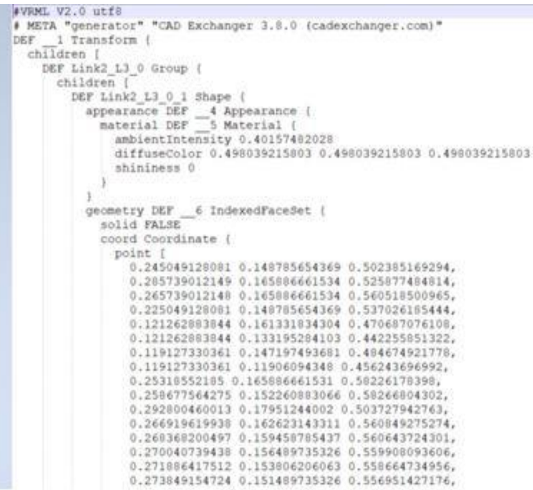


Robot Link_1Right CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link1_R3_0 Group {
      children [
        DEF Link1_R3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.251314729452
              diffuseColor 0.784313738346 0.800000011921 0.78823530674
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.112640586657 -0.0761278838382 0.498731778213,
                0.116671443856 -0.102806451764 0.472329719512,
                0.116671443856 -0.130943001965 0.500760944297,
                0.112640586657 -0.104264434039 0.527163002998,
                0.0355784405931 -0.0725177662098 0.446110441635,
                0.0715784405934 -0.0725177662098 0.383756612563,
                0.0456691211657 -0.0616286848728 0.368797860017,
                0.00966912116536 -0.0616286848728 0.431151689089,
                0.163012323288 -0.113492290421 0.503866406396,
                0.163117388106 -0.114535538738 0.503461283126,
                0.162645219585 -0.113462890264 0.50378065333,
                0.162759203417 -0.114506853441 0.50337761255,
                0.070142172355 -0.122706221421 0.482975875861,
                0.0701006708838 -0.123024364255 0.484934411442,
                0.0715919502001 -0.129328643944 0.491258836831,
                0.116746516942 -0.150432059648 0.519460301487,
                0.116724832693 -0.150288531866 0.519602324886,
```

Robot Link_1Right VRML File

Figure A 2 STEP & VRML files of Link_1 Right (without features addition)

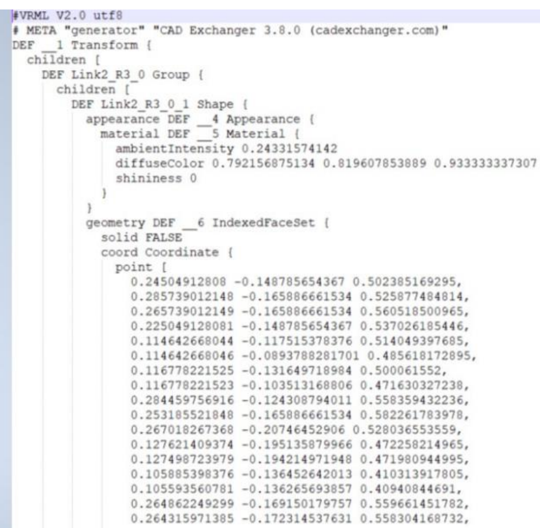


```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link2_L3_0 Group {
      children [
        DEF Link2_L3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.40157482028
              diffuseColor 0.498039215803 0.498039215803 0.498039215803
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.245049128081 0.148785654369 0.502385169294,
                0.285739012149 0.165886661534 0.525877484814,
                0.265739012148 0.165886661534 0.560518500965,
                0.225049128081 0.148785654369 0.537026185444,
                0.121262883844 0.161331834304 0.470687076108,
                0.121262883844 0.133195284103 0.442255851322,
                0.119127330361 0.147197493681 0.484674921778,
                0.119127330361 0.11906094348 0.456243696992,
                0.25318552185 0.165886661531 0.58226178398,
                0.258677564275 0.152260883066 0.58266804302,
                0.292800460013 0.17951244002 0.503727942763,
                0.266919619938 0.162623143311 0.560849275274,
                0.268368200497 0.159458785437 0.560643724301,
                0.270040739438 0.156489735326 0.559908093606,
                0.271886417512 0.153806206063 0.558664734956,
                0.273849154724 0.151489735326 0.556951427176,
```

Robot Link_2 Left CAD File

Robot Link_2 Left VRML File

Figure A 3 STEP & VRML files of Link_2 Left (without features addition)



```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link2_R3_0 Group {
      children [
        DEF Link2_R3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.24331574142
              diffuseColor 0.792156875134 0.819607853889 0.933333337307
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.24504912808 -0.148785654367 0.502385169295,
                0.285739012148 -0.165886661534 0.525877484814,
                0.265739012149 -0.165886661534 0.560518500965,
                0.225049128081 -0.148785654367 0.537026185446,
                0.114642668044 -0.117515378376 0.51404939768 5,
                0.114642668046 -0.0893788281701 0.485618172895,
                0.116778221525 -0.131649718984 0.500061552,
                0.116778221523 -0.103513168806 0.471630327238,
                0.284459756916 -0.124308794011 0.558359432236,
                0.253185521848 -0.165886661534 0.582261783978,
                0.267018267368 -0.20746452906 0.528036553559,
                0.127621409374 -0.195135879966 0.472258214965,
                0.127498723979 -0.194214971948 0.471980944995,
                0.105885398376 -0.136452642013 0.410313917805,
                0.105593560781 -0.136265693857 0.40940844691,
                0.264862249299 -0.169150179757 0.559661451782,
                0.264315971385 -0.172314537631 0.558304168732,
```

Figure A 4 STEP & VRML files of Link_2 Right (without features addition)

Robot Link_3 Left CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link3_L3_0 Group {
      children [
        DEF Link3_L3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.40157482028
              diffuseColor 0.498039215803 0.498039215803 0.498039215803
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.367594562489 0.188254769832 0.516912379902,
                0.364070899243 0.164933107803 0.539992325257,
                0.364070899243 0.186035520454 0.561315743847,
                0.367594562489 0.209357182483 0.538235798492,
                0.25047721484 0.151066952353 0.505519076644,
                0.230477214839 0.151066952353 0.540160092794,
                0.361481067981 0.186072910655 0.560958130711,
                0.358969927432 0.185788571707 0.560287432828,
                0.35661377733 0.185191143108 0.559324029014,
                0.354484208104 0.184298777422 0.558097191804,
                0.352645925637 0.183138588729 0.556644198028,
                0.351154785212 0.181745828775 0.555009196166,
                0.350056094375 0.180162815868 0.553241864922,
                0.349383236293 0.178437649052 0.551395903762,
                0.349156655417 0.17662274665 0.549527401281,
                0.349383236293 0.174773253551 0.547693130982,
                0.350056094375 0.172945365668 0.54594882624,
```

Robot Link_3 Left VRML File

Figure A 5 STEP & VRML files of Link_3 Left (without features addition)

Robot Link_3 Right CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link3_R3_0 Group {
      children [
        DEF Link3_R3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.40157482028
              diffuseColor 0.498039215803 0.498039215803 0.498039215803
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.358198127166 -0.126063671091 0.578458900848,
                0.361721790412 -0.149385333118 0.555378955493,
                0.361721790412 -0.170487745769 0.576702374083,
                0.358198127166 -0.147166083742 0.599782319437,
                0.265393014225 -0.165741246693 0.560318738971,
                0.285393014225 -0.165741246694 0.525677722819,
                0.253733931623 -0.165884526306 0.581300177403,
                0.297733931613 -0.165884526306 0.505089941887,
                0.246001327592 -0.149185841563 0.502934921939,
                0.226001327593 -0.149185841563 0.537575938091,
                0.359131959151 -0.17052513597 0.576344760947,
                0.356620818602 -0.170240797022 0.575674063064,
                0.3542646685 -0.169643368423 0.57471065925,
                0.352135099274 -0.168751002737 0.57348382204,
                0.350296816807 -0.167590814044 0.572030828264,
                0.348805676381 -0.16619805409 0.570395826402,
                0.347706985545 -0.164615041183 0.568628495158,
```

Robot Link_3 Right VRML File

Figure A 6 STEP & VRML files of Link_3 Right (without features addition)

Robot Link_4 Left CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link4_L3_0 Group {
      children [
        DEF Link4_L3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.593932747841
              diffuseColor 0.317647069693 0.341176480055 0.349019616842
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.415199316995 0.114981846922 0.53770686492,
                0.41384731574 0.125787179178 0.523763829786,
                0.411793833102 0.112021213377 0.537791771605,
                0.409953197929 0.117842271469 0.529080443119,
                0.410123840208 0.124356429694 0.521749388968,
                0.441942882162 0.163772257307 0.510499883585,
                0.441543081562 0.160418626371 0.513633601803,
                0.509497526687 0.0917452020297 0.529354752498,
                0.509048551964 0.0884174131919 0.53255483096,
                0.508692884201 0.0878010165443 0.532899899024,
                0.507034242879 0.0857305406402 0.534392097549,
                0.507529494465 0.0890656280081 0.531205490437,
                0.450625401582 0.0923989110123 0.466591442952,
                0.451190824925 0.09564411675 0.463323719705,
                0.451868821487 0.0985918281961 0.462823194208,
                0.450514442623 0.100707530672 0.4639492763,
                0.448797048347 0.0999031242177 0.465308315371,
```

Robot Link_4 Left VRML File

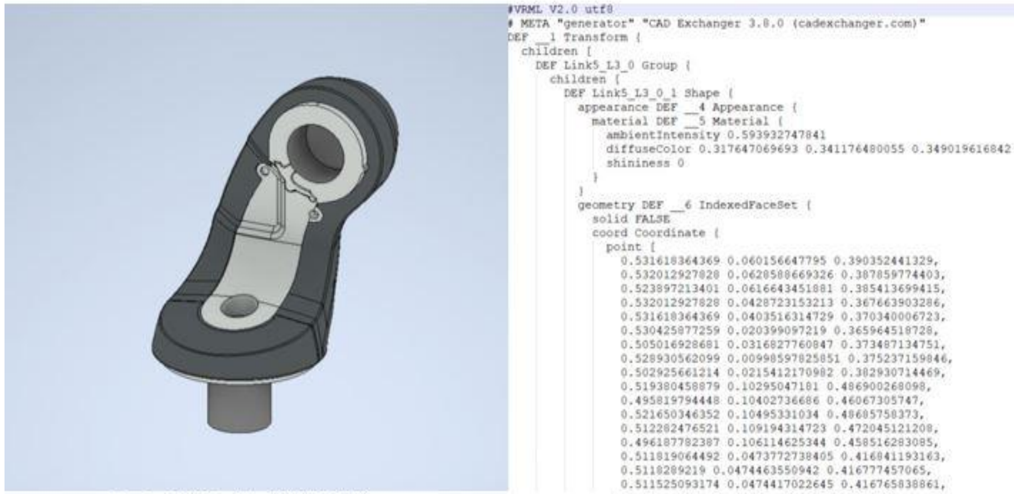
Figure A 7 STEP & VRML files of Link_4 Left (without features addition)

Robot Link_4 Right CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link4_R3_0 Group {
      children [
        DEF Link4_R3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.593932747841
              diffuseColor 0.317647069693 0.341176480055 0.349019616842
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.460658986488 -0.156912521769 0.537706864922,
                0.44154308166 -0.160418626425 0.513633601866,
                0.463844171439 -0.160028731324 0.537790638614,
                0.441937689845 -0.163728701702 0.51054058323,
                0.513289198929 -0.116840704234 0.504519351895,
                0.513830663096 -0.120068218629 0.501232069371,
                0.513627673168 -0.120462333097 0.50057709973,
                0.512542209827 -0.122185483941 0.498314990439,
                0.512046877823 -0.118964310372 0.501616692092,
                0.446286608899 -0.0636822462065 0.495010474754,
                0.445861298215 -0.060370197474 0.498232042676,
                0.44600419883 -0.0597763273368 0.501236390826,
                0.444529586351 -0.0610962478741 0.503150008275,
                0.443175741011 -0.0626980281476 0.502127800344,
                0.443655239205 -0.0658161401429 0.498931738515,
                0.384972335945 -0.223180809454 0.517977061629,
                0.355725183394 -0.210888914877 0.501091210232,
```

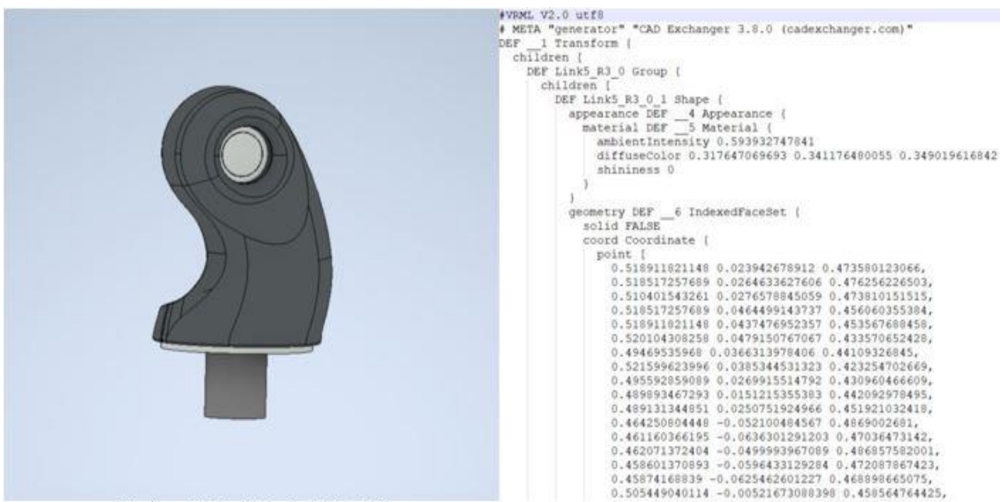
Robot Link_4 Right VRML File

Figure A 8 STEP & VRML files of Link_4 Right (without features addition)



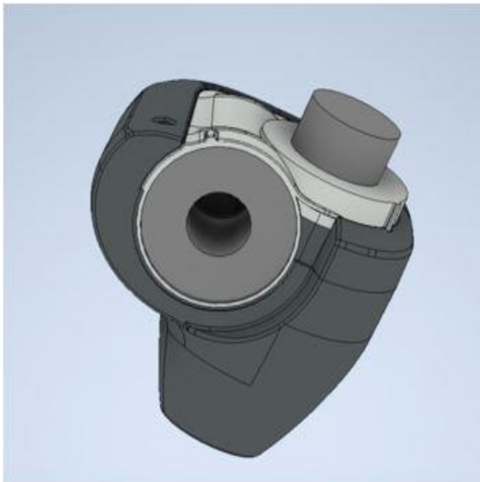
Robot Link_5 Left CAD File Robot Link_5 Left VRML File

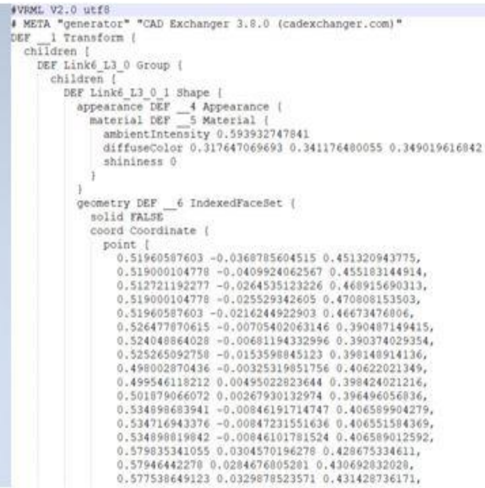
Figure A 9 STEP & VRML files of Link_5 Left (without features addition)



Robot Link_5 Right CAD File Robot Link_5 Right VRML File

Figure A 10 STEP & VRML files of Link_5 Right (without features addition)

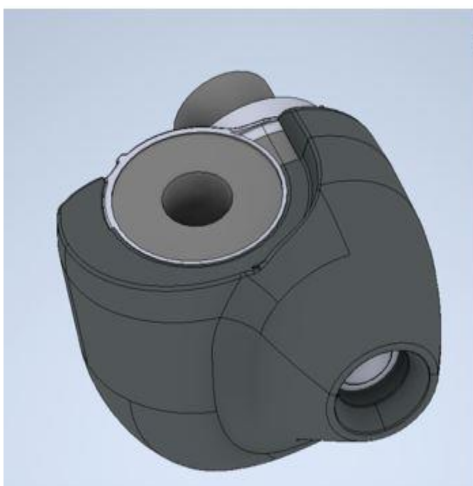




```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link6_L3_0 Group {
      children [
        DEF Link6_L3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.593932747841
              diffuseColor 0.317647069693 0.341176480055 0.349019616842
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.51960587603 -0.0368785604515 0.451320943775,
                0.519000104778 -0.0409924062567 0.455183144914,
                0.512721192277 -0.0264535123226 0.468915690313,
                0.519000104778 -0.025529342605 0.470808153503,
                0.51960587603 -0.0216244922903 0.46673476806,
                0.526477870615 -0.00705402063146 0.390487149415,
                0.524048864028 -0.00681194332996 0.390374029354,
                0.525265092758 -0.0153598845123 0.398148914136,
                0.498002870436 -0.00325319851756 0.40622021349,
                0.499546118212 0.00495022823644 0.398424021216,
                0.501879066072 0.00267930132974 0.396496056836,
                0.534898683941 -0.00846191714747 0.406589904279,
                0.534716943376 -0.00847231551636 0.406551584369,
                0.534898819842 -0.00846101781524 0.406589012592,
                0.579835341055 0.0304570196278 0.428675334611,
                0.57946442278 0.0284676805281 0.430692832028,
                0.577538649123 0.0329878523571 0.431428736171,
```

Robot Link_6 Left CAD File Robot Link_6 Left VRML File

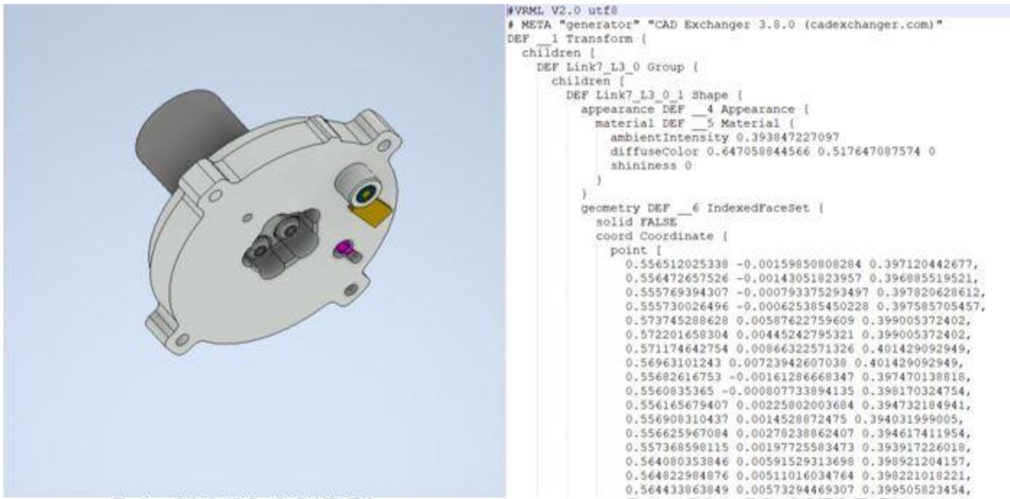
Figure A 11 STEP & VRML files of Link_6 Left (without features addition)



```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF Link6_R3_0 Group {
      children [
        DEF Link6_R3_0_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.593932747841
              diffuseColor 0.317647069693 0.341176480055 0.349019616842
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.530924309487 -0.038033444851 0.377185361727,
                0.531530080739 -0.041938295165 0.373111976284,
                0.525251168239 -0.0564771890987 0.386844521683,
                0.531530080739 -0.0574013588162 0.388736984872,
                0.530924309487 -0.0532875130111 0.392599186011,
                0.511600043788 0.0183665564836 0.415940404811,
                0.51169393224 0.0182274569015 0.416087175491,
                0.511666726917 0.0182177567001 0.416095334914,
                0.511572838972 0.0183568025361 0.415948513654,
                0.513887825808 0.0195662828956 0.415486479667,
                0.513887013094 0.0195718235168 0.415492290853,
                0.51408403156 0.01941432566 8 0.415669973092,
                0.514084994773 0.0194086616759 0.415664301244,
                0.51404976556 0.0192616654269 0.415554409776,
                0.513859479108 0.0194137877782 0.415382787588,
                0.581642901529 -0.019371159839 0.420642196561,
```

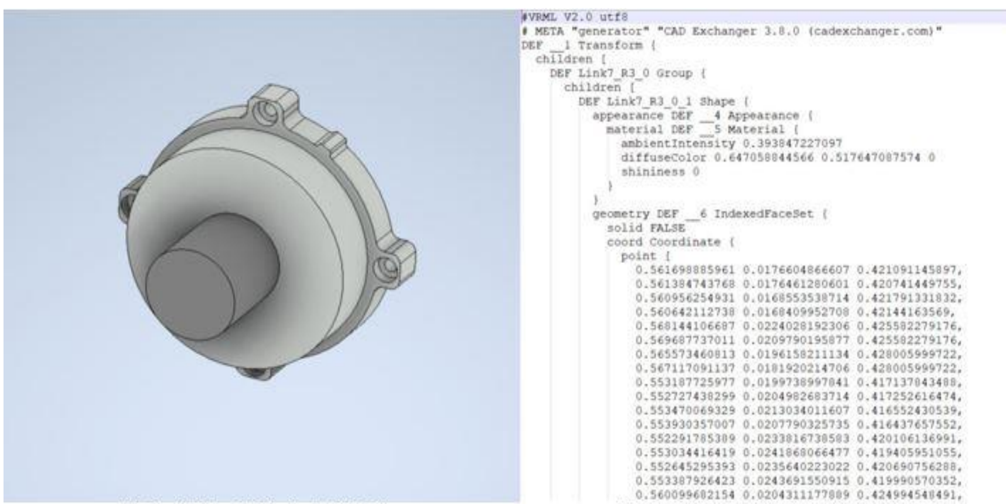
Robot Link_6 Right CAD File Robot Link_6 Right VRML File

Figure A 12 STEP & VRML files of Link_6 Right (without features addition)



Robot Link_7 Left CAD File　　Robot Link_7 Left VRML File

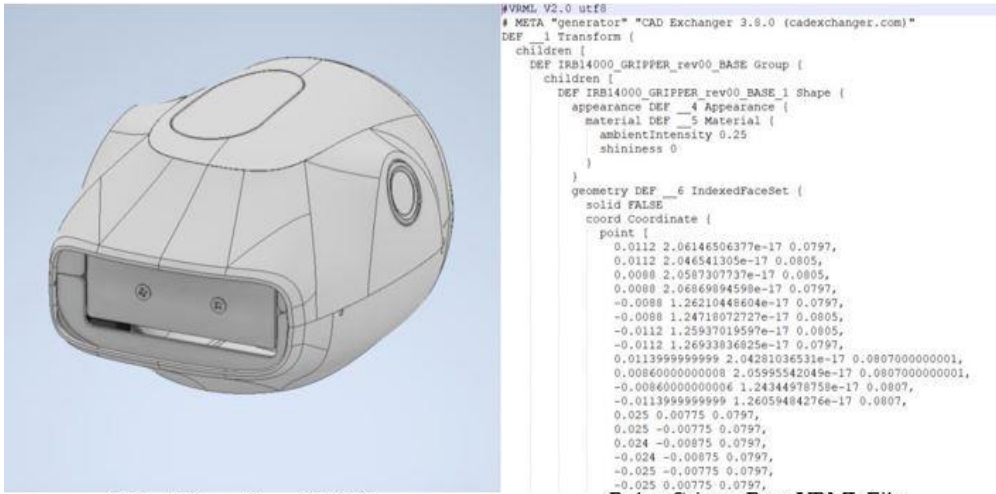
Figure A 13 STEP & VRML files of Link_7 Left (without features addition)



Robot Link_7 Right CAD File　　Robot Link_7 Right VRML File

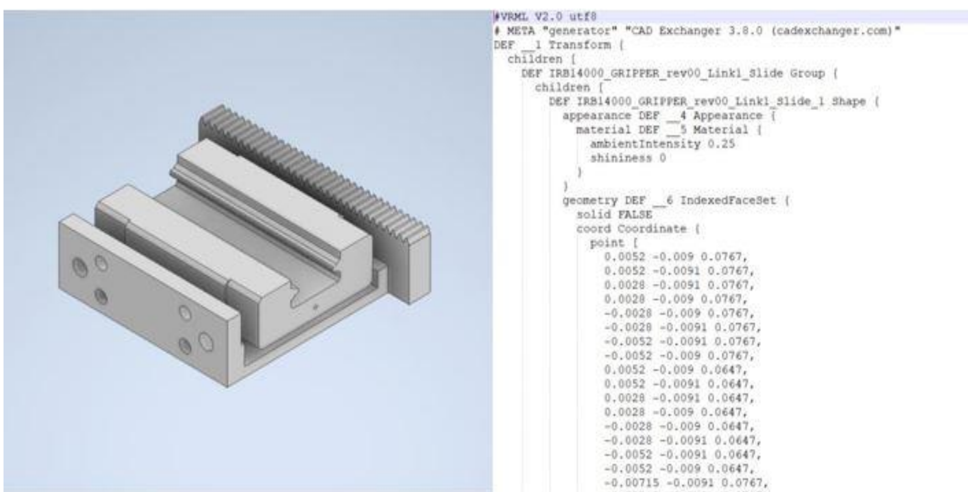
Figure A 14 STEP & VRML files of Link_7 Right (without features addition)

ABB Robot Gripper CAD Files without Rotation and Translation



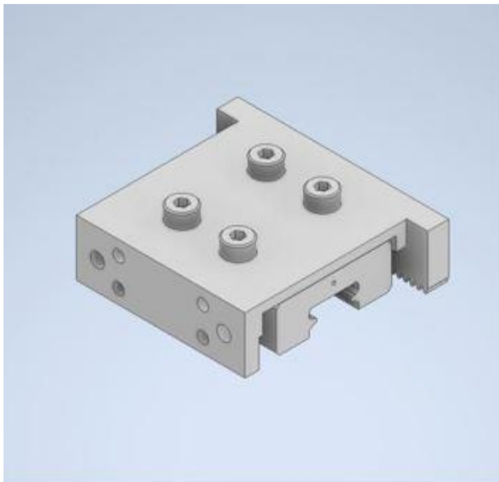
Robot Gripper Base CAD File Robot Gripper Base VRML File

Figure A 15 STEP & VRML files of Link_7 Right (without features addition)



Robot Gripper Link_1 Slide CAD File Robot Gripper Link_1 Slide VRML File

Figure A 16 STEP & VRML files of Link_7 Right (without features addition)

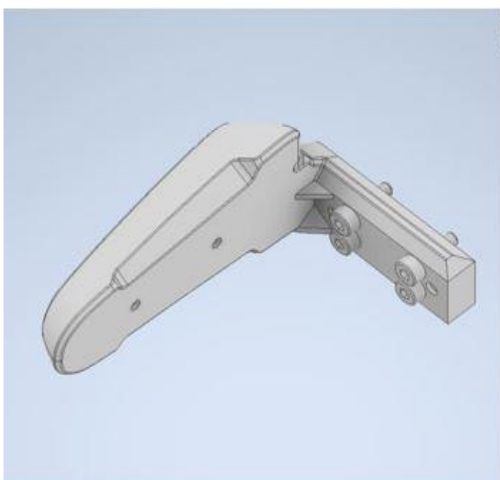


```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF IRB14000_GRIPPER_rev00_Link2_Slide Group {
      children [
        DEF IRB14000_GRIPPER_rev00_Link2_Slide_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.25
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                -0.0028 0.0091 0.0647,
                -0.0028 0.009 0.0647,
                -0.0052 0.009 0.0647,
                -0.0052 0.0091 0.0647,
                0.0052 0.0091 0.0647,
                0.0052 0.009 0.0647,
                0.0028 0.009 0.0647,
                0.0028 0.0091 0.0647,
                -0.0028 0.0091 0.0767,
                -0.0028 0.009 0.0767,
                -0.0052 0.009 0.0767,
                -0.0052 0.0091 0.0767,
                0.0052 0.0091 0.0767,
                0.0052 0.009 0.0767,
                0.0028 0.009 0.0767,
                0.0028 0.0091 0.0767,
                0.00715 0.009 0.0647,
```

Robot Gripper Link_2 Slide CAD File

Robot Gripper Link_2 Slide VRML File

Figure A 17 STEP & VRML files of Link_7 Right (without features addition)

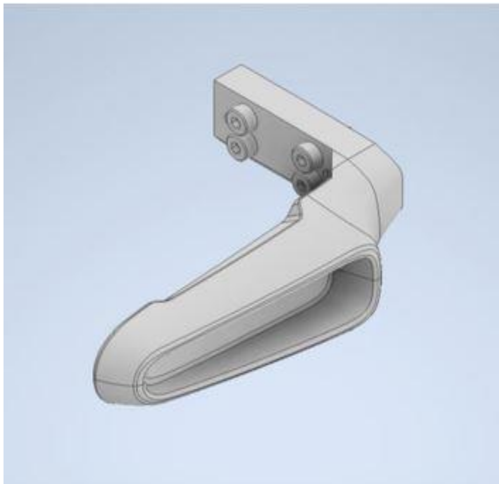


```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF IRB14000_GRIPPER_rev00_Tool1_Finger1 Group {
      children [
        DEF IRB14000_GRIPPER_rev00_Tool1_Finger1_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.25
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                -0.0125000120166 -0.00248938607087 0.0894667835719,
                -0.0125000052515 -0.00248606551817 0.092199986329,
                -0.00976679558585 -0.00249051015688 0.0894667835719,
                -0.0128000923838 0.00232035745539 0.0914753628708,
                -0.0128000924101 0.0023206971873 0.090579082522,
                -0.0155000180371 1.81598389593e-05 0.1209,
                -0.0155000199578 0.000818159838957 0.1217,
                -0.0125000199578 0.000818167041705 0.1217,
                -0.0125000180371 1.8167041707e-05 0.1209,
                -0.0125000161163 -0.000781832958291 0.1217,
                -0.0155000161163 -0.000781840161038 0.1217,
                -0.0155000180371 1.81598389593e-05 0.1059,
                -0.0155000199578 0.000818159838957 0.1067,
                -0.0125000199578 0.000818167041705 0.1067,
                -0.0125000180371 1.8167041707e-05 0.1059,
                -0.0125000161163 -0.000781832958291 0.1067,
                -0.0155000161163 -0.000781840161038 0.1067,
                -0.008 -0.00649999903963 0.0836999999827,
```

Robot Gripper Tool_1 Left Finger CAD File

Robot Gripper Tool_1 Left Finger VRML File

Figure A 18 STEP & VRML files of Link_7 Right (without features addition)



Robot Gripper Tool_1 Right Finger CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF IRB14000_GRIPPER_rev00_Tool1_Finger2 Group {
      children [
        DEF IRB14000_GRIPPER_rev00_Tool1_Finger2_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.25
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.0125000120166 0.00248938607087 0.0894667835719,
                0.0125000052515 0.00248606551816 0.092199986329,
                0.00976679558585 0.00249051015688 0.0894667835719,
                0.0155000180371 -1.81598389623e-05 0.1209,
                0.0155000199578 -0.00081815983896 0.1217,
                0.0125000199578 -0.000818167041707 0.1217,
                0.0125000180371 -1.81670417096e-05 0.1209,
                0.0125000161163 0.000781832958288 0.1217,
                0.0155000161163 0.000781840161035 0.1217,
                0.0155000180371 -1.81598389623e-05 0.1059,
                0.0155000199578 -0.00081815983896 0.1067,
                0.0125000199578 -0.000818167041707 0.1067,
                0.0125000180371 -1.81670417096e-05 0.1059,
                0.0125000161163 0.000781832958288 0.1067,
                0.0155000161163 0.000781840161035 0.1067,
                0.01 0.0065000096036 0.0836999999822,
                0.01 0.0065000096036 0.0887,
                0.008 0.00649999759908 0.0887,
```

Robot Gripper Tool_1 Right Finger VRML File

Figure A 19 STEP & VRML files of Link_7 Right (without features addition)

Robot Gripper Tool_2 Left Finger CAD File

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF IRB14000_GRIPPER_rev00_Tool2_Cup Group {
      children [
        DEF IRB14000_GRIPPER_rev00_Tool2_Cup_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.25
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                -0.031 0.0123552857148 0.036357640007,
                -0.031 0.0185 0.04375,
                -0.03 0.0133384400005 0.0365404176059,
                -0.03 0.0236615599995 0.0384595823941,
                -0.031 0.0246447142852 0.038642359993,
                -0.044 0.0185 0.03125,
                -0.044 0.0185 0.04375,
                -0.031 0.0185 0.03125,
                -0.045 0.0185 0.03175,
                -0.045 0.0185 0.04325,
                -0.044 0.0185 0.04325,
                -0.044 0.0185 0.03175,
                -0.029 0.0185 0.040863277845,
                -0.029 0.0185 0.034136722155,
                -0.03 0.0185 0.034136722155,
                -0.03 0.0185 0.040863277845,
                -0.0254 0.0185 0.04,
                -0.0254 0.0185 0.035,
```

Robot Gripper Tool_2 Left Finger VRML File

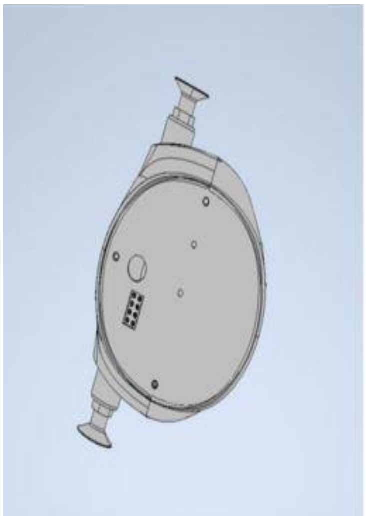
Figure A 20 STEP & VRML files of Link_7 Right (without features addition)

```
#VRML V2.0 utf8
# META "generator" "CAD Exchanger 3.8.0 (cadexchanger.com)"
DEF __1 Transform {
  children [
    DEF IRB14000_GRIPPER_rev00_Tool3_Cup Group {
      children [
        DEF IRB14000_GRIPPER_rev00_Tool3_Cup_1 Shape {
          appearance DEF __4 Appearance {
            material DEF __5 Material {
              ambientIntensity 0.25
              shininess 0
            }
          }
          geometry DEF __6 IndexedFaceSet {
            solid FALSE
            coord Coordinate {
              point [
                0.031 0.0200970408396 0.0435425127684,
                0.031 0.0185 0.04375,
                0.03 0.0198415143053 0.0425757107255,
                0.03 0.0171584856947 0.0324242892745,
                0.031 0.0169029591604 0.0314574872316,
                0.044 0.0185 0.04375,
                0.044 0.0185 0.03125,
                0.031 0.0185 0.03125,
                0.045 0.0185 0.04325,
                0.045 0.0185 0.03175,
                0.044 0.0185 0.03175,
                0.044 0.0185 0.04325,
                0.029 0.0185 0.034136722155,
                0.029 0.0185 0.040863277845,
                0.03 0.0185 0.040863277845,
                0.03 0.0185 0.034136722155,
                0.0254 0.0185 0.035,
                0.0254 0.0185 0.04,
```

Robot Gripper Tool_2 Right Finger CAD File   Robot Gripper Tool_2 Right Finger VRML File

Figure A 21 STEP & VRML files of Link_7 Right (without features addition)

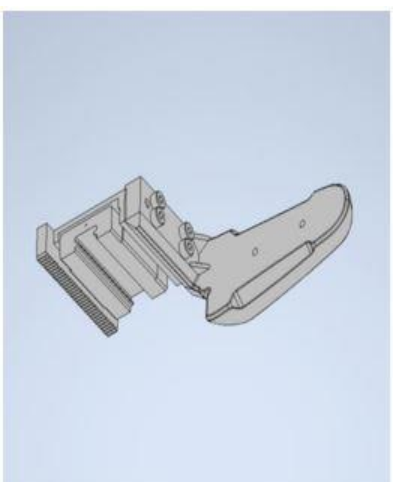
ABB Robot Gripper CAD Files with Rotation and Translation Features



Right Gripper CAD File

Right Gripper VRML File

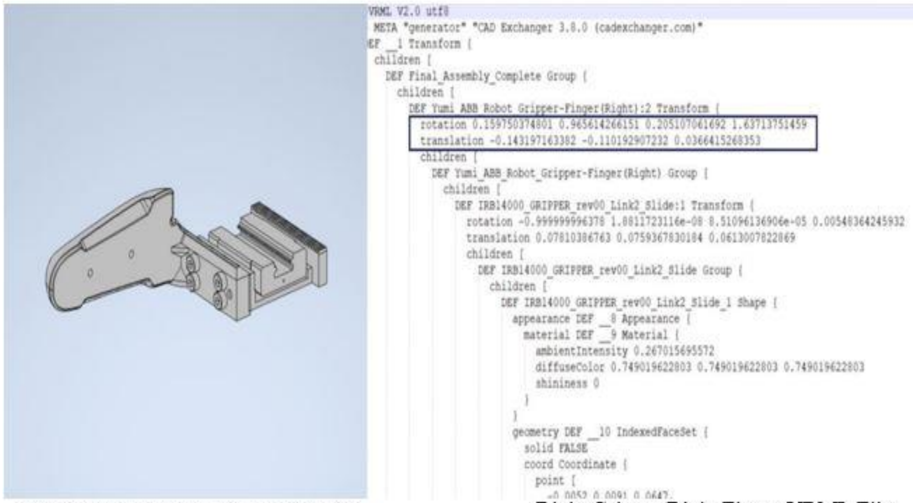
Figure A 22 Right Gripper CAD &VRML File (features added)



Left Gripper Right Finger CAD File

Left Gripper Right Finger VRML File

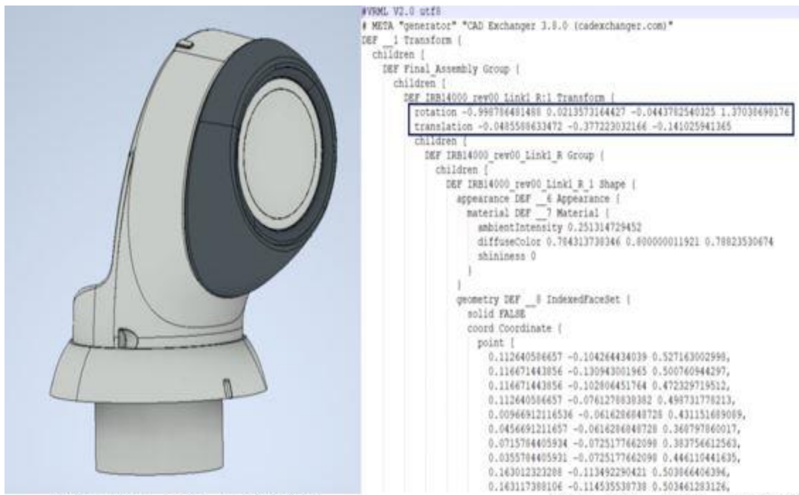
Figure A 23 Left-Gripper Right Fingers CAD &VRML File (features added)



Right Gripper Right Finger CAD File　　Right Gripper Right Finger VRML File

Figure A 24 Right-Gripper Right Finger CAD &VRML File (features added)

ABB Robot Link CAD Files with Rotation and Translation Features



Robot Link_1 Right CAD File Robot Link_1 Right VRML File

Figure A 25 CAD & VRML file of Link_1 Right (with features)

Robot Link_2 Left CAD File Robot Link_2 Left VRML File

Figure A 26 CAD & VRML files of Link_2 Left (with features)

Robot Link_2 Right CAD File

Robot Link_2 Right VRML File

Figure A 27 CAD & VRML files of Link_2 Right (with features)

Robot Link_3Left CAD File

Robot Link_3 Left VRML File

Figure A 28 CAD & VRML files of Link_3 Left (with features)

Robot Link_3 Right CAD File　　Robot Link_3 Right VRML File

Figure A 29 CAD & VRML files of Link_3 Right (with features)

Robot Link_4 Left CAD File　　Robot Link_4 Left VRML File

Figure A 30 CAD & VRML files of Link_4 Left (with features)

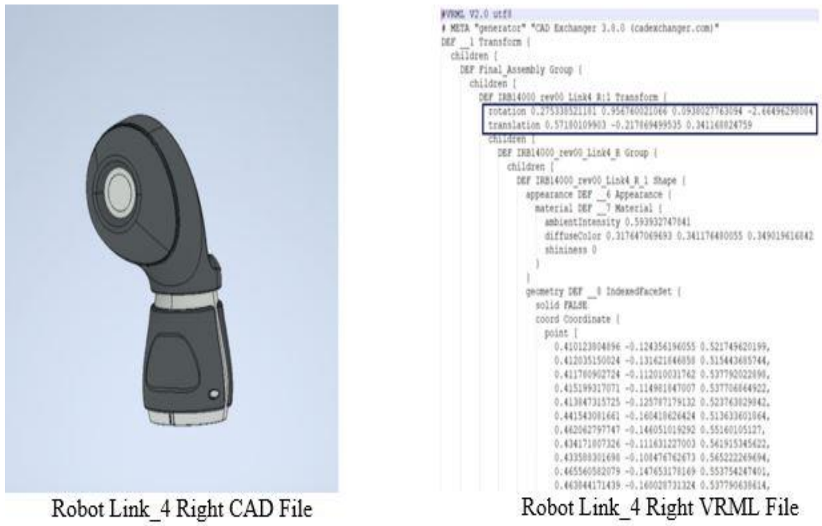
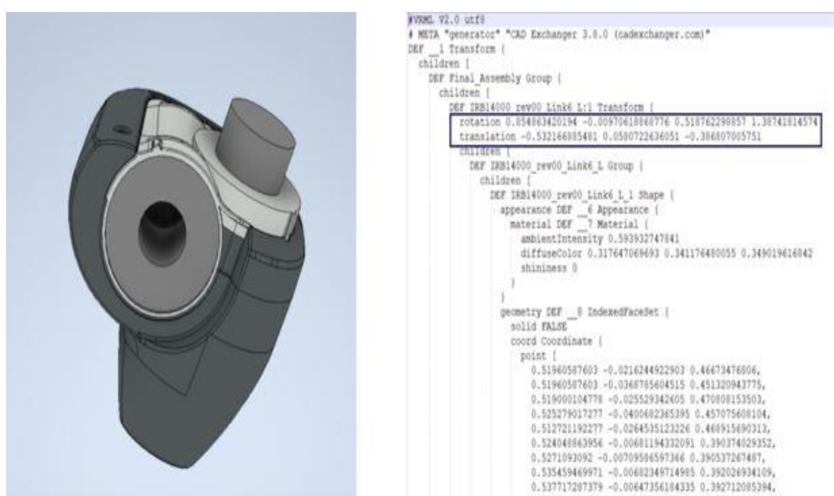


Figure A 31 CAD & VRML files of Link_4 Right (with features)

Figure A 32 CAD & VRML files of Link_5 Left (with features)

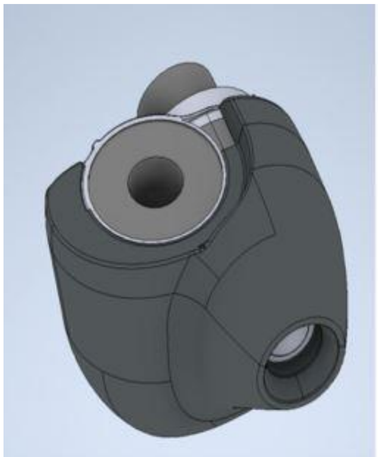
Robot Link_5 Right CAD File Robot Link_5 Right VRML File

Figure A 33 CAD & VRML files of Link_5 Right (with features)



Robot Link_6 Left CAD File Robot Link_6 Left VRML File

Figure A 34 CAD & VRML files of Link_6 Left (with features)

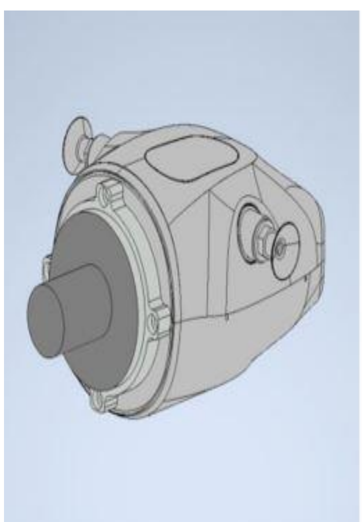


Robot Link_6 Right CAD File

Robot Link_6 Right VRML File

Figure A 35 CAD & VRML files of Link_6 Right (with features)

After modifying the link 1 to 6, Link_7 is added to the Gripper Fingerless part as they are fixed to each other. So, the features are added to the Gripper and the link_7 after combining them.



Link_7 Left with Gripper CAD File

Link_7 Left with Gripper VRML File

Figure A 36 CAD & VRML files of Link_7 Left with Fingerless - Gripper (with features)

APPENDIX B

ROBOT FILES ROTATION AND TRANSLATION CO-ORDINATE DATA

ABB Robot Part Files Rotation and Translation Co-ordinate Data

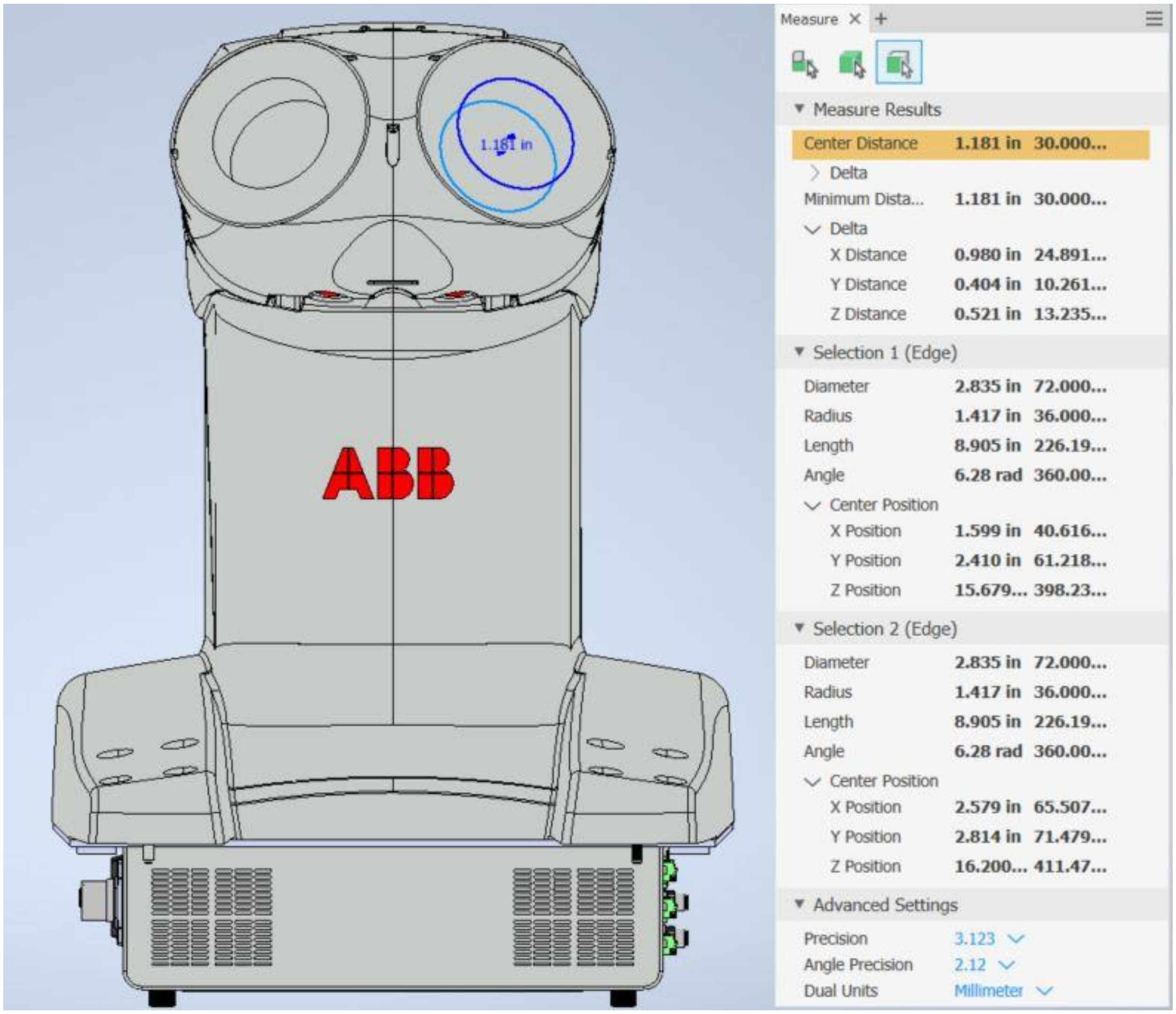


Figure B 1 Base (Right Hole) coordinate Data for rotation & Translation

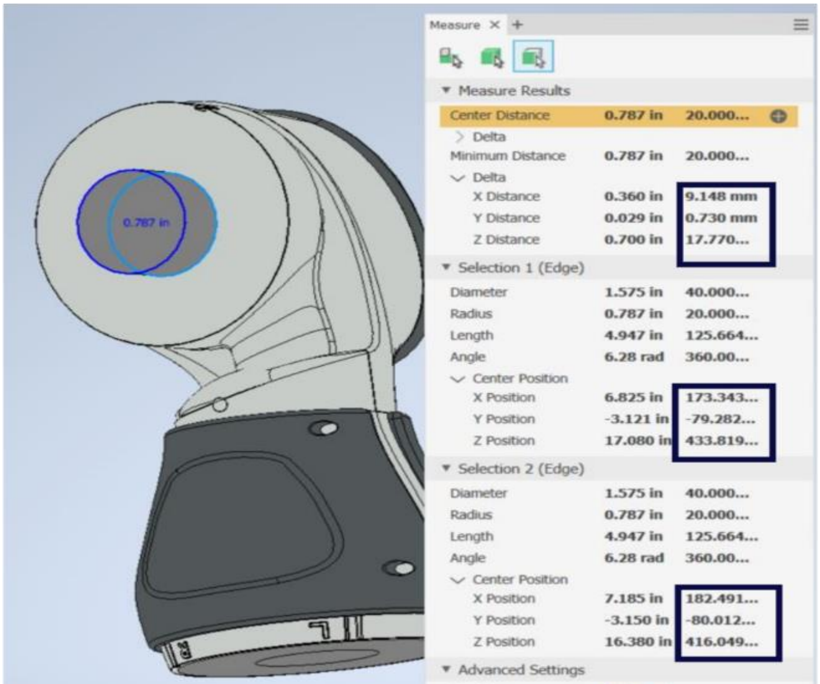


Figure B 2 Link_2 Left coordinate Data for Rotation & Translation

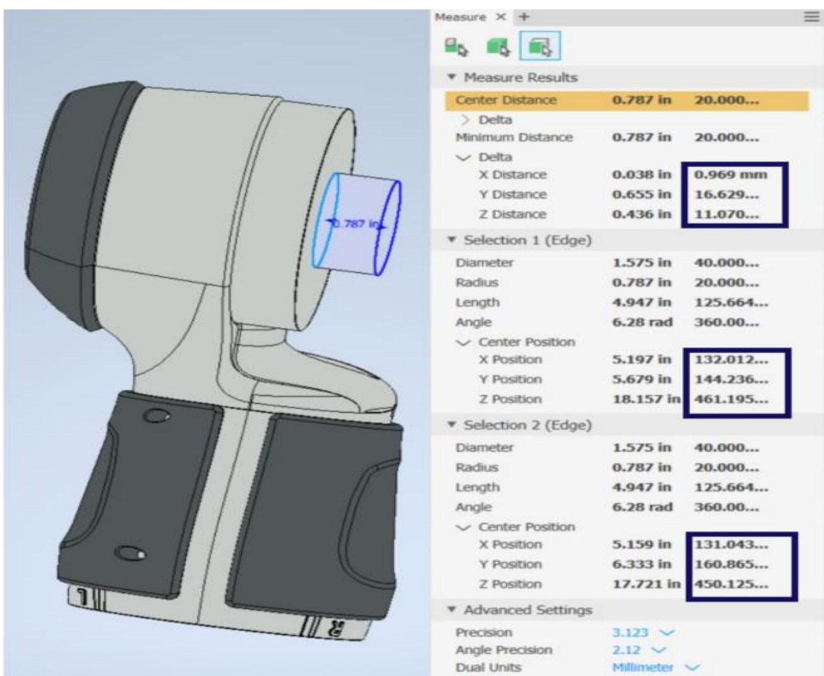


Figure B 3 Link_2 Right coordinate Data for Rotation & Translation

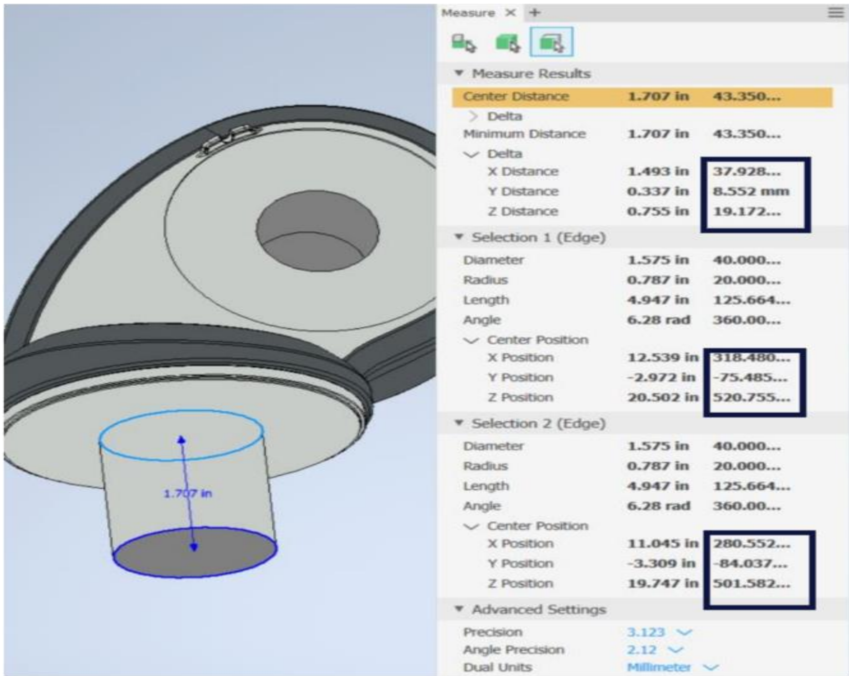


Figure B 4 Link_3 Left coordinate Data for Rotation & Translation

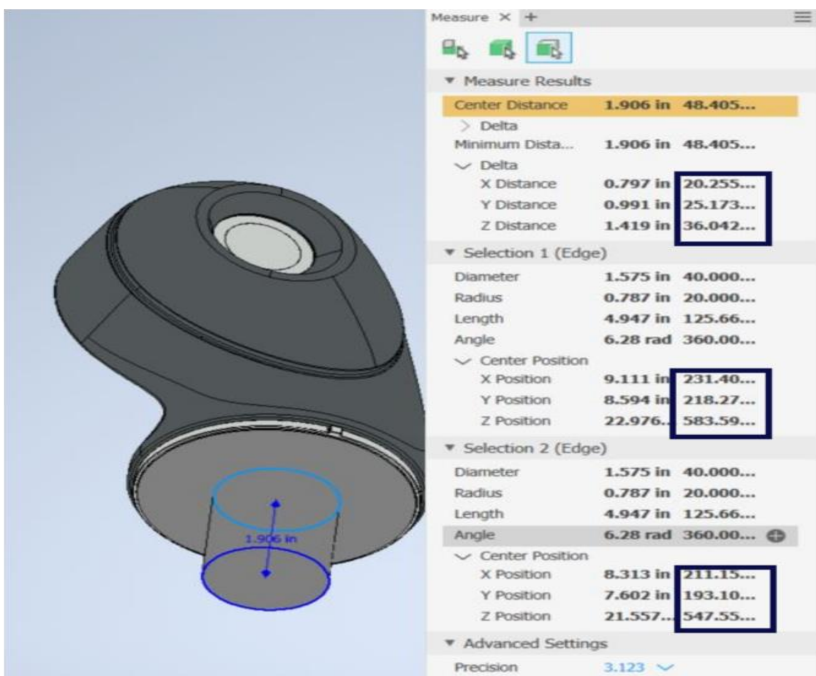


Figure B 5 Link_3 Right coordinate Data for Rotation & Translation

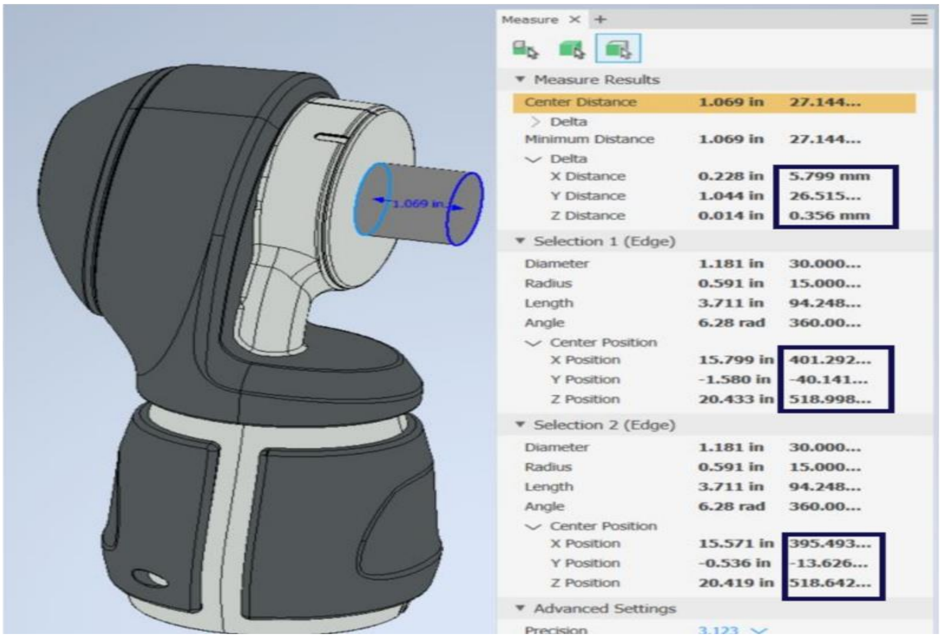


Figure B 6 Link_4 Left coordinate Data for Rotation & Translation

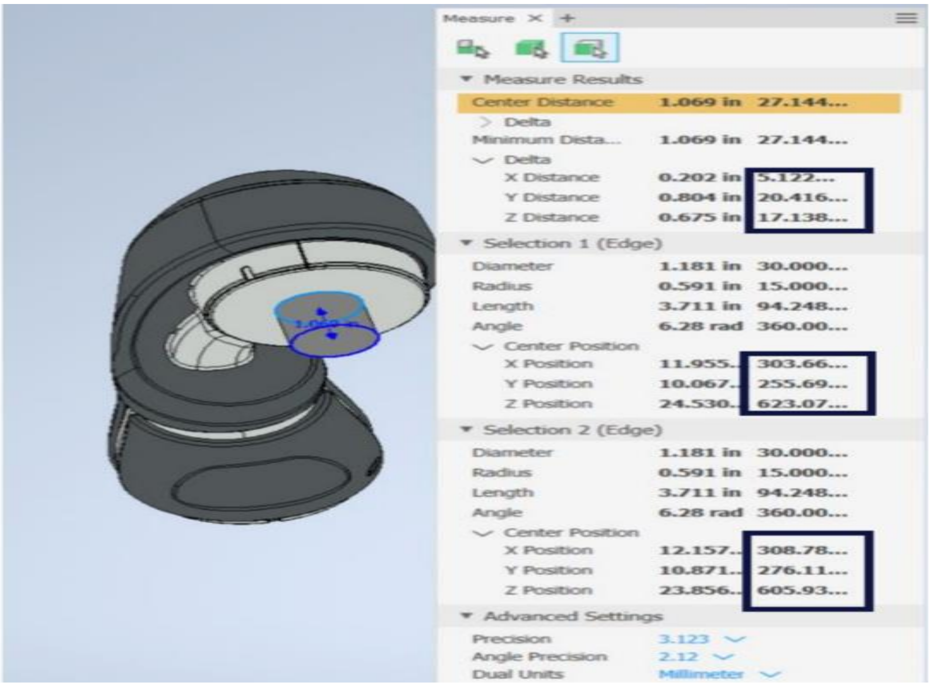


Figure B 7 Link_4 Right coordinate Data for Rotation & Translation

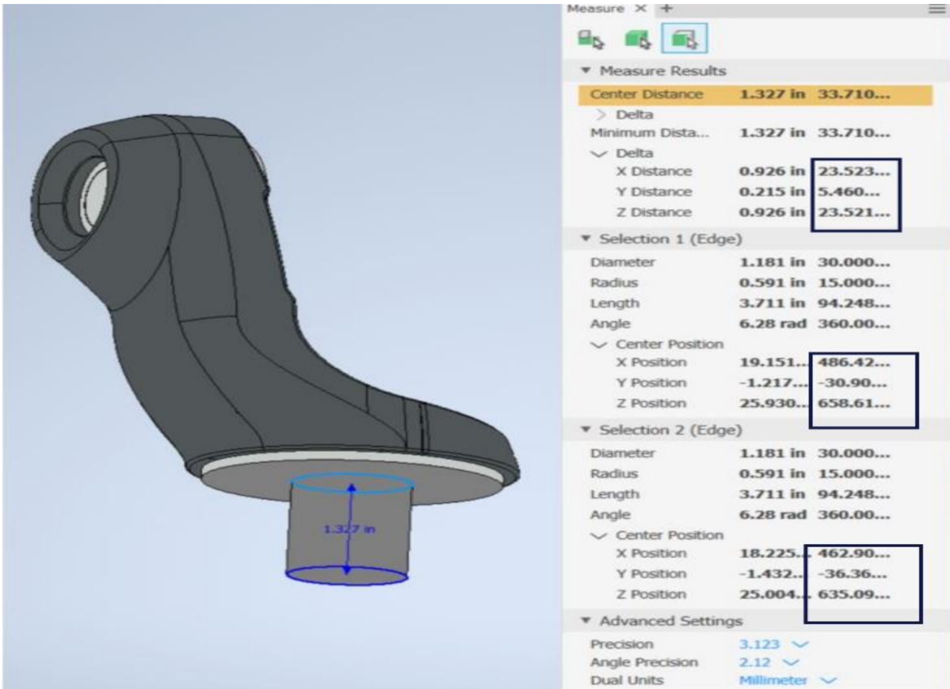


Figure B 8 Link_5 Left coordinate Data for Rotation & Translation

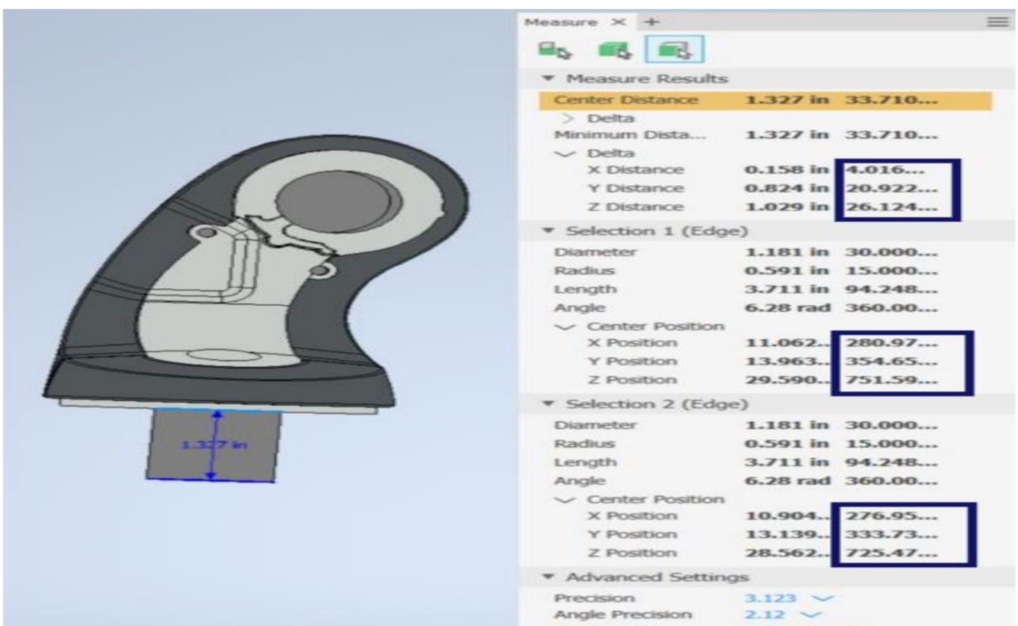


Figure B 9 Link_5 Right coordinate Data for Rotation & Translation

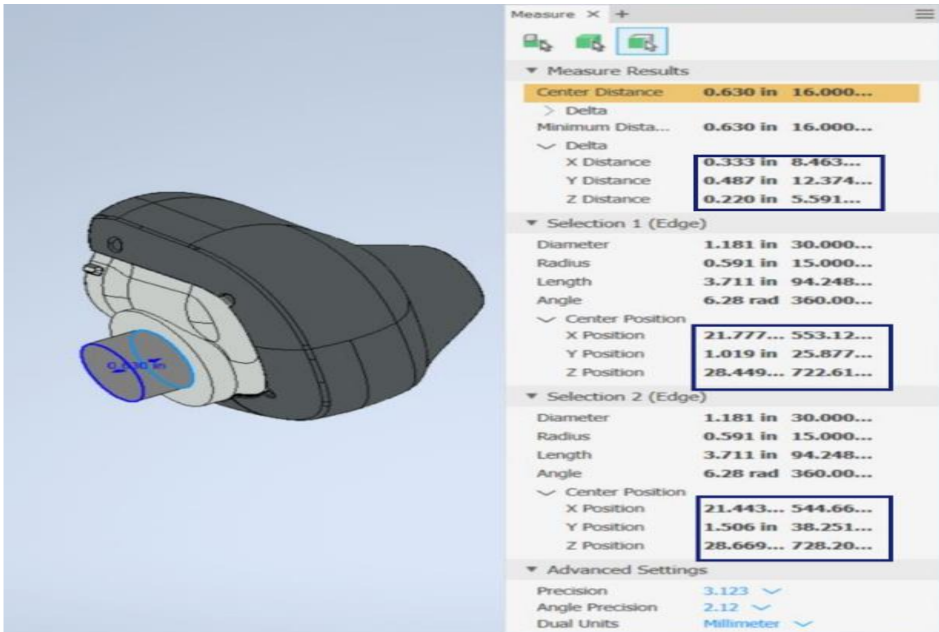


Figure B 10 Link_6 Left coordinate Data for Rotation & Translation

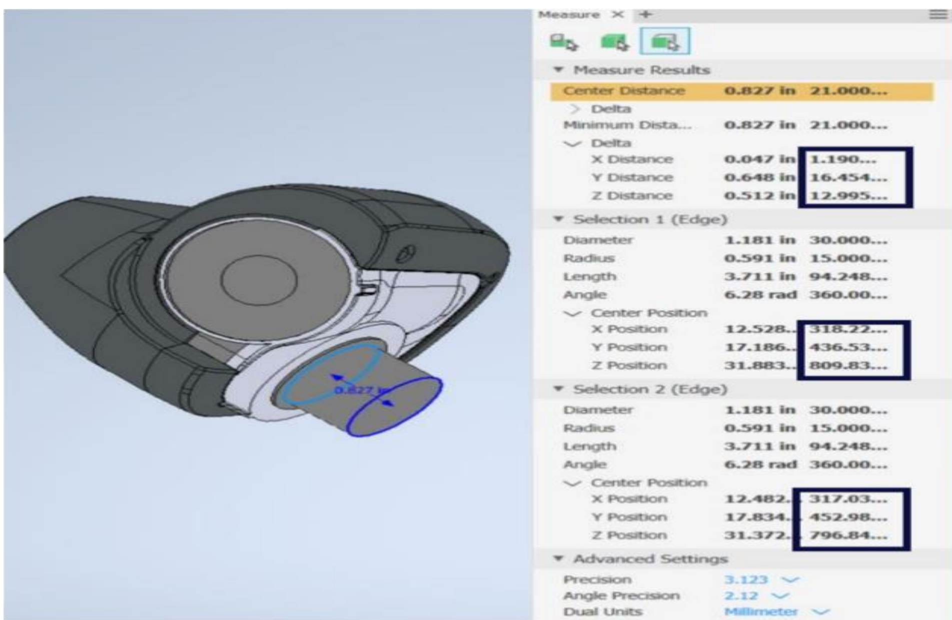


Figure B 11 Link_6 Right coordinate Data for Rotation & Translation

ABB Robot Gripper Files Rotation and Translation Co-ordinate Data

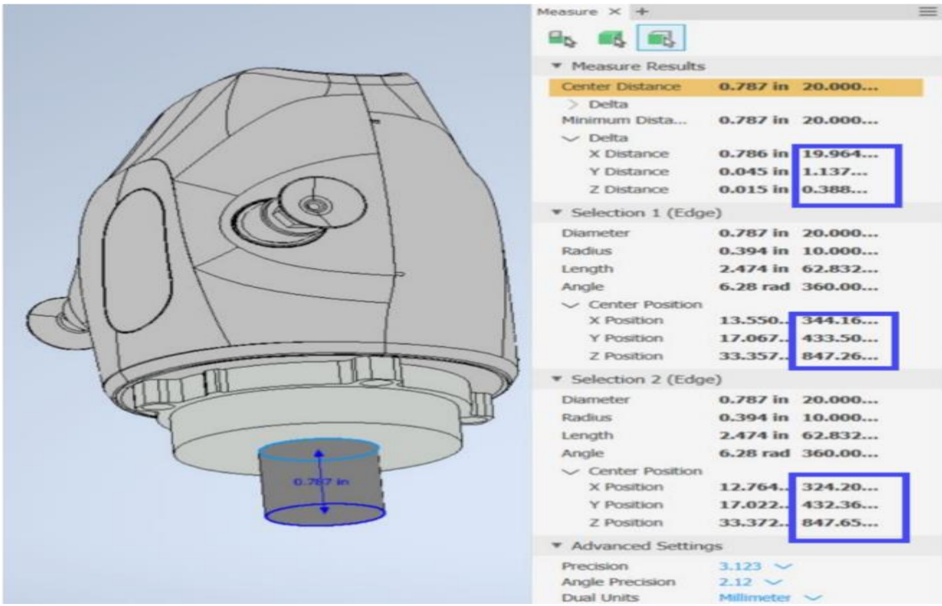


Figure B 12 Link_7 Right & Fingerless Gripper coordinate Data for Rotation & Translation

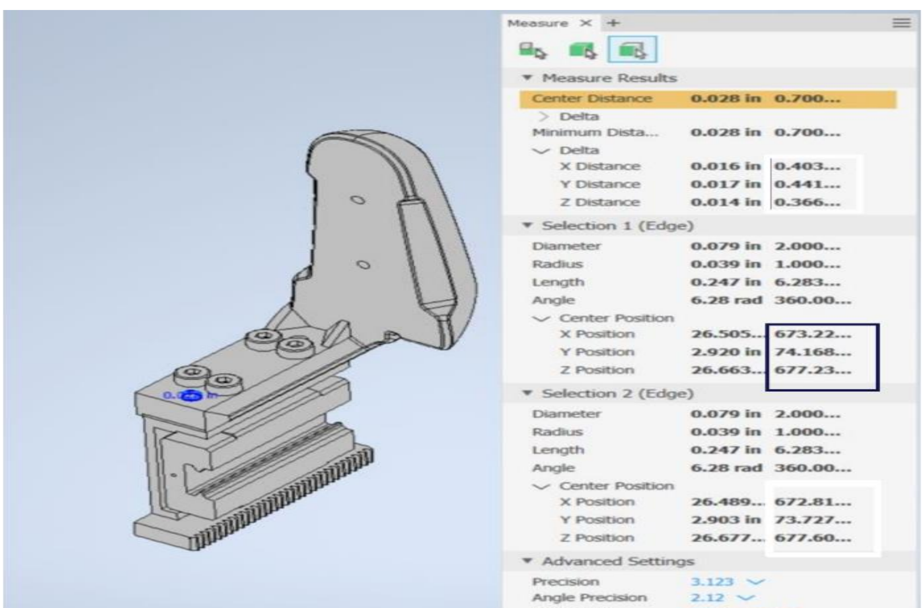


Figure B 13 Left Gripper_Right Finger (close) coordinate Data for Rotation & Translation

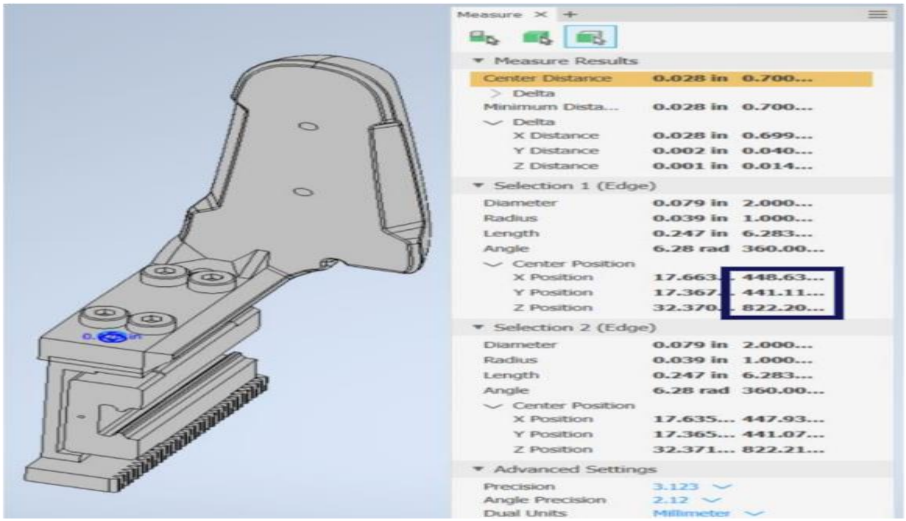


Figure B 14 Right Gripper Left Finger (close) coordinate Data for Rotation & Translation

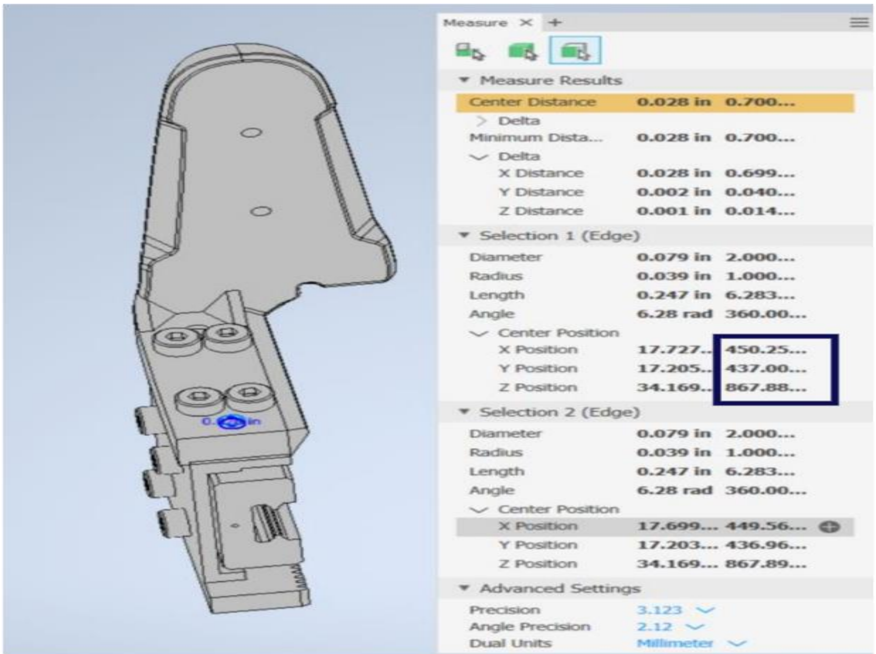


Figure B 15 Right Gripper_Right Finger (close) coordinate Data for Rotation & Translation

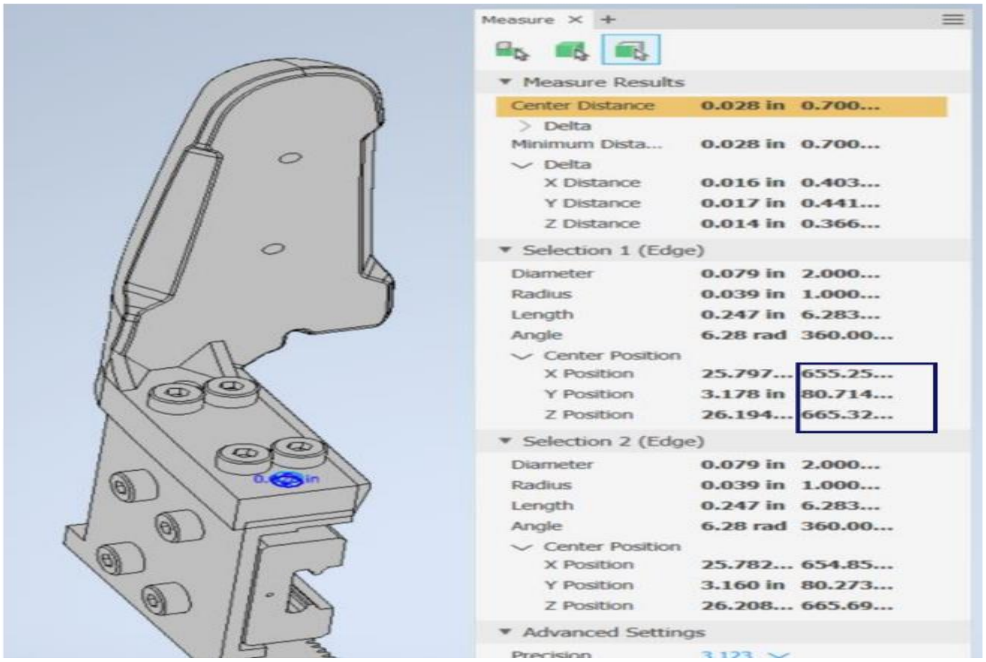


Figure B 16 Left Gripper Left Finger (Open) coordinate Data for Rotation & Translation

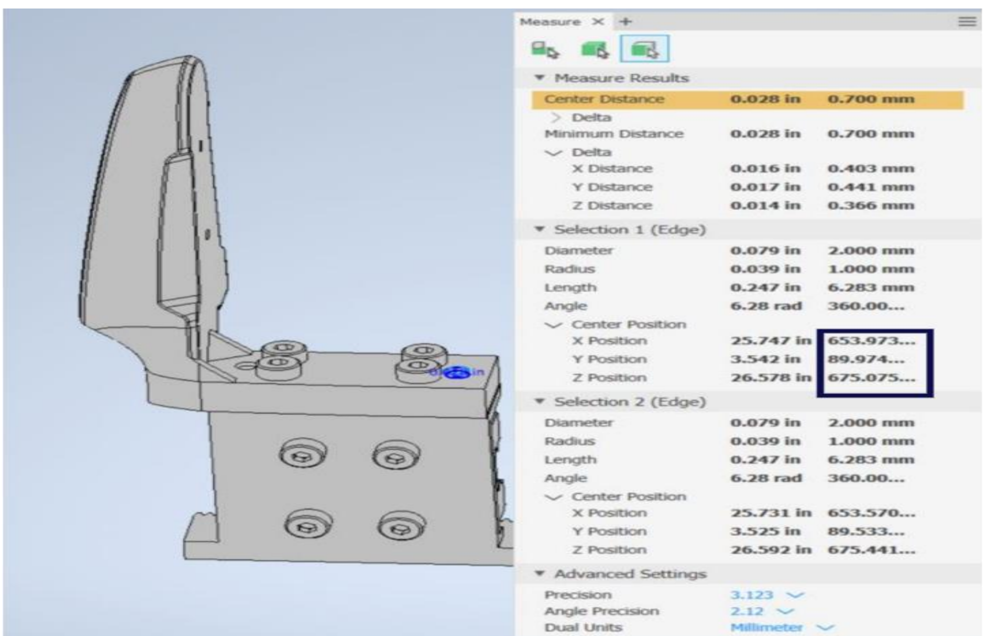


Figure B 17 Left Gripper Right Finger (Open) coordinate Data for Rotation & Translation

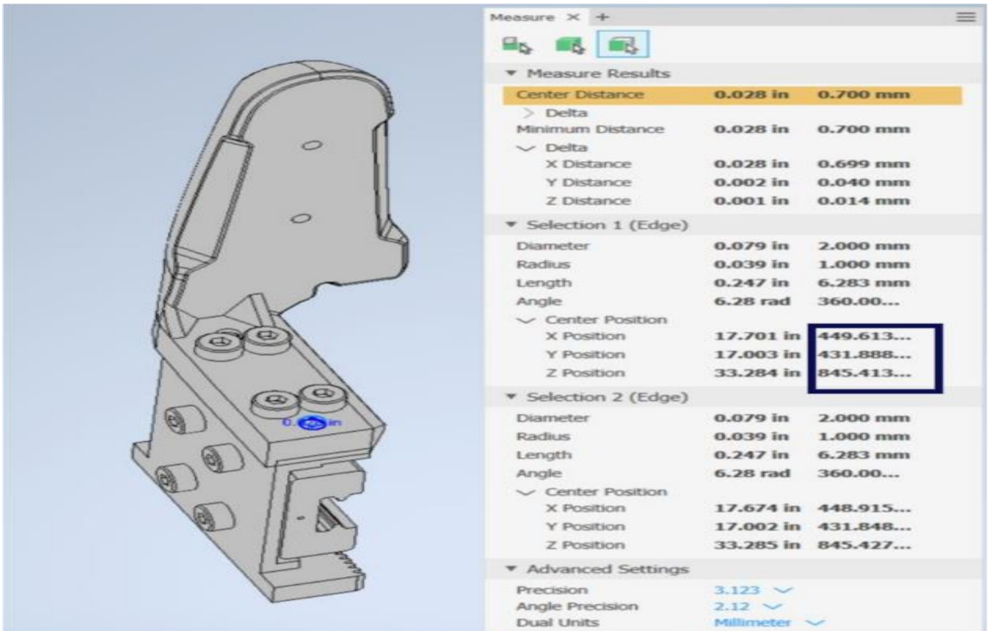


Figure B 18 Right Gripper Left Finger (Open) coordinate Data for Rotation & Translation

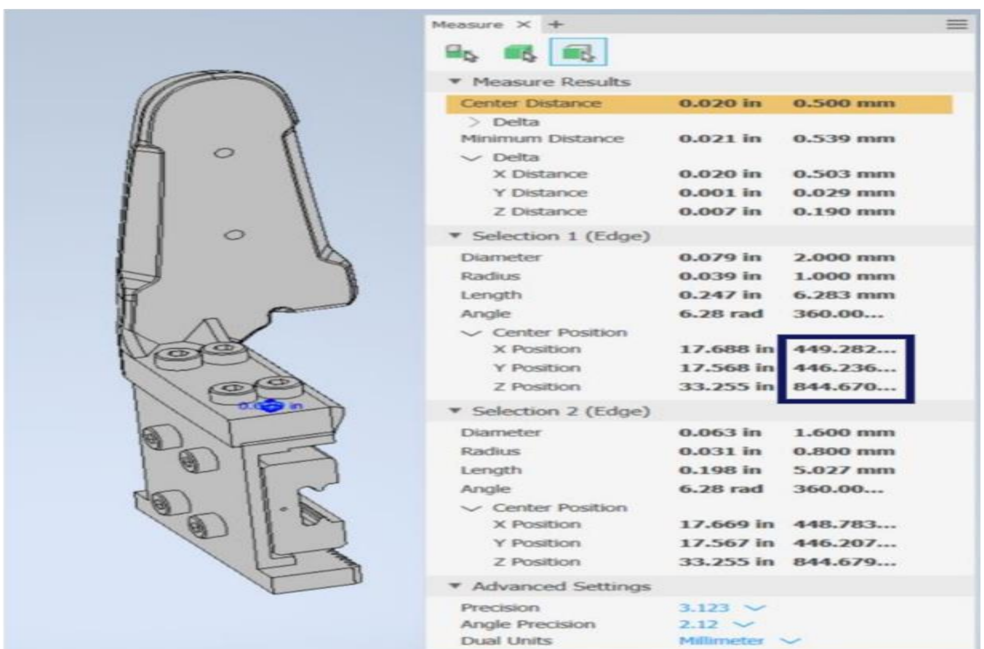


Figure B 19 Right Gripper Right Finger (Open) coordinate Data for Rotation & Translation

Table B5 ABB Robot Rotation & Translation Data

ROBOT LEFT ARM				ROBOT RIGHT ARM		
X	Y	Z	ROTATION & TRANSLATION	X	Y	Z
LINK_1 LEFT				LINK_1 RIGHT		
67.031	-72.107	412.282	PART ORIGIN (SELECTION 1) (mm)	67.031	72.107	412.282
40.616	-61.218	398.236	NEXT LINK JOINT (SELECTION 2) (mm)	40.616	61.218	398.236
26.415	-10.889	14.046	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	26.415	10.889	14.046
0.040616	-0.06122	0.398236	TRANSLATION (SELECTION 2)/1000 (m)	40.616	61.218	398.236
LINK_2 LEFT				LINK_2 RIGHT		
173.343	-79.282	433.819	PART ORIGIN (SELECTION 1) (mm)	132.012	144.236	461.195
182.491	-80.012	416.049	NEXT LINK JOINT (SELECTION 2) (mm)	131.043	160.865	450.125
-9.148	0.73	17.77	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	0.969	-16.629	11.07
0.141875	-0.01879	0.017813	TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.090427	0.099647	0.051889
LINK_3 LEFT				LINK_3 RIGHT		
318.48	-75.485	520.755	PART ORIGIN (SELECTION 1) (mm)	231.407	218.277	583.592
280.552	-84.037	501.582	NEXT LINK JOINT (SELECTION 2) (mm)	211.152	193.103	547.55
37.928	8.552	19.173	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	20.255	25.174	36.042
0.098061	-0.00403	0.085533	TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.080109	0.032238	0.097425
LINK_4 LEFT				LINK_4 RIGHT		
401.292	-40.141	518.998	PART ORIGIN (SELECTION 1) (mm)	303.666	255.698	623.071
395.493	-13.626	518.642	NEXT LINK JOINT (SELECTION 2) (mm)	308.789	276.114	605.933
5.799	-26.515	0.356	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	-5.123	-20.416	17.138
0.114941	0.070411	0.01706	TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.097637	0.083011	0.058383
LINK_5 LEFT				LINK_5 RIGHT		
486.426	-30.908	658.615	PART ORIGIN (SELECTION 1) (mm)	280.972	354.656	751.594
462.903	-36.368	635.094	NEXT LINK JOINT (SELECTION 2) (mm)	276.956	333.733	725.47
23.523	5.46	23.521	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	4.016	20.923	26.124
0.06741	-0.02274	0.116452	TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	-0.03183	0.057619	0.119537
LINK_6 LEFT				LINK_6 RIGHT		
553.125	25.877	722.611	PART ORIGIN (SELECTION 1) (mm)	318.222	436.532	809.838
544.662	38.251	728.201	NEXT LINK JOINT (SELECTION 2) (mm)	317.032	452.986	796.844
8.463	-12.374	-5.59	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	1.19	-16.454	12.994
0.081759	0.074619	0.093107	TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.040076	0.119253	0.071374
LINK_7 LEFT (GRIPPER BASE)				LINK_7 RIGHT (GRIPPER BASE)		
598.105	23.532	721.477	PART ORIGIN (SELECTION 1) (mm)	344.167	433.506	847.264
582.568	6.537	735.576	NEXT LINK JOINT (SELECTION 2) (mm)	324.203	432.369	847.652
15.537	16.995	-14.099	ROTATION COORDINATES (SEL. 1 - SEL. 2) (mm)	19.964	1.137	-0.388
0.037906	-0.03171	0.007375	TRANSLATION [(SEL. 2 - SEL 2(PREV. LINK)/1000] (m)	0.007171	-0.02062	0.050808
LEFT GRIPPER_LEFT FINGER				RIGHT GRIPPER_LEFT FINGER		
636.007	96.52	663.168	CLOSED (SELECTION 1) (mm)	448.636	441.117	822.2
645.6315	88.617	664.2475	HALF OPEN (SELECTION 1) (mm)	449.1245	436.5025	833.8065
655.256	80.714	665.327	OPEN (SELECTION 1) (mm)	449.613	431.888	845.413
LEFT GRIPPER_RIGHT FINGER				RIGHT GRIPPER_RIGHT FINGER		
673.222	74.168	677.235	CLOSED (SELECTION 1) (mm)	450.259	437.007	867.883
663.5975	82.071	676.155	HALF OPEN (SELECTION 1) (mm)	449.7705	441.6215	856.2765
653.973	89.974	675.075	OPEN (SELECTION 1) (mm)	449.282	446.236	844.67

APPENDIX C

3D WORLD EDITOR LINK-ROTATION AND TRANSLATION DATA

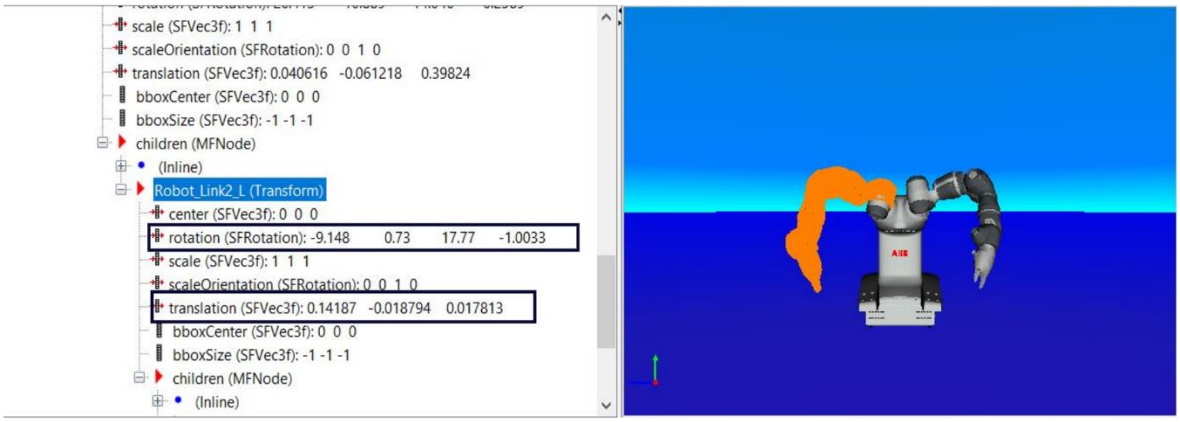


Figure C 1 Link_2 Left adjusted in 3D World Editor using Rotation & Translation Data

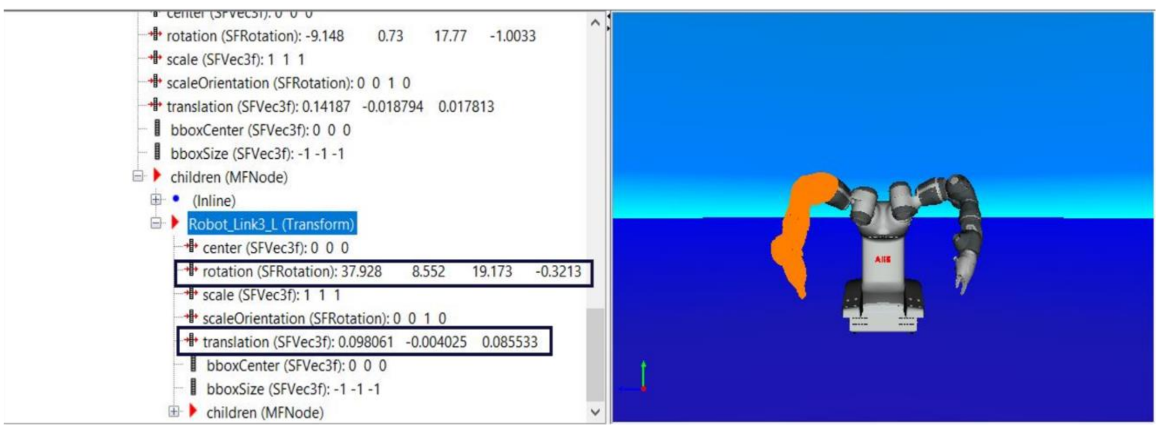


Figure C 2 Link_3 Left adjusted in 3D World Editor using Rotation & Translation Data

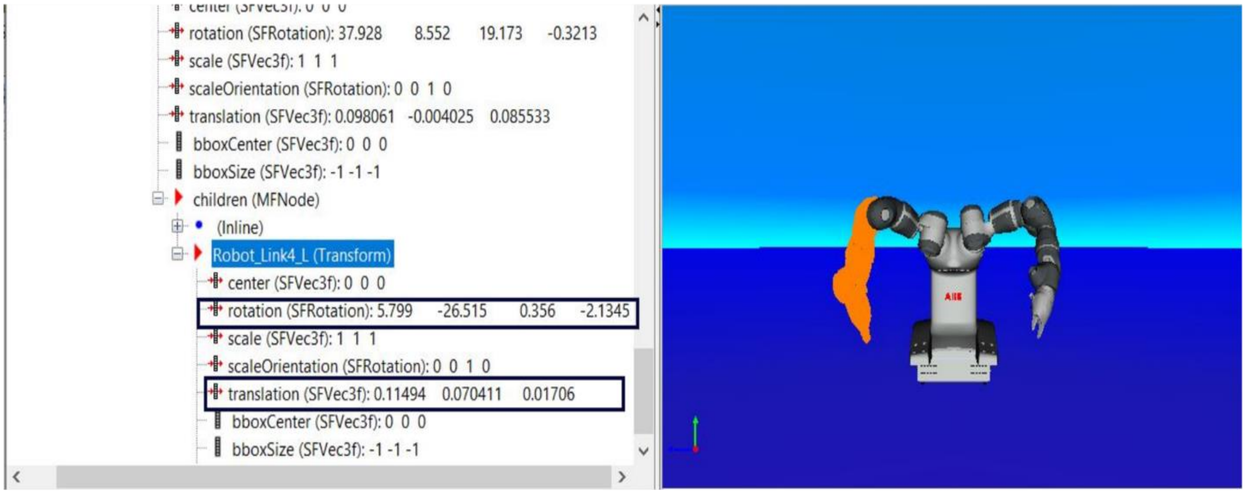


Figure C 3 Link_4 Left adjusted in 3D World Editor using Rotation & Translation Data

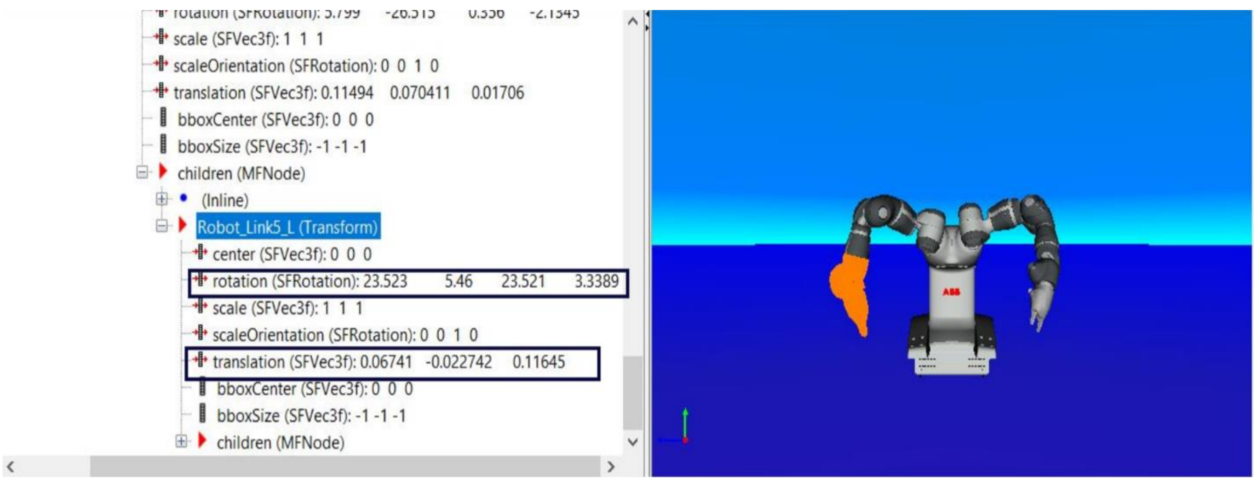


Figure C 4 Link_5 Left adjusted in 3D World Editor using Rotation & Translation Data

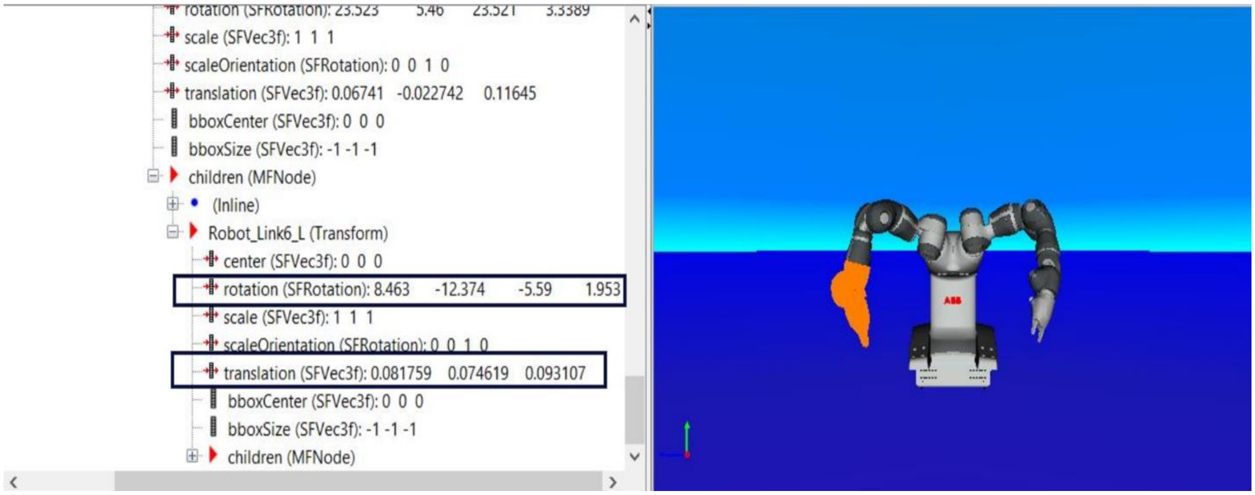


Figure C 5 Link_6 Left adjusted in 3D World Editor using Rotation & Translation Data

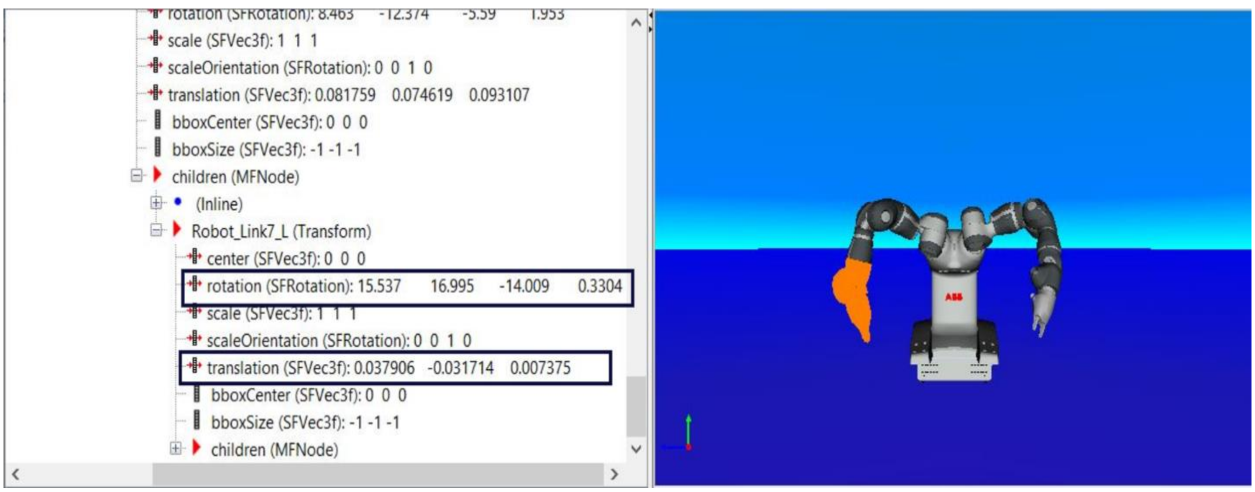


Figure C 6 Link_7 Left adjusted in 3D World Editor using Rotation & Translation Data

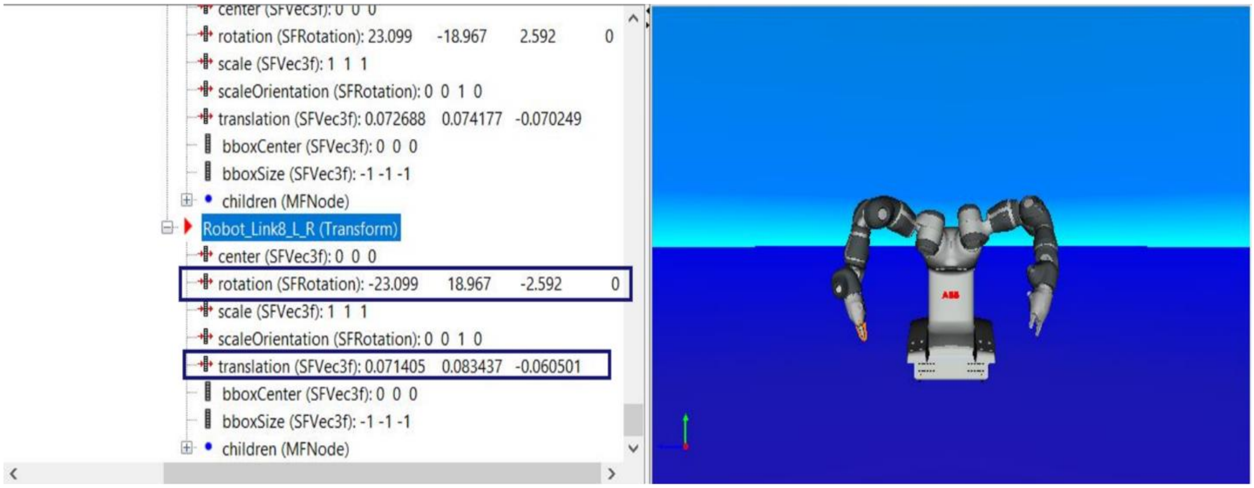


Figure C 7 Link_8 Left (Right Finger) adjusted using Rotation & Translation Data

Figure C 8 Link_2 Right adjusted in 3D World Editor using Rotation & Translation Data

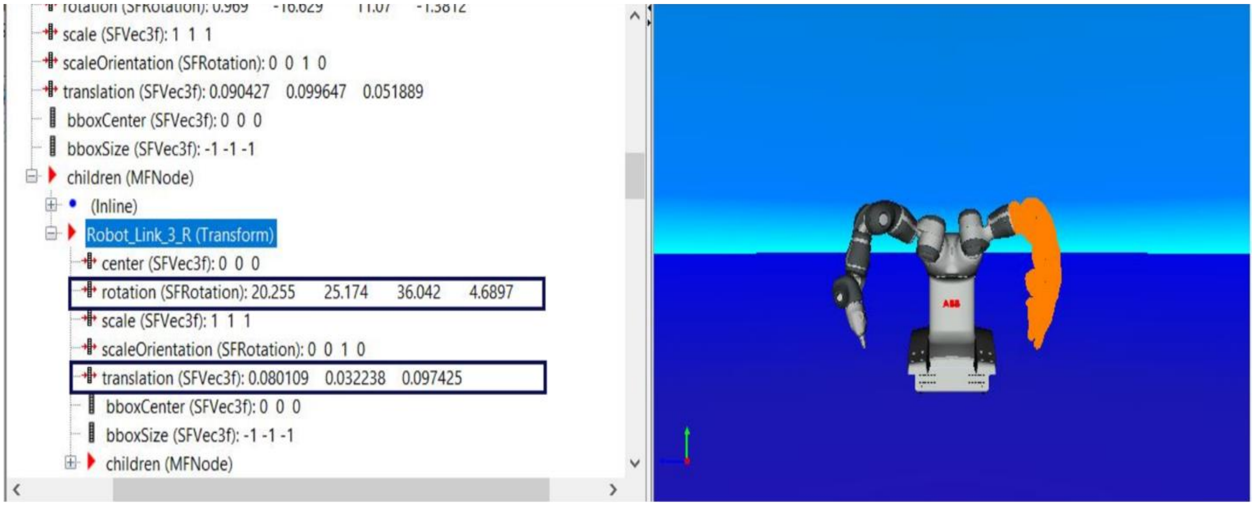


Figure C 9 Link_3 Right adjusted in 3D World Editor using Rotation & Translation Data

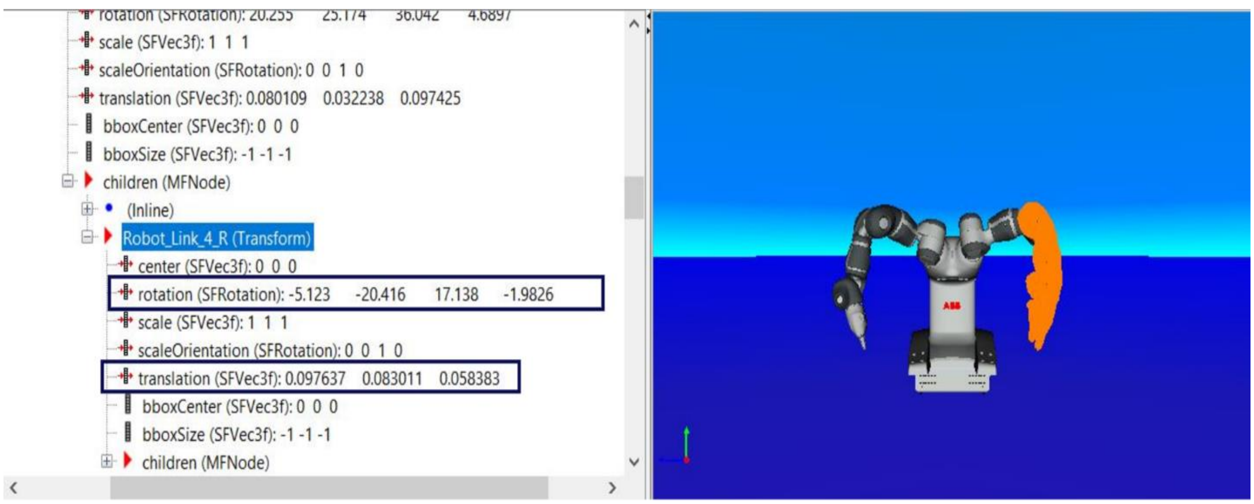


Figure C 10 Link_4 Right adjusted in 3D World Editor using Rotation & Translation Data

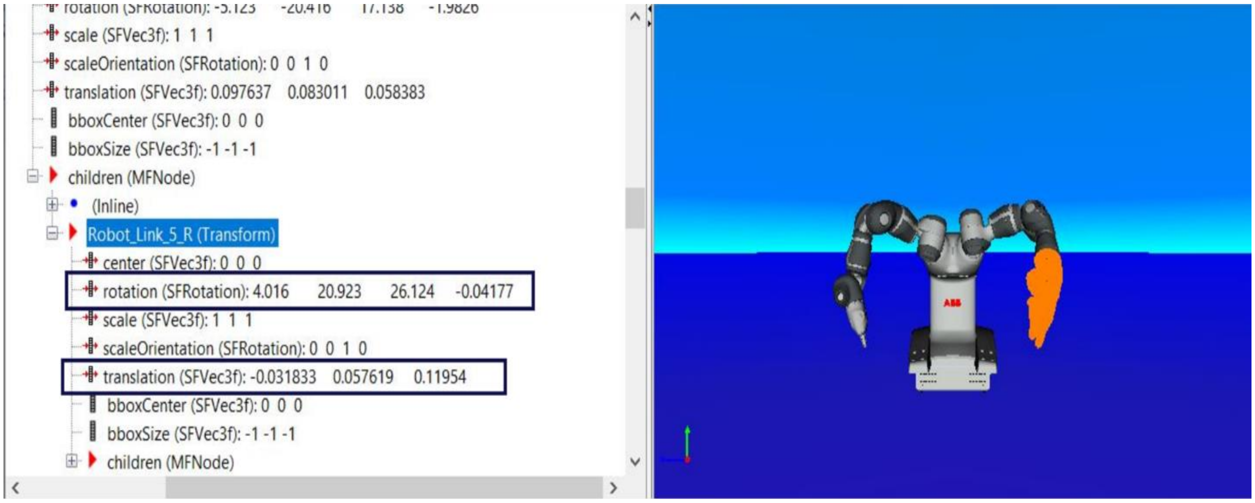


Figure C 11 Link_5 Right adjusted in 3D World Editor using Rotation & Translation Data

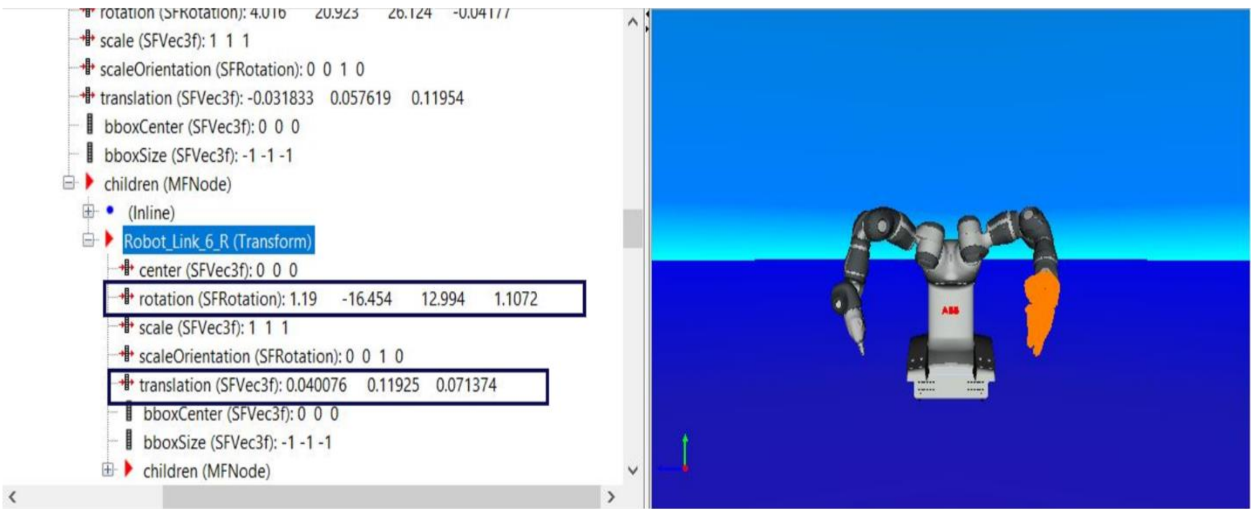


Figure C 12 Link_6 Right adjusted in 3D World Editor using Rotation & Translation Data

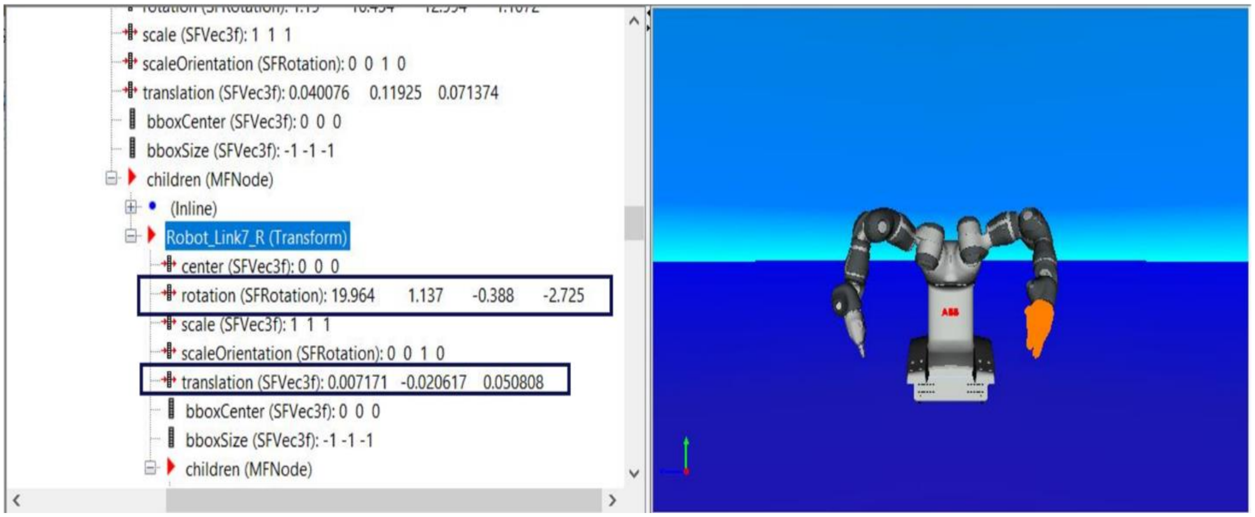


Figure C 13 Link_7 Right adjusted in 3D World Editor using Rotation & Translation Data

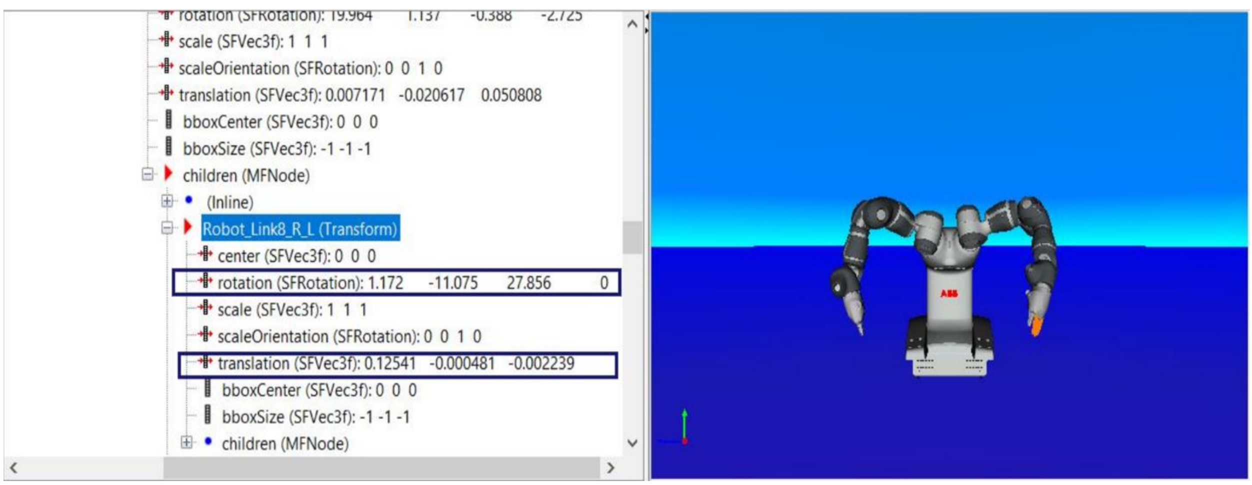


Figure C 14 Link_8 Right (Left Finger) adjusted using Rotation & Translation Data

VITA

Mohammad Aman Ullah Al Amin came from Chittagong, Bangladesh, the youngest son to the beloved parents of Kabir Ahmed Chowdhury and Jahan Ara Begum. He is the fifth of five children, four brothers. He attended Nasirabad Govt. High School and continued to Hajera-Taju College, Chittagong, Bangladesh. After college graduation with excellent results, he got an opportunity to study in Industrial & Production Engineering at the Khulna University of Engineering and Technology (KUET) in 2011 where he got an award for outstanding results. He completed his undergraduate thesis on Optimization of Vendor Managed Inventory (VMI) of Multiproduct EPQ model with Service Level and Quantity discount applying Genetic Algorithm.

After graduation, in 2017 Aman started his career as an Industrial Engineering in a renowned Chemical Industry, where he worked as Production Engineer & supply chain analyst. During his Two-year career, he implemented lead time in supply chain, lean and 6S concept in the production system to enrich his knowledge in the diverse Industrial Engineering field.

Aman joined the University of Tennessee at Chattanooga (UTC) in August 2019 after having a Research Assistantship from the Department of Engineering Management & Technology Department. In his 2nd semester 2020, he was honored by the UTC Outstanding Graduate Student SEARCH Award for his excellent performance in research works. He will be graduating in August 2021. Fortunately, he got several fully funded assistantships to offer from several top-ranking universities for his Ph.D. studies.