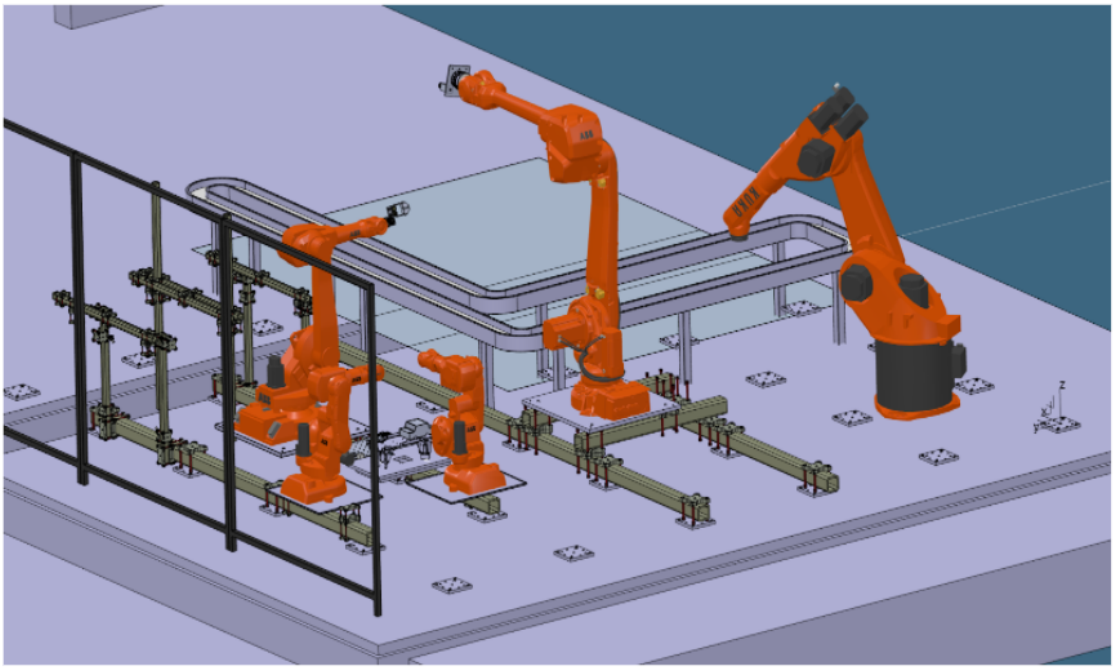
CHALMERS
UNIVERSITY OF TECHNOLOGY



Validation Of Production System at PSL

Master's thesis in Production Engineering

SHREYAS BHANDESH

LUIS MIGUEL SAINZ DE LA FUENTE DELON

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Validation Of a Production System at PSL

SHREYAS BHANDESH

LUIS MIGUEL SAINZ DE LA FUENTE DELON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
Gothenburg, Sweden 2021

Validation of a Production System at PSL
SHREYAS BHANDESH, LUIS MIGUEL SAINZ DE LA FUENTE DELON

Supervisor: Per Nyqvist, Department of Industrial and Materials Science
Examiner: Björn Johansson, Department of Industrial and Materials Science

Master's Thesis 2021
Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Digital Twin of the Production systems lab.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Validation of a Production System at PSL
SHREYAS BHANDESH
LUIS MIGUEL SAINZ DE LA FUENTE DELON
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

The production system lab at Johanneberg, Goteborg is a research lab that addresses projects related to industrial automation and virtual commissioning. This master thesis project aimed to analyse, verify, and validate the complete chain from design/assembly of a physically manufactured product virtually and physically. The assembly of the product is carried out by two ABB robots i.e., IRB4600 and IRB1600. A digital twin of the current state of the PSL is developed to perform all the tests for the design/assembly of the product before its implementation in the real world.
For the design/construction of a digital twin the following methodology is used; data collection, equipment design, layout design, calibration, product design, process planning and virtual validation. The physical validation is also implemented in order to verify and compare the results obtained in the virtual platform. The design and construction of the digital twin as well as the simulation of assembly operation was done using 3DExperience. For the design of the product and equipment needed Solidworks was used whereas 3Dprinting was the manufacturing process chosen for the physical implementation. Several calibration methods were performed on all the equipment in the PSL to increase the accuracy of both the physical model and the virtual model. Along with the methodology followed, this project aimed to answer the research questions that are presented in the report. The results of the project suggest that the virtual model and the physical model are alike but at the end a virtual model is a virtual representation of something real, therefore 100% accuracy can not be achieved, nevertheless virtual manufacturing proven to be a very efficient tool that can be used by companies to improve the performance of the process, decrease time to market as well as save on cost due to errors in manufacturing.

Keywords: Digital twin, Calibration, 3D experience, online programming, ABB Robot.

Acknowledgements

We would like to extend our deepest gratitude to our project supervisor, Per Nyqvist, for his generous support, constant motivation and patience during the whole duration. We would like to appreciate all the help from providing computer rooms and all required software access to helping us conduct experiments at PSL. We would like to express our gratitude to Björn Johansson, the examiner of the Master thesis project, for supporting and encouraging us throughout the project duration. Further, We would like to thank Goran Stigler for proving us all the help and guidance for the design and 3D printing the parts. We would also like to thank Henrik Kihlman and Chethan Shivaraju for sharing us your knowledge on 3D Experience and solving all our douts throught out the thesis.
Lastly, We would like to express our deepest gratitude and love to our family and friends for their constant support during our graduate studies at Chalmers University and, for believing and guiding us throughout.

Shreyas Bhandesh, Luis Sainz, Gothenburg, December 2021

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis are listed

TCP	Tool Center Point
RMS	Root Mean Square
CAD	Computer Aided Design
AM	Additive manufacturing
PLC	Programmable Logic Controller
FDM	Fused deposition modelling
LAN	Local Area Network
PSL	Production System Laboratory

Terminologies

Below is the list of terminologies that have been used throughout this thesis that are important for the understanding of this report are listed

- **Quaternions**
  Quaternions is an alternative to represent 3D rotation. It is denoted with the form H = a+bi+cj+dk where a, b, c and d are real numbers and i, j, k are unit vectors. [22]
- **Singularity**
  Singularity occurs when a robot reaches a configuration where the robot's end-effector gets constrained in certain directions. When a robot enters a singularity, it loses 1 or more degrees of freedom. [23]
- **Work Object coordinate system**
  It is composed of 2 coordinate systems, the object frame and the user frame. When programming a robot, all targets in the program are with respect to the work object coordinate system defined, if the work object coordinate system is not defined, all targets will be with respect to the default Wobj0, which matches with the base of the robot. Work objects are used for calibrating and adjusting offline programs. [24]
- **Robot configuration**Robot configuration
  Robot configuration represents a specific state of the robot. It represents a specific posture of reaching a certain point with the robot. When moving the robot linerly, the configuration of the robot cannot change, if the configuration of the robot changes between two points a joint movement will be needed. For a 6 degrees of freedom robot, there are usually 8 feasible configurations for every position. [25]

Contents

List of Figures

List of Tables

1
Introduction

This chapter provides a brief introduction to the thesis project. This chapter provides the background, aim and limitations of the project. The research questions that are intended to be answered through this thesis are also provided in this chapter.

1.1 Background

The rise of competition and time to market among all industries emphasises manufacturers to design and produce good quality products at the least time possible. To keep up with the market demands, manufacturers have to continuously improve their production systems. To minimise the errors and to improve the production, a prototype of the production system is required to create and simulate any new improvements. Hence, a computer based platform to create and simulate a physical model, a digital twin can be created to verify each process and simulate them.

The digital twin can help evaluate the performance, functionality and safety of a system or process in the virtual environment before being implemented in the physical world. This master thesis aims to create a digital twin of the production platform and validate the platform in both virtual and physical environments. This involves creating a digital twin of the Production Systems Laboratory(PSL) that is situated at Chalmers Johanneberg Campus, Gothenburg, Sweden.

The PSL is an arena for research in the next generation of workplaces that strive to achieve intelligent production systems with flexible and competitive solutions. PSL focuses on highly automated production, semi automated production, human-robot collaboration and manual production and assembly. The laboratory hosts a number of industry robots from various suppliers alongside a few collaborative robots which are mobile. This thesis focuses on automated production with the assistance of the industry robots. The thesis primarily focuses on the validation of the selected production platform in a virtual environment. The process will be validated in the physical environment.

1.2 Aim

The aim of this master thesis is to develop a virtual simulation platform of a real production cell at PSL. Moreover to show, analyse, verify and validate the complete

chain from design/construction to the physically manufactured product.

1.3 Project Limitations

- This thesis only covers a part of the production line. i.e. a production cell. To include the whole line would be out of our project scope.
- Real time feedback of data to or from the physical to the virtual platform is precluded in order to avoid a complex system.
- The scope of the project will be adjusted to complete the project within the available time frame.

1.4 Research Questions

The master thesis aims to find results and solutions for the following questions.

- What are the different methods and resources required to create a digital twin?
- How flexible is the designed system to change?
- What are the differences between the virtual model and the physical model and how do they change the outcome?
- Which tools are most suited to create a digital twin?
- What are the pros and cons of using a digital twin?

2
Theory

This chapter includes all the relevant theories that were used or implemented in this thesis. This chapter provides detailed explanation on automation concepts, robotics, digital twin and additive manufacturing.

2.1 Industry 4.0

The industry 4.0 that is more commonly known as the fourth industrial revolution has gained enormous developments in recent years. Although the fourth industrial revolution is built on the previous industrial revolution, it is mainly based on the digital transformation. The industry 4.0 is targeted to convey the highest degree of automation, digitalisation, virtualisation and decentralisation among all industries. [2]

Industry 4.0 was coined by the German federal government with other private companies and universities. To understand Industry 4.0, the concept can be divided into two layers: the front-end and the base technologies as seen in figure 2.1. The front-end technologies or the first layer of industry 4.0 constitutes the four main divisions of smart manufacturing, smart working, smart supply chain and smart product. The second layer or the base technologies constitute the technologies that provide the connectivity to the front-end such as the internet of things, big data, analytics, etc. The combination of two technologies can be used to define the various concepts of industry 4.0 and their implication of implementation of these technologies in a company can provide its maturity index. [3]

Smart manufacturing is the beginning of industry 4.0. The method of smart manufacturing allows production lines to automatically adjust to variation and to have mass production with higher degree of flexibility and customisation. This can provide high productivity and better quality of the product. The related technologies for smart manufacturing can be divided into six different purposes:

1. Vertical integration
2. Virtualization
3. Automation
4. Traceability
5. Flexibility
6. Energy management

This project includes the study of digital twin, simulation and virtual commissioning,

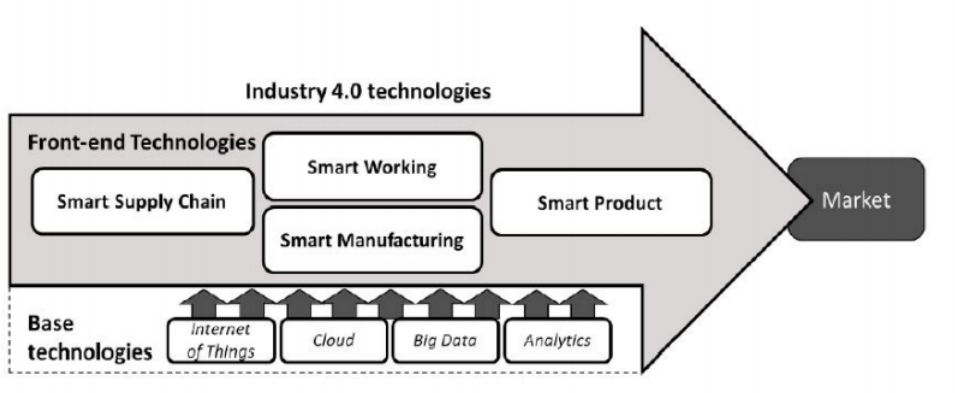


Figure 2.1: Theoretical framework of Industry 4.0 technologies[3]

which is a part of smart manufacturing. This includes different purposes such as virtualisation, automation, traceability and vertical integration.

2.2 Automation

The trend of mass producing multiple variants of the product in the same production space is increasing. In order to keep up with the mass customisation, the process has to be automated to some extent depending on the needs and cost of the product. Automation according to Frohm is defined as "automatic control of the manufacture of a product through several successive stages; the application of automatic control to any branch of industry or science; by extension, the use of electronic or mechanical devices to replace human labour" [4]. Automation does not always mean the elimination of human labour. It also means to support and assist human labour, such as co-bots that can work alongside humans.

Automation is used to ensure a more precise execution of a task that can yield a better quality of the product. This can increase the stability of the output by removing humans from repetitive and monotonous tasks, that helps in achieving a breached capacity limits of control tasks and thereby increase speed, efficiency and security. [5]

The level of automation has to be decided based on the complexity and cost of the product. High level of automation for product realisation is an important means to meet high customer demands. Both the physical and cognitive automation is required in realising the level of automation.

2.3 Process planning

Process planning is the interface between the design and the manufacturing stages. This planning includes the selection and sequencing of various processes and operations to convert the raw material into a finished product [6]. This is one of the

important stages that include the selection of material,fixtures and jigs and the process parameters. The various activities of this stage are depicted in figure 2.2.

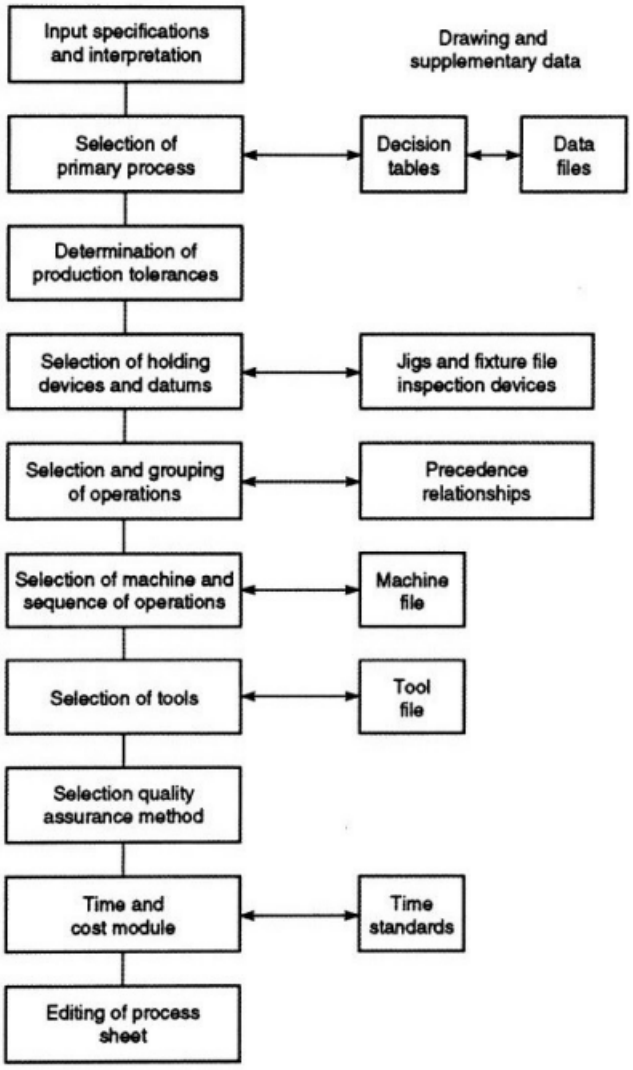
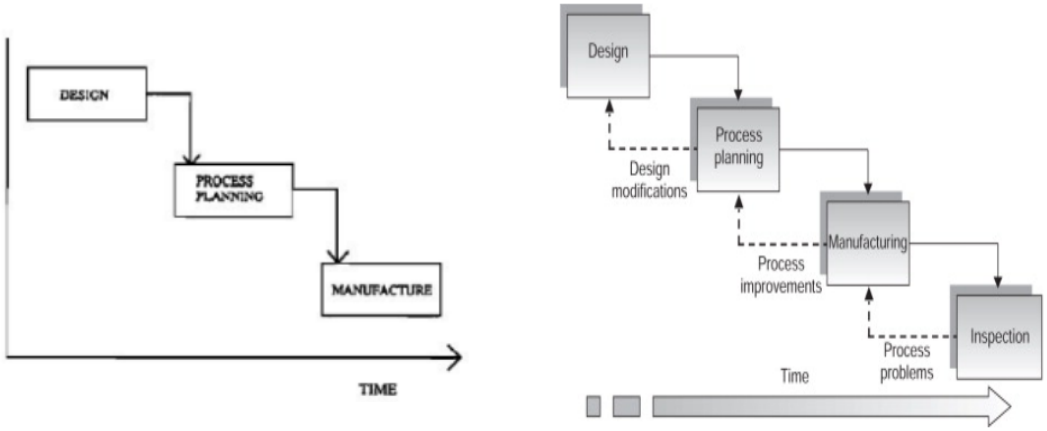


Figure 2.2: Process planning activities [7]

The traditional method of process planning takes place in a sequential manner i.e. the design stage generates information about the product and the operations that need to be carried out. This information is used in the planning stage to generate the manufacturing data and passed onto the next stage [8]. However, this approach is preferred for a stable product with longer life cycles. The more advanced approach used in today's competitive environment is concurrent or simultaneous or parallel engineering. This is a cooperative approach that utilises cross-functional teams to reduce the time to market. This project also used this parallel engineering and this can clearly be understood by the figure 2.3. The method of offline programming and simulations are included in this stage before manufacturing. The simulation of the process can be used to test various parameters in order to obtain the highest possible number of products. Offline programming and simulation of the assembly can help improve the process of assembly. This project uses a similar method of

offline programming to test the product and its assembly in the virtual environment. The results from simulation and offline programming have been used for process improvements and design modification.



a) Sequential Engineering

b) Parallel Engineering

Figure 2.3: Planning strategies

2.4 Robotics

The robots in this project are referred to as industrial robots. Based on the industrial organisation of standardisation (ISO8373) an industrial robot can be defined as "automatically controlled, reprogrammable multipurpose manipulator programmable in three or more axes", which can either be fixed or mobile for use in industrial automation applications. [9]

Reprogrammable : designed in such a way that the program or the function of the robot can be changed without any physical alterations
MultipurposeMultipurpose: capable to adapt to different applications with physical alterations.
Physical alteration : altering the mechanical system
Axis: specific direction of motion- rotary or linear

2.4.1 Robot classification

The international federation of robotics have classified the industrial robots based on the mechanical structure of the robot. [9] The robots are classified into five categories:

- Cartesian robot: robot whose arm has three prismatic joints and whose axes are correlated with a cartesian coordinate system
- SCARA robot: a robot, which has two parallel rotary joints to provide compliance in a plane

- Articulated robot: a robot whose arm has at least three rotary joints
- Parallel/Delta robot: a robot whose arms have concurrent prismatic or rotary joints
- Cylindrical robot: a robot whose axes form a cylindrical coordinate system

2.4.2 Robot anatomy

A traditional industrial robot consists of an arm with an actuator. The actuator of a joint generates the motion. The actuator of a joint consists of a power mechanism to affect the motion of the link, usually a motor. [10]

The traditional design concept of a robot is with a serial sequence of links connected to each other with joints, each joint has a single degree of freedom. The motion of the robot is by the motors of each joint. The joints connecting the links are either prismatic joints or revolute joints. Prismatic joints have linear motions of the links in space in relation with the others, whereas revolute joints have rotational motion of the links in space with relation to the other links in space. [10]

The robot consists of a manipulator which can be divided into two: arm and wrist. The arm or the primary axes support the position and motion of the wrist in space. The wrist of the secondary axes support the position and orientation of the end effector. The primary and secondary axes are interconnected, and to gain high accuracy of motion,coordination of all joints is required which affect the position and orientation of the end effector which is to be controlled.

End effectors are special tool attachments that are attached to the robot wrist to perform a specific task. The end effectors can be of two types, namely grippers and tools. Grippers are attachments used to grasp and manipulate an object or workpiece during the work cycle and the tools are attachments used to perform a specific process, tools such as spray guns and welding guns are used for painting and spot welding respectively.

2.4.3 Geometry and Motion

In order to move the end effector to a particular position in space along a specified trajectory the robot arms are intended to have complex movements. To achieve this complex motion the robot has system control with a set of algorithms which describes and calculates how the segment of the robot manipulator has to be moved based on the command instructions. [10]

The instructions to move the arm can be defined in a cartesian space and to do so the control model has to change the move instruction by specifying movement of each joint to achieve the robot target position. This transformation of instruction in the cartesian system is illustrated by the kinematics of the robot. Determining the motion of the tool or end effector given the motion of links is called forward kinematics and the determining the motion of the links given the position and orientation

of the end effector is called the inverse kinematics. [11]

2.4.4 Coordinate system

A coordinate system is used to define the position and orientation of an object or point in space. The coordinate systems that are used for industrial robotics are listed below along with figure 2.4.

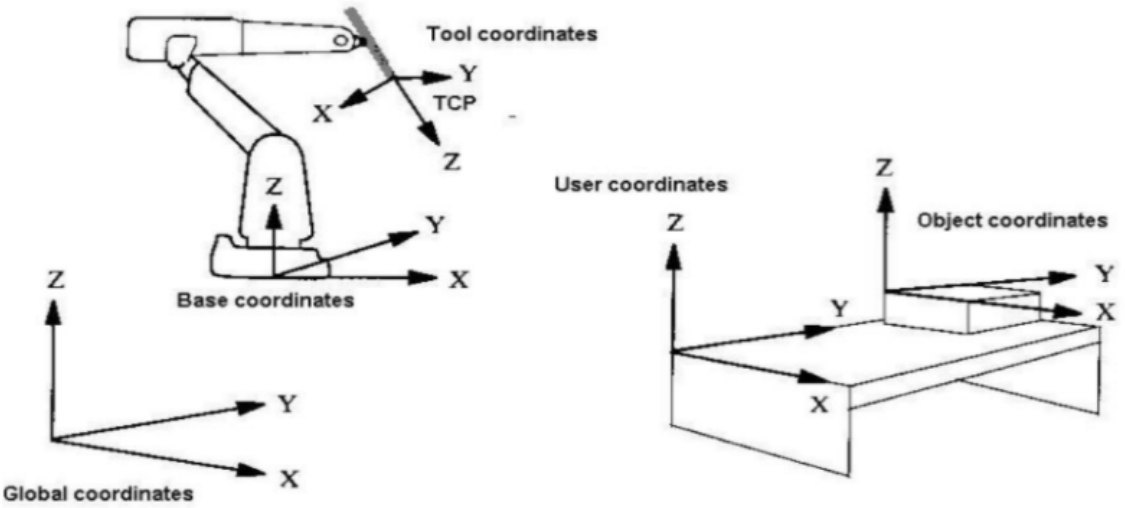


Figure 2.4: Coordinate system of ABB robots [12]

- World coordinate system: the world coordinate system is a permanently defined cartesian coordinate system and it is the base for all other coordinate systems.
- Robot base coordinate system: the robot base coordinate system is the cartesian coordinate system located at the base of the robot and it defines the position of the robot relative to the world coordinate system.
- User coordinate system: User coordinate system is a reference coordinate system used to define motions in Cartesian space. The world coordinate system by default is used as the user coordinates, but it is possible to define an additional user coordinate relative to the world coordinates.The user coordinate system can be observed at the corner of the work bench in the figure 2.4.
- Object coordinate system: Object coordinate system defines the position and orientation of the object relative to the user coordinate system. The object coordinate system can be observed on the workpiece in the figure 2.4.
- Flange coordinate system: Flange coordinate system defines the position and orientation of the robot flange and it is always moving with the robot. Flange coordinate system acts as a base for the coordinate systems which describe the tool mounted on the flange.
- Tool coordinate system: Tool coordinate system defines the position and orientation of the tool. The origin of the tool coordinate system is called Tool centre point (TCP).

2.5 Coordinate Transformation

The coordinate transformation is a calculation method used to determine the position or coordinate of an entity with respect to another coordinate system or entity. The figure 2.5 shows a robot with different coordinate systems.The origin of the coordinate system at the base of the robot can be defined as $Ox_oy_0z_0$. In order to determine a relation between the origin and the end effector transformation from the base coordinate to the tool coordinate at the end effector is carried out. The successive transformation from the base coordinate to the link 1, link1 to link 2 and so on until the transformation to the end effector is complete [11].This provides us the relation between(x_0,y_0,z_0) i.e. base coordinate system and the tool coordinate system (x_n,y_n,z_n) .The transformation of a coordinate system can occur in two ways i) translation, ii) rotation.

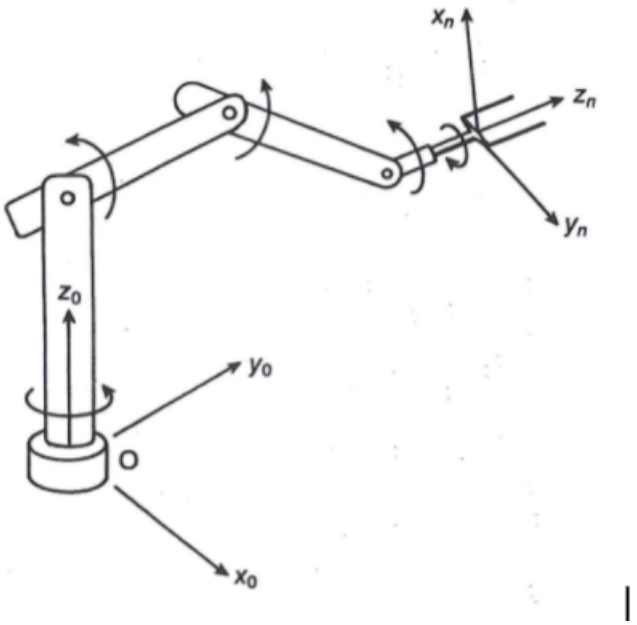


Figure 2.5: Robot with different coordinates [11]

Translation is the form of transformation that occurs on axes that are parallel and do not coincide. This is a simpler form of transformation and it covers the transformation along three degrees of freedom i.e. along x,y,z axes. The figure 2.6 show the axes of two system $Ox_0y_0z_0$ and $Px_1y_1z_1$. If (p_x;p_y;p_z) represent the coordinate of p with respect to O. Then the translation matrix would be represented as equation 2.1.

$$T = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

The transformation related to the other three degrees of freedom is covered by rotation.[11]. This type of transformation is done when two coordinates do coincide

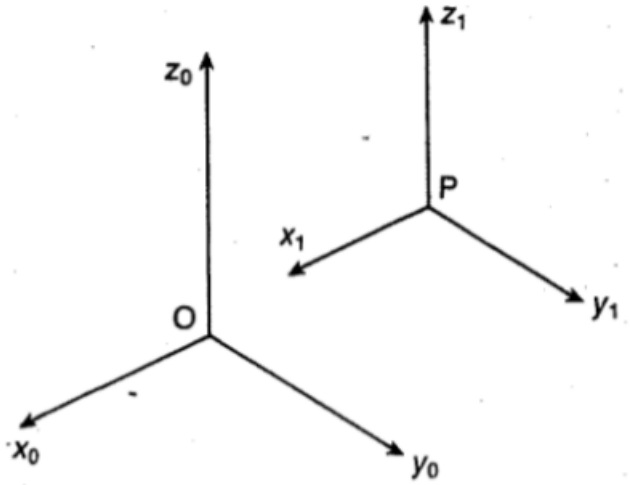


Figure 2.6: Parallel coordinates [11]

and axes are not parallel as shown in figure 2.7. The axes of system with coordinates $x_1y_1z_1$ are rotated with respect to the axes of the origin with coordinates $Ox_0y_0z_0$. Then, the corresponding transformation is defined by a rotation matrix R. The rotation about different axes have different formulas and are represented in equations 2.2,2.3,2.4.

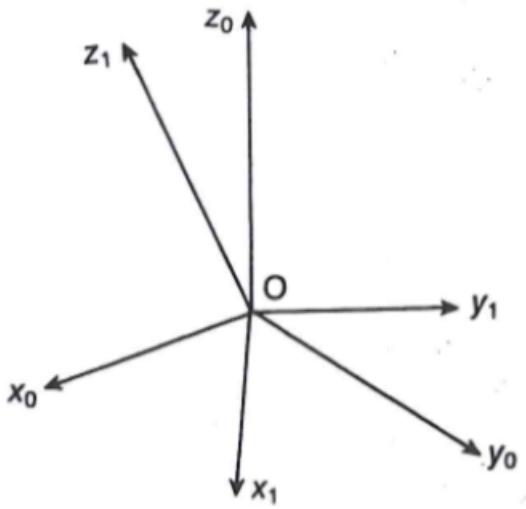


Figure 2.7: Coordinates with the same origin [11]

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \tag{2.2}$$

$$T = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{2.3}$$

$$T = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The homogeneous transformation matrix for the rotation equation can be represented in the equation 2.5. The coefficients $R_{11}R_{12}R_{13}.....R_{ij}$ are determined by the angles between the coordinate axes.

$$T = \begin{bmatrix} \mathrm{R}_{11} & \mathrm{R}_{12} & \mathrm{R}_{13} & 0 \\ \mathrm{R}_{21} & \mathrm{R}_{22} & \mathrm{R}_{23} & 0 \\ \mathrm{R}_{31} & \mathrm{R}_{32} & \mathrm{R}_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

The total transformation from the base coordinates to the tool coordinates are a combination of translation and rotation. The general relationship is seen in equation 2.6.

$$T_0^n = T_0^1 T_1^2 T_2^3 ....T_{n-1}^n \tag{2.6}$$

2.6 Robot programming

Robots have a special advantage compared to other machines that are only capable of doing a single task,i.e. programablity. The robots and their support systems have to be programmed in the lowest possible time and best quality to obtain the highest efficiency. The robot programming can be classified into two categories: online robot programming and offline robot programming.

2.6.1 Online programming

Using the robot system to program the robot is called online programming. Generally online programming is used for simple and less cumbersome tasks. Since the programming is done in the real environment, it is easier to correct errors. The major drawback of using this method is, it can affect the production efficiency as the production is stopped during the programming.

Online programming advantages

- Simplicity
- Fast to implement
- Calibration of objects in the robot cell is not needed as the programming is done in accordance to the real position of them.

- Modify the robot configuration, problems with singularity or problems with reachability are avoided.

Online programming disadvantages

- The production has to be stopped which might incur some financial loss.
- Operator requires prior knowledge regarding programming and safety.
- Difficult programs or more structured programs might be difficult to program using this method.

This is a more traditional way of programming the robot also known as the "teach-in" method. This method of programming generally uses a teach-pendant or flexpendant that is used to control and program the robot. Depending on the robot supplier is the flexpendant model, every flexpendant is independent of each brand, during this project ABB robots are used. ABB flexpendant is shown on figure 2.8

Figure 2.8: Flex pendent of ABB robot

In order to create a program online the arm of the robot is jogged to the desired location. Once the robot is set in the desired location and orientation, the position in the coordinate system is saved in the flexpendant by adding a motion instruction. Similarly, the robot is moved to the next desired location and continued until the task is completed. The movement/ motion of the robot are specified as joint or linear depending on the requirement. The speed, zone, and accuracy of the arm are also specified in the program and are chosen depending on the task needs. [13]
This method of programming was used in the project to familiarise us with the robot programming and operation of the robot in the real world. Whereas, most of the programming in the later stages of the project were carried out offline and deployed to the teach pendant to run and verify physically.

2.6.2 Offline programming

In this method the programs for the robot system are programmed on a computer while the robot is in operation in the production system. The offline programming

also requires some final checks and the production has to be stopped during the deployment but the time required is considerably low. This is best suited to carry out complex tasks since the programming takes longer time and the production system need not stop during this time. Offline programming also requires the knowledge of computer and computer programming to program the robot and the support systems. The offline programming has better structure and documentation which make it easier to share among programmers. Since the programming is not done in the real world there are some errors concerning the geometric tolerances. These errors are discussed in the following chapters of this project.

Offline programming advantages

- Productivity is increased
- Short changeover
- Changes can be made on the program quickly and easy
- Several programs can be created and simulated at the same time in the same platform

Offline programming disadvantages

- Virtual models are just an approach of the real model, therefore it will never be able to represent the real model with 100
- Time consuming when modelling and calibrating all the equipment needed for the virtual workcell
- Offline programming software tend to be expensive

This method of programming is generally done on computer softwares such as ABB Robotstudio, 3DExperience, Visual components, etc. During this project the 3D Experience was used to create the programs. In order to create a program in this software the robot arm or the tool can be moved to the required target and saved. Similar to online programming the speed, zone, type of movement can be altered based on task requirements. The robot can be given multiple targets to create the entire program. This program can be simulated in the virtual environment. These instructions are translated into a robot program with a post processor.

2.7 Calibration

Simulation softwares and offline programming are tools that allow the users to reduce the time to production, changeover time and reduce risks and unnecessary errors. In order to create a simulation or offline program both the virtual and physical model should mirror each other. Therefore, the real model must be calibrated so the gap between the virtual model and real model is closed. Methods such as tool measurement and cell alignment are performed in order to increase the accuracy of the model. [14]

In a robot's end effector there are two types of tools, stationary tools and non-stationary tools. When it comes to non-stationary tools such as arc welding guns, the TCP of the weld gun in the robot arm is changed continuously as the geometry of the tool varies during the production. On the other hand, for stationary tools

such as grippers and vacuums, the TCP is considered to be stable and therefore it is not changed unless there is a collision or the needs of the task demand it.

Cell alignment is another method that must be performed in order to increase the accuracy. There must be zero deviation between the position of the robot and all the equipment in the virtual and real model. Therefore several methods are performed in order to assure that the position of every resource in the cell is accurate and does not deviate from model to model. The various methods used during this project for calibrating tools, base of the robots, etc. are explained in the further parts of the report.

2.8 Digital twin

A digital twin is a replica of a product or process in a virtual environment used to simulate the various scenarios to test and improve the product or process. Although the concept of the digital twin was first recorded in 1991 by David Gelernter. It was NASA in 2010 who were able to create and simulate a digital twin of a space capsule and craft. With the development of various technologies (such as computers, internet, sensors, etc.) over the years it is now easier to create a digital twin to collect data to predict how the product or process will perform. [1] The digital twin can be integrated with the internet of things (IoT), analytics and artificial intelligence to improve the output of the simulation. With the growth in the field machine learning and big data in industries, the digital twin can be integrated to amplify its performance. [15]

Digital twin is also being used in manufacturing industries today due to its numerous advantages. Digital twin can be created for the entire production plant of an industry.Since the digital twin is reprogrammable and smart it is easier to make changes in the model and test it before implementing it in the real world. With the feature of modularity it is easier to test and simulate individual modules to improve the bottlenecks and enhance the performance of the product or process [16]. This improves the quality of the product as well the production. This project also focuses on creating digital twin of PSL, and validating the production cell containing multiple robots.

2.9 Additive Manufacturing(AM)

The technique of adding layers to build/create parts or products is known as additive manufacturing. This method of manufacturing products uses computer aided design(CAD) data or data from 3D scanners to precisely create the product. The data fed to the additive manufacturing software will create individual slices of each layer and the support structure required for the product. Then the individual slices or layers are printed one top of the other with geometric accuracy to form the final product.

This method of adding material has a lot of advantages over traditional machining. Additive manufacturing uses far less material than conventional manufacturing processes and also has lesser wastage. This technique is best suited for prototyping. The use of multiple machines or tools used to carry out different operations i.e. grinding, drilling, milling, etc. are completely eliminated with the use of AM. More complex designs can be printed with AM that saves cost and time to produce the parts.

There are several methods of AM that can be used to create a product. The methods vary in type of heat source used and the type of material printed. Each method has its pros and cons and the method is selected based on the product being printed.In our case depending on the features and the availability of the printer we have used the Fused Deposition Modelling(FDM) method of printing.

The figure 2.9 shows the FMD process. In the FDM method the thermoplastic materials are extruded through a nozzle tip that heats up the material. The melted material is selectively deposited on the predetermined path or the slice data of the layer. Similarly, each layer is built one over the other according to the slice data or the STL file [17]. Once all the layers are built the part or the product is detached from the printer base and the support structures are removed.

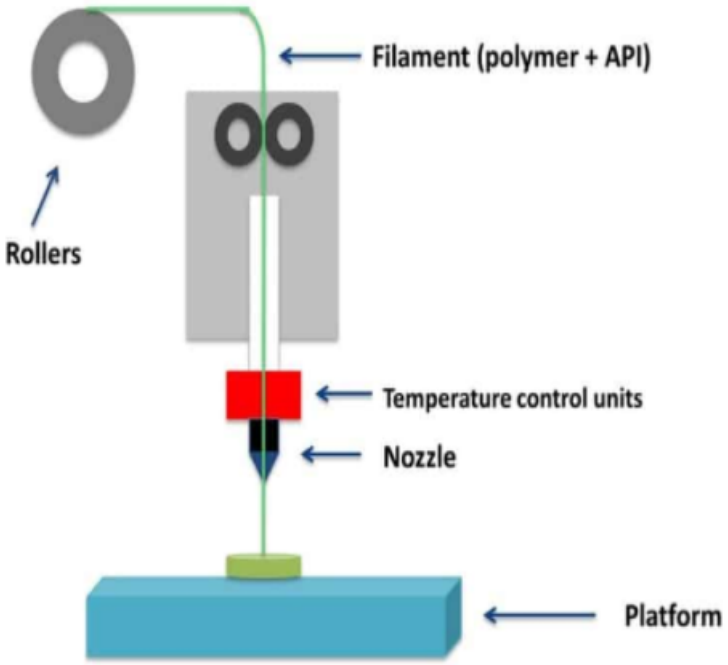


Figure 2.9: Fused Deposition Modelling process

2.10 Fixtures

These are attachments that are used to secure the workpiece in the right position, location, and orientation. These are critical manufacturing tools used to constrain the degrees of motion of the workpiece in order to restrict any unnecessary movement during the operation or assembly of the workpiece/ product [18]. The fixtures are usually customised based on the operation and the working parameters. However, fixtures can also be brought off-the-shelf for more common operations. The fixtures used in this project were designed based on the requirements and also used some common fixtures such as clamps and pins. More explanation regarding this topic is covered in the future chapter.

3
Methods

This chapter discusses the methodology followed for the analysis, verification and validation of the complete chain from design/assembly of a physically manufactured product using a virtual platform. This methodology included all the stages for the designing, simulation and validation of a virtual production line. The stages involved in the methodology are stated below. Moreover, an additional stage was done, in order to fully validate every stage of the chain physically. The physical validation was performed in the PSL of Chalmers University of Technology.

- Data collection
- Equipment design
- 3D layout design
- Calibration
- Product and tool design
- 3D printing
- Robot programming
- Robot simulation
- Virtual validation
- Physical validation

3.1 Data collection

3.1.1 Literature study

The data collected in the literature research of this project included topics associated with digital twin, industry 4.0, validation, methods and processes involved in each stage of creating a digital twin. The research also included selection of software that could be used to create a digital twin. The data collection was done through various sources such as Chalmers library website, google scholar and science direct. The data collected mainly included research papers that included the keywords digital twin, industry 4.0, physical and virtual validation, automation. Data regarding the robots were collected from ABB website.

3.1.2 Technical Data

Besides the data collected from the research, data regarding the layout of the PSL and the robots were collected through other sources. The layout of the PSL was

received from a 3D scan conducted previously. The CAD data of the robots were collected either from the robot library or from the manufacturers website. Similarly the CAD data used in the digital twin was retrieved from the manufactures websites. All the data collected were verified with the physical model and a large amount of the data collected has been carried out physically from the production platform.

3.1.3 3DExperience

This is the software engineering tool decided for the project. This software is developed by Dassault systems and can be used to design and simulate a model in a 3D platform. This is a collaborative and data driven platform that allows users to create a personalised dashboard and collaborate in real time. This software allows users to have access to the required data into their workstation since all the data is related to the product or a project is placed in a central location. This software provides a vast range of application(apps) for all the different needs of the project. Moreover, 3DExperience contains a vast number of robot libraries and post-processors.

3.1.4 Industrial robots tools and equipment

Currently the PSL production cell has 5 industrial robots; IRB 140, IRB 4600, IRB 1600 and KUKA Robot. During the project only IRB 1600 and IRB 4600 are used for the product assembly. The production cell consists of multiple working platforms and a conveyor system. A platform that is easily reachable by most of the robots are selected for this project, more regarding this is discussed in the future chapters.

3.2 Equipment design

This stage involved the set of tools that were used to model and design the mechanical resources needed for the manufacturing process. The mechanical systems required for this manufacturing process included industrial robots, robotic end-effectors (grippers, drilling tools, screwdrivers, weld guns and so on).

3.2.1 Resource creation

Before placing all the mechanical resources needed in the manufacturing cell, some mechanical ports and parameters have to be assigned. The basic mechanical ports for an industrial robot or a robot tool are base port, mount port and tool center point (TCP). The base port and the mount port are defined for every industrial robot and for every robot tool, the mount port of the robot and the base port of the tool allows the user to attach both together. The base port of the robot indicates the place where the robot will be mounted in the manufacturing cell. Finally, TCP is defined for every tool device used by the robots and they are defined independently depending on the tool and the task.

3.2.2 Motion controller

A motion control is a device created to define all the robot parameters and attributes such as, travel limits, home positions and joint speed and acceleration. The robots used were exported from the library, therefore the motion controller parameters and attributes are set by default, nevertheless the attributes and parameters were changed depending on our needs.

3.2.3 Tool design

The tools attached to the robot arm are explained in this stage. The robot arm was equipped with a SCHUNK tool changer. The SCHUNK tool changer consists of two parts, a master that is fixed to the robot arm and a slave/adapter. The tools such as the vacuum, gripper, etc. are attached to the adapter as shown in the figure 3.1. The tools are designed to fit the tool changer. During the project only one tool was used per robot. The IRB 1600 and IRB 4600 used the gripper and vacuum tool respectively. The two parts of the SCHUNK tool are attached to each other with pneumatic pressure. The pneumatic valves on the master help hold the adapter in place and the guiding pins aid the positioning of the adapter. The tool such as the gripper tool or the vacuum tool are attached to the adapter/slave with fasteners or screws.

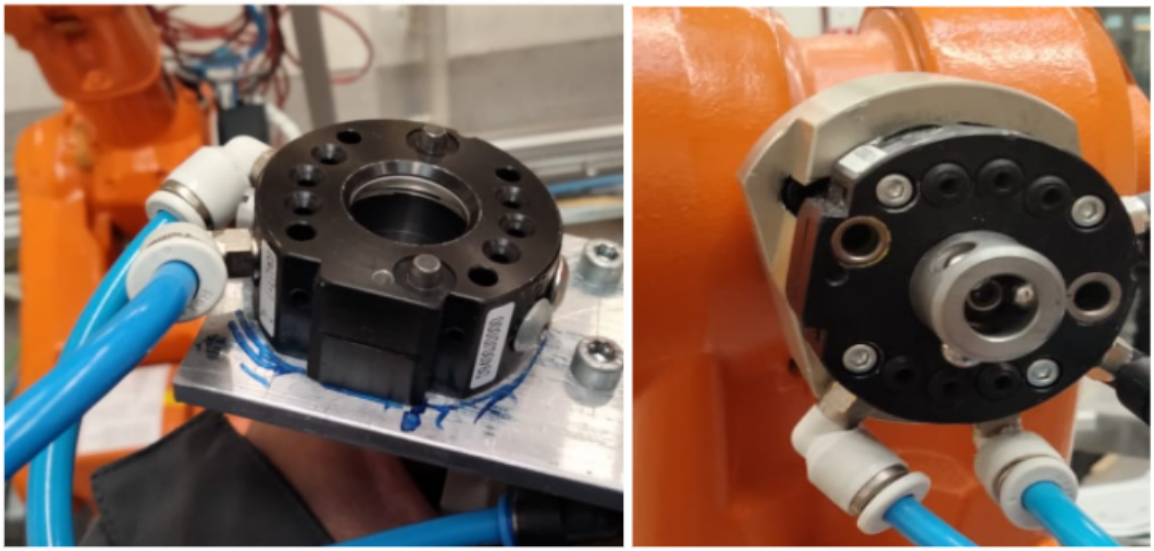


Figure 3.1: Tool changer

Gripper

The gripper used in this project was a SCHUNK 0308620 gripper with 2 finger parallel gripper. This gripper was attached to the IRB 1600 robot. The gripper was powered by pneumatic tubes and control signals were provided through the flex pendant. The signals activated the pneumatics to open or close the grippers. The figure 3.2 below shows the gripper used along with the pneumatic tubes. A CAD model of the gripper was used in the virtual model.

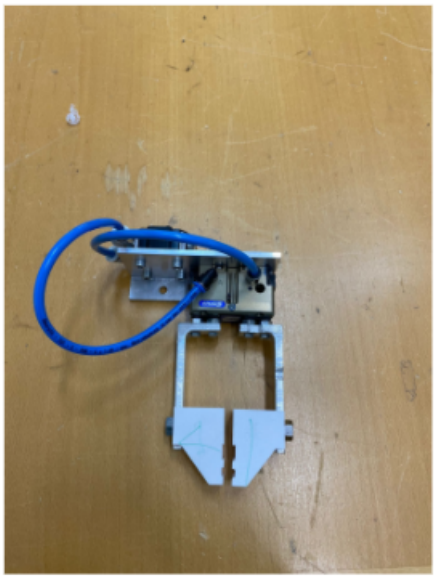


Figure 3.2: Gripper tool

Vacuum
The vacuum tool used during this project was a PIAB tool with a fcf35p suction cap and a vacuum pump. The vacuum pump was powered by pneumatics which in turn activated the vacuum suction at the suction cap. The vacuum was attached to an aluminium plate and powered by pneumatics. The signals to activate and deactivate the vacuum tool were provided by the flex pendant. The figure 3.3 below shows the vacuum tool along with its aluminium attachments. A CAD model of the tool was used in the virtual model.

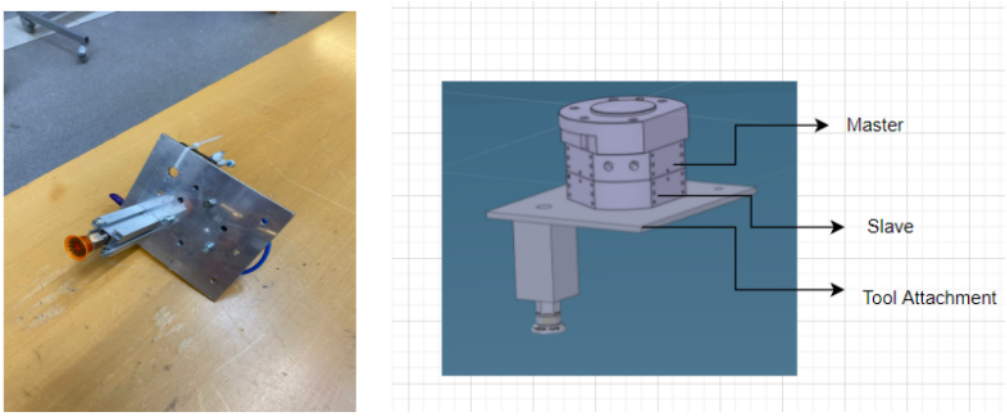


Figure 3.3: Vacuum tool

3.2.4 Working plate

The working plate is the location where all the assembly would occur, hence this should be placed in the right way that is accessible by all the robots involved in the assembly. The working plate used in our project was an aluminium plate one

sq.mt in dimension. The plate consists of holes with 10mm diameter. All the holes make a pattern such that each hole is 100mm apart from the other as seen in figure 3.4. This plate was placed in a position that could be easily accessible by IRB 1600 and IRB 4600. A CAD model for the same dimension was created and placed in an exact position in the virtual model.

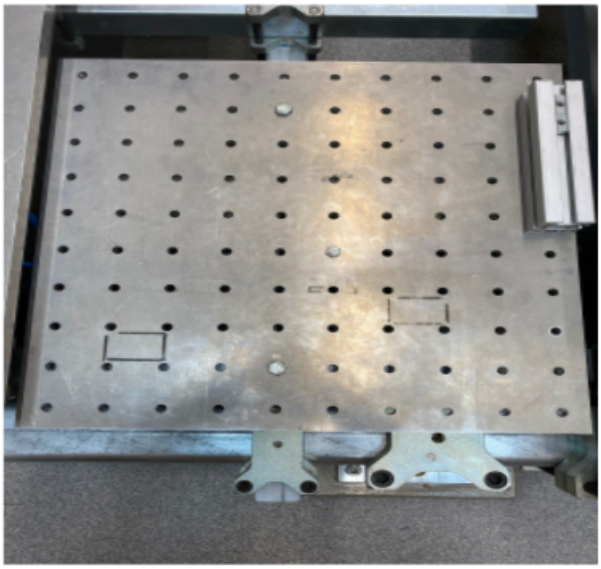


Figure 3.4: Working plate

3.3 3D layout design

In the previous step all the manufacturing resources needed for the manufacturing process were designed, the next step is to position those mechanical resources in the right place. When it comes to developing and designing the layout for a manufacturing robot cell, the position of the robots is very important. The robots must be put in a position in which they can reach the positions needed to perform the task according to its own limits. Based on that, mechanical resources are placed within the feasible limits of the robot [19].

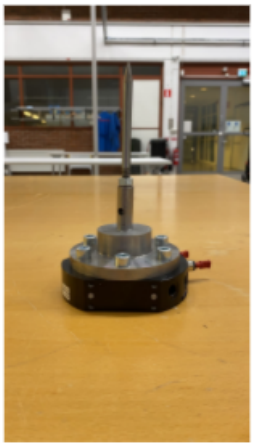
When developing a virtual model of a real model, accuracy plays an important role. It is important that all the mechanical resources are placed accurately in the correct position, else there will be a mismatch between what is happening in the virtual model and what will happen in the real model. Having said that, the virtual model must be calibrated in order to close the gap between it and the real mode as close as possible. Thus, before starting with the programming and simulation, every resource in the layout must be calibrated and positioned in the correct position [19]. Nonetheless, it is still a virtual model, which means that the virtual model will be slightly different than the real model, 100% accuracy cannot be assured. The methods used for calibrating the location of the resources in reference to the world coordinate system are explained in further stages.

3.4 Calibration

In order to adjust the position of the robots and all the resources included in the PSL model, several calibration methods are used. This stage presents the calibration methods used for the calibration of the virtual model as well as the equipment and tools needed for the performance of the method.

3.4.1 Equipment for Calibration

This section discusses all the equipment needed for the performance of the calibration method. The equipment used for the performance of the calibration is shown in figure 3.5.



a) Calibration tool IRB 4600

b) Flexpendant

c) Calibration tool IRB 1600

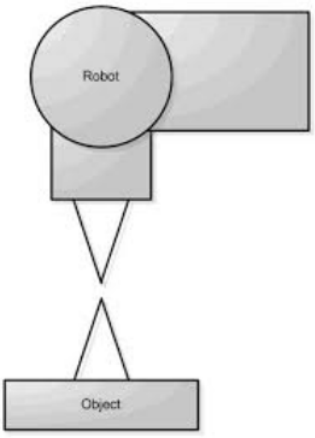
d) Calibration tip

Figure 3.5: Equipments required for calibration

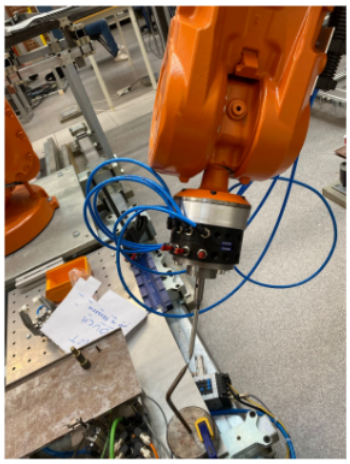
3.4.2 Tool center point calibration tip

For the preparation of this method, the calibration tool is attached to the robot. The calibration tip is placed in the working plate, it is used to define the TCP of the calibration tool. The method recommended is "TCP (default orientation)" and it is done online using the flexpendant.

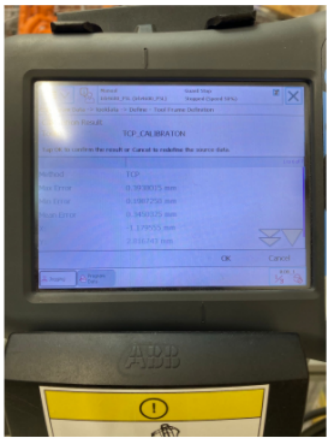
This method is done by jogging the robot to a position in which both the end of the calibration tool and the end of the calibration tip are accurately on each other as shown in figure 3.6. Once the robot has been jogged into a position in which both the calibration tool and the calibration tip converge and the position is saved. This step is repeated for as many points as chosen. The orientation of the calibration tool must be different for every reading. Once all the readings are obtained, the flexpendant automatically displays the TCP as shown in figure 3.6. The TCP displayed in the flexpendant also contains a RMS error. For the result to be validated, the error must be less than 1mm. If the error is greater than 1mm some points must be recalculated. The step by step methodology followed is explained in appendix 4.



a) Representation of calibration



b) Calibration



c) Calibration using flex pendant

Figure 3.6: Tool centre point calibration

3.4.3 Performance test of the calibration point tools

The base frame position for the robots as well as the reference holes coordinates were provided by the supervisor. This data was a result from an earlier cell alignment where a high precision Leica laser tracker was used. In order to check the reliability of the calibration point tools a performance test was carried out. In this test the calibration point tools previously defined were used for measuring the position of the reference holes. Once the position of the reference holes were obtained, the results were compared with the results obtained using the Leica laser tracker method.

In order to measure the exact position of the reference holes a calibration mark is specifically manufactured so it fits perfectly in the center of the reference holes. The calibration mark is then placed in the center of a reference hole desired to be measured in a way that allows us to put the calibration tool accurately in the center of it as shown in figure 3.7. As soon as the calibration tool is put accurately in the center of the reference hole, the position of it is saved. This procedure is repeated for as many reference holes as required. This method was done for those reference holes that were actually reachable by the robot.

Once all the reference holes have been measured, the position and orientation of it with respect to the base coordinate system is displayed in the flexpendant. Then, the position of the reference holes with respect to the world coordinate system is measured using the cell alignment data given from the previous data obtained using the Leica laser tracker. This is done by obtaining the homogeneous transformation matrix following the next formula P_2= T*R_z*R_y*R_x*P_1; where P_1 is the position of the reference hole with respect to the base coordinate system, R_x is the rotation along the X-axis, R_y is the rotation along the Y-axis, R_z is the rotation along the Z-axis, T is the translation and P_2 is the position of the reference hole with respect to the world coordinate system. Once all the reference holes with respect to the world coordinate system were measured they were compared with the results obtained using the Leica laser tracker. The results are presented in further stages.

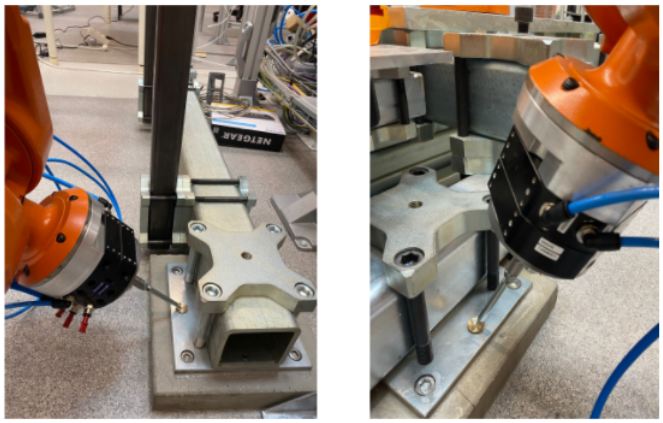


Figure 3.7: Measuring reference holes

3.4.4 Working plate position definition

In order to define the position of the working plate various measurements are carried out. All measurements contain the coordinate positions of the corner points. One of them is carried out by the Romer measurement arm. The measuring device provides the position of the steel plate with respect to the world coordinate system. This device has high precision in measurements of about 60 microns.

The other measurements that are used to define the working plate are the robots. Once the reliability of the calibration point tool has been validated these can be used to measure. The calibration point used to measure work object position with respect to the robot. This helps to define user frames as explained in the later stages. This measurement is carried out by jogging the robot with the calibration tool tip to the corner of the working plate and accurately positioning it. This position is saved and repeated for two corners. Since three points are required in defining a work object. Once the required corners are measured the position and orientation of the working plate with respect to the robot is known. Similar measurements are carried out to all robots that are using this steel plate, i.e. in this case two robots(IRB1600 and IRB 4600).

In order to place the working plate in the virtual environment the position obtained from the Romer arm is used due to its high accuracy.

3.4.5 Gripper TCP definition

A tool that was designed and used during this project was a gripper. For the calibration of the gripper XYZ four point method is used. The gripper finger tips are considered for calibration. The four edges of the finger tips form a rectangle and the center of this rectangle would be the TCP of the gripper tool. This is done by defining the four corners of the rectangle i.e the inner corners of the fingertips. The TCP of the tool can be calculated as the center of the rectangle.
The definition of every edge is carried out similar to that of the calibration tool. It is carried out by jogging the robot into a position in which one of the inner edges of the gripper and the calibration tip are accurately on each other as shown in figure 3.8. Once the robot has reached that point, the position is saved. This procedure is done four times and the orientation of the tool is changed for every point. After all four points have been defined the flexpendant will display the TCP for that point.

Once all the TCP are defined, the TCP of the gripper can be calculated. In Addition to the calibration of the gripper the values obtained are updated in the virtual model to improve the accuracy of the model. The results are presented in further stages.

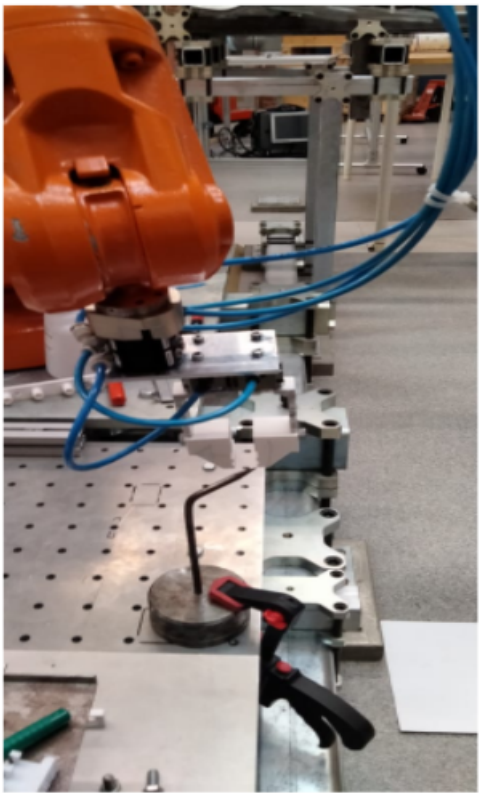


Figure 3.8: Gripper TCP definition

3.4.6 Vacuum TCP definition

A tool that was designed and used during this project was a Vacuum. For the calibration of it, the XYZ 4 point method is used. One corner of the plate where the vacuum is mounted is considered for the calibration. This is done by defining a TCP in one of the corners of the plate where the vacuum is mounted.

The definition of the TCP in the corner is carried out similar to the one of the calibration tip tool. It is carried out by jogging the robot into a position in which one of the corners where the vacuum is mounted and the calibration tip are accurately on each other as shown in Figure 3.9. Once the robot has reached that point, the position is saved. This procedure is done four times and the orientation of the tool is changed for every point. After all four points have been defined the flexpendant will display the TCP for that point. Once the TCP is obtained, the data given by the flexpendant is taken into the virtual model in order to attach the tool correctly in the virtual model. Once the position of the tool in the real world with respect to the virtual world matches accurately, another TCP is defined in the virtual world in the nozzle of the vacuum. The results are presented in further stages.

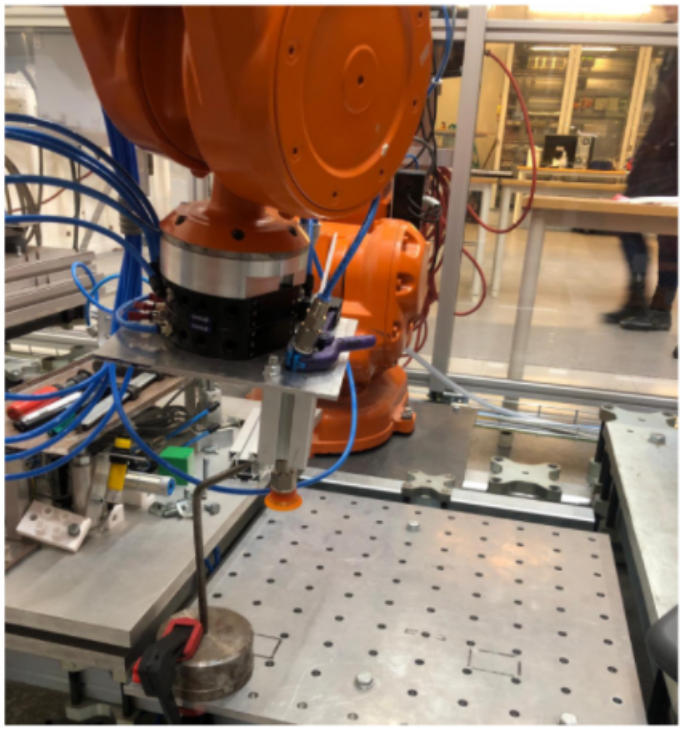


Figure 3.9: Vacuum tool TCP definition

3.4.7 Laser Tracker

A laser tracker is a device used to measure. The device uses a laser and an optical target or reflectors to measure a point as seen in figure 3.10. The laser tracker emits the laser beam to a retro reflective target and the laser is reflected back to the laser. The reflected laser hits the distance meter which can provide an absolute distance [21]. The device has absolute accuracy when performing static measurements of about 10 microns. The measurement results in a single point .i.e the point only has (x,y,z,) values and does not have orientation for a point. This device has been used to measure the position of the reference holes. All the positions of reference holes were measured with respect to the world coordinate. The position of the reference holes obtained during the base position calibration using the robot, would be compared to the positions obtained by the laser tracker. Comparing both the values shows the error in calibration.

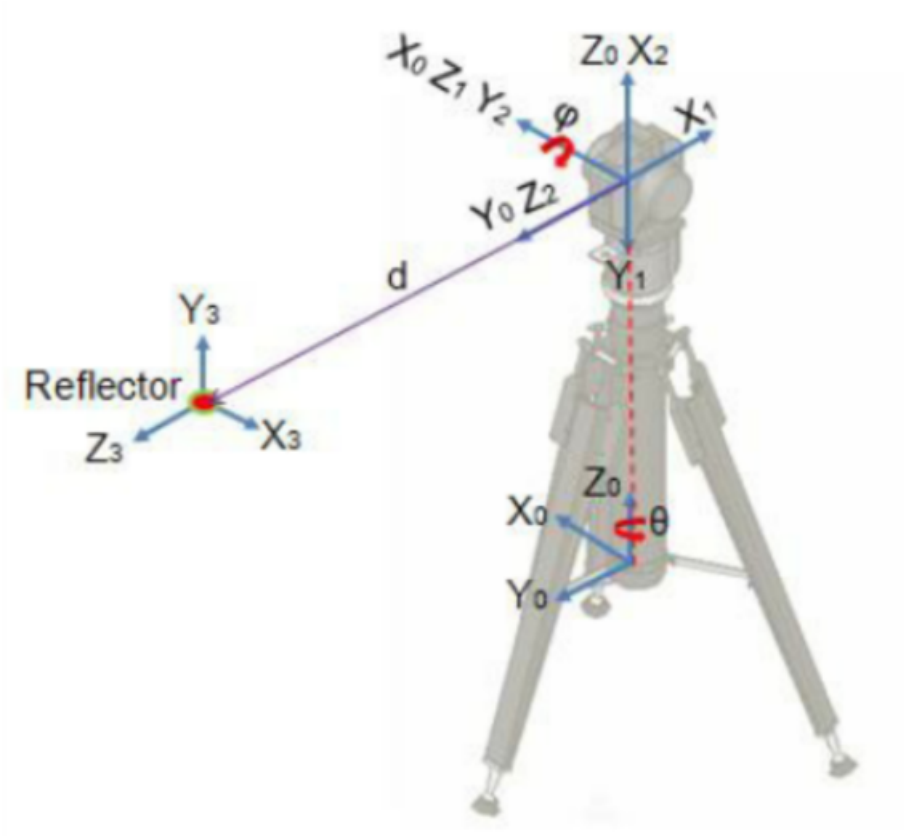


Figure 3.10: Image of a laser tracker [21]

3.4.8 Point to Point program

This test was used to evaluate the accuracy of the virtual model compared to the physical model and to check how 3DExperience deals with work object definition. When defining a new work object in 3DExperience, it is defined in reference to the world coordinate system defined by default in the virtual model. Nevertheless, the world coordinate system in the real model could be different, thus the work object data i.e. the position and orientation has to be modified in order to match it with the actual real model.

This test consists of creating a program in the virtual model in which both robots IRB4600 and IRB1600 converge at the same points. A new work object is defined in the origin with respect to the world coordinate system so both robots work under the same coordinate system. Then a program is created in the virtual model where both of the robots move with reference to the work object that was previously created. The four points of the robot are the start point , then a point moving 500mm in X-axis and similarly moving 500mm in Y-axis and Z-axis. Once the program is created in the virtual model, it is deployed in the real world. As it was said before, the work object must be updated before the program is tested.

The alignment in the cell is perfect for the robot and the tool if both calibration tips

of the robots converge at the same point with respect to the work object previously defined. However, the gap in between both calibration tips yields small differences as seen in figure 3.11. The gap in between can be measured by moving one of the robots manually in a way that both the tips converge precisely. The coordinates of the robot moved compared to its original position yields the distance between the two tips. This procedure is done for every point in the program.

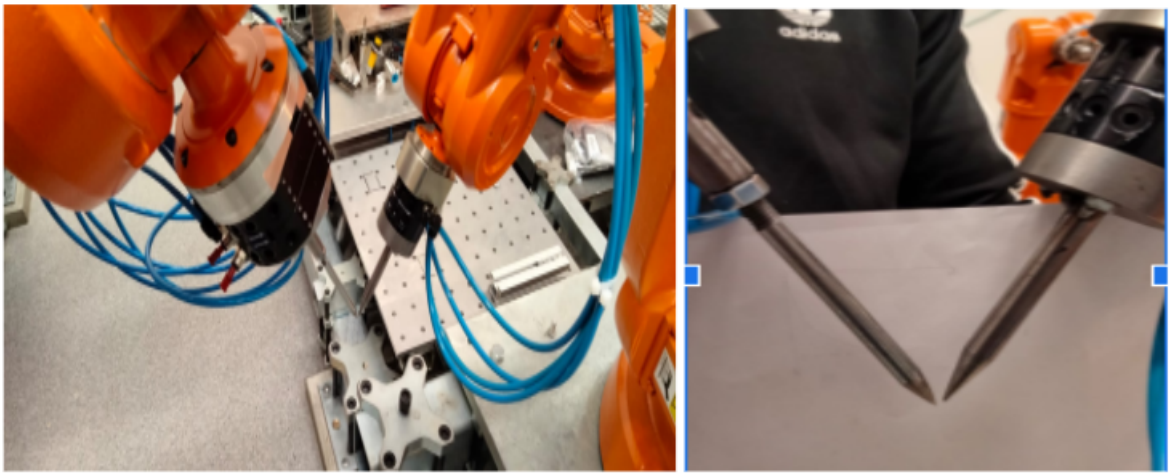


Figure 3.11: Point to point program

3.4.9 Work Object coordinate system definition

Robots can be programmed by using a default work object coordinate system (wobj0). However, in order to increase accuracy when programming it is important to define a new work object coordinate system. The work object can be defined in both the virtual and real environment. Defining the work object in the virtual simulation is much simpler and only requires creating a user fram in the required point. Creating a work object coordinate system using online programming is done by using the 3 point method. This consists of jogging the robot manually using the flexpendant into 3 points in the space, two points in the X-axis and one point in the Y-axis. In order to define a new work object using the flexpendant follow the following path; menu-jogging-work object-new-edit-define-3 point. The first point will be the origin of the work object and it will be one corner of the working plate, the second point will be another corner of the working plate moving along the X-axis and finally the third corner will be another corner of the working plate but this one moving from the origin corner along the Y-axis. Once all 3 points have been modified the work object is defined as seen in figure 3.12. Work object are important because the accuracy is increased as the new point defined is closer to the working area, secondly all targets in the program will be with reference to that point which helps out when collaborating with 2 or more robots, finally if the position of the object is changed, the targets of the program will not have to be modified, only create a new work object in the new position of the object and update it in the program.

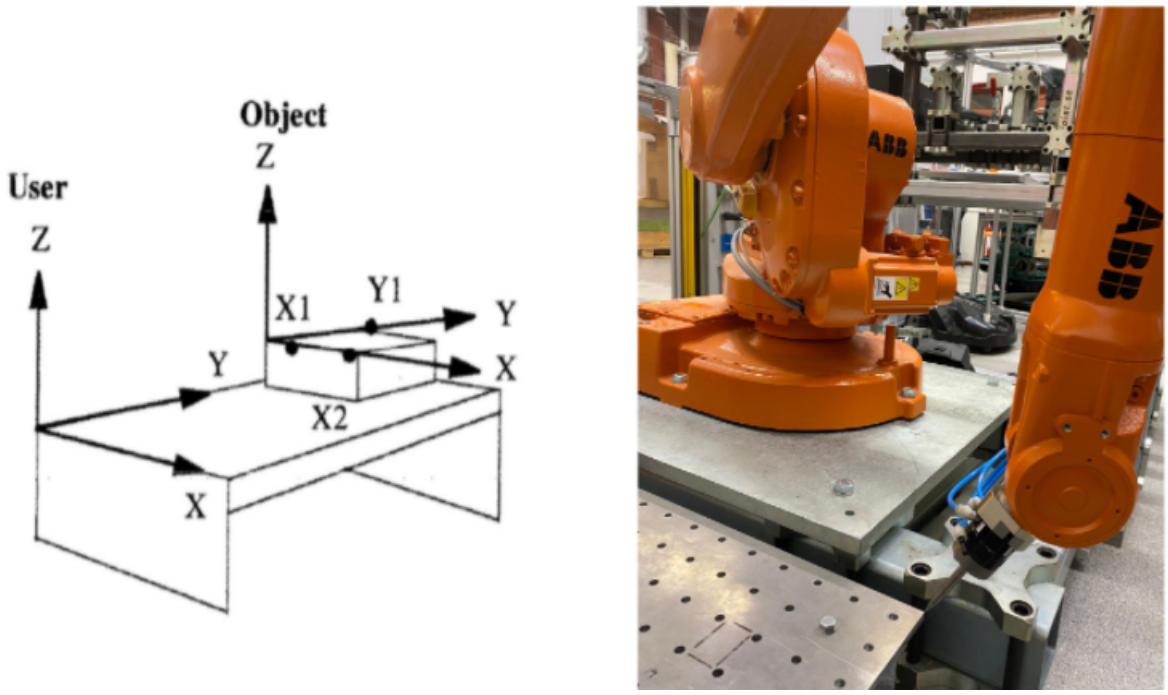


Figure 3.12: Work object coordinate system

During this project a work object at the corner of the steel plate/working plate has been created. A four point program has been built using the work object in the corner of the steel plate. This program has been tested with both the robots IRB 1600 and IRB 4600 to check the accuracy. This helps validation of the robot accuracy and any offsets found can be updated and re-evaluated to attain accuracy. This can help create easier and better assembly of the product that is carried out in the future stage.

3.5 Product design

3.5.1 Design

The design was created based on the functional requirements and test carried out for the validation. The tests such as sliding and snap fit mechanisms had certain requirements to be fulfilled to achieve functioning mechanisms. The design was created using computer aided design(CAD) software called Solid Works that is developed by Dassault Systems. The designs created for the products are limited by size, since the products were 3D printed with a finite printable dimension. All the parts designed were transferred to 3D Experience as a .sldprt file to build the virtual model. The design parts that needed to be printed were converted to .stl file which was the most suitable file format for printing. Some designs of the tools such as SHUNK tool and the PIAB vacuum grippers were directly downloaded from the company website in the required file formats.

3.5.2 3D Printing

A more sustainable way for prototyping was carried out using additive manufacturing technique.Various parts, products and sub assemblies used during the project were created by 3D printing. A Fused Deposition Modelling(FDM) method of printing was used to print polylactic acid of high tensile variant (PLA-HT). The .stl file from the design was used to print. While printing the parts were oriented in such a way that required no or minimal support structures. The different orientation also reduced the time to print and the material consumed. The figures in appendix[] show the various parts printed using this method.

3.5.3 Fixtures

The fixtures designed for the experiments were also built using 3D printing and the holes in the fixtures were made to coincide with the hole in the steel plate. The holes in the metal plate helped to secure the fixture in place with the help of bolts. The holes in the fixtures were given a tolerance of 0.5 mm considering the shrinkage during printing. The pins or groves in the fixture were designed to secure the workpiece in place. Although the workpiece was not fixed to the fixture, the groves or pin hindered the workpiece from moving and can be seen in the figure 3.13. The fixtures designed satisfied the intended purpose i.e. restricting the movement in the required degrees of freedom.

Figure 3.13: Fixture for car assembly

3.6 Virtual validation

This stage describes the method followed to approve the virtual model. It includes the process and tools used to test and simulate the model in different environments and conditions in order to ensure that the model works properly. Some tools and tests that were used are described below.

3.6.1 Robot Simulation

Once the programs are written, its performance can be tested in the simulation software. For every simulation the layout, tools needed and product parts have to be prepared and put in position. In order to prepare everything for the test, simulation states are created. Simulation states allow the user to go back to the original state of the simulation once the simulation is terminated, by doing so, the user can run the test as many times as needed to validate its performance.

3.6.2 Reachability test

While programming, the reachability test checks if all the points and positions are actually reachable for the robot. This method consists of running the simulation point by point and following the travel of the joints. If a point is not reachable the simulation will stop. The reason why a point is not reachable is due to various reasons such as, singularity, a robot exceeding joint travel limits, the motion type chosen or simply because the robot cannot reach that point. During the simulation, the singularity state is detected as when the robot enters this state it turns red as shown in figure 3.14.

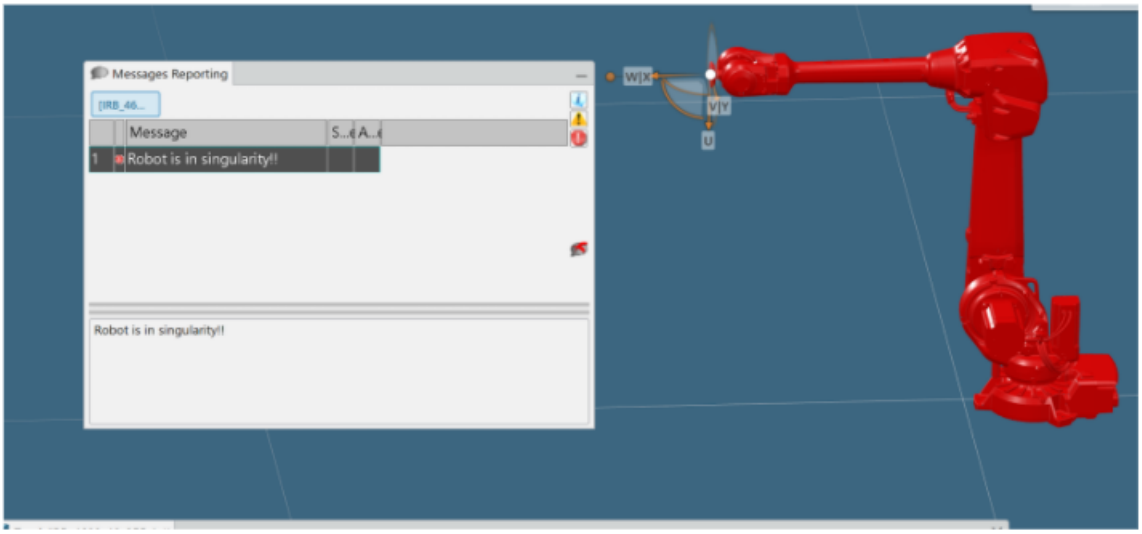


Figure 3.14: Robot in singularity

On the other hand, when the travel limit of the joint is exceeded, the joint in the flexpendant will change from green to red as shown in the figure 3.15. In order for

the robot to reach a point all the joints have to be marked as green.

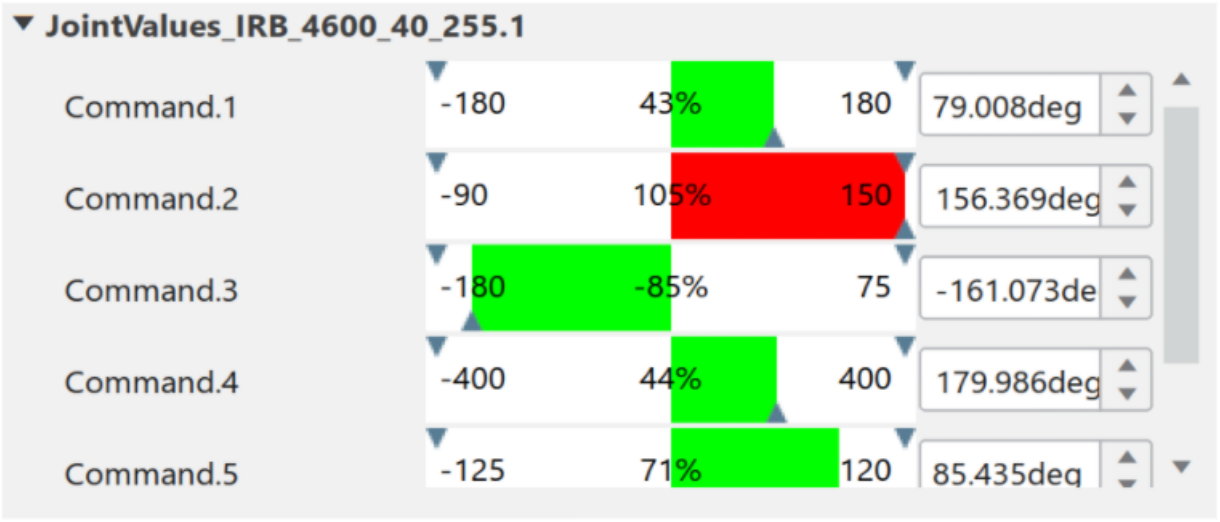


Figure 3.15: Travel limits in 3D experience

In order to solve reachability problems the configuration of the robot can be changed. By changing the configuration of the robot its posture is changed, thus the robot can reach that point with a posture that might not exceed the travel limits of the joints nor entering in singularity. Another way to solve reachability problems is by changing the motion type chosen. The motion type can be changed for either linear, joint or circular movement.

3.6.3 Dynamic Clash Test

When running the simulation, the collision detection tool will highlight the parts that are colliding or stop the simulation when a collision is tracked down. The two or more parts that are converging will be highlighted as shown in figure 3.16. This tool allows the user to prevent and avoid collisions in the real tests.

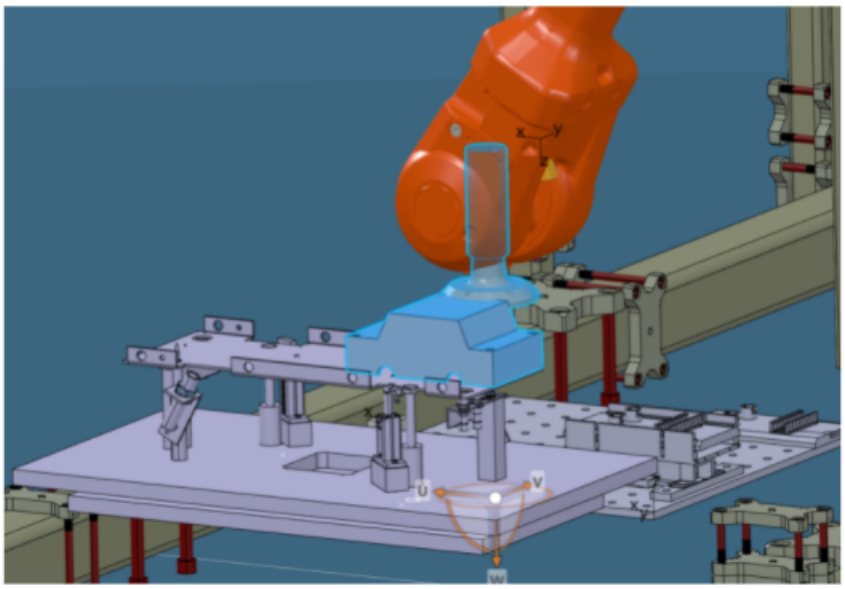


Figure 3.16: Dynamic clash test

3.7 Physical validation

This stage involves all the methods used for the physical validation of every test that was previously run and validated virtually. Once the virtual test has been approved, it is ready for testing in the real world. In order to test in the real world, the program has to be translated into a language that the robot understands as previously mentioned.The post processor used depends on the supplier of the robot. For this project ABB robots were used, therefore the post-processor used was "DELMIA Rapid Translator R2021xFP2042 vst-R1132101022211-00000044 A.1".

As soon as the programs have been translated they are ready to be tested. The flexpendant of the robot is used for running each test. There are several ways of importing the robot programs into the robot flexpendant. A physical connection such as a USB drive or LAN cable are used.

Once the program is imported in the robot flexpendant it can be run. To be validated, it is run line by line and compared with the virtual model. The program is expected to run exactly the same in both the virtual and physical environment. This is only possible in an ideal condition. However, in reality the robot may not perform exactly as programmed in the virtual environment. The validation of the robot program in both environments yields a comparison on how well the robot performs in the real environment to what is expected. The difference between the physical and virtual environments can be reduced with improvement in each iteration.

Robot Sequencing
Use of multiple robots in the same production cell requires sequencing. Sequencing provides the order in which the tasks have to be carried out. This avoids collision between the robots and/or damaging the work pieces. During online programming the signals to activate the sequence can be through a socket that connects the robots together by using the string function or can be done with the help of PLC signals. During offline simulation using 3DExperience there are several ways to mimic the sequencing. A virtual PLC connection can be built or direct signal from I/O can be used. The figure **??** shows the sequence of tasks carried out in series and in parallel. However, in both cases the robots have to be programmed with reference to the same workobject, such as the world coordinate system.

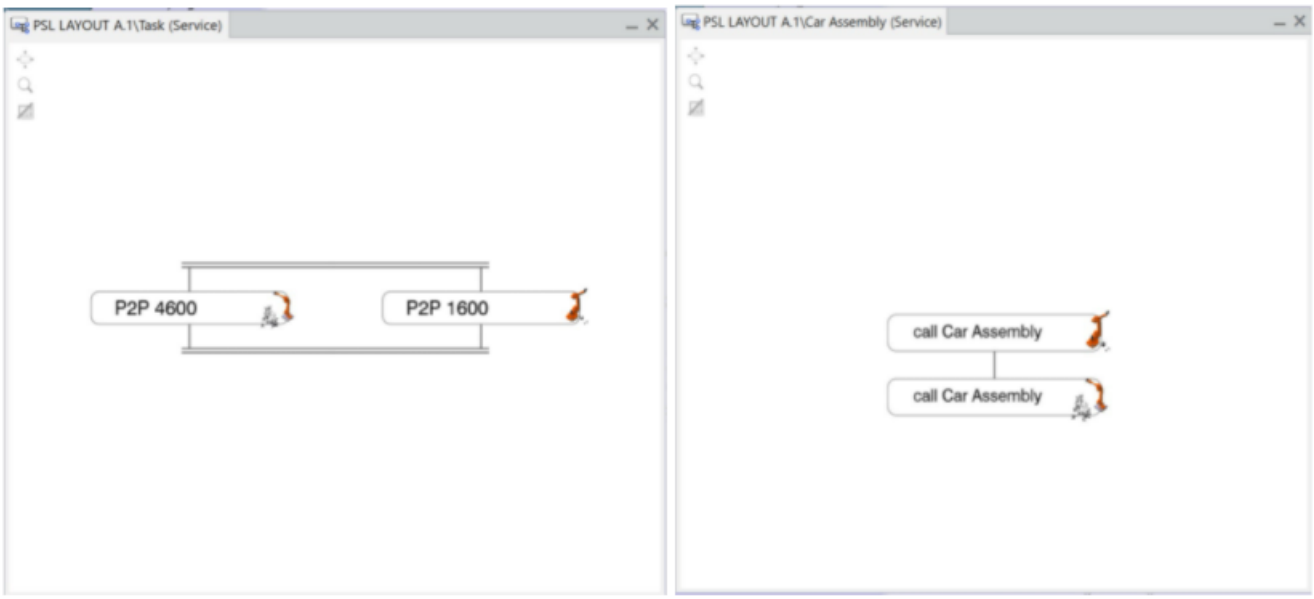


Figure 3.17: Sequencing in 3D experience

4
Results

The results obtained during this project work are presented in this chapter. The results are presented in section, as it was done in the methods part.

4.1 Equipment design

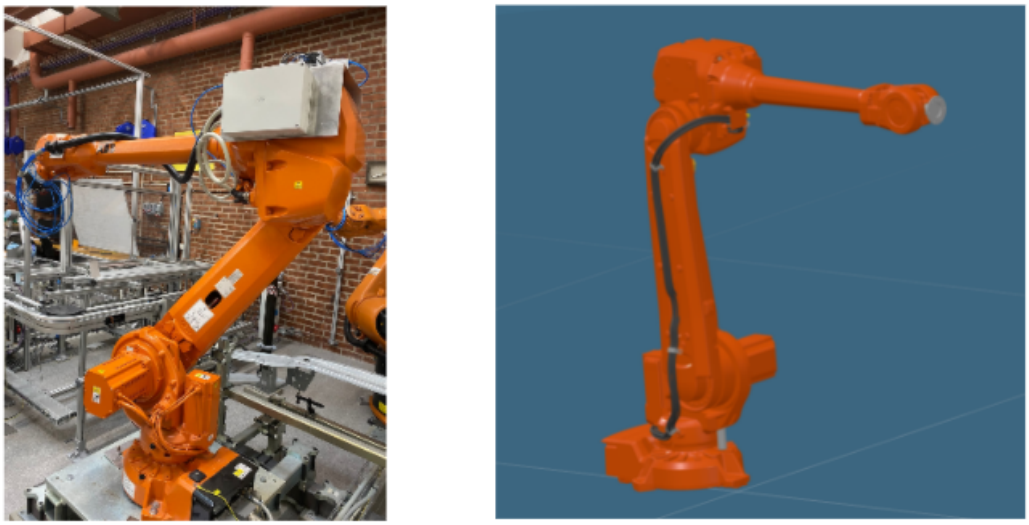
The results for the equipment design are presented in this section. This section includes all the virtual and physical representation of all the resources used during this project.

4.1.1 Physical and virtual robot

This section shows the virtual and physical representation of the industrial robots used during this project. The physical and virtual representation of the IRB 1600 can be seen in figure 4.1 (a),(b). On the other hand, the physical and virtual representation of the IRB 4600 is seen in figure 4.1(c),(d). Both virtual models are an accurate representation of the physical model.

a) Physical IRB 1600

b) Virtual IRB 1600

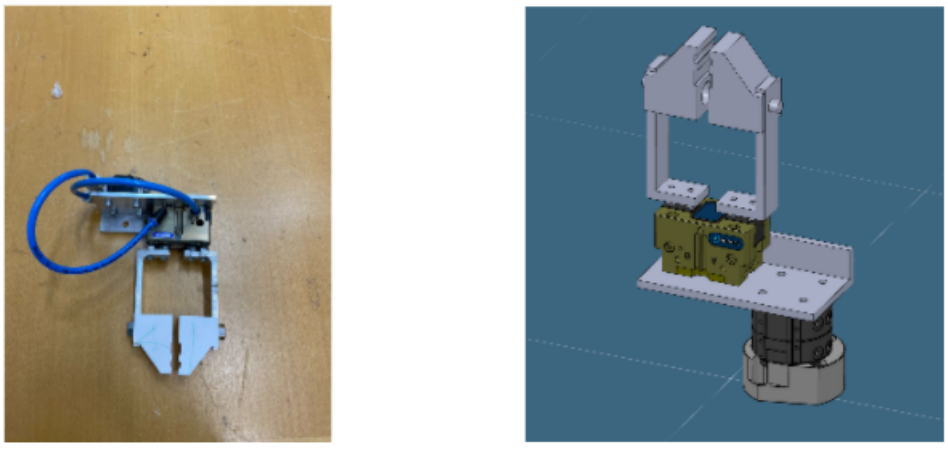


c) Physical IRB 4600 d)Virtual IRB 4600

Figure 4.1: Equipments required for calibration

4.1.2 Physical and virtual gripper

In this section the physical and virtual representation of the gripper are shown. The physical and virtual representation of the gripper can be seen in figure 4.2. The virtual model of the gripper is an accurate representation of the physical gripper, however some parts were 3D printed, therefore there is a slight difference between. the virtual and physical model. The difference between the virtual model of the gripper with the physical model is explained in further stages.



a) Physical Gripper b) Virtual Gripper

Figure 4.2: Gripper tool

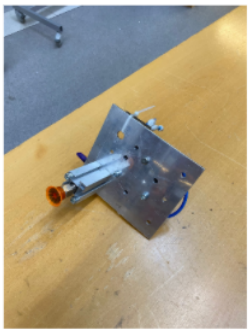
4.1.3 Gripper Tool Profile

This section presents the tool profile of the gripper. The tool profile includes information about the gripper such as TCP, mass of the gripper and tool type. The tool profile data is shown in figure 4.3

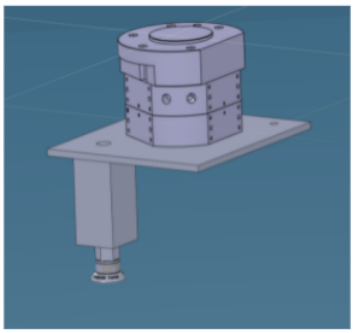
Figure 4.3: Gripper tool profile

4.1.4 Physical and virtual vacuum

In this section the physical and virtual representation of the vacuum are shown. The physical and virtual representation of the vacuum can be seen in figure 4.4. The virtual model of the vacuum is an accurate representation of the physical vacuum.



a) Physical Vacuum



b) Virtual Vacuumr

Figure 4.4: Vacuum tool

4.1.5 Vacuum Tool profile

This section presents the tool profile of the vacuum. The tool profile includes information about the vacuum such as TCP, mass of the gripper and tool type. The tool profile data is shown in figure 4.5.

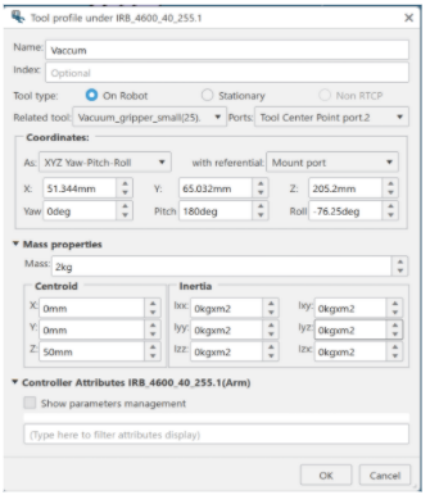
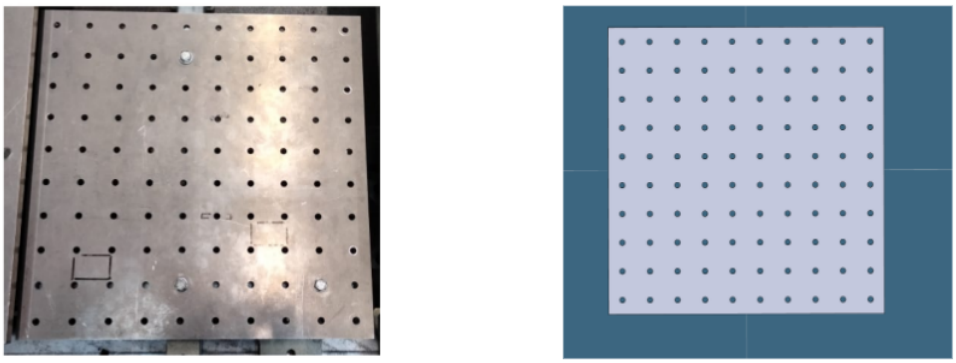


Figure 4.5: Vacuum tool profile

4.1.6 Physical and virtual working plate

In this section the physical and virtual representation of the working plate are shown. The physical and virtual look of the working plate can be seen in figure 4.6. The virtual model of the working plate is a dimensionally accurate representation of the physical working plate. Details such as the bolts have not been added in the virtual model.



a) Physical working plate b) virtual working plate

Figure 4.6: Working plate

4.2 Layout design

The results for the layout design are presented in this section. This stage includes the virtual and physical representation of the production cell layout.

4.2.1 PSL physical

In this section it can be seen the current state of the PSL in Johanneberg.The current state of the PSL is shown in figures 4.7.

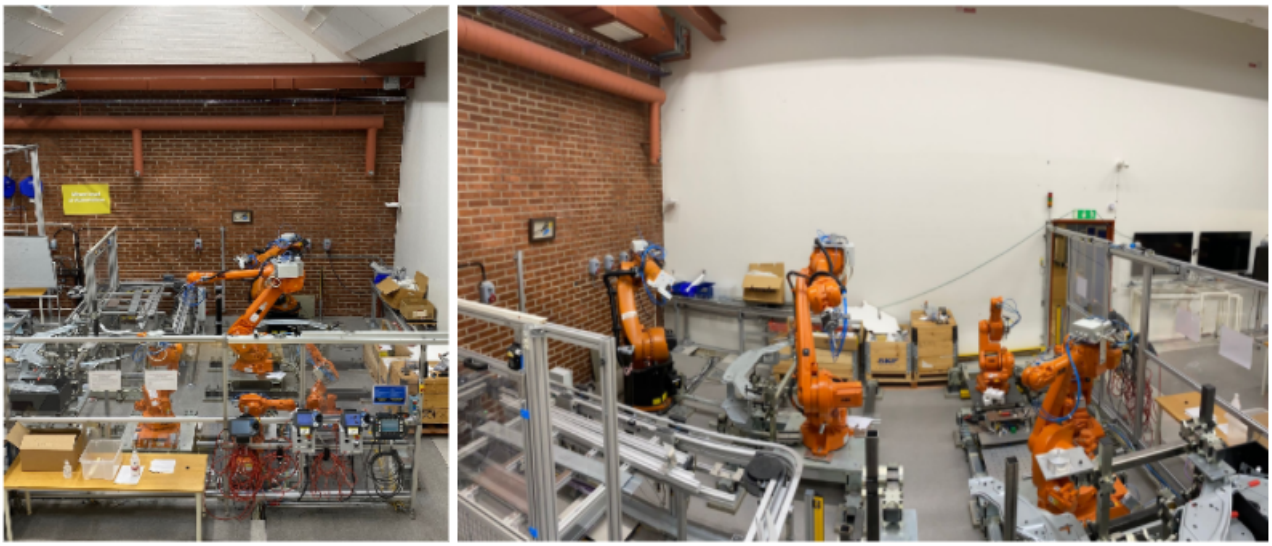


Figure 4.7: Physical state of PSL

4.2.2 PSL digital twin

This section presents the virtual representation of the PSL. Once all the equipment and resources included in the current PSL layout were designed, they all were imported into the virtual model and placed accordingly to the current layout of the PSL. The position of every robot and the working plate in the space are adhering to calibration values shown in the following chapter. The virtual layout created in 3DExperience can be seen in figures 4.8.

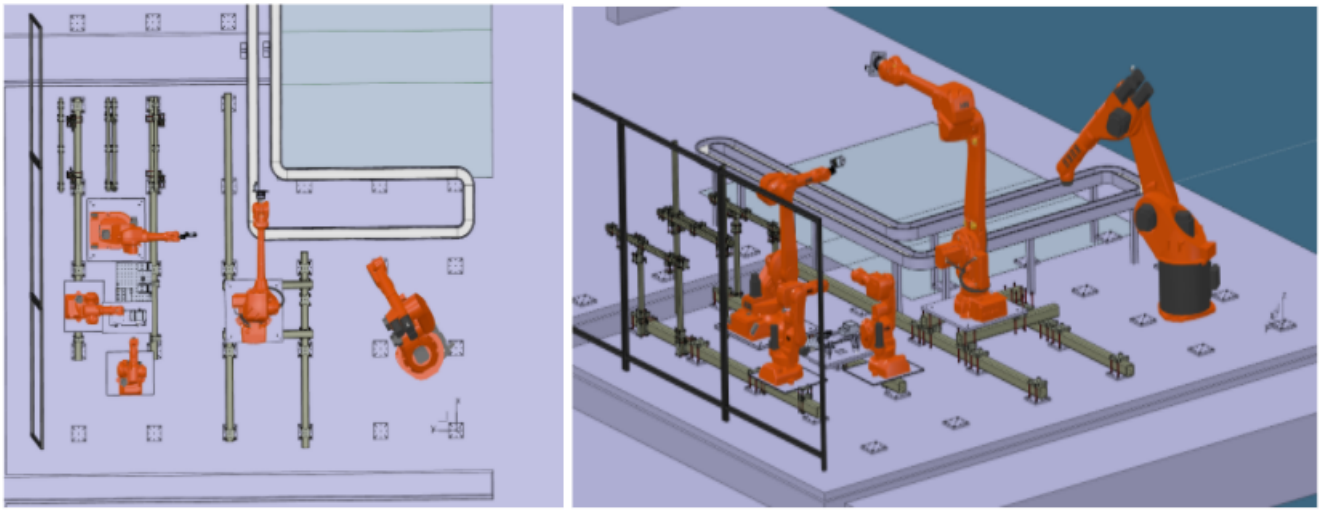


Figure 4.8: Virtual state of PSL

4.3 Calibration

The results for the calibration methods performed for the development of the digital twin are presented in this section.

4.3.1 Tool center point calibration tip

This section discusses the method used for the definition of the TCP of the calibration tool. The method was carried out for both calibration tools, the one for IRB1600 and the one for IRB 4600. The TCP data for the calibration tools obtained are shown in table 4.1 and 4.2. This data was imported into the virtual model and used for the point to point program.

Table 4.1: TCP calibration tool IRB 1600

	TCP calibration tool IRB 1600
X	-0.130289
Y	-1.3375
Z	164.868

Table 4.2: TCP calibration tool IRB 4600

	TCP calibration tool IRB 4600
X	-0.983006
Y	2.988368
Z	256.369

4.3.2 Performance test of the calibration point tools

This section discusses a rough method for the definition of the robot base position. The step by step methodology followed for the performance of this method is explained in appendix 3. The method was carried out for IRB 1600 and IRB 4600. For the calibration of IRB 4600 seven reference holes were measured and the results are shown in table 4.3 and table 4.4. On the other hand for IRB 1600 two calibration reference holes were measured and the results are shown in table 4.5 and table 4.6.

Table 4.3: Coordinate of the reference holes with respect to the base of the robot IRB4600

IRB 4600	X	Y	Z
B5_3	-527.18	1288.49	-294.54
C5_3	521.48	1290.6	-295.06
D5_3	1571.7	1292.33	-293.57
D4_1	1575.13	402.58	-296.05
C4_1	522.7	399.29	-296.89
B4_1	-525.66	395.76	-296.09
A4_1	-1573.09	394.23	-294.66

Table 4.4: Coordinate of the reference holes with respect to the world coordinate system

	X	Y	Z
B5_3	1045.3	4148.4	1.1
C5_3	2093.9	4130.1	1.3
D5_3	3144.1	4131.5	3.5
D4_1	3147.3	3241.7	3.1
C4_1	2094.8	3238.8	1.5
B4_1	1046.5	3235.7	1.6
A4_1	-1	3234.5	2.3

Table 4.5: Coordinate of the reference holes with respect to the base of the robot IRB1600

IRB 1600	X	Y	Z
C5_3	610.26	-527.8	-299.27
D5_3	607.51	522.09	-302.03

Once all the reference holes were calculated, the results obtained are compared with the provided values based on the laser tracker method. The results of the comparison are shown in table **??**.

Table 4.6: Coordinate of the reference holes with respect to the world coordinate system

	X	Y	Z
C5_3	2095.49	4128.43	0.33
D5_3	3145.38	4131.18	-2.43

Table 4.7: Comparison of results of robot with results of laser tracker

a)Comparison with IRB 4600

Reference hole in IRB 4600			
	X	Y	Z
B5_3	1045.3	4128.4	1.1
C5_3	2093.9	4130.1	1.3
D5_3	3144.1	4131.5	3.5
D4_1	3147.3	3241.7	3.1
C4_1	2094.8	3238.8	1.5
B4_1	1046.5	3235.7	1.6
A4_1	-1	3234.5	2.3
Reference hole in Laser Tracker			
	X	Y	Z
B5_3	1045.28	4129.036	0.07516
C5_3	2096.075	4131.035	0.1568
D5_3	3145.339	4131.75	0.2383
D4_1	3148.265	3241.961	0.251
C4_1	2096.882	3240.034	0.1692
B4_1	1047.496	3237.594	0.0872
A4_1	0.22264	3235.645	0.0062
Error			
	X	Y	Z
B5_3	0.0201	0.636	1.02484
C5_3	2.175	0.9347	1.1432
D5_3	1.2386	0.325	3.2617
D4_1	0.965	0.261	2.849
C4_1	2.0822	1.234	1.3308
B4_1	0.9963	1.8939	1.5128
A4_1	1.22264	1.1453	2.2938

b) Comparison with IRB 1600

Reference hole in IRB 1600			
	X	Y	Z
C5_3	2095.6	4129.2	-1.6
D5_3	3145.5	4131.02	-1.7
Reference hole in Laser tracker			
	X	Y	Z
C5_3	2096.075	4131.035	0.1568
D5_3	3145.339	4131.75	0.2383
Error			
	X	Y	Z
C5_3	0.475	1.8347	1.7568
D5_3	0.1614	0.155	1.9383

4.3.3 Gripper TCP definition

This section discusses the method used for the definition of the TCP of gripper. The calibration method of the gripper was then compared with the virtual values to increase accuracy during robot programming and robot simulation. The results of the definition of TCP for every corner of the gripper are shown in table 4.8 and table 4.9.

Table 4.8: TCP Real values 4 corners gripper

Real	X	Y	Z
Corner 1	-57.805	16.002	223.1588
Corner 2	-46.9537	43.74803	222.705
Corner 3	-41.795	39.364	223.802
Corner 4	-52.1125	12.5307	223.11

Table 4.9: TCP Virtual values 4 corners gripper

Real	X	Y	Z
Corner 1	-60.666	17.309	223.728
Corner 2	-48.703	44.821	223.728
Corner 3	-41.454	41.025	223.646
Corner 4	-53.417	13.513	223.646

Once all 4 corners are measured, the TCP of the gripper can be defined. The TCP is defined in the virtual model with the use of 3DExperience, nevertheless the calibration is done so that TCP is updated with respect to the real values. The virtual TCP defined data can be seen in table 4.10.

Table 4.10: Virtual TCP gripper

TCP data virtual	
X	-51.069
Y	29.35
Z	186.291

In order to define the real TCP, real values are taken into consideration. For the coordinate in X, X_4 plus the average of (X_1-X_4)/2 and (X_2-X_3)/2. Similarly it is done for the coordinate in Y, Y_1 plus the average of (Y_1-Y_2)/2 and (Y_3-Y_4)/2. The results can be seen in tables 4.11 and 4.12.
Once the real TCP is defined it is updated in the virtual model. The reason why the virtual and the real model differ is because some parts of the gripper were 3D printed, therefore there is a difference between the real parts and the virtual parts. The reason why there is a difference between a real 3D printed part and a virtual part is explained in further stages.

Table 4.11: TCP Calculations

	X	Y	
$(X_1$-$X_4)/2$	2.84626	13.873	$(Y_1$-$Y_2)/2$
$(X_2$-$X_3)/2$	2.57935	13.4167	$(Y_3$-$Y_4)/2$
average	2.71281	13.6448	average

Table 4.12: Real TCP gripper

TCP data real	
X	-49.4
Y	29.6468
Z	186.194

4.3.4 Vacuum TCP definition

This section discusses the method used for the definition of the TCP of the vacuum. The data obtained by the XYZ 4 point method is then taken into the virtual model so the tool is attached properly to the virtual robot. Once the tool is attached to the robot in the virtual world a virtual TCP is defined in the same corner as it was defined in the real world. The definition of the virtual TCP is shown in Figure 4.9. Both TCP virtual and real are compared. The results of the comparison are shown in table 4.13.

Table 4.13: Comparison virtual and real TCP

Real	X	Y	Z
Corner 1	67.57343	91.16629	172.0199
Virtual			
Corner 1	67.822	91.937	172.2
error	-0.24857	-0.77071	-0.1801

Once the calibration of the real model was compared with the virtual model, the virtual model of the vacuum was validated. Once it is validated, another TCP can be defined in the virtual model. The new TCP is defined in the center of the nozzle of the vacuum. The TCP data of the vacuum is shown in table 4.14.

Table 4.14: TCP vacuum

TCP Vacuum	
X	51.344
Y	65.032
Z	205.2

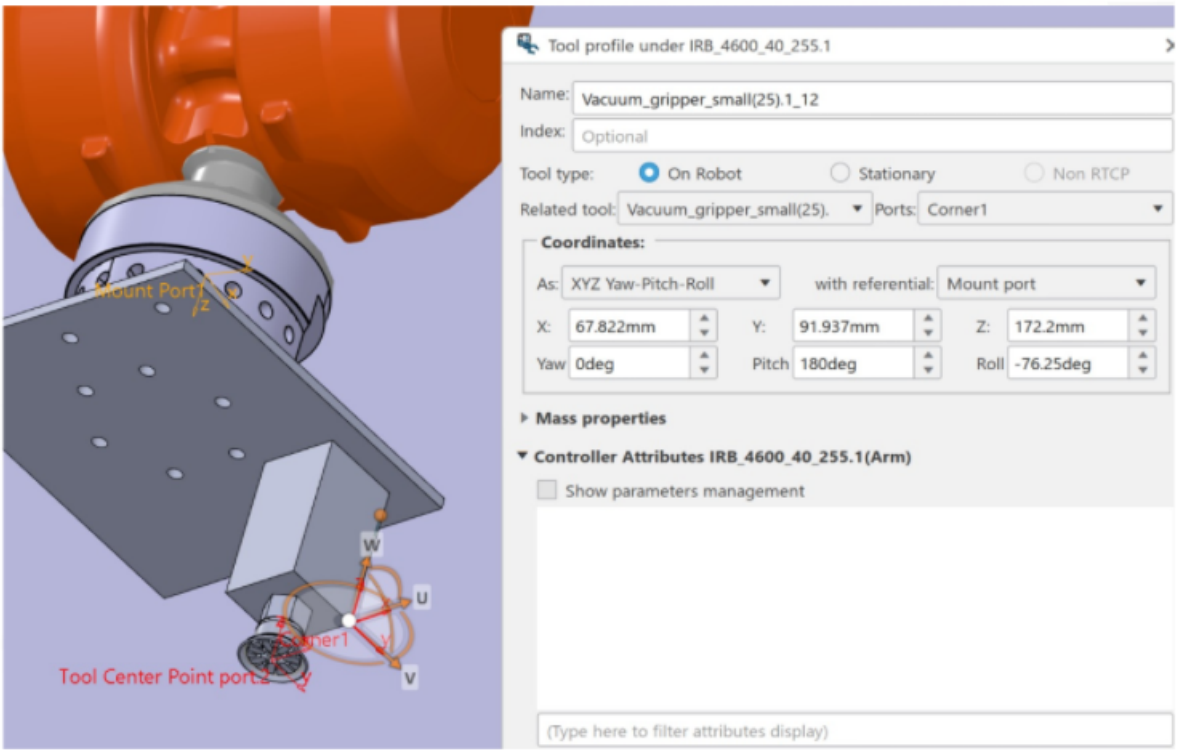


Figure 4.9: Defining TCP virtually

4.3.5 Point to point program

This stage includes the results of the point to point program. The offset in between the two calibration tips are found by jogging a robot manually so both calibration tips converge precisely. Once all points were modified, they were compared with the original values in order to evaluate the accuracy of them. The results obtained are shown in the table 4.15 and 4.16.

Table 4.15: Position of calibration tool tip before and after movement

OLD	Point 1	move Y	move X	move Z
X	1948.445	1948.445	2448.445	1948.445
Y	4041.461	4541.461	4041.461	4041.461
Z	659.474	659.474	659.474	1159.474
NEW	Point 1	move Y	move X	move Z
X	1948.42	1947.96	2448.46	1948.76
Y	4040.28	4539.77	4040.82	4039.17
Z	657.74	656.17	656.99	1157.59

Once the offset is measured, it is updated in the work object data. The new program with the work object updated is then run again. The behaviour of the new program is tracked and compared with the first program. The accuracy of the program happened to be increased. The performance of the new program can be seen in Figure 4.10.

Table 4.16: Position of calibration tool tip before and after movement

Offset	Point 1	move Y	move X	move Z	Average error
X	0.025	0.485	-0.15	-0.315	0.21
Y	1.181	1.69.1	0.641	2.291	1.451
Z	1.736	3.304	2.484	1.884	2.352

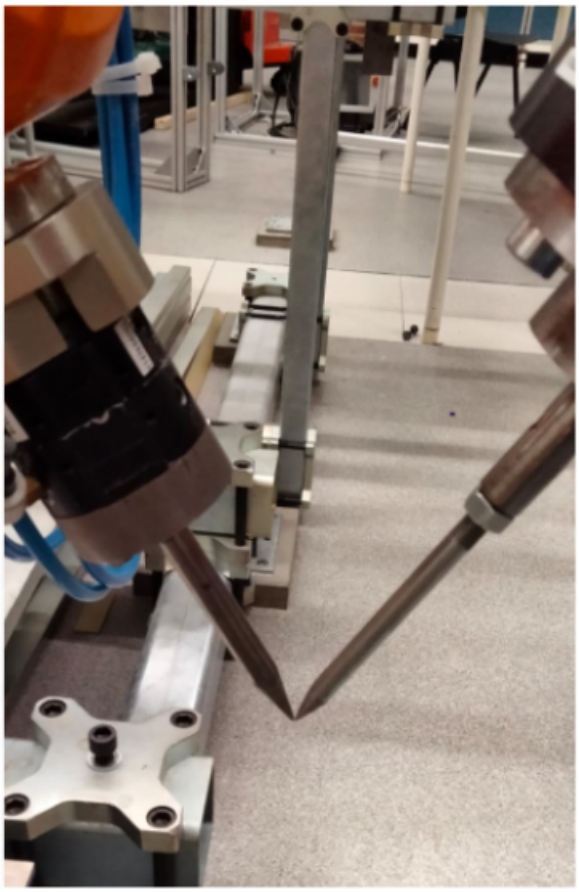


Figure 4.10: Improvement in point to point program

4.3.6 Four Point Program

A four point program has been created to validate the accuracy of the robot with a work object in the steel plate. Similar to the point to point program both the robots equipped with the calibration tool tip are programmed to converge at point and continue to move 400mm in the x-axis and then same distance in y-axis and continue to form a square as seen in the RAPID code in the appendix. The figure 4.11 below shows the convergence of both the robots at the same point. A tolerance of 1 mm was accepted. Any error greater than 2.5 mm is not accepted, leading to redefining the work object and recalibrating the calibration tool tips. However, conducting this program yielded all acceptable values. This program aids to ensure accuracy in carrying out assembly operations of the toy car.

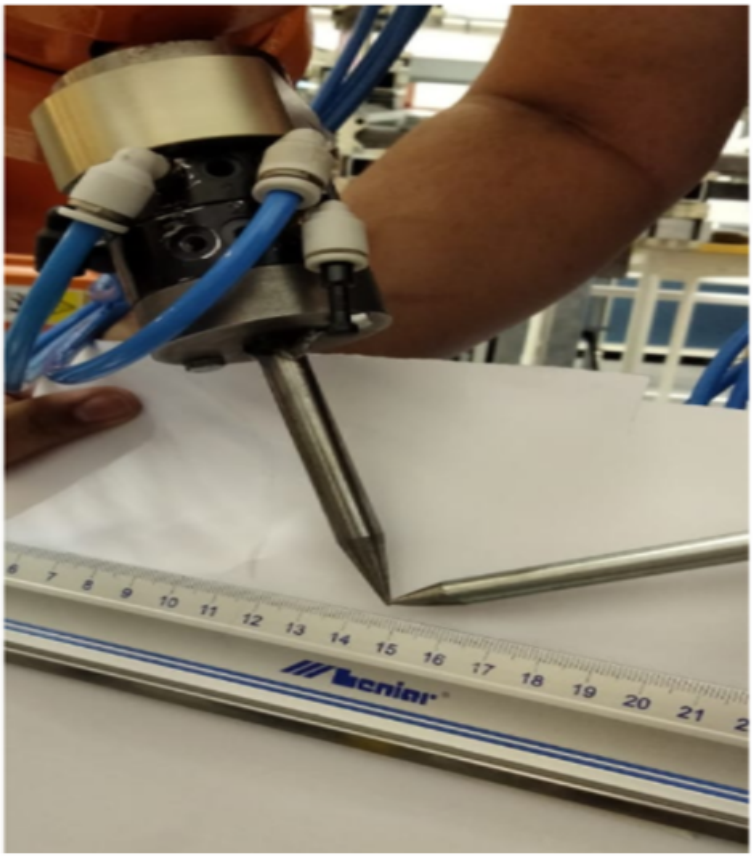


Figure 4.11: Result obtained from Four point program

4.4 Product design

The main aim of the design of the product was to fulfill the functional aspect of the product. The product design was limited in size, by the size of the printer and the support material used, which is discussed in the next section. Each robot had a single tool or gripper attached to the arm. The design of the product also included few features to ease the fixturing during assembly i.e. adding holes in products to pin them to the base or supports during assembly.

The designs were improved in each iteration after the test. This meant redesigning the parts, fixtures and gripper attachments to be more suitable for assembly. The toy car was designed to accommodate different assembly methods such as sliding, snapping or clips and the key- hole. The different assembly methods also included providing the parts with precise geometric tolerances. The increase in tolerance increased the assembly inaccuracy but made it easier for robot assembly i.e. the higher tolerance made it possible for robots with higher calibration error to complete the task. Whereas, lowering the tolerance would improve the assembly between the parts, but the robots had to carry out the task with more precision.

4.4.1 3D printing

As mentioned in the previous chapter the parts for assembly, gripper attachments and the fixtures used during the project were made with additive manufacturing.Since, most of the products were prototypes, 3D printing was most suitable. Although 3D printing is a more environmentally friendly method of manufacturing compared to traditional material methods, there are a few drawbacks. The FDM method deposits the material to build the layers and these layers tend to shrink when cooled, this yields lower geometric accuracy. This could be seen prominently as the parts increase in size as seen in figure 4.13. The two pictures show the dimension in the actual model and the CAD design. Similar dimensional comparison of parts are provided in table 4.17. The smaller parts printed have more dimensional accuracy but lag in sharp corners, the edge or corners of the parts have to be greater than the size of the filament of material used. The parts printed with different orientation tend to yield different results. This could easily be seen in part with a snap fit mechanism . The part when printed in a horizontal orientation was less flexible and would fracture between the layers when minimal force applied. The same part when reorientated and printed is more flexible. The different results can be seen in the figure 4.12. The orientation is an important aspect to consider while printing to obtain the required function or features in the part.

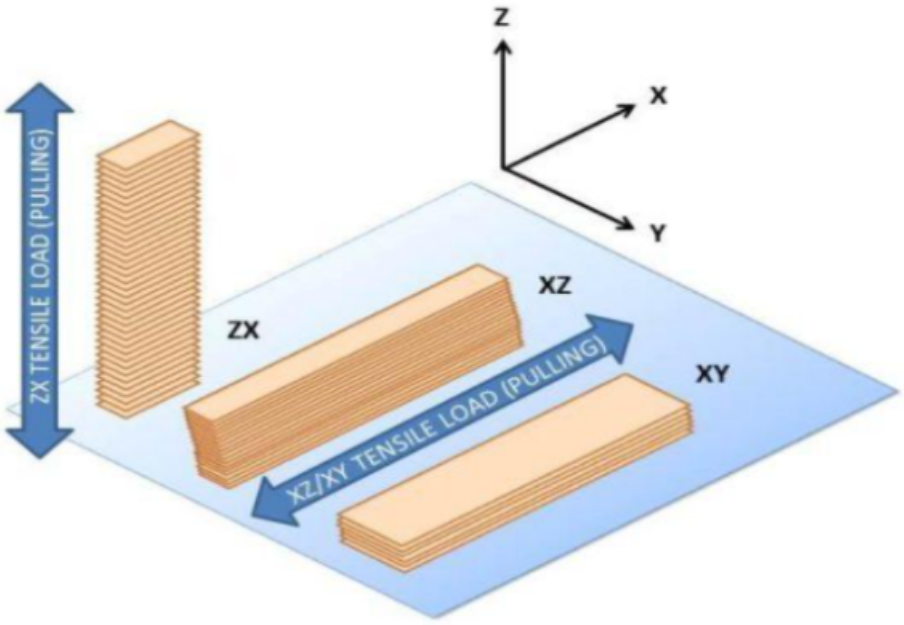


Figure 4.12: Results of printing in different orientation [17]

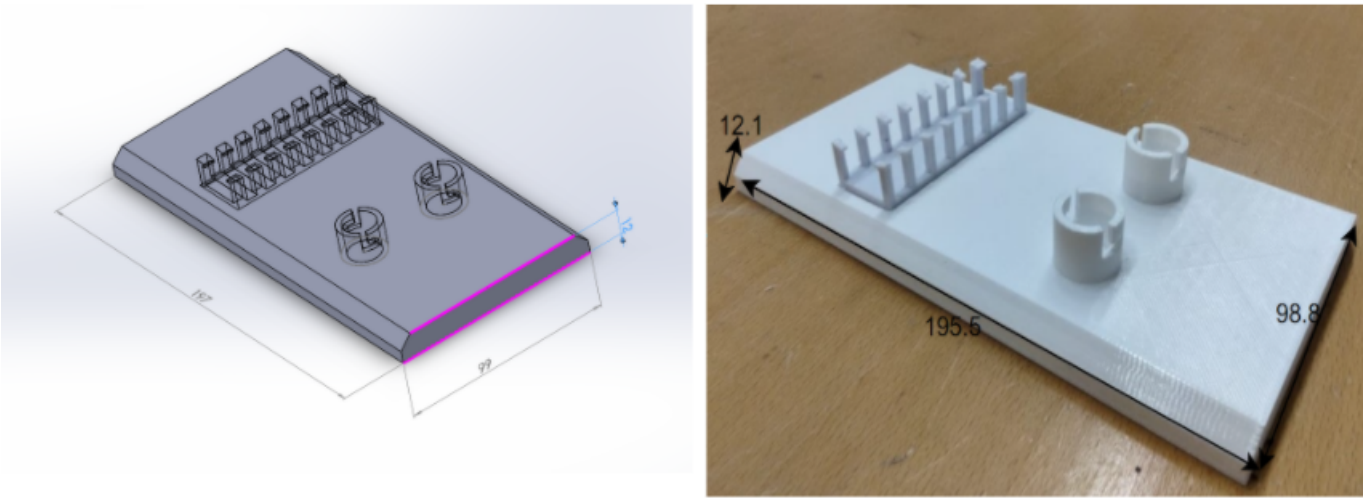


Figure 4.13: Comparison between CAD and physical model

Table 4.17: Comparison of dimensions of actual and CAD model

	Actual model	CAD MOdel
MID PART		
Length	195.5	197
Width	98.8	99
Heigth	12.1	12
BASE PART		
Length	200.2	200
Width	111.5	111
Heigth	39.95	40
CAR BODY		
Length	211	212
Width	122.75	123
Heigth	45.4	46
BACK SEAT		
Length	74.85	75
Width	25	25
Heigth	32.35	32.5

4.4.2 Fixtures

The fixtures designed during this project carried out its intended purpose of holding the workpiece in place and can be in figure 4.14. Certain operations such as sliding and key-hole mechanisms required high precision. The fixtures when bolted to the metal plate had the tendency to move from its position due to the tolerances and lack of treads in the fixture and working plate. The movement was minimal and always less than 1mm and had to be readjusted to the correct position. The tolerances for the pins and the grooves to hold the workpiece also resulted in minor movement. However reducing the tolerances would yield difficulty in assembly due to tight fits between the parts. The fixtures could be redesigned in various methods such as

using a combination of elliptical and round holes to improve fixture position in the plate.

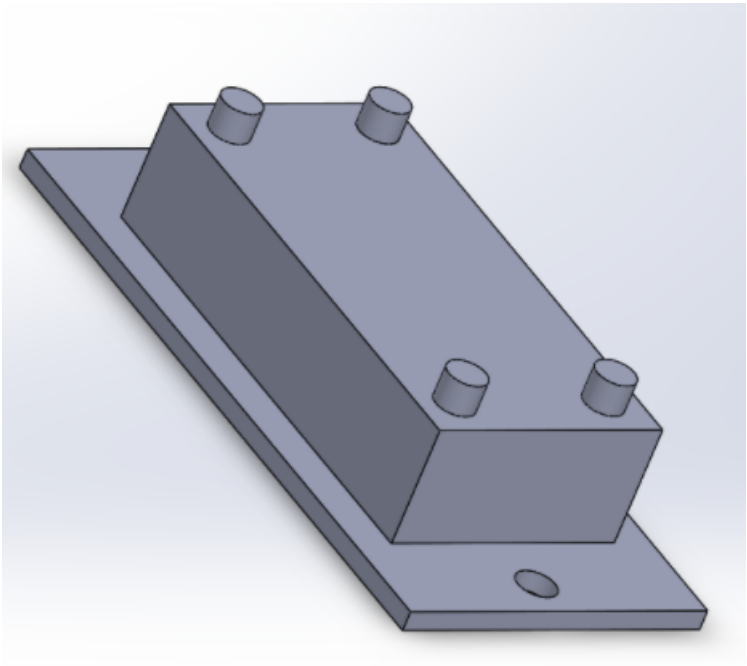


Figure 4.14: CAD model of the fixture

4.5 Virtual validation

For the virtual validation every program was simulated in the virtual model and several tests were made. Clash detection tests and reachability tests were mainly performed for each program. If an error was found, the program is updated depending on the problem found. Multiple simulations are run before a program is validated and ready for physical test.

4.6 Physical validation

For the physical validation, all the programs that were previously validated in the virtual model are tested in the real model. It was expected for the programs to perform as they performed in the virtual model, nevertheless there are some mismatches between the real parts of the assembly and the virtual parts of the assembly. Besides there was an offset between the position of the target in the real model and in the virtual model.

When assembling the parts physically, some of them did not match in the same way they did virtually, this is due to problems with the design or problems with the manufacturing process chosen. Therefore, in order to fix those part design problems, changes on the design are made in relation with the manufacturing process chosen. For the target position error, the offset is measured and compensated in the work object data for all the programs. The offset is also updated in the virtual model for future programs made on it. Once a change on the program is made, it has to be run again to track the behaviour of it. Multiply iterations are conducted before the program is validated.

5
Discussion

This chapter reflects on the methodology used to create a digital twin, validating the production platform and the results obtained.

The digital twin created during this project was for an existing production cell. The perks of this are that all the resources and their placements are already known. However, the exact position and orientation of the resources have to be checked and verified before creating a virtual model. The errors that occur in this stage are reflected when the programs are deployed from a virtual model to the real model. On the contrary, creating a digital twin for a new production cell has the advantage of placing the resources and reorientation that suits the best. For creating the virtual model for PSL the position of each robot was known, however all the resources in the PSL had to be measured before their implementation in the virtual model. Comparing the results from the laser tracker and the calibration, the position differed as seen in the results section. The reason was because an online method was used for the verification of the position of the robot, which in comparison with the laser tracker method is less accurate. Therefore, the positions given from the laser tracker are used to place the resources in the virtual model. Even Though the results obtained with the online method do not differ significantly, the ones obtained using laser tracker were used as accuracy plays a very important role when programming the robots.

As mentioned before, when deploying the robots programs into the real model, the programs have to be slightly modified in order for the robot to understand them. There are several reasons why the programs written in the virtual model did not perform as close as expected in the real model.

One of the reasons being the manufacturing process used for creating the parts and fixtures to assemble the product. 3D printing has been proven to be ideal for prototyping, however the dimensional inaccuracies in the parts make it harder to assemble. This has been overcome by optimising the tolerances. Manufacturing processes with better accuracy can yield better results. The possibility of adding weight and physics to CAD to analyse the force and stress can improve the design such as in the snap fit assembly the right dimension of the clips could be analysed through the force and stresses acting during the assembly.

Furthermore, the digital twin is an approximation of the real model, i.e. there are differences between the virtual and real model. The cell alignment of the robots and

all the resources in the PSL was calibrated in order to make them more accurate, nevertheless there is still a difference between the virtual cell alignment and the real cell alignment. Besides the level of detail in the virtual model is not high, there are some parts in the model as well as in the robot that are not included in the virtual model such as the pneumatic control, tubes, safety sensors, electronic cabinets, etc. In order to deal with this mismatch between what is real and what is virtual some changes in every program are made. As it was mentioned before, every program is run line by line and the performance of it is evaluated. The difference in position in the between the virtual and real environment are noted. These differences have been taken into account and the offset in the workobejct has been made in the virtual environment to compensate for the error. In this way the gap between what is real and what is virtual is closed.

The software used for the virtual simulation i.e. 3DExperience is comparatively less evolved or matured compared to simulation softwares such as ABB Robotstudio. There are features in the model i.e. coordinate frames and work objects that are defined in the virtual model by default and cannot be adjusted so they match with the real model. Therefore some data in the programs had to be modified manually in order to adjust the program to the real model features.

In order to close the gap between the virtual model and the real model some further studies must be done. The calibration of the model must be performed using a more accurate method. In this way the cell alignment of every resource in the model will improve. The improvements can be made in the fixture design, that avoids the movement of the workpiece completely during assembly. Use of spring loaded fixtures helps hold workpiece constraint all degrees of freedom. The use of sensor and programmable logic control(PLC) can improve positioning of the workpiece and communicating with the robot to perform certain tasks. The assembly of the parts can be improved by optimising the geometric dimensioning and tolerances. This can also include testing newer assembly methods such as self aligning parts. Workpiece and fixtures can be created with other material and manufacturing techniques and compared with the 3D printing.

6
Conclusion

The aim of the master thesis project was to analyse, verify and validate the complete chain from design/assembly of a physically manufactured product. For this project, simulations and tests were executed in both virtual and real models and compared. For the virtual tests and simulations, a digital twin of the current state of the PSL was developed. The software 3D experience was used to create the digital twin of the PSL. This is an engineering tool that has various features that help in creating a digital twin and is flexible to change. The flexibility to change aids easier updates in the virtual model. The use of digital twin has proven to be efficient to validate the production platform. The virtual and the real model always have some differences, the level of detail may be high in a virtual model but never exactly the same. Similarly, any simulation in the virtual environment is run in ideal world conditions and similar results may not be obtained in the real world. Reducing these differences will help close the gap between the two environments.

The use of a digital twin has proven to be an efficient tool that can be used by manufacturing companies to increase the performance of its operation. A digital twin could help companies to improve planning, production and cost saving without production stoppage. The difference between the real and virtual world and an approach to overcome these differences have been discussed in this project.The project has successfully implemented the assembly of the car using both the robots in the production cell at PSL. The master thesis has presented topics that can be explored for further work to close the gap between the virtual and real world. However, the suggested improvements provided in this thesis can be implemented and tested to close the gap.

Bibliography

[1] Michael Batty(2018), Digital twins. Journal of Environment and Planning B: Urban Analytics and City Science,Volume: 45 issue: 5, page(s): 817-820. https://doi.org/10.1177/2399808318796416

[2] M. Ghobakhloo, M. Fathi, M. Iranmanesh, P. Maroufkhani, M. E. Morales,(2021) Industry 4.0 ten years on: A bibliometric and systematic review of concepts, sustainability value drivers, and success determinants. Journal of Cleaner Production,Volume 302,127052,https://doi.org/10.1016/j.jclepro.2021.127052 (https://www.sciencedirect.com/science/article/pii/S0959652621012713)

[3] A. G. Frank, L.S. Dalenogare, N. F. Ayala(2019), Industry 4.0 technologies: Implementation patterns in manufacturing companies. International Journal of Production Economics, Volume: 210, page(s): 15-26. Retrieved from https://doi.org/10.1016/j.ijpe.2019.01.004 (https://www.sciencedirect.com/science/article/abs/pii/S0925527319300040)

[4] Frohm Lindström Winroth Stahre, J. V. and M. J. (2008). Ergonomia - International Journal of Ergonomics and Human Factors. Levels of Automation in Manufacturing, 30, 181–207. Retrieved from http://publications.lib.chalmers.se/records/fulltext/76667/local76667.pdf

[5] Mattsson, S. (2018, September). Level of Automation (LoA) in a production system 14 / 71 [Slides]. Retrieved from https://chalmers.instructure.com/courses/4249

[6] Peter Scallan(2003), What is process planning?. Journal of Process Planning: The design/Manufacturing Interface,page(s) 35-62. Retrieved from https://doi.org/10.1016/B978-075065129-5/50003-9 (https://www.sciencedirect.com/science/article/pii/B9780750651295500039)

[7] Halevi, G. (2013). Process and Operation Planning: Revised Edition of The Principles of Process Planning: A Logical Approach (Revised ed.).Retrieved from https://doi.org/10.1007/978-94-017-0259-1

[8] BROWNEt, J., TIERNEY, K., WALSH, M. (1991). A two-stage assembly process planning tooJ for robot-based flexible assembly sys. International Journal of Production Research, 29(2), 247–266. Retrieved from

https://doi.org/10.1080/00207549108930068

[9] Industrial robots, form International Federation of Robotics, Retrieved from https://ifr.org/industrial-robots/

[10] Gunnar Bolmsjö (2014), Definitions and terminology. Journal of Industrial Robots, page(s) 1-19, Retrieved from https://chalmers.instructure.com/courses/9395/files/445909?module_item_id=69833

[11] Per-Åke Jansson , Ragnar Grahn(1995). Journal of Engineering Mechanics: Dynamics.

[12] Per Nyqvist. (2019, April). Offline programming of an ABB industrial robot. Gothenburg, Sweden: Chalmers University of Technology

[13] Musab Al Hayani (2013), Technical Journal of Offline Programming of Robots in Car Seat Production, University West,Trollhättan, Sweden. Retrieved from http://www.diva-portal.org/smash/get/diva2:635121/FULLTEXT01.pdf

[14] S. Axelsson(October 2002), Technical research paper of Off-Line Programming of Robots at Volvo Cars – Floor shop preparation. Robots and Mechanical Systems. Gothenburg, Sweden. Retrieved from https://chalmers.instructure.com/courses/9395/files/397160?module_item_id=69855

[15] What is a digital twin and how does it work?. TWI. Retrieved from https://www.twi-global.com/technical-knowledge/faqs/what-is-digital-twin

[16] Digital Twin, Wikipedia, Retrieved from https://en.wikipedia.org/wiki/Digital_twin#: :text=A

[17] Alkaios Bournias Varotsis, Introduction to FDM 3D Printing. on HUBS a Proto Labs Company from knowledge base . Retrieved from https://www.hubs.com/knowledge-base/introduction-fdm-3d-printing/

[18] Y. K. Rong (1999). Fixture Design principle. Journal of Computer-Aided Fixture Design: Manufacturing Engineering and Materials Processing Series/55, page(s) 14-48.

[19] Allan Tubaileh (2015).Layout of robot cells based on kinematic constraints. International Journal of Computer Integrated Manufacturing 28(11):1142-1154. Retrieved from (PDF) Layout of robot cells based on kinematic constraints (researchgate.net)

[20] Define a Tool(TCP), from Robo DK, Retrieved from https://robodk.com/doc/en/General.html

[21] Conte, Javier Santolaria, Jorge Majarena, Ana Aguado, Sergio. (2015). Laser Tracker Kinematic Error Model Formulation and Subsequent Verification under Real Working Conditions. Procedia Engineering. 132. 788-795. Retrieved from https://www.researchgate.net/figure/Laser-tracker-kinematic-model_fig1_289996976

[22] Steven M. LaValle(2006), Quaternions. From the Cambridge University press Planning Algorithm. Retrieve From http://planning.cs.uiuc.edu/node151.html

[23] Ilian Bonev(2019), What are Singularities in a Six-Axis Robot Arm?. From Mecademic. Retrieved from https://www.mecademic.com/en/what-are-singularities-in-a-six-axis-robot-arm

[24] Operating manual RobotStudio 5.12 (2009), Revision B, Retrieved from https://library.e.abb.com/public/792b4432c1402c40c1257b4b00521525/3HAC032104-001_RevB_en.pdf

[25] RoboDK. Robot configurations. Retrieved from https://robodk.com/doc/en/Interface-Robot-Configurations.html

A

Appendix 1

A.1 Calibration

A.1.1 Calibration tool 4 point method

1. Get all the equipment needed, flexpendant and 2 calibration tips.
2. Place 1 calibration tip in the working place in a position where it is reachable by the robot.
3. Fix the calibration tip with a clamp
4. Place the calibration tip tool in the robot
5. In the flexpendant create a new tool
6. Define new parameters for the tool
7. For the TCP of the tool select 4 point method
8. Jog the robot till a position in which both ends of the tips are accurately on each other.
9. Save the position as point 1
10. For the next point, change the orientation of the tool in the robot. Once the orientation of the tool has been changed, repeat step 8 and save the position as point 2.
11. For the points 3 and 4 repeat steps 10 and 8.
12. Once all the points have been obtained, the flexpendant will automatically display the tool TCP.
13. In order to approve the value, it has to have a mean value lower than 1. If not, some points must be modified.
14. Save the tool data

A.1.2 Working plate calibration 3 point method

1. Mount the calibration tool on the robot.
2. On The flexpendant select the calibration tool previously defined.
3. Define a work object with 3 points on the working plate. The origin will be 1 corner on the working plate, a second point moving along the X-axis and a 3third point moving from the origin along the Y-axis.
4. Jog the robot till a position in which the end of the tip is accurately above the origin corner chosen on the working plate.
5. Save the position by adding a motion instruction. A matrix with the position and orientation with respect to the base coordinate system is displayed.
6. Repeat steps 4 and 5 for the 2 other corners of the working plate.

7. Once the position and orientation of the 3 corners of the working plate with respect to the base coordinate system are obtained, the position of the working plate with respect to the world coordinate system can be obtained.
8. Calculate the position and orientation of the working plate with respect to the world coordinate system by multiplying the homogenous matrix of the base of the robot and the homogenous matrix of 1 corner of the working plate (steps 3 and 4).
9. Calculate the position and orientation with respect to the world coordinate system for every corner of the working plate.
10. Once the position and orientation of every every point with respect to the world coordinate system is obtained, all the points can be compared. The value in x for the second point must be almost the same as the value in x for the origin point as well as the value in y for the third point.
11. Compare the values obtained for every corner of the working plate.
12. The difference between the value of every corner of the working plate must not be greater than +-1mm.

A.1.3 Gripper calibration 4 point method

1. Mount the gripper tool on the robot.
2. Place the calibration tip in the working plate in a position that it is reachable by the robot.
3. Fix the calibration tip with a clamp.
4. In the flexpendant create a new tool.
5. Define the tool parameter for the gripper tool.
6. Use the 4 points method.
7. Jog the robot til a position in which 1 corner of the gripper is accurately on the end of the calibration tip.
8. Save the position as point 1.
9. For the next point, change the orientation of the gripper tool in the robot. Once the orientation of the tool has been changed, repeat step 7 and save the position as point 2.
10. For the points 3 and 4 repeat steps 8 and 7.
11. Once all the points have been obtained, the flexpendant will automatically display the gripper tool parameters.
12. In order to approve the value, it has to have a mean value lower than 1. If not, some points must be modified.
13. Save the tool data
14. In order to get the tool parameter for a different corner of the gripper, another tool must be created in the flexpendant.
15. Once the new tool is created, obtain the tool parameter following the same method.
16. Obtain the position and orientation of 4 corners of the gripper.
17. Compare the values obtained for every corner of the gripper obtained with the virtual values in the 3DExperience model.

A.1.4 Vacuum calibration 4 point method

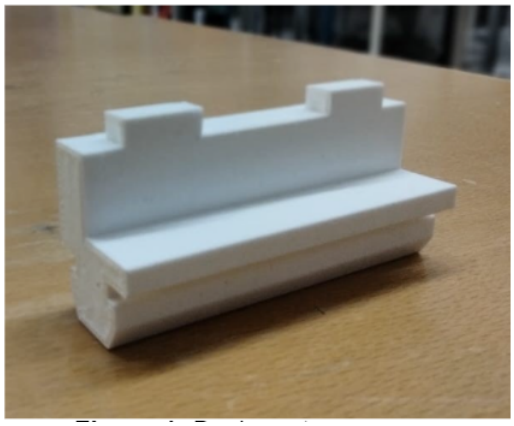
1. Mount the vacuum tool on the robot.
2. Place the calibration tip in the working plate in a position that it is reachable by the robot.
3. Fix the calibration tip with a clamp.
4. In the flexpendant create a new tool.
5. Define the tool parameter for the vacuum tool.
6. Use the 4 points method.
7. Jog the robot til a position in which 1 corner of the vacuum is accurately on the end of the calibration tip.
8. Save the position as point 1.
9. For the next point, change the orientation of the vacuum tool in the robot. Once the orientation of the tool has been changed, repeat step 7 and save the position as point 2.
10. For the points 3 and 4 repeat steps 9 and 7.
11. Once all the points have been obtained, the flexpendant will automatically display the vacuum tool parameters.
12. In order to approve the value, it has to have a mean value lower than 1. If not, some points must be modified.
13. Save the tool data
14. In order to get the tool parameter for a different corner of the vacuum, another tool must be created in the flexpendant.
15. Once the new tool is created, obtain the tool parameter following the same method.
16. Compare the values obtained for every corner of the vacuum obtained with the virtual values in the 3DExperience model.

B

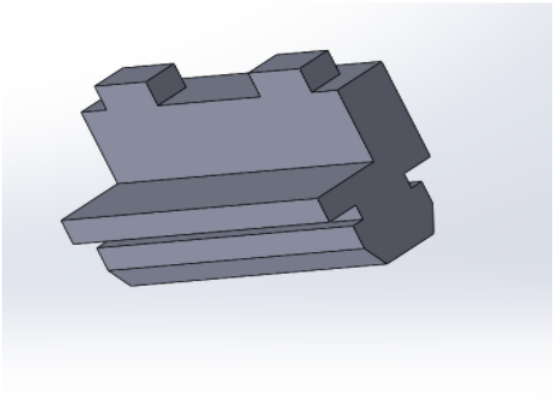
Appendix 2

B.1 CAD models and the 3D printed parts

B.1.1 Back Seat

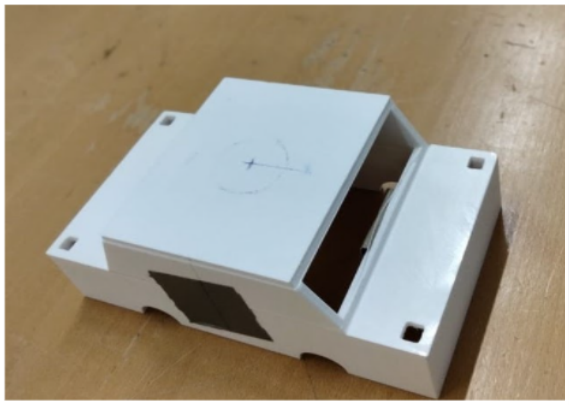


3D Printed model

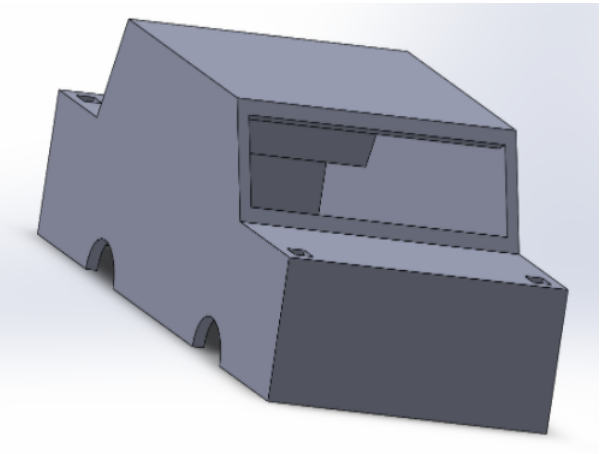


CAD model

B.1.2 Car body of toy car

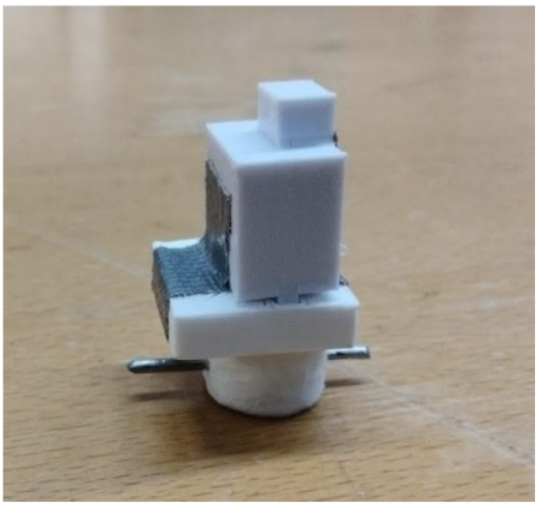


3D Printed model

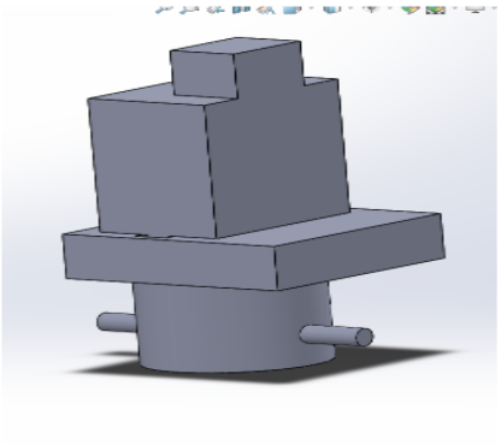


CAD model

B.1.3 Front seats

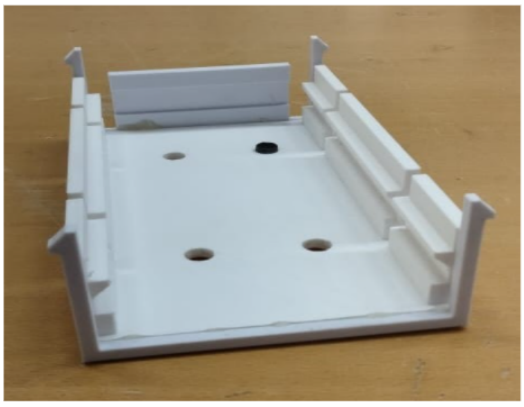


3D Printed model

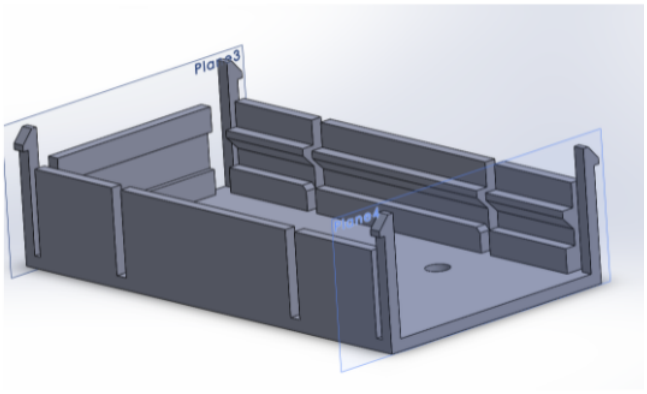


CAD model

B.1.4 Chassis or base frame of toy car

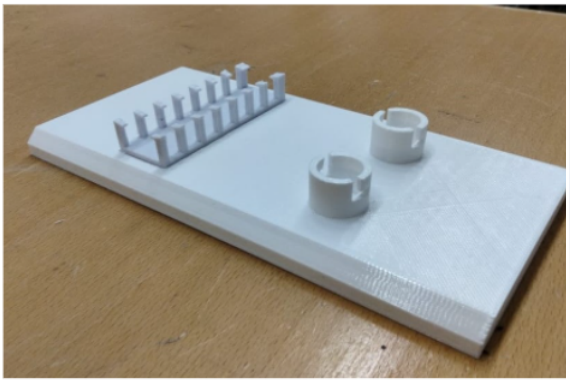


3D Printed model

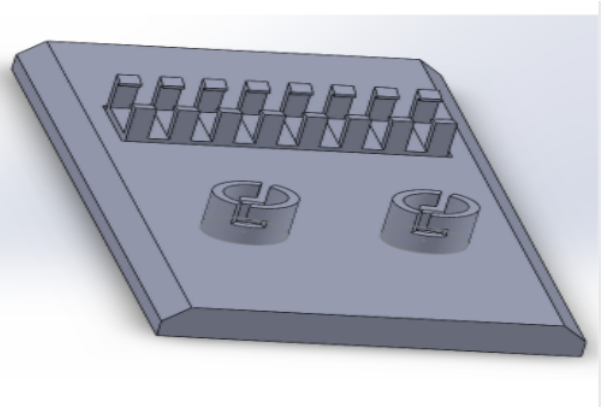


CAD model

B.1.5 Mid part of toy car



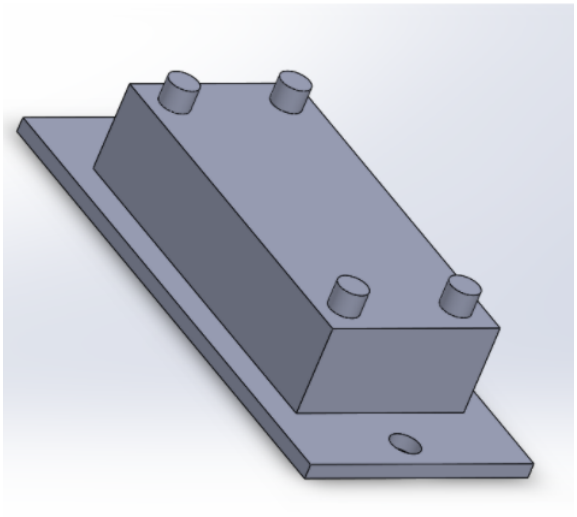
3D Printed model



CAD model

B.1.6 Fixture for toy car assembly

3D Printed model



CAD model

C

Appendix 2

C.1 RAPID CODE

C.1.1 Mid part sliding assembly

MODULE FloorSliding

PERS tooldata Vaccum_:=[TRUE,[[51.344,65.032,205.200],[0.000000,0.617379,0.786666,0.000000]], [2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[-1571.133,-2838.977,-303.026],[0.999999, 0.001171, 0.000348,0.000178]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PERS wobjdata New_Worldsteelplate:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[125.1667,1901.237,-33.13485],[0.00264684,-1,-1.074E-06,-1.089E-05]]];

PROC main()

MoveJ [[1254.685,1845.504,-1472.419],[0.001472,0.999999,-0.000178,0.000348],[0,0,-1,0],[9E+09, 9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

ReleaseBox;

MoveJ [[-176.428,517.305,-1469.495],[0.000795,0.706981,-0.707231,0.001292],[1,0,-1,0],[9E+09, 9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveJ [[450.994,400.035,-516.370],[0.001475,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09, 9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveL [[450.646,401.511,-16.375],[0.001475,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09, 9E+09,9E+09,9E+09,9E+09,9E+09]],v40,fine,Vaccum_\$WObj := New_Worldsteelplate$;

GrabBox;

MoveL [[450.994,400.035,-516.370],[0.001475,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09, 9E+09,9E+09,9E+09,9E+09,9E+09]],v40,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveJ [[171.502,693.408,-568.934],[0.001475,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v40,fine,Vaccum_\$WObj := New_Worldsteelplate;

MoveL [[171.152,694.890,-66.818],[0.001473,0.999999,-0.000177,0.000355],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v40,fine,Vaccum_\$WObj := New_Worldsteelplate;

MoveL [[171.049,405.000,-65.962],[0.001473,0.999999,-0.000177,0.000358],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v40,fine,Vaccum_\$WObj := New_Worldsteelplate;

ReleaseBox;

MoveL [[171.399,403.517,-568.079],[0.001475,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v40,fine,Vaccum_\$WObj := New_Worldsteelplate;

MoveJ [[1254.685,1845.504,-1472.419],[0.001472,0.999999,-0.000178,0.000348],[0,0,-1,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate;

ENDPROC

PROC GrabBox()

Set DO10_3;
Reset DO10_5;
WaitTime 1;

ENDPROC

PROC ReleaseBox()

Set DO10_5;
Reset DO10_3;
WaitTime 1;

ENDPROC

ENDMODULE

C.1.2 Car body assembly program

MODULE CarBody

PERS tooldata Vaccum_:=[TRUE,[[51.344,65.032,205.200],[0.000000,0.617379,0.786666,0.000000]],
[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[-1571.133,-2838.977,-303.026],[0.999999,
0.001171, 0.000348, 0.000178]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PERS wobjdata New_Worldsteelplate:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[125.1667,1901.237,-
33.13485],[0.00264684,-1,-1.074E-06,-1.089E-05]]];

PROC main()

MoveJ [[1254.685,1845.504,-1472.419],[0.001472,0.999999,-0.000178,0.000348],[0,0,-1,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

ReleaseBox;

MoveJ [[-176.428,517.305,-1469.495],[0.000795,0.706981,-0.707231,0.001292],[1,0,-1,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveJ [[-169.790,297.249,-580.226],[0.001476,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveL [[-169.999,298.135,-280.228],[0.001476,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

GrabBox;

MoveL [[-169.790,297.249,-580.226],[0.001475,0.999999,-0.000177,0.000362],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveJ [[177.695,394.001,-368.043],[0.001475,0.999999,-0.000177,0.000351],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

MoveL [[177.535,394.680,-138.036],[0.001474,0.999999,-0.000177,0.000355],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

ReleaseBox;

MoveL [[178.022,392.613,-838.015],[0.001471,0.999999,-0.000177,0.000365],[0,0,0,0],[9E+09,
9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Vaccum_\$WObj := New_Worldsteelplate$;

ENDPROC

```
PROC GrabBox()

Set DO10_3;
Reset DO10_5;
WaitTime 1;

ENDPROC

PROC ReleaseBox()

Set DO10_5;
Reset DO10_3;
WaitTime 1;

ENDPROC

ENDMODULE
```

C.1.3 Back seat assembly

```
MODULE seatonfloor

PERS tooldata Gripper_Screws:=[TRUE,[[-51.448,28.545,215.728],[0.693520,0.693520,-
0.137950,-0.137950]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],
0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[4741.57,-2619.33,-288.4],[0.706756,
-0.001231, 0.000539, 0.707456]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PERS wobjdata New_World2:=[FALSE,TRUE,"",[[0.548694,-925.7507,-40.06818],[0.00176765,
0.705955, 0.708253, -0.0002938]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[895.272,722.425,-931.974],[0.706714,0.707498,0.000945,0.000654],[-1,-2,-2,1],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws:=New_World2;

ReleaseBox;

MoveJ [[199.073,5.771,-932.480],[0.509736,0.510503,0.489505,0.489837],[-2,-2,-2,1],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\WObj := New_World2;

MoveJ [[138.786,203.135,-126.623],[0.707599,0.706614,0.000247,0.000651],[-1,-2,-3,1],[0.000,0.000
,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\WObj := New_World2;
```

MoveL [[138.828,203.024,-26.629],[0.707599,0.706614,0.000247,0.000651],[-1,-2,-3,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

GrabBox;

MoveL [[138.786,203.135,-126.677],[0.707599,0.706614,0.000247,0.000651],[-1,-2,-3,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveJ [[448.157,446.312,-235.484],[0.000243,0.000655,-0.707599,-0.706614],[-1,-2,3,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveJ [[448.227,446.123,-65.483],[0.000243,0.000655,-0.707599,-0.706614],[-1,-1,1,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveL [[448.239,446.090,-35.485],[0.000243,0.000655,-0.707599,-0.706614],[-1,-1,1,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

ReleaseBox;

MoveL [[448.227,446.123,-65.485],[0.000243,0.000655,-0.707599,-0.706614],[-1,-1,1,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveJ [[220.854,171.020,-289.373],[0.500522,0.500111,-0.500173,-0.499192],[-1,2,0,1],[6.000,-6.000, 9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

ENDPROC

PROC GrabBox()

Set DO10_3;
Set DO10_6;
Reset DO10_4;
Reset DO10_5;
WaitTime 1;

ENDPROC

PROC ReleaseBox()

Set DO10_4;
Set DO10_5;
Reset DO10_6;
Reset DO10_3;
WaitTime 1;

ENDPROC

ENDMODULE

C.1.4 Front seat assembly

MODULE frontseat

PERS tooldata Gripper_Screws:=[TRUE,[[-51.448,28.545,215.728],[0.693520,0.693520,-0.137950,-0.137950]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[4741.57,-2619.33,-288.4],[0.706756,-0.001231, 0.000539, 0.707456]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PERS wobjdata New_World2:=[FALSE,TRUE,"",[[0.548694,-925.7507,-40.06818],[0.00176765,0.705955, 0.708253, -0.0002938]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[882.162,603.671,-932.100],[0.014472,0.014780,0.706566,0.707345],[-1,0,-2,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

ReleaseBox;

MoveJ [[221.348,21.005,-189.394],[0.500523,0.500111,-0.500173,-0.499192],[-1,0,-2,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveL [[221.416,20.822,-24.412],[0.500523,0.500111,-0.500173,-0.499192],[-1,0,-2,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

GrabBox;

MoveL [[221.333,21.044,-224.391],[0.500523,0.500111,-0.500173,-0.499192],[-1,0,-2,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveJ [[151.773,452.353,-267.541],[0.153394,0.153575,-0.690772,-0.689724],[-1,0,-3,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveL [[151.846,452.159,-92.540],[0.153394,0.153575,-0.690772,-0.689724],[-1,0,-3,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

MoveJ [[151.846,452.159,-92.540],[0.000243,0.000654,-0.707599,-0.706614],[-1,0,-3,0],[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

ReleaseBox;

MoveL [[151.773,452.353,-267.541],[0.000243,0.000654,-0.707599,-0.706614],[-1,0,-3,0],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Gripper_Screws\$WObj := New_World2$;

ENDPROC

PROC GrabBox()

Set DO10_3;
Set DO10_6;
Reset DO10_4;
Reset DO10_5;
WaitTime 1;

ENDPROC

PROC ReleaseBox()

Set DO10_4;
Set DO10_5;
Reset DO10_6;
Reset DO10_3;
WaitTime 1;

ENDPROC

ENDMODULE

C.1.5 Four point program for IRB 1600

MODULE newprogramcorner

PERS tooldata Calibration_tool:=[TRUE,[[-0.130,-1.337,164.860],[1.000000,0.000000,0.000000,
0.000000]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[4741.57,-2619.33,-288.4],[0.706756,
-0.001231, 0.000539, 0.707456]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PERS wobjdata New_World2:=[FALSE,TRUE,"",[[0.548694,-925.7507,-40.06818],[0.00176765,
0.705955, 0.708253, -0.0002938]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[10,-10,-250],[0.0976462,0.130398,-0.195047,-0.96717],[-2,1,-4,0],[0.000,0.000,

9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World2$;

MoveL [[410,-10,-250],[0.0976462,0.130398,-0.195047,-0.96717],[-2,1,-4,0],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World2$;

MoveL [[410,390,-250],[0.0976462,0.130398,-0.195047,-0.96717],[-2,1,-4,0],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World2$;

MoveL [[10,390,-250],[0.0976462,0.130398,-0.195047,-0.96717],[-2,1,-4,0],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World2$;

MoveL [[10,-10,-250],[0.0976462,0.130398,-0.195047,-0.96717],[-2,1,-4,0],[0.000,0.000,
9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World2$;

ENDPROC

ENDMODULE

C.1.6 Four point program for 4600

MODULE newpointtopoint_new_corner

PERS tooldata Calibrated_tool:=[TRUE,[[-0.983,2.988,256.369],[1.000000,0.000000,
0.000000,0.000000]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000, 0.000000,0.000000],
0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[-1571.133,-2838.977,-303.026],[0.999999,
0.001171, 0.000348, 0.000178]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PERS wobjdata New_Worldsteelplate:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[125.1667,1901.237,-
33.13485],[0.00264684,-1,-1.074E-06,-1.089E-05]]];

PROC main()

MoveJ [[10,-10,-250],[0.45876,0.434456,-0.260236,0.730112],[0,-2,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, Calibrated_tool\$WObj :=
$New\_Worldsteelplate$;

MoveL [[410,-10,-250],[0.45876,0.434456,-0.260236,0.730112],[0,-2,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, Calibrated_tool\$WObj :=
$New\_Worldsteelplate$;

MoveL [[410,390,-250],[0.45876,0.434456,-0.260236,0.730112],[0,-2,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, Calibrated_tool\$WObj :=
$New\_Worldsteelplate$;

MoveL [[10,390,-250],[0.45876,0.434456,-0.260236,0.730112],[0,-2,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, Calibrated_tool\$WObj$:= $New\_Worldsteelplate$;

MoveL [[10,-10,-250],[0.45876,0.434456,-0.260236,0.730112],[0,-2,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, Calibrated_tool\$WObj$:= $New\_Worldsteelplate$;

ENDPROC

ENDMODULE

C.1.7 Point to point program for IRB 1600

MODULE newnewnew

PERS tooldata Calibration_tool:=[TRUE,[[-0.130,-1.337,164.860],[1.000000,0.000000,
0.000000,0.000000]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],
0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[4741.57,-2619.33,-288.4],[0.706756,
-0.001231, 0.000539, 0.707456]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[1948.445,4041.461,659.474],[0.183013,0.683013,0.683013,-0.183013],[-1,0,-2,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj$:= New_World;

MoveL [[1948.445,4541.461,659.474],[0.183013,0.683013,0.683013,-0.183013],[-1,0,-2,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj$:= New_World;

MoveL [[2448.445,4041.461,659.474],[0.183013,0.683013,0.683013,-0.183013],[-1,1,-2,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj$:= New_World;

MoveL [[1948.445,4041.461,1159.474],[0.183013,0.683013,0.683013,-0.183013],[-1,0,-1,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj$:= New_World;

ENDPROC

ENDMODULE

C.1.8 Point to point program for IRB 4600

MODULE pppppppppp

PERS tooldata Calibrated_tool:=[TRUE,[[-0.983,2.988,256.369],[1.000000,0.000000, 0.000000,0.000000]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000], 0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[-1571.19,-2841.731,-306.922],[0.999999, 0.001171, 0.000348, 0.000178]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[1948.443,4041.457,659.466],[0.183011,-0.683014,-0.683013,-0.183010],[0,0,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

MoveL [[1948.445,4541.461,659.474],[0.183013,-0.683013,-0.683013,-0.183013],[0,0,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

MoveL [[2448.445,4041.461,659.474],[0.183013,-0.683013,-0.683013,-0.183013],[0,0,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

MoveL [[1948.445,4041.461,1159.474],[0.183013,-0.683013,-0.683013,-0.183013],[0,0,1,0], [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

ENDPROC

ENDMODULE

C.1.9 Point to point program for IRB 1600 after updating the offset (no change)

MODULE newnewnew

PERS tooldata Calibration_tool:=[TRUE,[[-0.130,-1.337,164.860],[1.000000,0.000000,0.000000, 0.000000]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[4741.57,-2619.33,-288.4],[0.706756, -0.001231, 0.000539, 0.707456]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[1948.445,4041.461,659.474],[0.183013,0.683013,0.683013,-0.183013],[-1,0,-2,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World$;

MoveL [[1948.445,4541.461,659.474],[0.183013,0.683013,0.683013,-0.183013],[-1,0,-2,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World$;

MoveL [[2448.445,4041.461,659.474],[0.183013,0.683013,0.683013,-0.183013],[-1,1,-2,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World$;

MoveL [[1948.445,4041.461,1159.474],[0.183013,0.683013,0.683013,-0.183013],[-1,0,-1,0],
[0.000,0.000,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibration_tool\$WObj := New_World$;

ENDPROC

ENDMODULE

C.1.10 Point to point program for IRB 4600 after updating the offset (change in work object data)

MODULE ppppppppp

PERS tooldata Calibrated_tool:=[TRUE,[[-0.983,2.988,256.369],[1.000000,0.000000,
0.000000,0.000000]],[2.000,[0.000,0.000,50.000],[1.000000,0.000000,0.000000,0.000000],
0.000,0.000,0.000]];

PERS wobjdata New_World:=[FALSE,TRUE,"",[[-1571.133,-2838.977,-303.026],[0.999999,
0.001171, 0.000348, 0.000178]],[[0.000,0.000,0.000],[1.000000,0.000000,0.000000,0.000000]]];

PROC main()

MoveJ [[1948.443,4041.457,659.466],[0.183011,-0.683014,-0.683013,-0.183010],[0,0,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

MoveL [[1948.445,4541.461,659.474],[0.183013,-0.683013,-0.683013,-0.183013],[0,0,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

MoveL [[2448.445,4041.461,659.474],[0.183013,-0.683013,-0.683013,-0.183013],[0,0,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj := New_World$;

MoveL [[1948.445,4041.461,1159.474],[0.183013,-0.683013,-0.683013,-0.183013],[0,0,1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,Calibrated_tool\$WObj :=

New_World;

ENDPROC

ENDMODULE

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY