

Savan Anghan

**NTNU**  
Norwegian University of  
Science and Technology  
Faculty of Engineering  
Department of Ocean Operations and Civil Engineering

Savan Anghan

# Digital Twin in Automated Manufacturing – MANULAB Case Study

December 2021





Norwegian University of  
Science and Technology

# Digital Twin in Automated Manufacturing – MANULAB Case Study

**Savan Anghan**

Product and System Design  
Submission date: December 2021  
Supervisor: Ola Jon Mork  
Co-supervisor: Robert Skulstad

Norwegian University of Science and Technology  
Department of Ocean Operations and Civil Engineering



***“SCIENCE IS NOT SUPPOSED TO BE  
TRUSTED,  
SCIENCE IS SUPPOSED TO BE  
TESTED”***

***- UNKNOWN***

# Acknowledgement

I would like to thank Professor Ola Jon Mork of the Department of Ocean Operations and Civil Engineering at NTNU Ålesund for supervising me and giving me freedom to work on my own and giving me access to use MANULAB. I would also like to thank Henrique Gasper for suggesting this topic to me.

Along with them I would like to thank Robert Skulstad, Markus Iltad, Mohmad Nikku, Rahulnath, Malin Sæbø Nes for helping me at the time of need and motivating me to finish my thesis.

Last but not least, I would like to thank my family and all my friends who were part of this journey by just being there and being them and believing in me throughout my masters journey. This accomplishment would not be possible without all this kind and wonderful people. THANK YOU.

# Summery

The Project explores the term Digital Twin in Automated Manufacturing. A digital twin can be described as a near real time digital image of a physical object or process. The goal of this project was to study different possibilities to create digital twin for automated manufacturing process at MANULAB at NTNU Ålesund.

RoboDK Simulation software was used to create three different level of integrated digital twin of automated manufacturing process. First level of integration was defined when the transfer of data was manual from digital system to physical system. The second level of integrated digital twin was defined when flow of data was digital from digital system to physical system. And third level of integration was defined when flow of data was digital from physical system to digital system.

Most of the work done in this project was to set up this different level of integrated digital twins at manulab.

It was concluded that the implementation of digital twin in automated manufacturing was successfully studied and in-dept knowledge of development of different level of integration of digital twin at MANULAB was obtained. In the end, areas which had scope for future development were suggested.

# Contents

1	Introduction .....	1
1.1	Motivation and Scope.....	2
1.2	Objective and Research Question .....	2
1.2.1	Objective .....	2
1.2.2	Research Question .....	3
1.3	Structure of the Thesis.....	3
2	Literature review.....	4
2.1	Digital Twin .....	4
2.2	Automation.....	9
2.3	ModBus.....	11
2.4	Digital twin technology in automated manufacturing .....	13
2.5	Robot Kinematics .....	14
2.6	MANULAB at NTNU Ålesund .....	16
3	Method And Solution .....	19
3.1	Methodology: - .....	19
3.2	Case Study: - .....	21
4	Analysis And Discussion .....	36
5	Conclusion and future work .....	37
6	Reference .....	38
7	Appendix .....	40
7.1	Apendix A.....	40
7.2	Apendix B.....	43

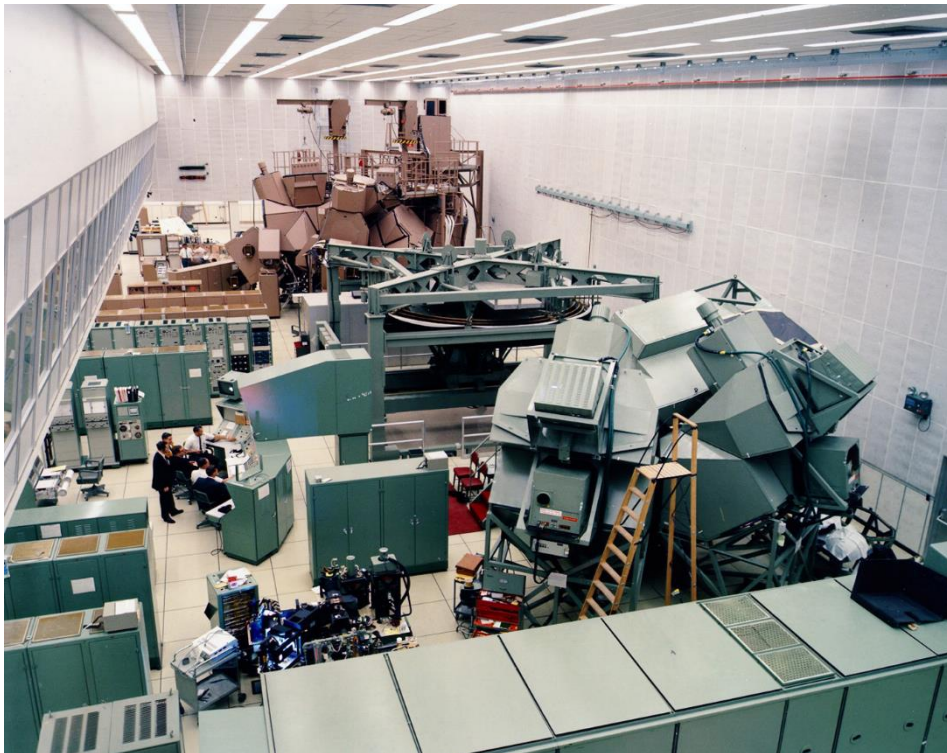


# List Of Figures

Figure 1 Physical Twin .....	1
Figure 2 Scope .....	2
Figure 3 Level of Integration .....	5
Figure 4 Modified Level of Integration .....	6
Figure 5 Input & Output in DT .....	7
Figure 6 Components of Automated robot .....	9
Figure 7 Human Automation Comparison .....	9
Figure 8 OMRON Collaborative Robot .....	11
Figure 9 Master Slave Protocol .....	11
Figure 10 Modbus TCP port on control box .....	12
Figure 11 Software comparison .....	13
Figure 12 Robot Manipulator Schematic .....	14
Figure 13 Direct and Inverse kinematics .....	15
Figure 14 MANULAB .....	16
Figure 15 Colaborative robot and Mobile robot .....	17
Figure 16 3D printers and VIPER .....	17
Figure 17 QUATTRO .....	18
Figure 18 Funnel Methodology .....	19
Figure 19 Research Flowchart .....	20
Figure 20 Assembly .....	21
Figure 21 Gripper Positioning .....	22
Figure 22 Assembly parts on table .....	23
Figure 23 Program Formation .....	25
Figure 24 Main Assembly Program .....	26
Figure 25 Post processor selection .....	27
Figure 26 Content of the zip file .....	28
Figure 27 Listen node in TM Flow .....	28
Figure 28 RoboDK live connection .....	29
Figure 29 Normal assembly project .....	30
Figure 30 Modbus Slave Setting .....	31
Figure 31 Code table example .....	31
Figure 32 Adding new Modbus Dev Setting .....	32
Figure 33 Variable creation in TM Flow .....	32
Figure 34 Log Thread along with Set Node setting .....	33
Figure 35 Log Node Setting .....	34
Figure 36 logged Data .....	34
Figure 37 Python API in RoboDK .....	35

# 1 INTRODUCTION

In current time, there is exponential growth of digitalization in all sector & manufacturing is one of them. INDUSTRI 4.0 is the name given to this digitalization in manufacturing sector. After steam engine, assembly line, automation, it is regarded as the fourth industrial revolution. (Haag 2018). Digital Twin is one part of this vast Indutri 4.0 revolution. In simple terms, a digital twin is near real time digital image of a physical object/process. Although digital twin was created in 2002 by Michael Grieves, but we can say that it has predecessor since 1960s by NASA which had similar concept, i.e., creating a twin of the product which they used in their space programming. They created a physical twin of APOLLO 13 at ground level and helped the astronauts of Apollo 13 which was up to 200,000 miles away by simulating the similar condition on the twin.

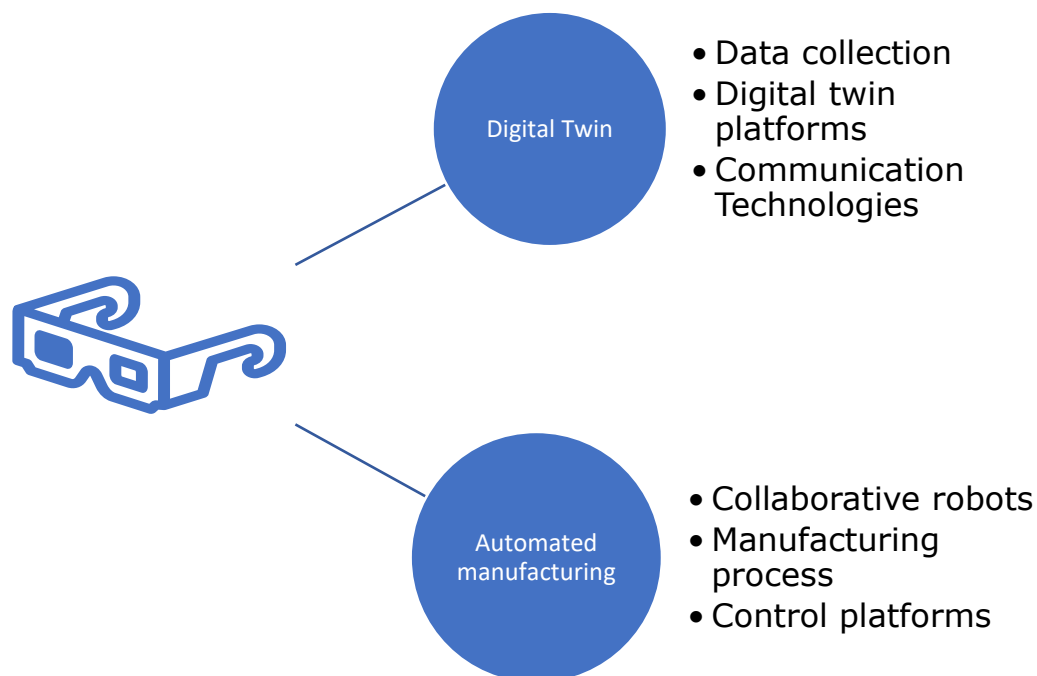


*Figure 1 Physical Twin*

<https://blogs.sw.siemens.com/simcenter/apollo-13-the-first-digital-twin/>

## 1.1 Motivation and Scope

To prove the promising concept of Digital Twin in the field of Automated Manufacturing. A Digital twin of MANULAB at NTNU Ålesund, which is a lab equipped with different automated manufacturing elements which helps making different manufacturing processes automated or semi-automated will be created. And to test this Digital twin of MANULAB, a case study which will take all elements automated manufacturing will be simulated.



*Figure 2 Scope*

## 1.2 Objective and Research Question

### 1.2.1 Objective

To have an overview of a digital twin system and to create a digital twin of an automated manufacturing process at MANULAB, which will include all the key elements of Digital Elements and Automated manufacturing available at the lab.

## 1.2.2 Research Question

- What are the different Digital Twin Technologies available for Automated Manufacturing?
- How to incorporate different digital twin technologies in the procedure at MANULAB?
- What are the steps to advance initial analysis?

## 1.3 Structure of the Thesis

Chapter 2: -

In chapter 2, relevant literature review of topics related to the content of this project has been carried out. It has covered all the information required for completion of this project

Chapter 3: -

In chapter 3, the methodology approach of the project is explained along with the research flow chart. Along with this it also describes the entire procedure to develop different integration of digital twin by using a case study.

Chapter 4: -

In chapter 4, analysis of the work carried out in this project has been done along with discussing on the case study for different level of integration in digital twin.

Chapter 5: -

In chapter 5, from experience from performing the case study, the work of the project is concluded while giving some recommendation for the future work is given.

## 2 LITERATURE REVIEW

### 2.1 Digital Twin

There are several different ways to define Digital Twin, one of the most recognized is from Glassgen, Stargel(2014) : the Digital twin is integrated multi-physics, multi-scale, probabilistic simulation of a complex product and uses the best available physical model and sensor to mirror the life of its corresponding twin.

It is also worth noting that most of the definition of Digital Twin is focused more on the product aspect whereas even the process aspect is a wide part of Digital twin.

A digital twin is different from the traditional computer aided design (CAD) and also different from sensor-enabled Internet of things(IOT) solutions. It is the combination of both the fields with a different level of complexity in each field for different product and process.

When Digital Twin is used to simulate specific complex products such as jet engines, wind farms, ROVs etc, it could give insight to its operations inefficiencies which could affect the future asset designs. But Digital Twin with respect to the production process, three main areas where it offers to simulate and optimize the process are (Kritzinger 2014)

- Production planning and control
- Maintenance
- Layout planning

A Digital Twin works in an Physical-digital-Physical loop. With the help of different elements of a digital Twin like sensors, Data, Integration, Analytics digital twin and actuators.

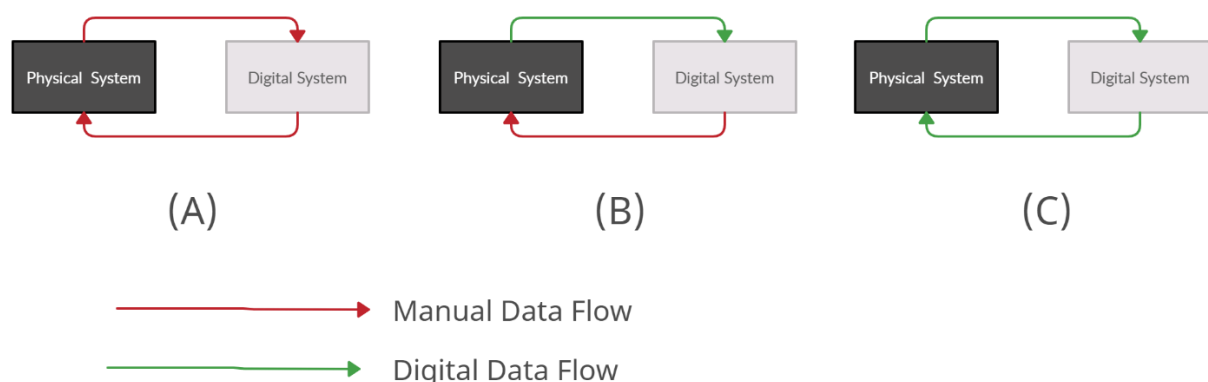
- Sensors : Sensors are distributed throughout the product or process that helps to take the operational and environmental data from physical process or product in real world for digital twin.
- Data: Data collected from real world using sensors and other available data such as engineering drawing, bill of material, etc are

stored so that it can be used in Digital twin. A data base is formed out of this collected data for the future use.

- Integration: The technology used for the communication of the data from the data base of the physical world to digital world and vice versa is carried out with the help of different integration technologies which uses python or C++ or some other programming language.
- Digital Twin: An software/ application has to be used to create digital model of product/process which can interact with available data and give insight result and plots.
- Analytics: Analytics techniques are used to analyse the data through some sort of algorithmic simulation to produce insight.
- Actuators: To produce a action from digital world to physical world, actuators are used. Digital twin send signal to respective actuators in real world, which produce the required action.

With the help of sensor, data is collected from real world production process, and along with other available data, it is communicated and aggregated to digital world i.e., in Digital Twin. In Digital twin platform, analysis of this data takes place with the help of different analytical techniques which is different for different use of software, to give insight to the particular case. Thus, physical-digital-physical loop is completed.

We can get different level of integration in digital twins based on how data flows or transferred.



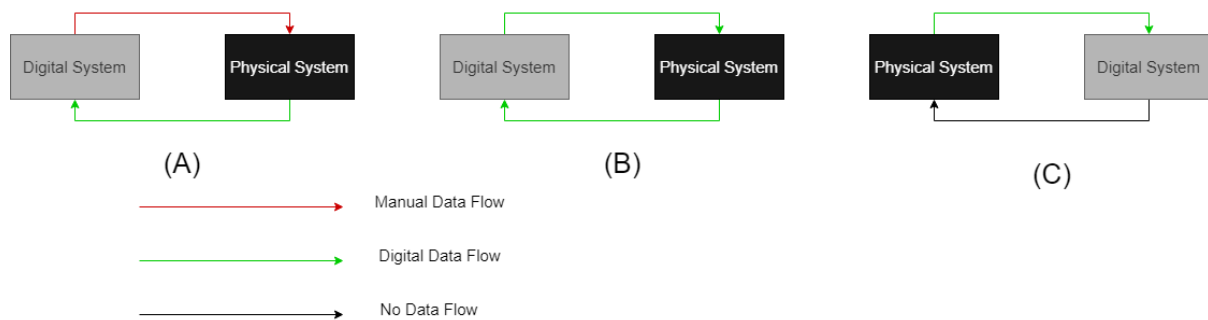
*Figure 3 Level of Integration*

In case (a), all data exchange is done manually. Thus, sudden change in the state of physical product/process will not have any direct effect on digital twin and vice-versa.

In case (b), the flow of data from physical object to digital object is automatic but from digital to physical is manual. Thus, change in state of physical object leads to change in digital object but not vice-versa.

In case (c), the flow of data is automatic both ways between physical and digital object. Thus, a change of state of object in physical and digital space will lead to change in other state as well.

In our case, as we will use the simulation software with library of all major component, i.e., pre stored data, we will modify this digital twin integration with respect to our requirement.



*Figure 4 Modified Level of Integration*

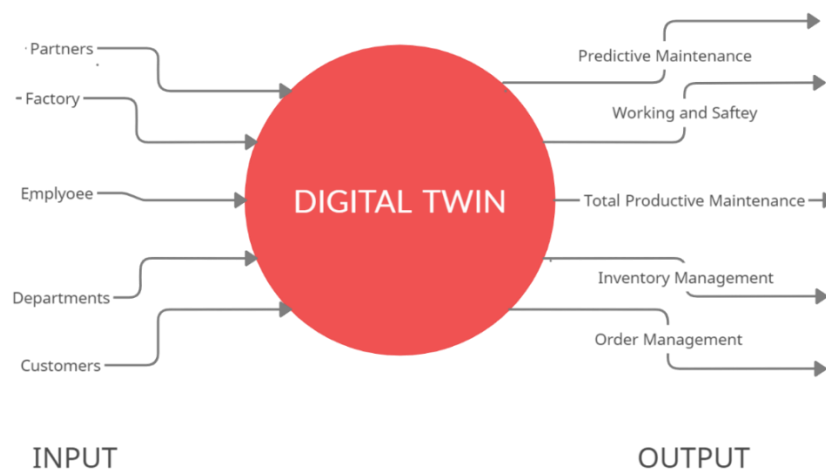
In case (a), the flow of data from physical system to digital system is digital, i.e., there is already predefined library that can be used, but flow of data from digital system to physical system is manual.

In case (b), similar to case (a), there is pre-defined library made from the data from physical world that can be used, but flow of data from digital system to physical system is digital as well.

In case (c), the flow of data is digital from physical system to digital system, but there is no flow of data from digital system to physical system i.e., digital system cannot do any changes in the physical system.

## Creating a Digital twin:

The role of the digital twin is to map the main features of the physical object/process enabling simulation, prediction and optimization in the area of interest. And thus, to map the main features of the product/process, it is important that digital twin only collects the data that is relevant for the case of interest rather than all available data from the physical system. Thus, we can also conclude that different digital twin of the same product/process can be made according to the way it is being used. So, it is important to know application, viewpoint, and context of the digital twin before starting to create one. Along with focusing on kind of information required across the life cycle of process/product, it is also important to structure this information such that it is reusable.



*Figure 5 Input & Output in DT*

The steps to create a digital twin can be done using following steps

- Create
- Communicate
- Aggregate
- Analyze

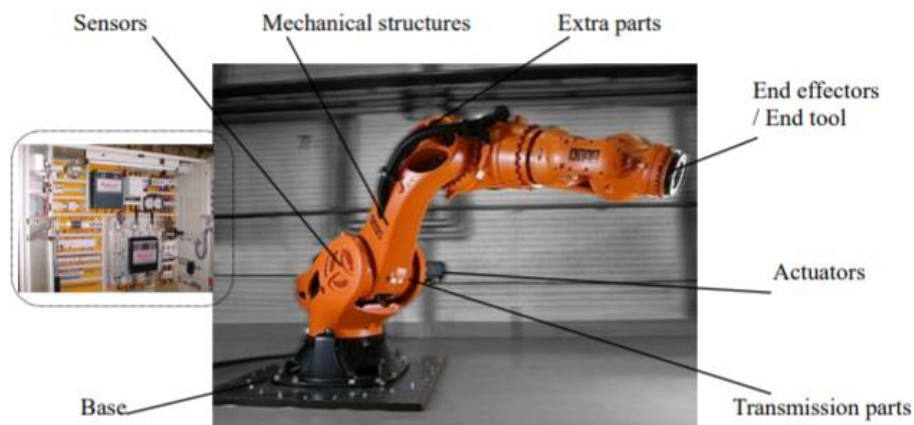


- Create: - The creation of product/process by use of data and sensors that measures critical input from the physical environment. The measurement of data can be classified into two types.
  - (1) Measurement of product/process assets- Measurement of operational data of product/process in the physical performance of the asset.
  - (2) Environmental or external data- Measurement of external data that affects the operation of the physical asset.
- Communication: - Communication step is the link between the physical world and the digital world. Based on level of integration as discussed before there can be three type of data transformation scenarios.
- Aggregate: - In order to analyze the product/process, aggregation of all communicated data is to be done using one of the available techniques
- Analyze: - After aggregation of all the data, the data is analyzed and visualized. To get the optimal results/output, iterative models can be developed by slightly manipulating the data.

This will help companies answer key strategical questions about their product/process. It will also help them answer many question in real time that could not be answered before. It will also help the companies to create and use more scenarios with high value than before.

## 2.2 Automation

Industrial automation is the use of control systems, such as computers or robots, and information technologies for handling different processes and machineries in an industry to replace a human being. According to Houxiang Zhang, it can be classified using following components



*Figure 6 Components of Automated robot*

As we are entering an era of smart factories and industrial automation, automated manufacturing has unleashed new ways that machines and human effort can collaborate to increase efficiency and drive down complexity. This is especially true for the engineering automation technologies which in the past has disrupted all kinds of traditional manufacturing process. Robotics and automation when implemented correctly, can reduce the time required to process an average production through the factory, leading to lower production lead time and bottlenecks.

The comparison between human and machine(automation) can be seen from below table.

HUMANS SURPASS MACHINES IN THE:	MACHINES SURPASS HUMANS IN THE:
<ul style="list-style-type: none"> <li>▪ Ability to detecting small amounts of visual or acoustic energy</li> <li>▪ Ability to perceiving patterns of light or sound</li> <li>▪ Ability to improvise and use flexible procedures</li> <li>▪ Ability to store very large amounts of information for long periods and to recall relevant facts at the appropriate time</li> <li>▪ Ability to reason inductively</li> <li>▪ Ability to exercise judgment</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ability to respond quickly to control signals, and to apply great force smoothly and precisely</li> <li>▪ Ability to perform repetitive, routine tasks</li> <li>▪ Ability to store information briefly and then to erase it completely</li> <li>▪ Ability to reason deductively, including computational ability</li> <li>▪ Ability to handle highly complex operations, i.e., to do many different things at once.</li> </ul>

*Figure 7 Human Automation Comparison*

To get the best of both the worlds, one common solution can be to integrate manual and automated operation to convert the process into semi-automated manufacturing system. Choosing the right level of automation out of following five levels of automation is one of the key decisions.

In the first level, everything is done manually. The operator loads the machine and starts the machine, and the machine cycles. Next the operator unloads the part and manually transfers it to the next production step. An example of this machine arrangement would be a manual press in which the operator loaded the press, pressed the part, unloaded the press, and took the part to the next station. The second level of automation is when the operator manually loads the machine, the machine automatically cycles, and the operator manually removes the part and takes it to the next station. In level three, the operator manually loads the part into the machine and the part automatically cycles. The part is automatically unloaded from the machine and the operator then moves the part. In level four automatically the part is loaded; it is automatically cycled, automatically unloaded, and then manually transferred to the next process. Finally, level five is entirely automatic. The machine is automatically loaded, cycled, and unloaded, and the part is transferred by automation (Harris and Harris, 2008).

In today's time where there is increased global competition and rapid changes in the market, third level of automation is beneficial for majority of the industries for which use of collaborative robots is most optimal. By working towards the same goal, collaborative robots give strong collaboration between humans and robots working in the same space (Ore 2015). In most applications, a collaborative robot is responsible for repetitive, menial tasks while a human worker completes more complex and thought intensive tasks. The accuracy, uptime, and repeatability of collaborative robots are designed to complement the intelligence and problem-solving skills of a human worker. Collaborative robots are also good to work by themselves, without any human interactions. Some of the major providers of collaborate robots are:

- Universal Robots
- FANUC
- TechMan
- Rethink Robotics
- AUBO
- ABB
- Kawasaki
- KUKA



Figure 8 OMRON Collaborative Robot

## 2.3 ModBus

ModBus is a method used for transmitting information over serial lines between electronic devices. Modbus is a communication protocol developed by Modicon systems in 1979. Modbus is an open protocol, thus it has become a very common protocol used widely by many manufacturers throughout many industries but commonly used in process and automation industry. Modbus uses serial protocol, i.e. it is a master/slave protocol, e.g. one master that controls the Modbus data transactions with multiple slaves that respond to the master's requests to read from or write data to the slaves.

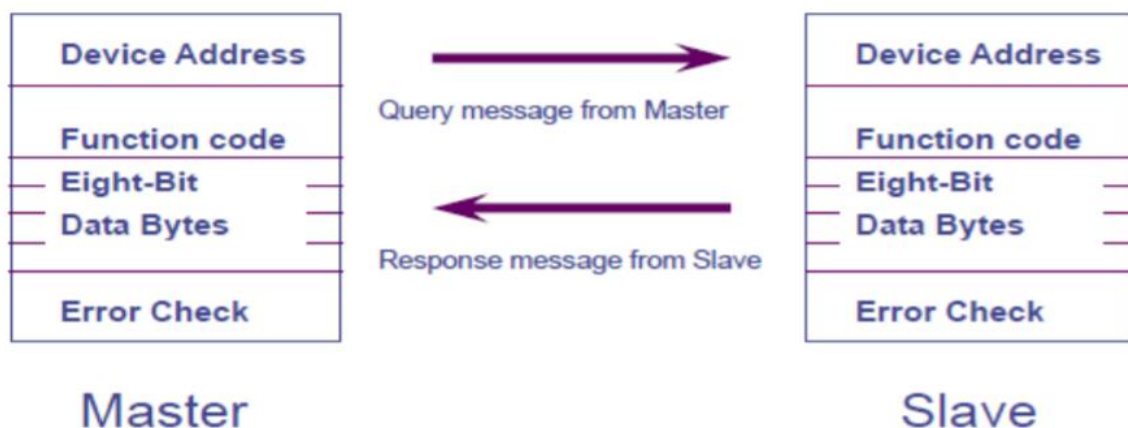


Figure 9 Master Slave Protocol

Omron uses Modbus for communication. Using modbus protocol in TM flow, users can use Modbus master to read or write the parameters and save them in the robot registers, such as positions, posture, temperature, IO

status etc which can be used to program the robot or monitor the robot. Omron provides two communication protocol Modbus TCP and Modbus RTU, the use of Modbus TCP will be done in this research. Modbus TCP uses RJ45 for communication and there is only one port available in the control box that can perform RJ45 to perform communication of Modbus TCP.



*Figure 10 Modbus TCP port on control box*

## 2.4 Digital twin technology in automated manufacturing

Based on different level of integration, different softwares are available

- Plant simulation (Siemens)
- Flextime
- Industrial ROS
- CoppeliaSIM
- AnyLogic
- RoboDK
- 3D-experience
- Tecnomatix Process
- Visual Components 4.0
- Simumatik 3D

The selection of the software depends upon area of use and level of integration needed in the digital twin technology. Comparison of some of this softwares is done by Al-Geddawy, T. (2020) which gives more insight into the application area of the softwares can be referred below

Features	3DEXPERIENCE	Simumatik 3D	Industrial ROS	CoppeliaSIM	RoboDK
<b>Cross platform</b>	Windows	Windows	Linux+RPI	All	All+RPI
<b>Opensource</b>	No	No	Yes	No	Yes
<b>Cost</b>	Very High	Free	Free	Low	Very Low
<b>Multiple programming languages</b>	Possible w difficulty	No	C++, python and MATLAB	Mainly C++	Integrated python, Visual studio, C++, MATLAB libraries
<b>Industrial Robots Focus</b>	Yes, with extensive robot library	Yes	Yes	No	Yes, with extensive robot library
<b>Physical world communication</b>	Possible w difficulty	OPC UA	Possible	No	OPC UA and python libraries
<b>Community support</b>	Not public	No	Large support	Small	Large support
<b>User-friendly</b>	Yes	Yes	No due to the need to establish nodes and other components first	No due to very small library of ready to use industrial robots	Yes due to large ready to use industrial robots
<b>Learning curve</b>	Slow due to learning multiple applications	Quick due to working in a single visual environment	Slow due to learning multiple components and working under Linux	Quick due to working in a single visual environment	Quick due to working in a single visual environment
<b>Building custom components</b>	Difficult	Difficult	Normal process	Normal process	Simple drag and drop

Figure 11 Software comparison

Based on the level of integration of this project and availability of the software, RoboDK will be used.

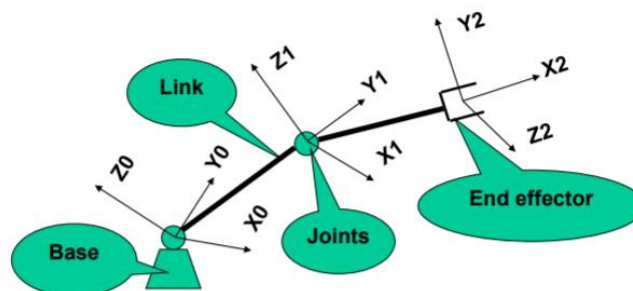
RoboDK:

RoboDK is software for simulation and offline Programming. Offline Programming means that robot programs can be created, simulated, and generated offline for a specific robot arm and robot controller. Robo DK is an open source which can be installed on the various platforms. RoboDK also has very large library of the industrial robots and grippers from different manufacturer along with options to import custom made CAD drawings. It has user friendly interface which makes it ideal choice for beginners. It also has features to plug in with all the major CAD softwares which gives ease on complex tasks. It also has a feature to be integrated with one of the universal programming language such as C++, Matlab and python to program any robots. Along with this, RoboDK also has the ability to communicate with the physical environment in real-time using OPC UA communication. RoboDK also has large community, along with all the online documentation and examples, which makes it quicker and easier to use.

## 2.5 Robot Kinematics

All the theory for robot kinematics are taken from the lecture notes of Houxiang Zhang from the course Mechatronics, robotics and deck machines.

A robot manipulator is an open-loop articulated chain with several rigid bodies connected in series. One end of the chain is attached to a supporting base while the other end is free which can be used to perform various tasks. Robot manipulator consist of base, links, joints and end effector. The schematic for the robot manipulator can be seen below



*Figure 12 Robot Manipulator Schematic*

The motion of the robot is done by motion of one link with respect to the previous one, which results in the unique pose of the robot. The analytical study of geometry and motion of the robot with respect to fixed reference co-ordinate system is robot kinematics. There are two type of robot kinematics, direct kinematics and inverse kinematics.

Direct kinematics gives the joint angle of the given robot. Direct kinematics is used to define the posture of the robot and find the position of end effector.

Inverse kinematics gives the position and orientation of the end effector with respect to the refence co-ordinate system. Inverse kinematics is used to see how many different robot configurations are possible and to see if its possible to reach the desired position.

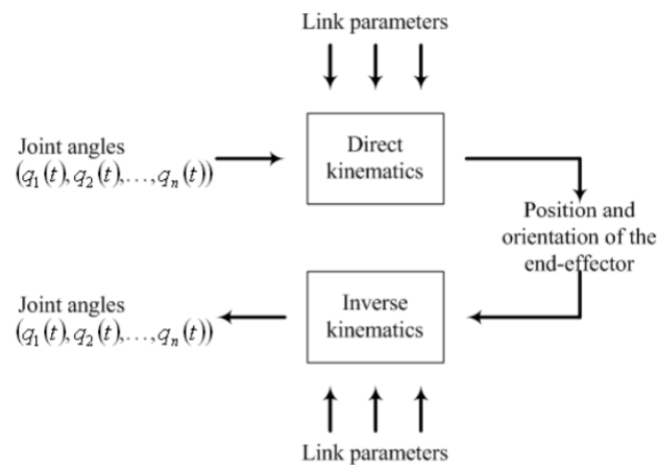


Figure 13 Direct and Inverse kinematics



## 2.6 MANULAB at NTNU Ålesund

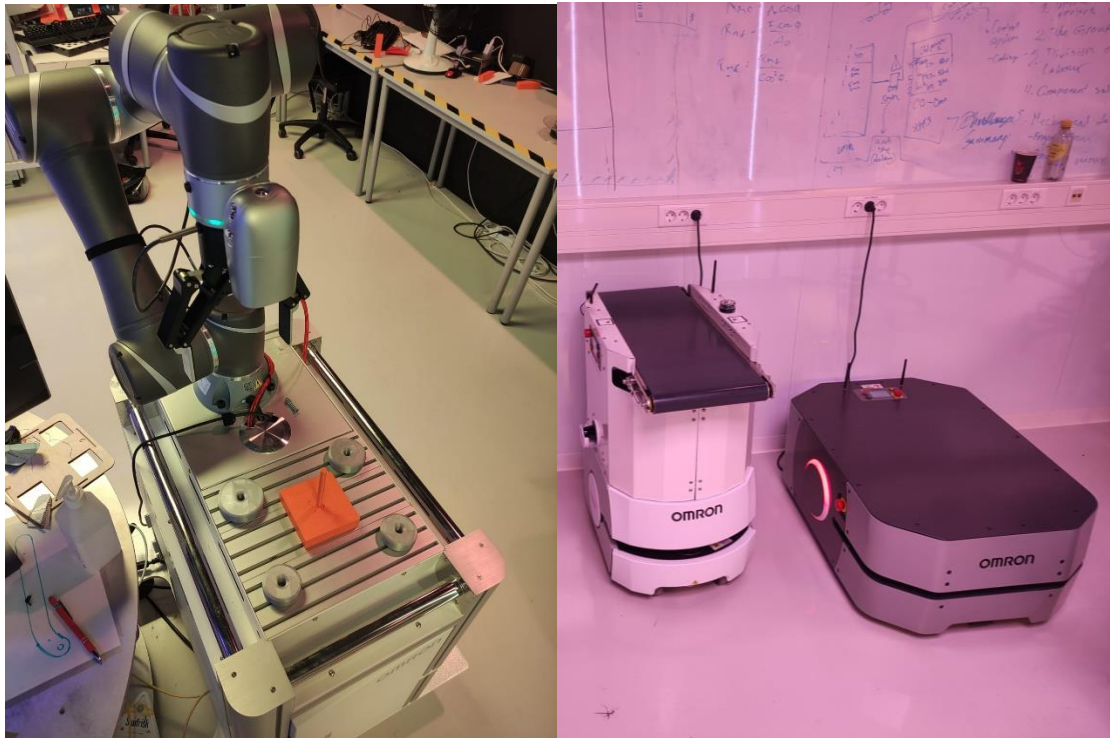
MANULAB is a infrastructure for basic research facility on manufacturing engineering. The equipment in the MANULAB is used for laser cutting, additive manufacturing, industry 4.0, learning factories and advance automation. MANULAB is a laboratory where students and companies can learn and discover new tools and technologies in manufacturing engineering.



*Figure 14 MANULAB*

The MANULAB at NTNU Ålesund consist of

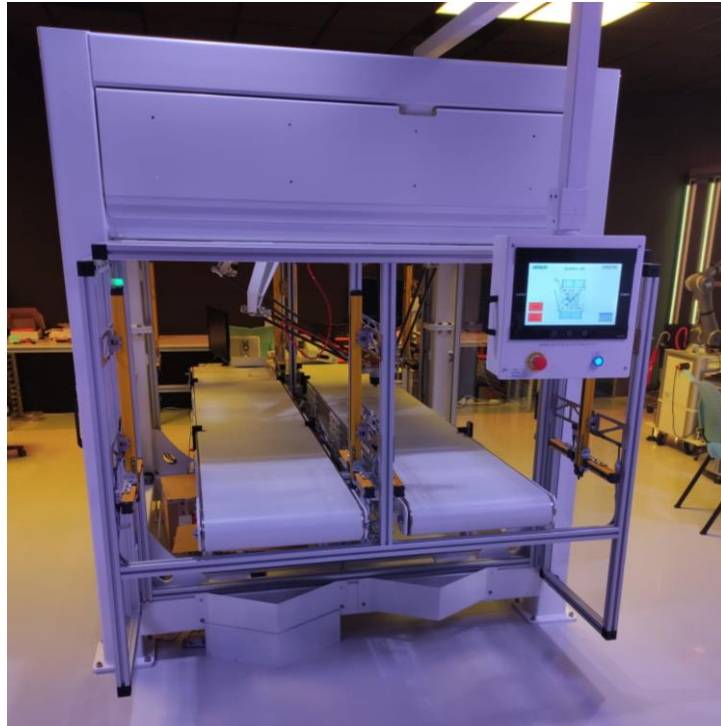
- Collaborative Robots
- Mobile Robots
- 3D Printers
- Viper Robot
- QUATTRO



*Figure 15 Colaborative robot and Mobile robot*



*Figure 16 3D printers and VIPER*



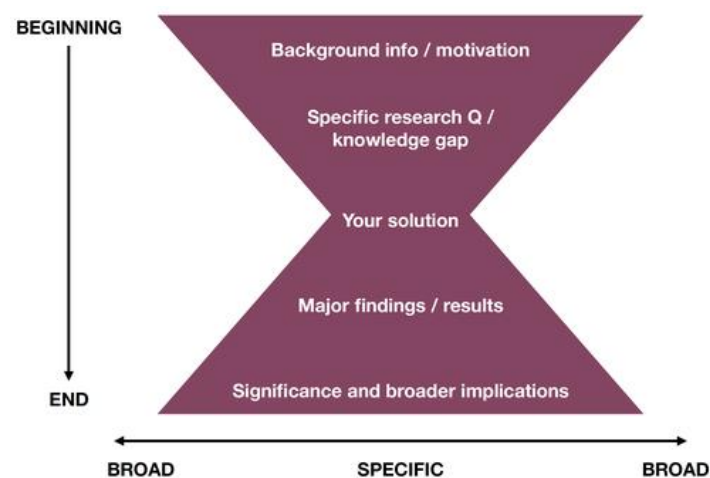
*Figure 17 QUATTRO*

### 3 METHOD AND SOLUTION

#### 3.1 Methodology: -

The methodology describes the broad philosophy of the research. This research was completed with the combination of both theoretical approach and experimental approach, where theoretical approach was of main concern in the beginning to funnel down the wide topic, and the experimental approach was centre of research method with occasionally shifting to theoretical method for brief every time somewhat of 'new-topic' was introduced or needed for experimental approach or more in-depth knowledge in a particular area was required.

The theoretical approach during different stages of the research consists of literature review from different papers, articles, journals, blogs, product manuals, software documentation, home pages of company websites etc. for all the topics used in the research.



*Figure 18 Funnel Methodology*

For the experimental approach, a case study methodology was used. "A case study is expected to capture the complexity of a single case, with a multitude of methods while simultaneously taking account of required context, variable and qualities. Along with this, the case study methodology also has the potential for further development through the combination of different level of techniques, methodologies, strategies, theories, and quality standards etc." Johansson, R (2007).

Figure below is the flowchart for the research methodology to visualize the steps from the beginning to the end of the thesis.

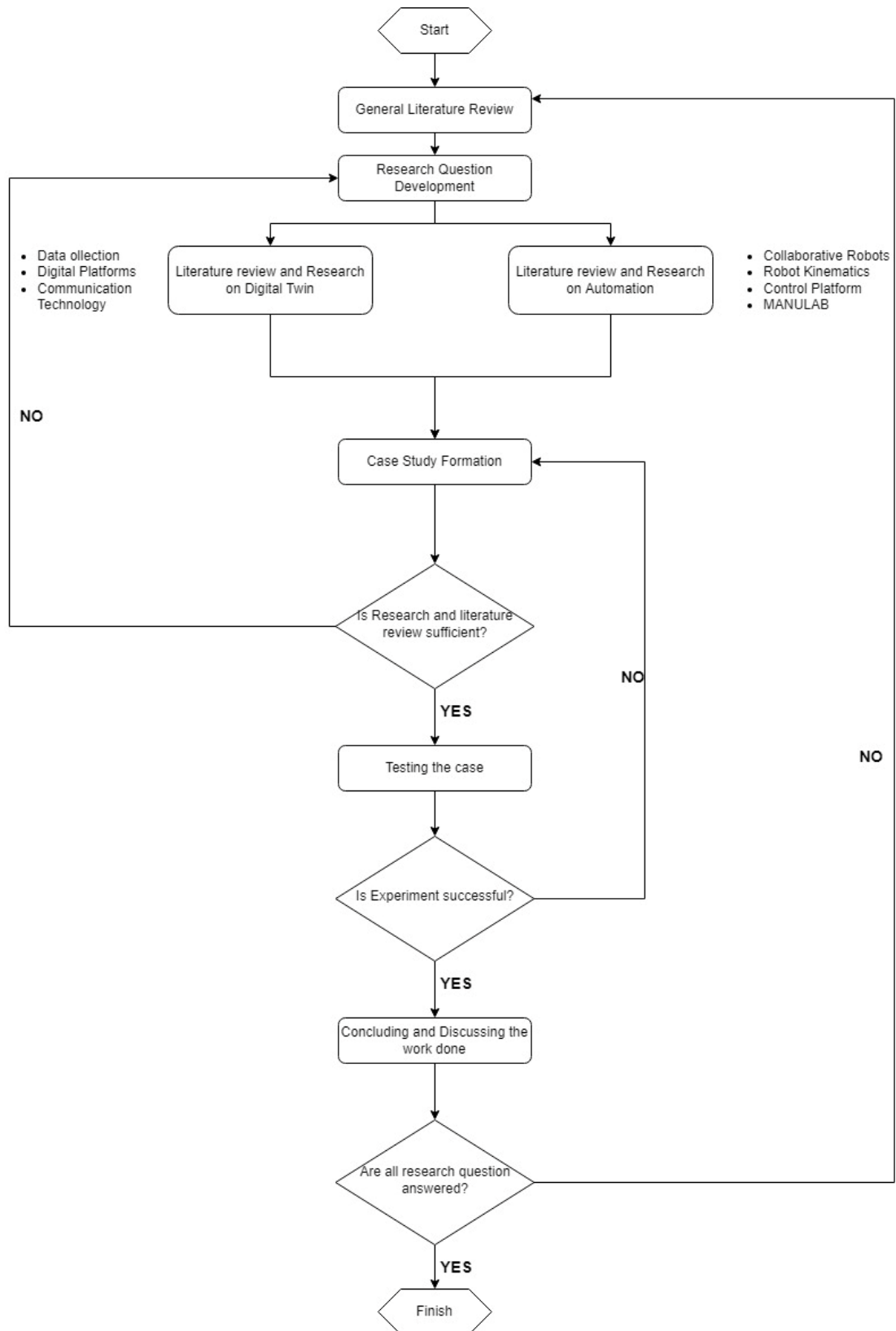


Figure 19 Research Flowchart

### 3.2 Case Study: -

The experimental approach of this research work is mainly focused on incorporation of digital twin technology in automated manufacturing at MANULAB. As we discussed earlier, RoboDK will be used to get different level of integration in digital twin of automated manufacturing at MANULAB.

As we know, MANULAB is equipped with multiple OMRON TM5 X series collaborative Robots from OMRON along with QUATTRO, VIPER, Mobile Robots and 3d-printers. In this case study, the process of communicating with single OMRON TM5-900 collaborative robot will be tried to establish using RoboDK to create a digital twin.

- Setting up the case study: -

For setting up the different level of integration in digital twin, standardization of the case study is of more importance than the complexity of the case study. For this purpose, a case study consist of assembly of donut shaped cylinders on top of each other was designed.

The dimensions of cylinders were as follows

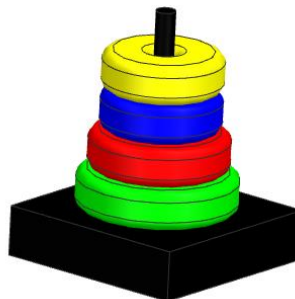
Cylinder 1 – 800mm

Cylinder 2 – 700mm

Cylinder 3 – 600mm

Cylinder 4 – 600mm

The assembly was in the order cylinder 4, cylinder 3, cylinder 2, cylinder 1 from top to bottom as seen below



*Figure 20 Assembly*

## First and second level of integration:

First and second level of integration of digital twin begins with creating a new station in RoboDK. The station in RoboDK contains all settings related to robots, tools, reference frames, targets, objects and other parameters including making a program along with making a robot program and robot script.

First step is to import the robot from offline or online library. If the robot in use is not available in offline library, then it could be downloaded from online library which contains robots from all the major robot providers as discussed before, which can be downloaded and loaded into the created new station, in our case we will download and import OMRON TM5-900. It is important to load the robot first onto the workstation so that the reference frame of the robot will become the base reference frame of the entire station. The next step is to import the tool, it can be both, a pre-defined tool from the offline or online libraries available in the market from different manufacturers of a customized tool whose 'stl' file is available with you. In our case its 2F adaptive gripper from Robotiq whose maximum distance between two arms is 85mm. When you load the tool, the tool will be added in robot base frame, to add the tool after last frame of the robot, in our case 6<sup>th</sup> joint, it is important to move the tool item inside the robot item in the station tree as seen below. This will move the tool along with the movement of robot. It is also important to check the orientation of the tool, if needed adjustment, it can be done using the control window of the tool. It has the ability to rotate in all three axis along with the possibility to have a translational movement in any of these axis in case of use of customized extension for the robot gripper.

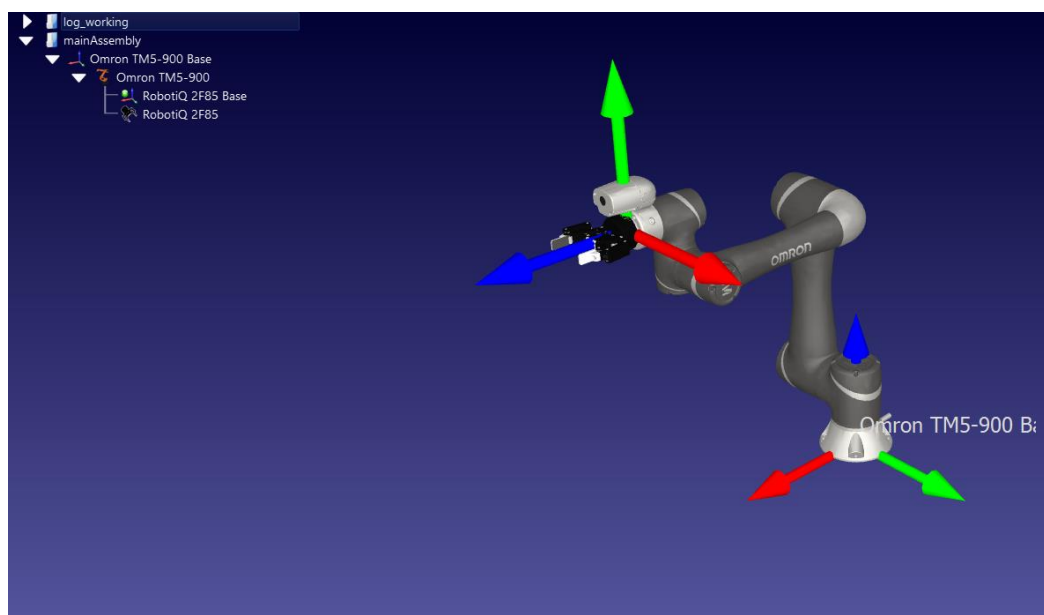


Figure 21 Gripper Positioning

Now the setting of robot and gripper is done, next step is to import a worktable, there are pre-designed tables available in the library, but as we are using a control box of OMRON TM5-900 as a table, we will create a CAD of its dimension and import it in our working station using a STL file. Before adding the station in to the working station it is important to add a new reference frame which can be used to orient the table perfectly w.r.t the OMRON TM5-900. Reference frame gives the benefit of quickly updating the translational and rotational orientations of the object inside it in all three axes. The reference frame can be renamed according to the use, in this case we will rename it table\_ref. we will move the table\_ref inside the reference frame of the robot and move the table inside table\_ref, so every time we move the position of robot, the table will move with it accordingly and orient itself.

Now, the robot with its control box as a table and gripper are set inside the station, the next step is to set up the assembly components. As discussed earlier, the assembly is standardized thus the position of each cylinder would be fixed for all three integrations, and it will be taken w.r.t to the base of the robot making it easier to set it up in real environment as well as digital environment. Similar to the setup of control box in Robodk, it would be same for all the assembly parts as well, we will have to import the STL file of all the assembly parts. Also, it become much more easier to orient them using reference frame for each individual cylinders, thus we created different reference frame with name donut\_1, donut\_2, donut\_3, donut\_4 and base for each of the different part of the assembly and orient the assembly position of each part as seen below.

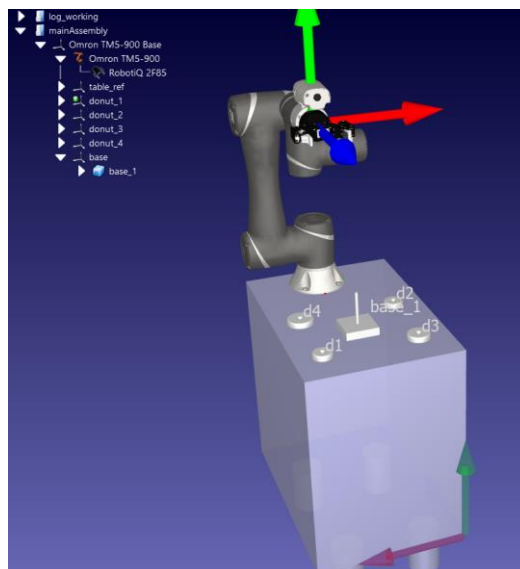


Figure 22 Assembly parts on table



The next step is to program the assembly process, but before starting the assembly process programming, we can add the program to save the relative position of all components in the workstation, which can be used to move all parts to their designated position after the finish of one cycle. This step is only for the digital world i.e., simulation. After this we can start with creating targets which will enable robotiq gripper to change its grip length, for this we will choose the reference frame of the gripper as the active frame and create new target. It is important to change the linked robot to Robotiq gripper from OMRON TM5-900 for the target which can be done by right clicking on the target and selecting Robotiq 2F85 in robot linked option. Then we can double click on the gripper and change its length to 80mm which is diameter of the first cylinder and can teach this position to the target. We can rename the target to donut1\_pick and will do the same for all the other cylinders and create donut2\_pick, donut3\_pick and donut4\_pick for grip length 70mm, 60mm and 50mm respectively. We will do the same thing for 85mm grip length which is largest length of the gripper to release all the cylinders on to the assembly base and call it place. What this targets will do is any time the any of this target is in command, it will change the length of the gripper to said length according to target in use irrespective of the position of the robot which will help us in picking and placing all the cylinders. After enabling the robotiq gripper to change it length, we can try positioning the robot to approach to each cylinders so the gripper can pick it and place it on assembly base. In order to do so we have to create the target in each reference frame of each cylinders. We can start with donut 1, for this we have to make reference frame of donut one active which can be done by right clicking the reference frame and selecting 'Active reference frame' option. It is better to go bit near the object then to directly go towards the object for picking it up to prevent the possibility to move the object or damage the object by the motion of the gripper. Thus a target would be created 200mm above the donut 1, this can be done by right clicking the donut 1 object file, selecting get with robot and choosing OMRON TM5-900, this will move the flange of the robot to reference frame of the donut 1, then the co-ordinate of z axis can be changed to 200mm to bring the robot to desired position. The target can be created at this position and be called D1, then we will move the gripper closer to the cylinder by changing the z co-ordinate to 160mm and create a new target which will position the robot to pick cylinder 1 and call this target D1\_pick. We will do the same for all the cylinders and call the targets D2, D2\_pick, D3, D3\_pick, D4, D4\_pick respectively. In addition to this, we will also create one more target above the assembly base and call it Drop.

Now we will create a program script by clicking on add new program linked to the selected robot, we will add the linked robot it is assigned to from

OMRON TM5-900 to Robotiq. Then we will click on donut1\_pick target and click add joint movement to program and we will rename the program to donut1\_pick. We will do the same for target place and call it donut\_drop. Then we will add a new program, we will make a joint movement to target D1 and then a linear movement to D1\_pick, then we will call for the program donut1\_pick by right clicking the program, selecting add instructions, choosing program call instruction and choosing donut1\_pick, this will reduce the length of the gripper at the position D1\_pick so that it can grip the cylinder, then we will make linear moment again to D1, then a joint movement to target drop and again call for program donut\_drop, this will release the cylinder on the assembly base. We will call this program part1. We will repeat the same process for cylinder 2, cylinder3, and cylinder4 and call it part2, part3, part4 respectively.

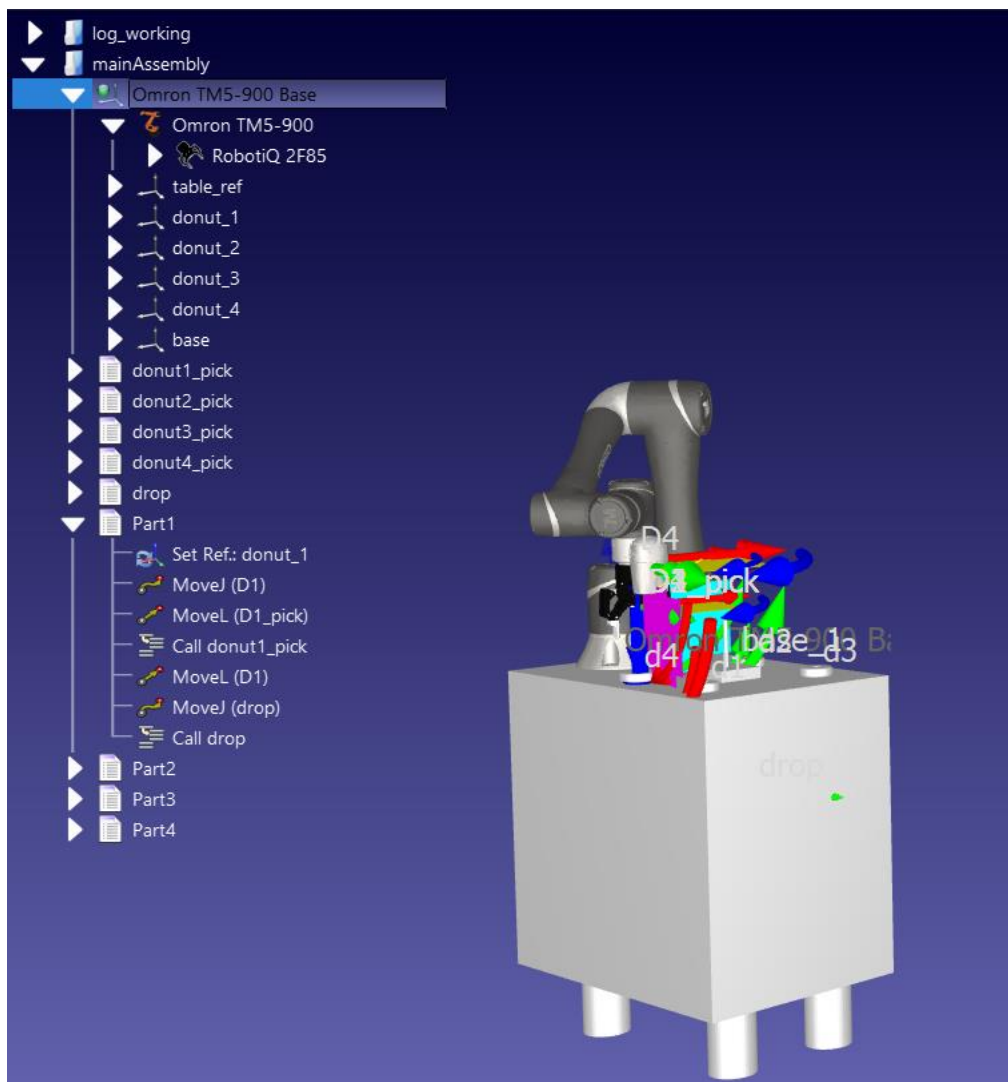


Figure 23 Program Formation

The last step is to combine all the part together, this can be done by adding the new program and we call for the programs part1, part2, part3 and part4 in chronological order. Thus, now the entire assembly process is completed. This program can be called MainAssembly.

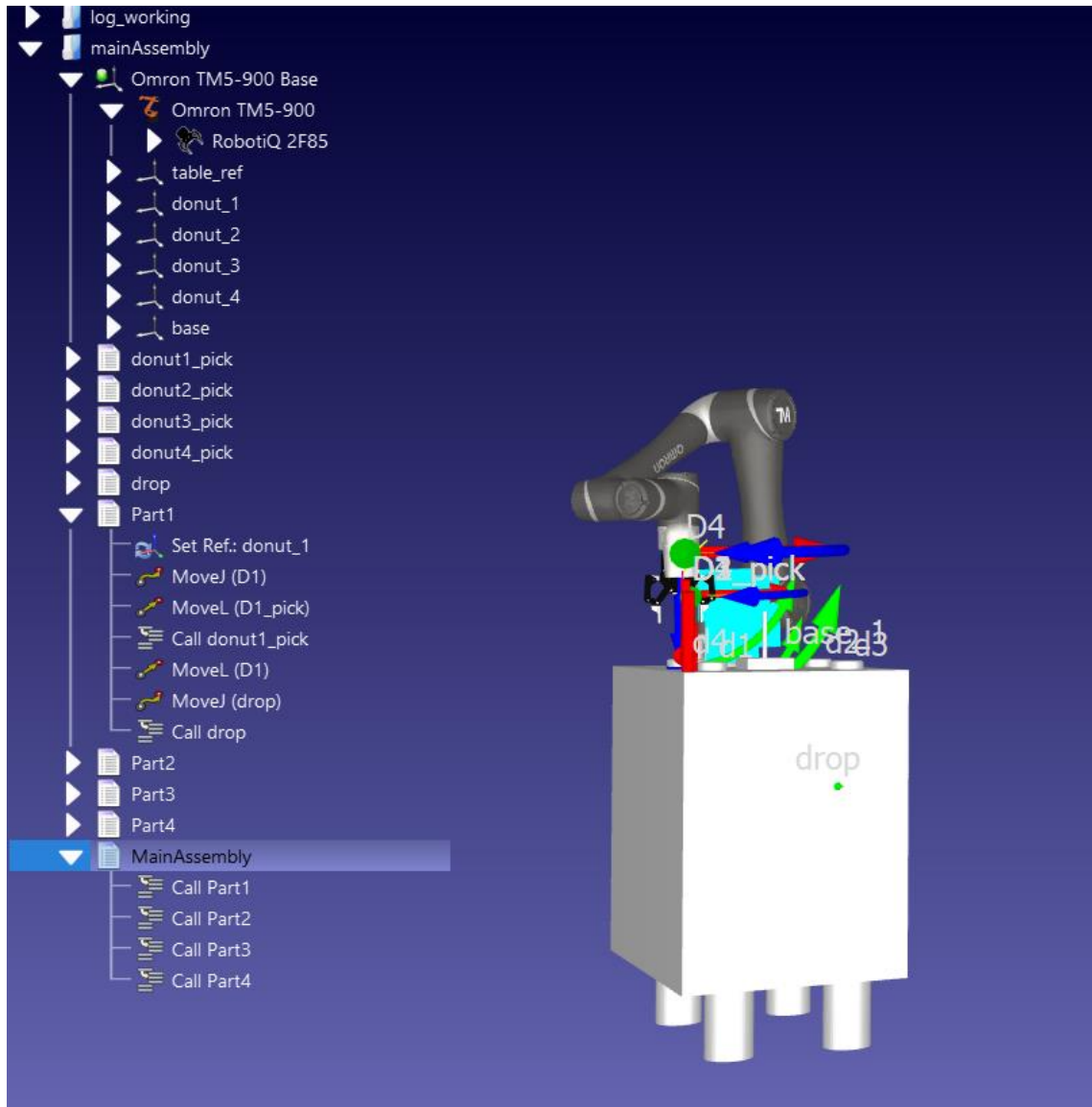
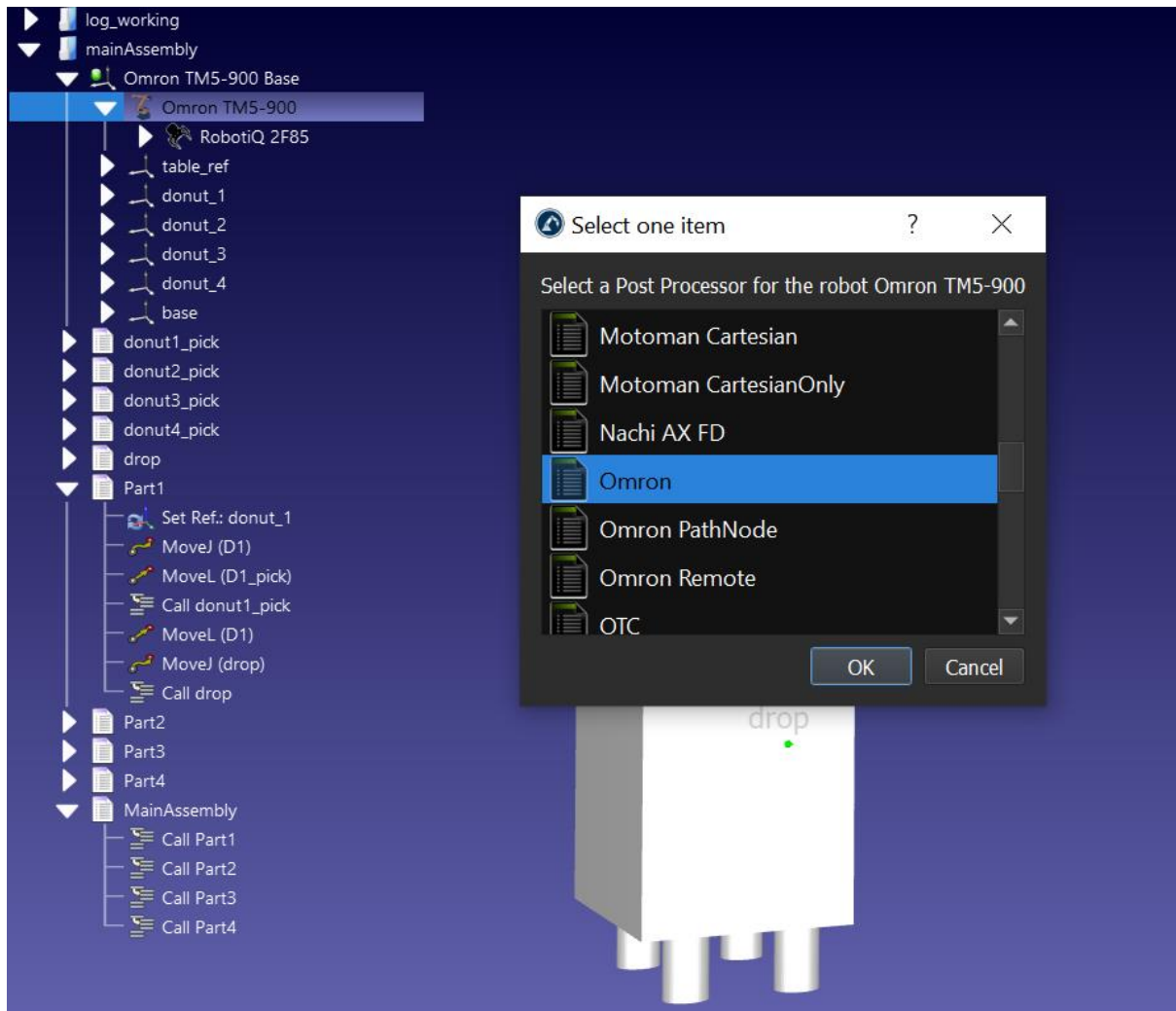


Figure 24 Main Assembly Program

Now the next step is to export this created program into the real OMRON TM5-900. As discussed earlier, the first integration is manual transmission of data from digital environment to real environment. For this we have to convert the program from RoboDK to a format that can be read by control box of OMRON TM5-900. Thus, it is important to check which post-processor is selected in RoboDK before converting the program. This can be done by right clicking the robot, select post-processor and select OMRON from the list.



*Figure 25 Post processor selection*

After selecting the post-processor, now we can generate the robot program of the MainAssembly program by right clicking the MainAssembly and selecting generate robot program as and select the file location. RoboDK will create a zip file that can be used to transfer the program on to the real OMRON TM5-900 robot. It is important to name the usb drive as TMROBOT and folder containing the zip file as TM\_Export. The zip file can be transferred under this folder, now to import the program on OMRON, we have to select the Import/Export window in the TM flow Software, select import, project and MainAssembly program zip file.

## ← MainAssembly

MainAssembly.zip ▸ Projects







<input type="checkbox"/>	 <b>MainAssembly.dat</b>  Type: KMP - MPEG Movie File	Date modified: 14/12/2021 18:26 Size: 1.39 KB → 674 bytes
<input type="checkbox"/>	 <b>MainAssembly.flow</b>  Type: FLOW File	Date modified: 14/12/2021 18:26 Size: 28.1 KB → 2.28 KB
<input type="checkbox"/>	 <b>MainAssembly.prog</b>  Type: PROG File	Date modified: 14/12/2021 18:26 Size: 58 bytes → 70 bytes

Figure 26 Content of the zip file

The second integration is the digital flow of data from digital environment to real environment. This can be achieved from connecting the RoboDK to OMRON TM5-900 in real time, for this it is important to put the OMRON TM5-900 into the listen mode. To do so we have to start a new project and use a listen node from the commands available on the left. And pushing the play button from the control stick to start the program.

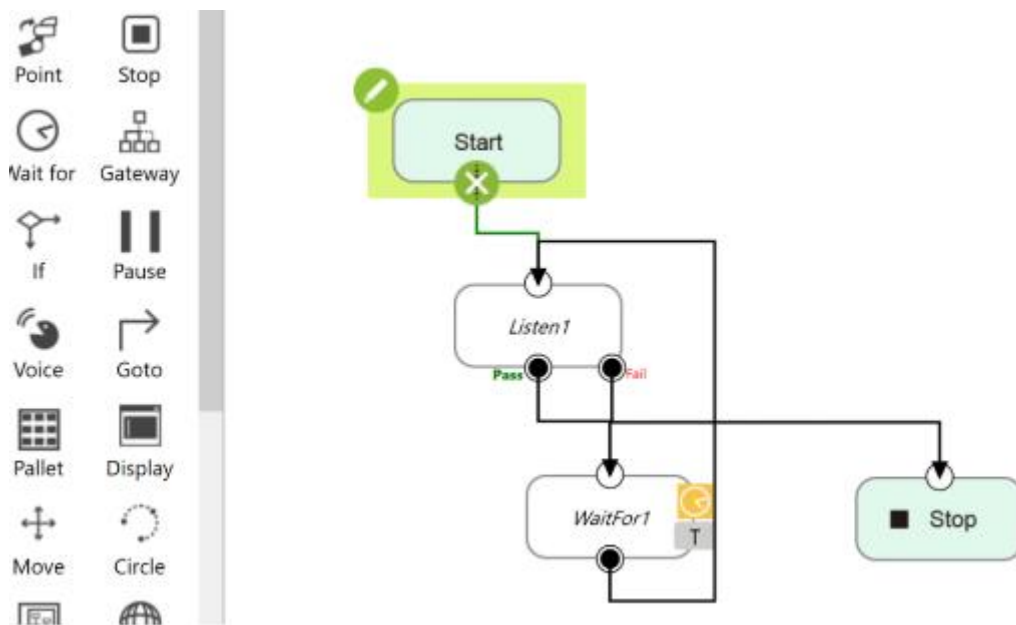


Figure 27 Listen node in TM Flow

To connect it from Robodk, we have to connect our computer to the same network. And in the software, we have to go connect robot in the connect setting on the top and select the OMRON TM5-900 from the available option. Then we will get option to put robot Ip and robot port, where we can put the Ip and the port of the robot in use, in our case the Ip of the robot was 10.0.12.59 and the port is 5890 for all the OMRON robots. Then we can run the MainAssembly program, which will make the simulation run as well as the robot in the real world as well.

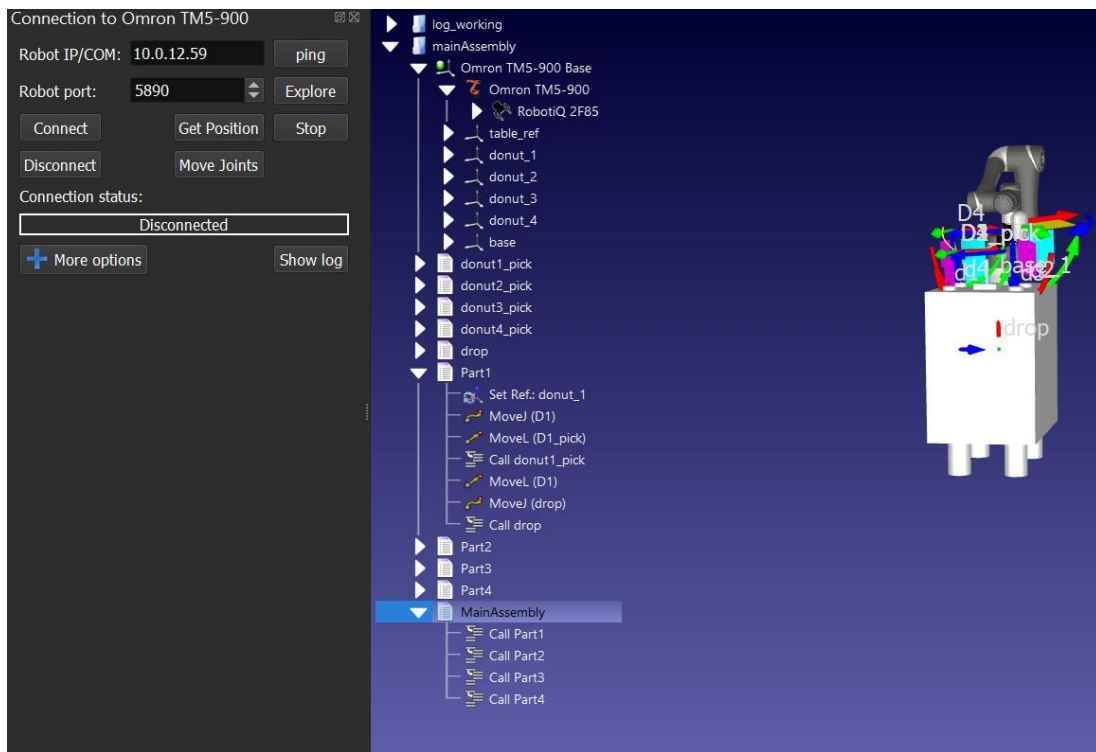


Figure 28 RoboDK live connection

### Third level of integration: -

The third level of integration is to send the data from real world to digital world. For this we have to set up the assembly operation in real time using the TM flow software for OMRON TM5-900, by creating points, joint and linear movement between this points which is straightforward.

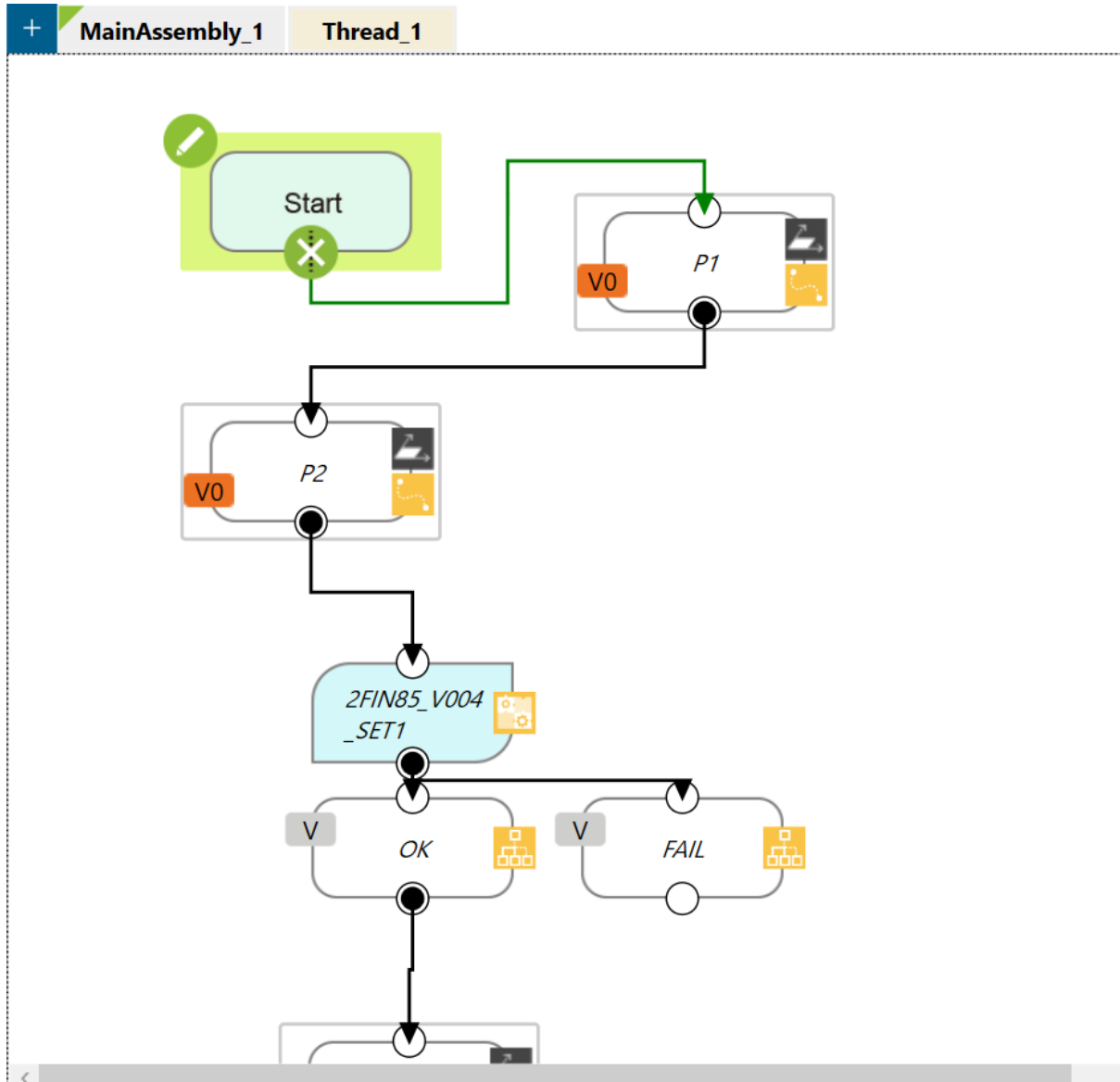


Figure 29 Normal assembly project

But, along with this we also have to extract data from OMRON TM5-900 which we can use to simulate the workstation on RoboDK. We will extract data using Modbus TCP communication. To enable the Modbus TCP communication, we have to go to setting in TMflow, go into connectivity and enable the TCP slave setting. The list of code table to get different robot

parameters can be found on the same page as well. For example, address 7049 is used to extract x co-ordinate with respect to robot base with tool.

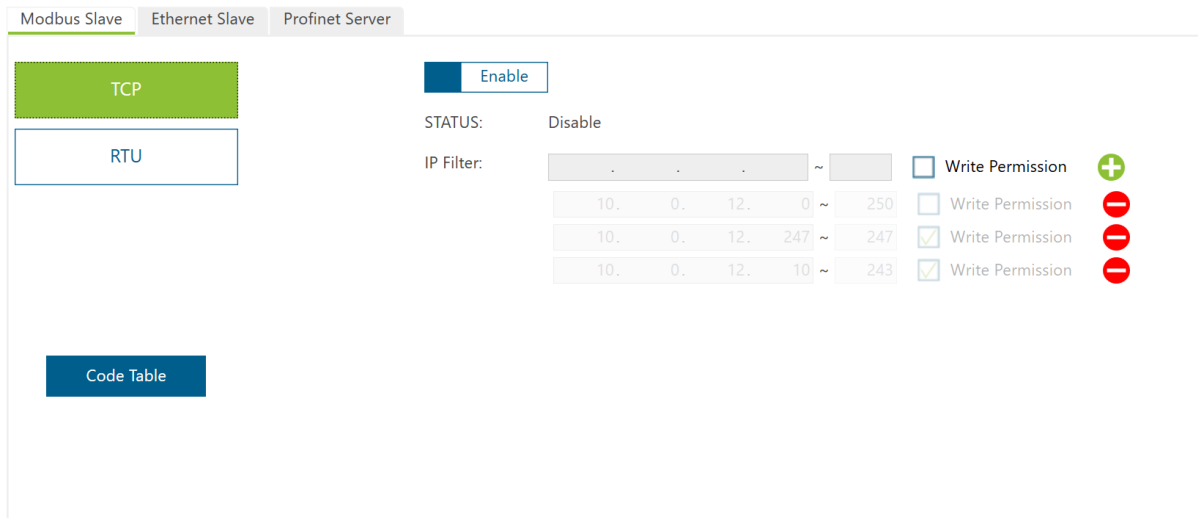


Figure 30 Modbus Slave Setting

Robot Coordinate	FC	Address <sub>10</sub>	Address <sub>16</sub>	Type	R/W	Note1	Note2
X (Cartesian coordinate w.r.t. current Base without tool)	04	7001~7002	1B59~1B5A	Float	R	Dword	mm
Y (Cartesian coordinate w.r.t. current Base without tool)	04	7003~7004	1B5B~1B5C	Float	R	Dword	mm
Z (Cartesian coordinate w.r.t. current Base without tool)	04	7005~7006	1B5D~1B5E	Float	R	Dword	mm
Rx (Cartesian coordinate w.r.t. current Base without tool)	04	7007~7008	1B5F~1B60	Float	R	Dword	degree
Ry (Cartesian coordinate w.r.t. current Base without tool)	04	7009~7010	1B61~1B62	Float	R	Dword	degree
Rz (Cartesian coordinate w.r.t. current Base without tool)	04	7011~7012	1B63~1B64	Float	R	Dword	degree
Joint 1	04	7013~7014	1B65~1B66	Float	R	Dword	degree
Joint 2	04	7015~7016	1B67~1B68	Float	R	Dword	degree
Joint 3	04	7017~7018	1B69~1B6A	Float	R	Dword	degree
Joint 4	04	7019~7020	1B6B~1B6C	Float	R	Dword	degree
Joint 5	04	7021~7022	1B6D~1B6E	Float	R	Dword	degree
Joint 6	04	7023~7024	1B6F~1B70	Float	R	Dword	degree
X (Cartesian coordinate w.r.t. current Base with tool)	04	7025~7026	1B71~1B72	Float	R	Dword	mm
Y (Cartesian coordinate w.r.t. current Base with tool)	04	7027~7028	1B73~1B74	Float	R	Dword	mm
Z (Cartesian coordinate w.r.t. current Base with tool)	04	7029~7030	1B75~1B76	Float	R	Dword	mm

Figure 31 Code table example



For our project, we will extract x co-ordinate, y co-ordinate, z co-ordinate, Rx, Ry, Rz, J1, J2, J3, J4, J5 and J6. To get this data after enabling the Modbus slave setting, we have to go on Modbus Dev, which can be found under top right menu in TM flow. In Modbus device, we have to add setting for all the parameters in local host by selecting add setting. For X cartesian co-ordinate, we will use name tcp\_X, slave Id as 1, address as 7049 from the code table, type as float and order as AB CD.

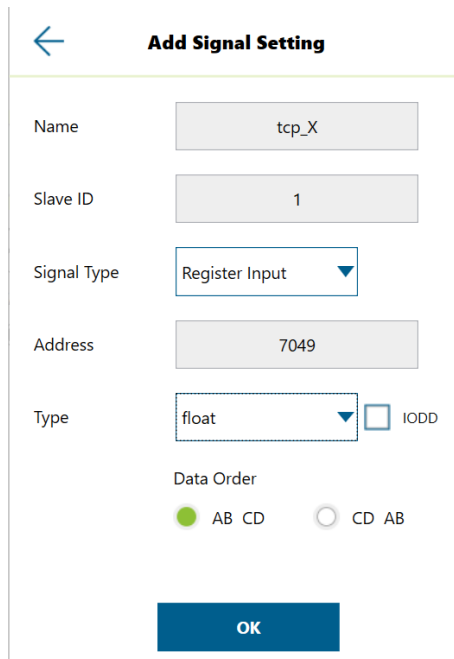


Figure 32 Adding new Modbus Dev Setting

Then we will create new float type variable with name pos\_X and value 0. We will do the same step for all the other parameters as well.

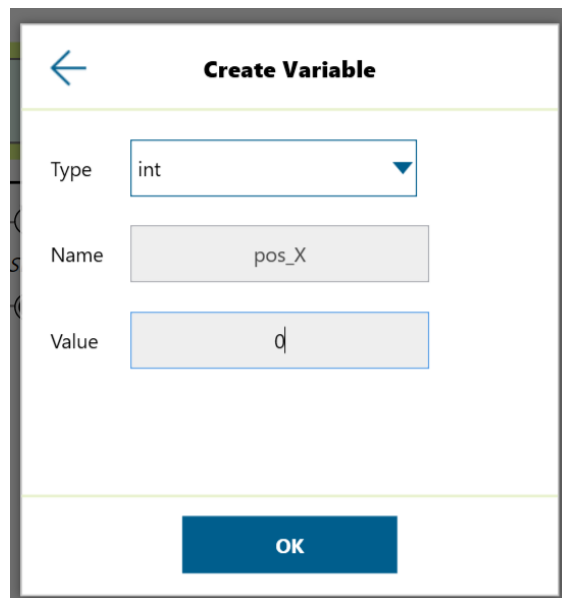


Figure 33 Variable creation in TM Flow

Next, we will create a logging operation to store the value of all the selected parameters after regular intervals, we will use USB stick to store all the logged data. To create a logging operation, in the assembly operation, we will create a new thread linked to the MainAssembly project. We will use the set node from left command window to link the variables to its local Modbus slave setting. This can be done by editing the set node, selecting variables on the left tab, we will select the pos\_X variable which is a float type, '=' expression as the operation, and on the right tab we will select tcp\_X with address 7049 and add it to the set node setting. This will read the value from Modbus slave setting in the variable in the float format. We will do the same for all the other parameters as well.

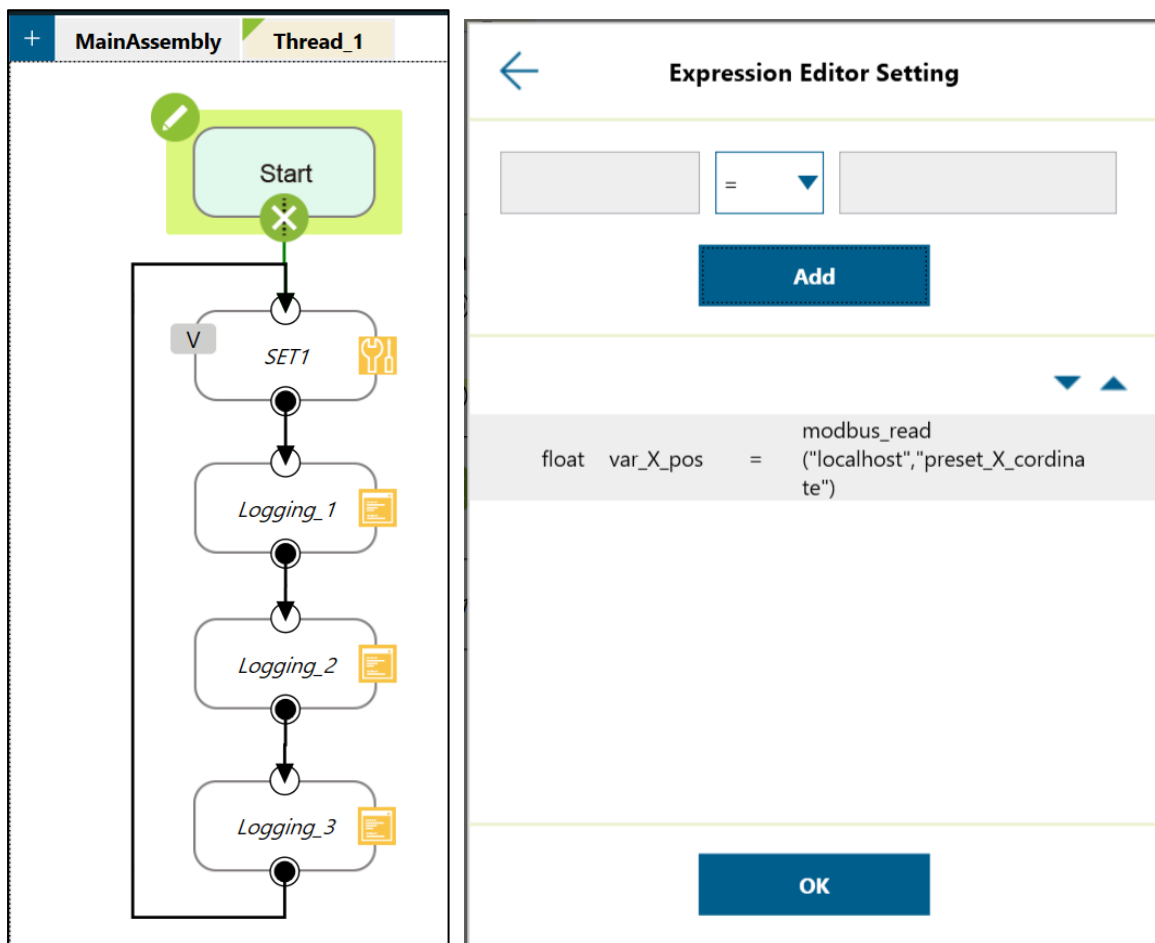


Figure 34 Log Thread along with Set Node setting

After setting up the set node expressions for all the parameters, we will edit the log node, we can name the file, choose the save file path and add the title against each variable to be read, for example we will use "X" for variable X and so on. We can create and edit 3 log node for all the variables in use. Then we just have to start the assembly operation using the control panel. Now the OMRON TM5-900 will do the assembly operation while saving all the data in txt format in usb stick.

**Log** ✕

**Node Name**

**Save Device**

**Save Directory**

Title	Content
"X"	var_X_pos <span style="float: right;">&gt;</span> <input type="text" value="Text"/>
"Y"	var_pos_Y <span style="float: right;">&gt;</span> <input type="text" value="Text"/>
"Z"	var_pos_z <span style="float: right;">&gt;</span> <input type="text" value="Text"/>
"Rx"	var_rot_x <span style="float: right;">&gt;</span> <input type="text" value="Text"/>
"Ry"	var_rot_y <span style="float: right;">&gt;</span> <input type="text" value="Text"/>

*Figure 35 Log Node Setting*

\*MainAssembly - Notepad

File Edit Format View Help

```

11/19/2021 12:52:14 X=364.3497
11/19/2021 12:52:14 Y=-114.331
11/19/2021 12:52:14 Z=424.356
11/19/2021 12:52:14 Rx=-173.255
11/19/2021 12:52:14 Ry=-3.710986
11/19/2021 12:52:14 Rz=88.1555
11/19/2021 12:52:14 J1=-0.01960373
11/19/2021 12:52:14 J2=-19.6838
11/19/2021 12:52:14 J3=109.9358
11/19/2021 12:52:14 J4=4.091111
11/19/2021 12:52:14 J5=94.06889
11/19/2021 12:52:14 J6=2.598333
11/19/2021 12:52:15 X=388.7875
11/19/2021 12:52:15 Y=-119.1395
11/19/2021 12:52:15 Z=429.6682
11/19/2021 12:52:15 Rx=-173.3933
11/19/2021 12:52:15 Ry=-3.709785
11/19/2021 12:52:15 Rz=85.95805
11/19/2021 12:52:15 J1=-0.7688293
11/19/2021 12:52:15 J2=-16.02535
11/19/2021 12:52:17 J3=106.7528
11/19/2021 12:52:17 J4=3.381667
11/19/2021 12:52:17 J5=94.15611

```

*Figure 36 logged Data*

Now, we will take help of Python API in ROBODK to read this data and simulate the workstation. The first part of the code is to read and segregate all the saved value and convert the string into float, i.e., segregate all X co-ordinates, Y co-ordinates and so on. The second part of the program is RoboDK API, which will command the RoboDK to simulate using the python script. We will use both the position and orientation of the end effector as well as the joint angles of each joints to set the robot in a particular pose, as s there is more than one possible position robot can reach a defined point in the working space. Thus, when we define each joint angle along with the end effector position, there is only one possible solution for the robot to reach the define point in the working space. By running this python script which contain both of this programs, we can simulate the OMRON TM5-900 in the digital environment using the data from the real environment.

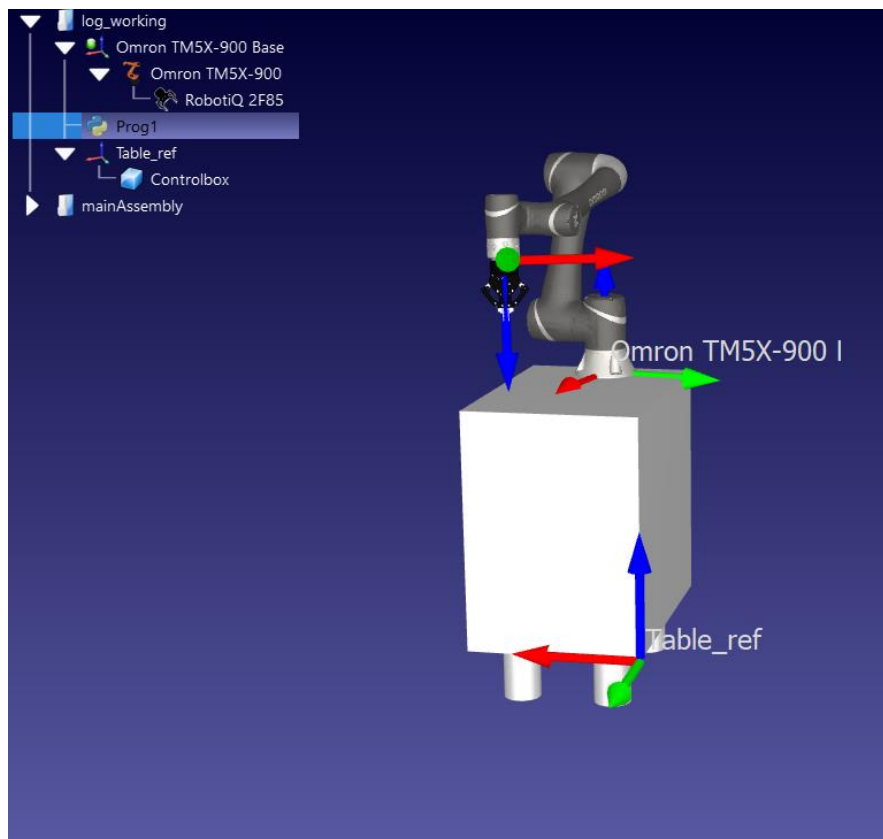


Figure 37 Python API in RoboDK

## 4 ANALYSIS AND DISCUSSION

The work on this project was started from February 2021, but because of some personal reasons, it was stopped in March 2021 and then continued again from August 2021 and finished in December 2021. During the first half period, a study to understand the concept of digital twin and to find the technologies used and available in implementation of digital twin was done. The latter half was used to setup the digital twin of an automated process at MANULAB on NTNU Ålesund.

All three level of integration were successful in terms of developing digital twin and establishing communication through different modes between digital system and physical system for collaborative robot, but in term of robotiq gripper, it was not successful to establish communication between digital system and physical system all thought it was successfully simulated in the digital system.

In third level of integration, use of usb stick was done to transfer the data into the digital system. However, it is possible to transfer and read live data with Modbus TCP. Pymodbustcp can be used as a python plugin to read the live registered data, but because of the restriction on transfer of data using NTNU IP server, it was not managed to establish the live reading of registered values.

## **5 CONCLUSION AND FUTURE WORK**

In conclusion implementation of digital twin in automated manufacturing was successfully studied, the project provided in-depth knowledge on the development of digital twin in automated manufacturing at MANULAB using a case study. During the different part of the thesis, different research questions were successfully addressed. Part of chapter 2 addressed the 1<sup>st</sup> research question, while 2<sup>nd</sup> research question was addressed in chapter 3. Part of chapter 4 and chapter 5 addressed the last research question.

More focus on successfully establishing the connection between digital system and physical system to create digital twin was given without any particular goal in mind, although this improves different values in life cycle of automated manufacturing, but it is in very broad prospect. In the future digital twin should be developed keeping the value in mind.

The work done during this project can be easily reproduced with the help of referring this document, and as digital twin and connection between only one robot was established, in the future establishing the connection with multiple robots at same time should be carried out and thus creating the digital twin of all collaborative robots at MANULAB should be done which will be 10x more beneficial and useful.

Also as discussed in the last chapter, we were unable to establish the connection between digital system and physical system for robotiq gripper, thus it provides the scope for the future work as well.

## 6 REFERENCE

1. Al-Geddawy, T. (2020). A Digital Twin Creation Method for an Opensource Low-cost Changeable Learning Factory. *Procedia Manufacturing*, 51, 1799-1805.
2. Erdei, T. I., Molnár, Z., Obianna, N. C., Gönczi, T., & Husi, G. Industrial KUKA Robot Manipulator in Simulation Environment and Position Read-back. In *1st International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS 2016)*.
3. Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11), 1016-1022.
4. RoboDK Documentation. <https://robodk.com/doc/en/Basic-Guide.html#Start>
5. Modbus TCP overview. <https://www.rtautomation.com/technologies/modbus-tcpip/>
6. Modbus TCP standard [http://www.dankohn.info/projects/Fieldpoint module/Open Modbus TCP Standard.pdf](http://www.dankohn.info/projects/Fieldpoint_module/Open_Modbus_TCP_Standard.pdf)
7. Omron Blog. <https://industrial.omron.eu/en/solutions/blog>
8. TM Flow software manual. [https://assets.omron.eu/downloads/manual/en/v1/tm flow software manual installation manual en.pdf](https://assets.omron.eu/downloads/manual/en/v1/tm_flow_software_manual_installation_manual_en.pdf)
9. Modbus TCP setup. [https://uchimurarobotics.com/wp-content/uploads/2020/01/1.64 ModBus V1.1 ENG.pdf](https://uchimurarobotics.com/wp-content/uploads/2020/01/1.64_ModBus_V1.1_ENG.pdf)
10. Omron Hardware Manual. [https://assets.omron.eu/downloads/manual/en/v2/i623 collaborative robots hardware installation manual en.pdf](https://assets.omron.eu/downloads/manual/en/v2/i623_collaborative_robots_hardware_installation_manual_en.pdf)
11. Robodk Python API documentation. <https://robodk.com/doc/en/PythonAPI/index.html>
12. Robodk Plugin interface. [https://robodk.com/doc/en/PlugIns/class item.html#a1294d328e0bc7e6577f0a70fce273260](https://robodk.com/doc/en/PlugIns/class_item.html#a1294d328e0bc7e6577f0a70fce273260)
13. RoboDK API. <https://robodk.com/doc/en/RoboDK-API.html#RoboDKAPI>
14. Pymodbus interface. <https://rodriguezanton.com/simple-tcp-readout-and-interface-using-pymodbus-sockets-and-flask/>
15. Deloitte university press. [https://www2.deloitte.com/content/dam/Deloitte/kr/Documents/insights/deloitte-newsletter/2017/26 201706/kr insights deloitte-newsletter-26 report 02 en.pdf](https://www2.deloitte.com/content/dam/Deloitte/kr/Documents/insights/deloitte-newsletter/2017/26_201706/kr_insights_deloitte-newsletter-26_report_02_en.pdf)
16. Haag, S., & Anderl, R. (2018). Digital twin–Proof of concept. *Manufacturing Letters*, 15, 64-66.

17. Malik, A. A., & Bilberg, A. (2018). Digital twins of human robot collaboration in a production setting. *Procedia manufacturing*, 17, 278-285.
18. Rosen, R., Von Wichert, G., Lo, G., & Bettenhausen, K. D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, 48(3), 567-572.
19. Zhou, G., Zhang, C., Li, Z., Ding, K., & Wang, C. (2020). Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing. *International Journal of Production Research*, 58(4), 1034-1051.
20. Shao, G., & Helu, M. (2020). Framework for a digital twin in manufacturing: Scope and requirements. *Manufacturing Letters*, 24, 105-107.
21. Burghardt, A., Szybicki, D., Gierlak, P., Kurc, K., Pietruś, P., & Cygan, R. (2020). Programming of industrial robots using virtual reality and digital twins. *Applied Sciences*, 10(2), 486.
22. Roy, R. B., Mishra, D., Pal, S. K., Chakravarty, T., Panda, S., Chandra, M. G., ... & Misra, S. (2020). Digital twin: current scenario and a case study on a manufacturing process. *The International Journal of Advanced Manufacturing Technology*, 107(9), 3691-3714.
23. Zhang, H., Liu, Q., Chen, X., Zhang, D., & Leng, J. (2017). A digital twin-based approach for designing and multi-objective optimization of hollow glass production line. *Ieee Access*, 5, 26901-26911.
24. Ibrion, M., Paltrinieri, N., & Nejad, A. R. (2019, October). On risk of digital twin implementation in marine industry: Learning from aviation industry. In *Journal of Physics: Conference Series* (Vol. 1357, No. 1, p. 012009). IOP Publishing.
25. Sivasankaran, P., & Karthikeyan, R. Simulation of Robot Kinematic Motions using Collision Mapping Planner using Robo Dk Solver.
26. Erdei, T. I., Molnár, Z., Obinna, N. C., & Gönczi, T. Industrial KUKA Robot Manipulator in Simulation Environment and Position Read-back. In *1st International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS 2016)*.



## 7 APPENDIX

### 7.1 Apendix A

#### Code list

Robot Coordinate	FC	Address <sub>10</sub>	Address <sub>16</sub>	Type	R/W	Note1	Note2
X (Cartesian coordinate w.r.t. current Base without tool)	04	7001~7002	1B59~1B5A	Float	R	Dword	mm
Y (Cartesian coordinate w.r.t. current Base without tool)	04	7003~7004	1B5B~1B5C	Float	R	Dword	mm
Z (Cartesian coordinate w.r.t. current Base without tool)	04	7005~7006	1B5D~1B5E	Float	R	Dword	mm
Rx (Cartesian coordinate w.r.t. current Base without tool)	04	7007~7008	1B5F~1B60	Float	R	Dword	degree
Ry (Cartesian coordinate w.r.t. current Base without tool)	04	7009~7010	1B61~1B62	Float	R	Dword	degree
Rz (Cartesian coordinate w.r.t. current Base without tool)	04	7011~7012	1B63~1B64	Float	R	Dword	degree
Joint 1	04	7013~7014	1B65~1B66	Float	R	Dword	degree
Joint 2	04	7015~7016	1B67~1B68	Float	R	Dword	degree
Joint 3	04	7017~7018	1B69~1B6A	Float	R	Dword	degree
Joint 4	04	7019~7020	1B6B~1B6C	Float	R	Dword	degree
Joint 5	04	7021~7022	1B6D~1B6E	Float	R	Dword	degree
Joint 6	04	7023~7024	1B6F~1B70	Float	R	Dword	degree
X (Cartesian coordinate w.r.t. current Base with tool)	04	7025~7026	1B71~1B72	Float	R	Dword	mm
Y (Cartesian coordinate w.r.t. current Base with tool)	04	7027~7028	1B73~1B74	Float	R	Dword	mm
Z (Cartesian coordinate w.r.t. current Base with tool)	04	7029~7030	1B75~1B76	Float	R	Dword	mm

Rx (Cartesian coordinate w.r.t. current Base with tool)	04	7031~7032	1B77~1B78	Float	R	Dword	degree
Ry (Cartesian coordinate w.r.t. current Base with tool)	04	7033~7034	1B79~1B7A	Float	R	Dword	degree
Rz (Cartesian coordinate w.r.t. current Base with tool)	04	7035~7036	1B7B~1B7C	Float	R	Dword	degree
X (Cartesian coordinate w.r.t. Robot Base without tool)	04	7037~7038	1B7D~1B7E	Float	R	Dword	mm
Y (Cartesian coordinate w.r.t. Robot Base without tool)	04	7039~7040	1B7F~1B80	Float	R	Dword	mm

---

Z (Cartesian coordinate w.r.t. Robot Base without tool)	04	7041~7042	1B81~1B82	Float	R	Dword	mm
Rx (Cartesian coordinate w.r.t. Robot Base without tool)	04	7043~7044	1B83~1B84	Float	R	Dword	degree
Ry (Cartesian coordinate w.r.t. Robot Base without tool)	04	7045~7046	1B85~1B86	Float	R	Dword	degree
Rz (Cartesian coordinate w.r.t. Robot Base without tool)	04	7047~7048	1B87~1B88	Float	R	Dword	degree

Z (Cartesian coordinate w.r.t. Robot Base without tool)	04	7041~7042	1B81~1B82	Float	R	Dword	mm
Rx (Cartesian coordinate w.r.t. Robot Base without tool)	04	7043~7044	1B83~1B84	Float	R	Dword	degree
Ry (Cartesian coordinate w.r.t. Robot Base without tool)	04	7045~7046	1B85~1B86	Float	R	Dword	degree
Rz (Cartesian coordinate w.r.t. Robot Base without tool)	04	7047~7048	1B87~1B88	Float	R	Dword	degree
X (Cartesian coordinate w.r.t. Robot Base with tool)	04	7049~7050	1B89~1B8A	Float	R	Dword	mm
Y (Cartesian coordinate w.r.t. Robot Base with tool)	04	7051~7052	1B8B~1B8C	Float	R	Dword	mm
Z (Cartesian coordinate w.r.t. Robot Base with tool)	04	7053~7054	1B8D~1B8E	Float	R	Dword	mm
Rx (Cartesian coordinate w.r.t. Robot Base with tool)	04	7055~7056	1B8F~1B90	Float	R	Dword	degree
Ry (Cartesian coordinate w.r.t. Robot Base with tool)	04	7057~7058	1B91~1B92	Float	R	Dword	degree
Rz (Cartesian coordinate w.r.t. Robot Base with tool)	04	7059~7060	1B93~1B94	Float	R	Dword	degree

Project Speed	FC	Address <sub>10</sub>	Address <sub>16</sub>	Type	R/W	Note
Project Speed	04	7101	1BBD	Int16	R	%
Change Project Speed	06	7101	1BBD	Int16	W	% (Auto Mode only) Project speed can only be written multiples of five. (5≤Project Speed≤100)

Running Timer	FC	Address <sub>10</sub>	Address <sub>16</sub>	Type	R/W	Note
Day	04	8200~8201	2008~2009	Int32	R	Current project running time. (If the project is stopped, it will return to 0.)
Hour	04	8202	200A	Int16	R	
Minute	04	8203	200B	Int16	R	
Second	04	8204	200C	Int16	R	

Up Time	FC	Address <sub>10</sub>	Address <sub>16</sub>	Type	R/W	Note
Day	04	8206~8207	200E~200F	Int32	R	Time elapsed since the controller was started
Hour	04	8208	2010	Int16	R	
Minute	04	8209	2011	Int16	R	
Second	04	8210	2012	Int16	R	

## 7.2 Appendix B

Python script for RoboDK

```
import pandas as pd
```

```
readfile =
```

```
pd.read_csv('C://Users//47463//Desktop//Log_omron//Savan//logg_2_success-2021-.log',sep=" ",skiprows=1)
```

```
listofvalues = readfile.values.tolist()
```

```
value_list = []
```

```
for i in range(len(listofvalues)):
```

```
    print(listofvalues[i][2])
```

```
    print(listofvalues[i][2].split("=",1))
```

```
    split_value = listofvalues[i][2].split("=", 1)
```

```
    value_list.append([split_value[0],split_value[1]])
```

```
X_list = []
```

```
Y_list = []
```

```
Z_list = []
```

```
Rx_list = []
```

```
Ry_list = []
```

```
Rz_list = []
```

```
J1_list = []
```

```
J2_list = []
```

```

J3_list = []
J4_list = []
J5_list = []
J6_list = []

print(len(value_list))

for i in range(len(value_list)):

    if 'X' in value_list[i][0]:
        X_list.append(float(value_list[i][1]))

    if "Y" in value_list[i][0]:
        Y_list.append(float(value_list[i][1]))

    if "Z" in value_list[i][0]:
        Z_list.append(float(value_list[i][1]))

    if "Rx" in value_list[i][0]:
        Rx_list.append(float(value_list[i][1]))

    if "Ry" in value_list[i][0]:
        Ry_list.append(float(value_list[i][1]))

    if "Rz" in value_list[i][0]:
        Rz_list.append(float(value_list[i][1]))

    if "J1" in value_list[i][0]:
        J1_list.append(float(value_list[i][1]))
    if "J2" in value_list[i][0]:
        J2_list.append(float(value_list[i][1]))
    if "J3" in value_list[i][0]:
        J3_list.append(float(value_list[i][1]))
    if "J4" in value_list[i][0]:
        J4_list.append(float(value_list[i][1]))
    if "J5" in value_list[i][0]:
        J5_list.append(float(value_list[i][1]))
    if "J6" in value_list[i][0]:
        J6_list.append(float(value_list[i][1]))

#uncomment to check the total as a verification measure
# print(len(X_list) + len(Y_list) + len(Z_list) + len(Rx_list) +
len(Ry_list) + len(Rz_list) + len(J1_list) +

```

```

# len(J2_list) + len(J3_list) + len(J4_list) + len(J5_list) +
len(J6_list))

from robolink import *           # import the robolink library
from robodk import *           # import the robodk library

RDK = Robolink()               # connect to the RoboDK API
robot = RDK.Item("", ITEM_TYPE_ROBOT) # Retrieve a robot
available in RoboDK
robot.setSpeed(200)

for i in range(100):
    pose1 = robodk.Pose ( X_list[i], Y_list[i], Z_list[i], Rx_list[i],
Ry_list[i], Rz_list[i])

    robot.MoveL(pose1)
robot.setJoints([J1_list[i], J2_list[i], J3_list[i], J4_list[i], J5_list[i],
J6_list[i]])

```