POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Meccanica

Tesi di Laurea Magistrale

VIRTUAL ANALYSIS AND SIMULATIONS OF INDUSTRIAL ROBOTS AND AUTOMATION IN INDUSTRY 4.0'S DEVELOPMENT FOR A CONVEYORS CELL'S APPLICATION

Relatori: Prof. Giovanni Belingardi
Prof. Khalid Mirza

Candidato: Carlo Frasca

Aprile 2019

To my mother,
my family,
Maryia, Narendra and Oussama.

Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The aim of this project is to propose a new innovative approach regarding to Industry 4.0 applications and its brand-new perspective. Industry 4.0 is commonly referred as the forth industrial revolution. It is the name given to a present trend of automation and data exchange in manufacturing technologies. In order to understand what Industry 4.0 means, it is useful to remember the previous industrial revolutions. [1] The three of the past were all triggered by technical innovations: the introduction of water- and steam- powered mechanical manufacturing at the end of the 18^{th} century, the division of labor at the beginning of the 20^{th} century and introduction of programmable logic controllers (PLC) for automation purposes in manufacturing, the born of Internet, computers, Industrial robotics and communication technology in the 1970s. According to experts from industry and research, the upcoming industrial revolution will be triggered by the Internet, which allows communication between humans as well as machines. Industry 4.0 focuses on the establishment of intelligent products and production process [2]. In future manufacturing, factories have to cope with the need of rapid product development, flexible production as well as complex environments, so that the factory of the future or "smart factory" will enable the communication between humans, machines and products alike. These structures allow to make changes to the production with a minimum effort in terms of time and cost. Actually, traditional manufacturing industry already uses robots and cutting-edge technologies. Most likely they are required for custom design and it is not easy to change it in the small to medium manufacturing industries, because the risk of putting even more advanced technologies might make their process not able to change in the quickest way. This project's main purpose is to give proof of how this is going to change whether small to medium manufacturing industries follow Industry 4.0's dictates.
As a result, many of the advances in technology that form the foundation for Industry 4.0 are already used in manufacturing, but with Industry 4.0, they will transform production: isolated, optimized cells will come together as a fully integrated, automated and optimized production flow, leading to greater efficiencies and changing traditional production relationships among suppliers, producers and costumers, as well as between human and machine.
Manufactures in many industries have long used robots indeed to tackle complex assignments, but robots are evolving for even greater utility. They are becoming more autonomous, flexible and cooperative. However, it is useful to make a comparison between traditional robots and collaborative robots in order to understand which advantages have to be exploited [3,4].
Robots and humans have been co-workers for years, but rarely have been truly working together. This is about to change with the rise of collaborative robots (cobots). Unlike traditional industrial robots, collaborative robots are not placed behind guards or in cages. Instead, they can be designed to work safely around people. Key differences between a robot and a cobot are: ease to set up because traditional robots require advanced programming skills, whereas training cobots can be as easy as setting them into servo-assist teach mode and guiding the arm through the tasks; and safe because industrial robots blindly perform their task until they are told to stop, whereas cobots can have pre-set speeds and torques, which can allow them to stop if they detect something in their way, so that they can be deployed within the traditional production working space. Indeed, cobots are incredibly versatile in terms of use, unlike older robots that are either purpose-built for a task or difficult to adapt to new tasks, so that cobots can be used where traditional industrial robots would

be too bulky, imprecise and unsafe: assembling delicate electronic components, packaging food or handling complex tasks that require 100% accuracy.
Eventually, advanced collaborative robots will interact with one another and work safely side by side with humans and learn from them. It easy to understand how cobots and industrial robots differ now, especially regarding an autonomous Industry 4.0's application, while collaborative robots are designed to work alongside human employees, industrial robots do work in place of those employees. By contrast, traditional robots are used to automate the manufacturing process almost entirely without human help on the manufacturing environment. Industry 4.0, in turn, frees up humans for more meaningful tasks that are less tedious and mundane and are less prone to repetitive motion injuries.
[5] In the design phase of the project, 3-D simulations of production processes are already extensively used, but, for a large project, in the future simulations will be used for most applications and in plant operations as well. These simulations will leverage real-time data to mirror the physical world in a virtual model, which can include machines, products, and humans. This allows operators to test and optimize the machine settings for the next product in line in the virtual world before the physical changeover, thereby driving down machine setup times and increasing quality. For example, Siemens and a German machine-tool vendor developed a virtual machine that can simulate the machining of parts using data from the physical machine. This lowers the setup time for the actual machining process by as much as 80 percent.
It is the first accomplishment of this project to indeed test an updated manufacturing system and to use Siemens' tools to simulate and predict its functionality while it has not built yet. The main simulation software that describes the project is going to be Siemens Tecnomatix Process Simulate Robotics v14.0 and it will mirror the Industry 4.0 suitable version of a Conveyor System that has been built and performed by Siemens PLM in the Oakland University Industrial and System Dept.'s Sharf Lab. Secondly, the Conveyor System's simulations will be used to enforce the candidature of a new class and activity at Oakland University that will let senior students deal with the system in the Lab as they will choose to attend "Industrial Robots and Automation" summer courses.
This work has 8 chapters and it is organized as follows. Chapter 2 contains the present system description, stock list, assumptions and the programming code to physically command the system. Chapter 3 briefly introduces the software and tools needed to import, re-model and eventually virtualize the Conveyor System's model. Chapter 4 presents the 6-axis robots' models that a Mechanical engineering company in Michigan (Applied Manufacturing Technologies) picked for the system. Chapter 5 aims to show the core of the work by illustrating the updated model of the entire system in the simulation environment. Chapter 6 is about the simulation process and the robotics planning paths operations. In Chapter 7 the results are shown and discussed. Chapter 9 concludes the whole work, commenting the project and discussing possible future improvements and applications.

Chapter 2
System description

A conveyor system is a common piece of mechanical handling equipment that moves materials from one location to another [6]. Conveyors are especially useful in applications involving the transportation of heavy or bulky materials. Conveyor systems allow quick and efficient transportation for a wide variety of materials, which make them very popular in the material handling and packaging industries. They also have popular consumer applications, as they are often found in supermarkets and airports, constituting the final leg of item/ bag delivery to customers. Many kinds of conveying systems are available and are used according to the various needs of different industries. There are chain conveyors (floor and overhead) as well. Chain conveyors consist of enclosed tracks, I-Beam, towline, power & free, and hand pushed trolleys.

Figure 0-1 The Conveyor System – view one

Figure 0-2 The Conveyor System – view two

2.1 Working assumption and Siemens stock list

The objective of the chapter is to give an idea of the activities and the learning meant to show how the PLC logic works and to get firsthand experience in programming a PLC conveyor system to perform a specific set of tasks. First, we had to choose what tasks Applied Manufacturing Technologies (AMT) wanted to perform. We chose to use the system as a post-assembly conveyor system where we checked the top portion of each part, ignoring some of the components we did not reckon helpful for the application [7].

We assumed a few preconditions for our system:

- Next station on the conveyor must not have any part for current station to release part
- Previous station will hold part if gripper arm is working
- Camera station will hold the part for enough time for camera to perform check

We also performed the following tasks for initialization:

- Any safety issue shuts down the system
- Reset button on HMI resets system
- Auto button on HMI starts system
- All conveyor motors start working
- Main air is turned on
- Dividers keep open path to main conveyor 1
- Gripper robot goes to home position (i.e. raised, retracted, and grip open)
- Camera programming/logic is pre-loaded to the system

BOM-STOCKLIST				BOM-STOCKLIST			
DETAIL	QTY.	MANUFACTURER	DESCRIPTION	DETAIL	QTY.	MANUFACTURER	DESCRIPTION
100	1	SIEMENS	IM 154-8 F CPU – PROFINET, PROFIBUS 512K	135	1	SIEMENS	ET200PRO MS/RSM CABLE – CONNECT BACKPLANE
101	1	SIEMENS	IM 154-8 F CPU – PROFINET, COMM MODULE	136	1	SIEMENS	SITOP PSU110P IP67 POWER SUPPLY 120/230VAC
102	1	SIEMENS	SIMANTIC S7, MICRO MEMORY CARD -F, ET200 V3	137	1	SIEMENS	POWER T-CLAMPING COUPLER 2.5mm/4mm
103	2	SIEMENS	F-SWICH MODULE PROFISATE	138	1	SIEMENS	ET200SP, PROFINET INTERFACE MODULE
104	1	SIEMENS	4/8 F-DI 4 F- DO 24VDC PROFISATE MODULE	139	1	SIEMENS	ET200SP, 4 F-DQ PROFISAFE OUTPUT MODULE, 24VDC
105	2	SIEMENS	16 DIGITAL INPUT MODULE – 24VDC STANDARD	140	1	SIEMENS	ET200SP, 8 F-DI PROFISATE INPUT MODULE, 24 VDC

106	2	SIEMENS	8 DIGITAL OUTPUT MODULE – 24VDC/0.5A STANDARD	141	1	SIEMENS	ET200SP DIGITAL INPUT MODULE 16x24VDC STANDARD
107	2	SIEMENS	16 DIGITAL OUTPUT MODULE – 24VDC	142	3	SIEMENS	ET200SP RELAY OUTPUT MODULE 4 – DQ
108	1	SIEMENS	CONNECTION MODULE – IM 154-8 CPU	143	2	SIEMENS	ET200SP – BASE UNIT – 8/4 F - INPUT/OUTPUT MODULE
109	1	SIEMENS	CONNECTION MODULE – IM PN M12, 7/8	144	3	SIEMENS	ET200SP – BASE UNIT – 4 OUTPUT MODULE
110	2	SIEMENS	CONNECTION MODULE – 8xM12 INPUTS	145	1	SIEMENS	ET200SP – BASE UNIT – 16 INPUT MODULE
111	2	SIEMENS	CONNECTION MODULE – 8xM12D OUTPUTS	146	1	FESTO	MANIFOLD VALVE – PROFINET- *32P-SCD-N-MAHAH-5MJJL
112	1	SIEMENS	CONNECTION MODULE – 4/8 F – DI/4 F -DO	147	1	FESTO	MANIFOLD VALVE – PN- *32P-SCD-N-MAA-3MJJNLL
113	2	SIEMENS	CONNECTION MODULE – F - SWITCH PROFISAFE	148	1	FESTO	MANIFOLD VALVE – PN- *32P-SCD-N-MA-3MJ
114	2	SIEMENS	MOUNTING RACK - 1000mm	149	2	FESTO	CONNECTOR CABLE
115	1	SIEMENS	RFID COMUNICATION MODULE RF170C	150	34	FESTO	FITTING MINI CONNECTOR W/INTERNAL HEX M7
116	1	SIEMENS	CONNECTION MODULE – RFID 2xM12 8PIN	151	10	FESTO	BLANKING PLUG HEX M7
117	1	SIEMENS	RFID READER ANTENNA – RF310R	152	3	FESTO	MAIN MANIFOLD SUPPLY FITTING
118	1	SIEMENS	SIMANTIC RF300 READER CABLE ASSY, 5M LENGTH	153	A/R	FESTO	PLASTIC TUBING
119	10	SIEMENS	MOBY D / RF300 MOBILE DATA MEMORY MDS D424	154	A/R	FESTO	PLASTIC TUBING
120	1	SIEMENS	REAPIR SWITCH – DISCONNECT	155	A/R	FESTO	PLASTIC TUBING BLUE 4MM O.D.
121	1	SIEMENS	400V ASM DISCONNECTING MODULE	156	1	SIEMENS	7’’ TOUCH HMI TP700 COMFORT PANEL
122	6	SIEMENS	DIRECT MOTOR STARTER 0.15/2.0A	157	1	RITTAL	HMI ENCLOSURE – FOR TOP COUPLING W/EMP PLATE
123	8	SIEMENS	BACKPLANE BUS MODULE 110mm	158	1	RITTAL	HMI ENCLOSURE – WITH 7’’

							CUTOUT / TOP COUPLING
124	1	SIEMENS	VFD FREQUENCY CONVERTER MODULE	159	1	RITTAL	CP120 ARM 2000mm
125	1	SIEMENS	BACKPLANE BUS MODULE – VFD	160	1	RITTAL	CP120 90 DEGREE COUPLING - ROTATING
126	7	SIEMENS	CONNECTOR – MOTOR 15mm 9POLE, PG16	161	1	RITTAL	CP120 90 DEGREE ELBOW
127	2	SIEMENS	CONNECTOR – POWER INFEED 6.0mm	162	1	RITTAL	CP120 WALL/BASE MOUNT COUPLING – NON-ROTATING
128	7	SIEMENS	POWER JUMPER PLUG – MOTOR	163	11	LUMBERG	M12 4 PIN MICRO CORD, 18 AWG, STR. MALE, RT ANG FEM.
129	3	SIEMENS	SEAL PLUG – MOTOR POWER HAN Q SOCKET	164	8	LUMBERG	M12 4 PIN MICRO CORD, 18 AWG, STR. MALE, RT ANG FEM.
130	1	SIEMENS	BLANK PLUG 7/8° FOR ET200 24VDC SUPPLY	165	22	LUMBERG	MICRO FIELD CONNECTOR 4 PIN, 18 AWG. SCREW TYPE
131	10	SIEMENS	BLANK PLUG 12mm FOR ET200 UNUSED I/O SOCKETS	166	5	LUMBERG	MINI 5-POLE CORD, STR MALE, STR FEM – 15FT LENGTH
132	6	SIEMENS	CONNECTOR 1800 DGR – METAL HOUSING – PROFINET	167	1	LUMBERG	MINI 3-POLE CORD, STR FEMALE, SGLE END – 6FT LENGTH
133	1	SIEMENS	PROFINET CABLE – CAT 5 S00 -20 M	168	1	LUMBERG	MINI 4-POLE CORD, STR MALE, SGLE END – 6FT LENGTH
134	2	SIEMENS	PROFINET CONNECTOR IP65 4WIRE/SHLD	169	16	LUMBERG	Y-CONNECTOR – 1 MALE – 2 FEMALES TO I/O BLOCK

Table 0-1 The Conveyor System's stock list

2.2 Process description

At first, the PLC HMI is used to place a request using a menu interface to specify amounts. The order is saved and associated with the RFID tag on the on the next tray to be released from the buffer. Once the order is confirmed, the conveyor system releases a tray from the buffer. There will be a Robot one that picks up from a container and place the part onto the moving tray. Robot one then releases the part and gets back to home position. According to RFID scan, the tray is filled with the order placed from the HMI. Robot two (that can possibly be both a 6-axis or a Scara robot) picks second part and places them onto the first one on passing trays. When the order is completed,

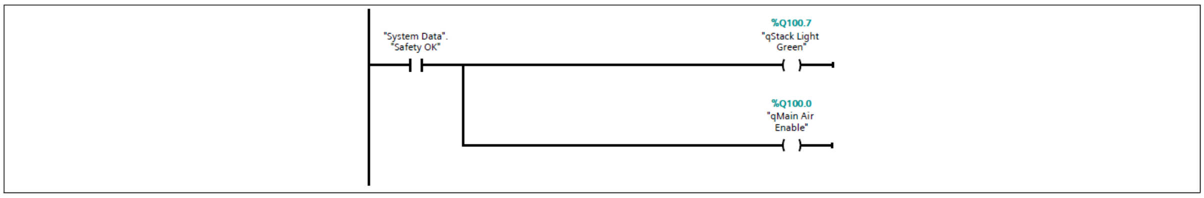
the tray is diverted out of the conveyor loop to pick-up location. Once the order is removed. The tray is diverted back into the loop and comes to rest at the back of the buffer.
Finally, purpose of this project will regard to a class activity in Oakland University and there will be different working groups that picks one task and perform it.

2.3 Ladder program

Normally Siemens PLC are programmed in Ladder logic code. [8] It is a graphic language that mainly represents for a programming environment the functioning of a power grid, in which electric coils are powered or not according to the state of the switches. Ladder logic's emergence is about helping to program and the ones used to cable the logic control circuits by using electromechanical relays. Here is the Ladder logic code written to help the understanding of the conveyor system motions and its Siemens PLCs computation.
Remember before going through that there are two lines for the robot (robot one and two). There are no i/o for the second robot, so it was used the same ones from the first robot. When a new robot gets added the tags can be switched to the new robots' tags.

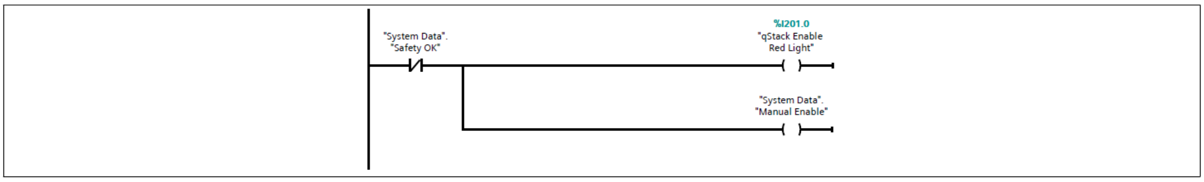
Network 1: Green light and main air on when safety enabled

If "Safety OK" is on then main air is enabled and the green stack light turns on



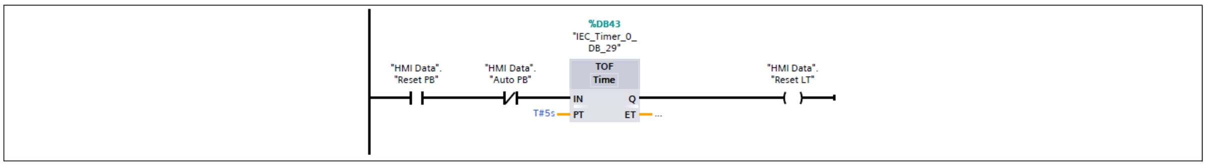
Network 2: Red light when safety disabled

If "Safety OK" is off, i.e. light screen has been broken, etc., red stack light turns on. It also sets the "Manual Enable" bit which is a dummy bit used to detect that safety has been turned off.

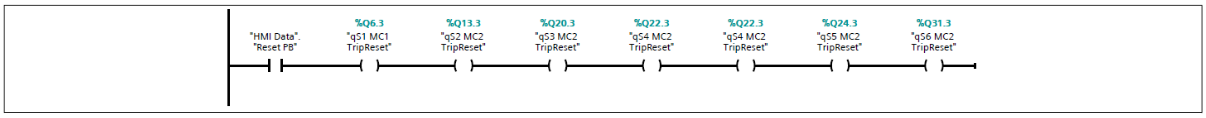


Network 3: Hold reset light on

Holds reset light on in HMI until auto button on HMI is pressed. Stack lights are not visible due to HMI blocking it.

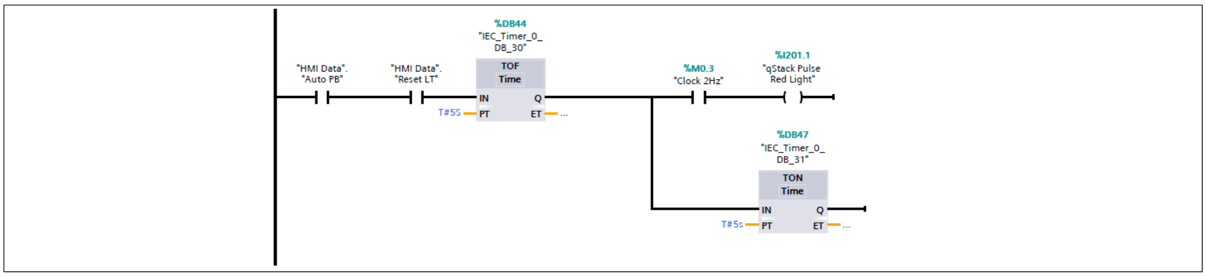


Network 4: Motor Resets



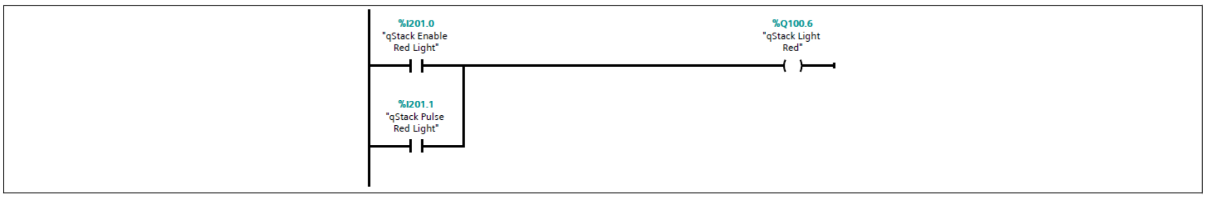
Network 5: Safety clear out

When auto button on HMI is pressed and the reset light on HMI is on the red stack light flashes in warning to allow people to clear out before the conveyors start.



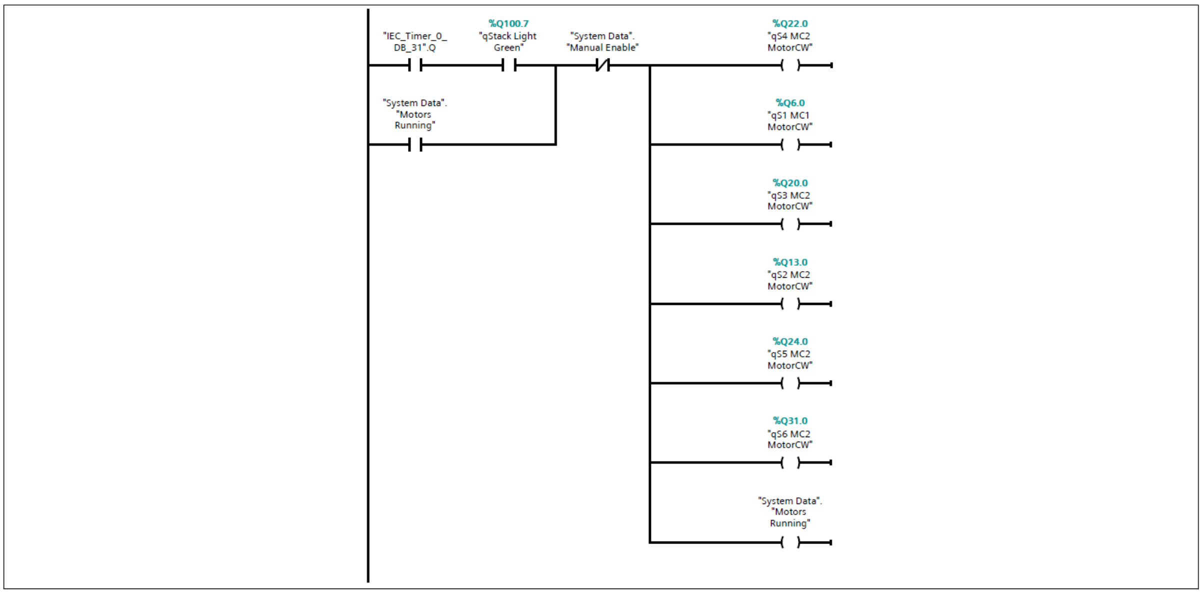
Network 6: Red light on

This controls how the red stack light is turned on during the safety clear out and during safety off condition.



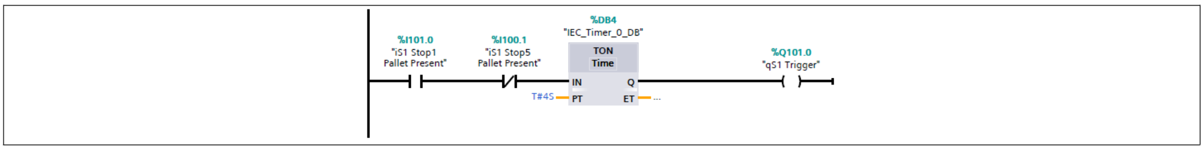
Network 7: Initialize conveyor motors

When flashing warning lights is done and green stack light is on, which says safety is OK, all motors are turned on. This also sets the dummy bit on "Motors Running" which turns off when safety is off and dummy bit "Manual Enable" is on.



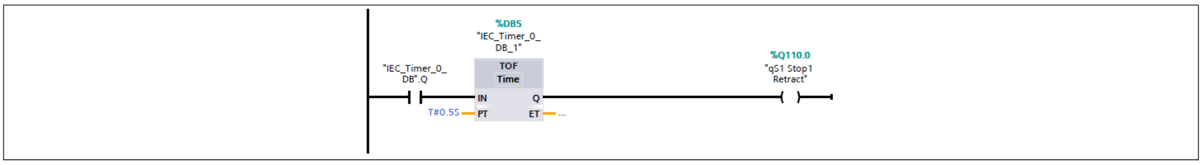
Network 8: Conveyor 1 stop 1 - Camera station

From here we start working on the operation of the stoppers on the conveyors. We only release a part from a station once the station next to it is clear. To perform this we will use the stoppers.
This rung starts the logic for stopper at conveyor 1 stop 1. Once there is a part on conveyor 1 stop 1 and no part on conveyor 1 stop 5 we start a timer. The function of the timer is to allow some time to pass before it releases the part and keep the stopper retracted for some time.
This rung also has the output "qS1 Trigger." This is a bit used to tell the camera to take a picture.



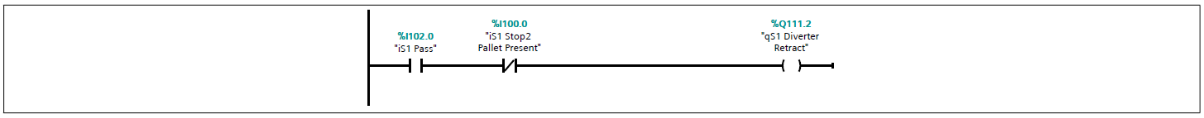
Network 9: Conveyor 1 stop 1 - Camera station

This is a continuation of the previous rung. Once the timers are done the stopper at conveyor 1 stop 1 is retracted.



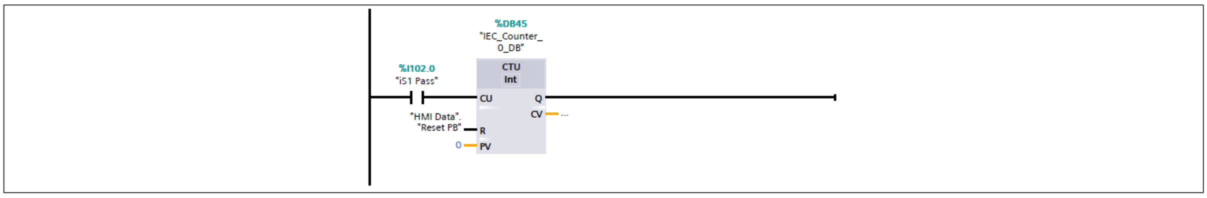
Network 10: Pass orientation check

This rung reads the output from the camera. When "iS1_Pass" is on it means that the part was checked by the camera and determined to be in the corrrect orientation. Then the diverter retracts to let the part pass through to the side conveyor 1, which we determined to be used as a "Shipping Bay." The passed parts collect here until the bay is full.
The bay is determined to be full if there is a part at conveyor 1 stop 2, because its the last stop on the conveyor.



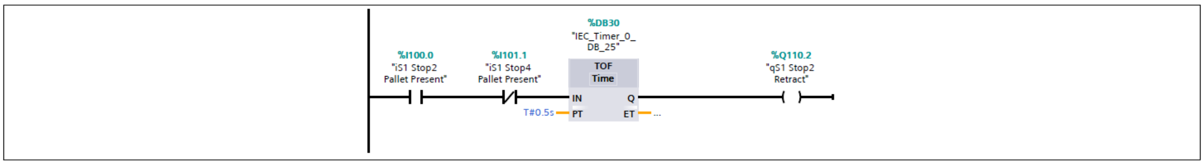
Network 11: Pass counter

If the part passed the check with the Camera then we count the part in a counter. This counter value is displayed at the HMI screen.



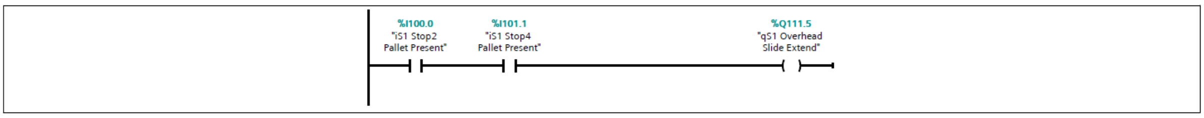
Network 12: Shipping Conveyor 1 stop 2

When there is a part on conveyor 1 stop 2 but no part on conveyor 1 stop 4, conveyor 1 stop 2 is retracted to let the part pass through. This is to allow parts to collect in the "Shipping Bay" properly.
If the shipping full button is pressed then the parts get released back into the main conveyor. This is just an added feature for testing purposes.



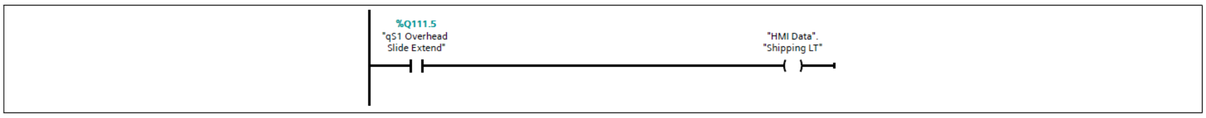
Network 13: Shipping full

If there is a part at conveyor 1 stop 2 and 4 that means that the shipping bay is full. Then the overhead slide extends giving a visual signal that shipping is full.



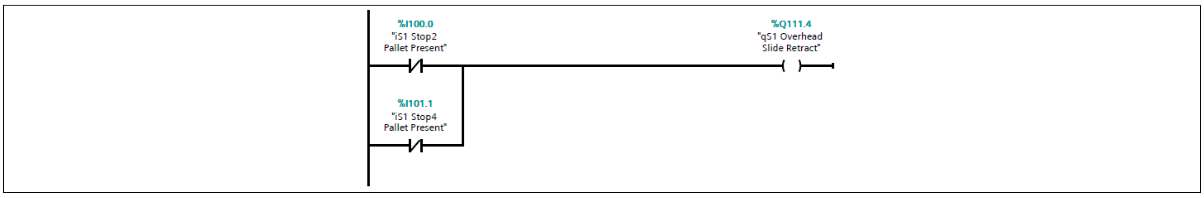
Network 14: Shipping full HMI light

The HMI display also turns on the light on the Shipping Full button.



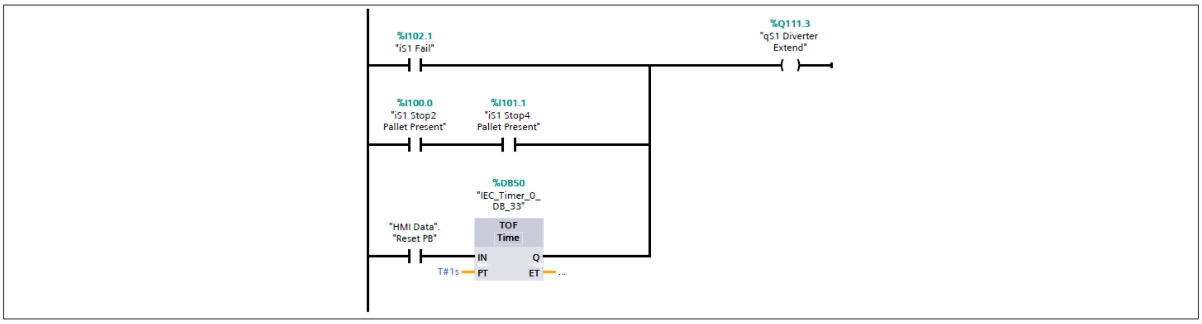
Network 15: Shipping not full

If the shipping is not full the overhead slide is retracted.



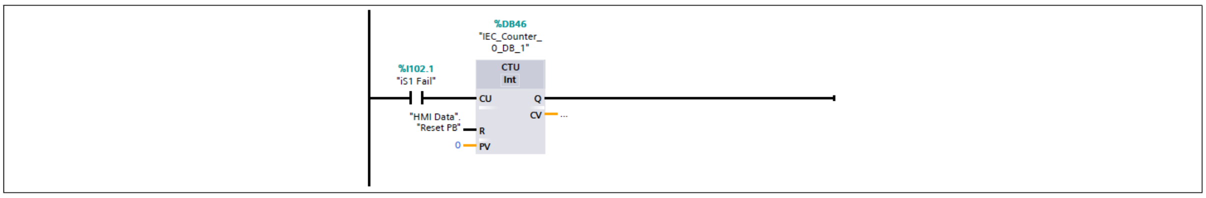
Network 16: Fail orientation check

This rung reads the output from the camera. When "iS1_Fail" is on it means that the part was checked by the camera and determined to be in the wrong orientation. Then the diverter extends to let the part pass through to the main conveyor 1 to let the part keep cycling until it passes.



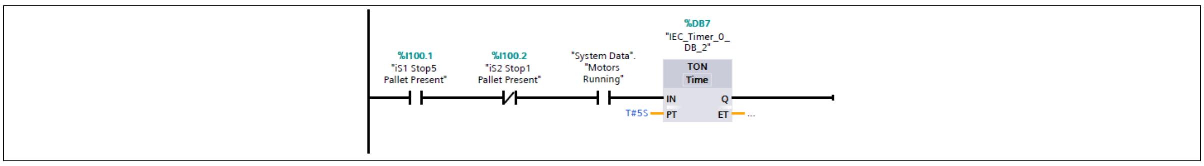
Network 17: Fail counter

If the part failed the check with the Camera then we count the part in a counter. This counter value is displayed at the HMI screen.



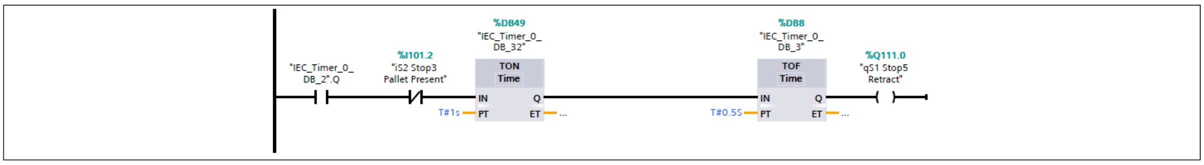
Network 18: Conveyor 1 stop 5

Once there is a part on conveyor 1 stop 5 and no part on conveyor 2 stop 1 we start a timer to allow some time to pass before it releases the part and keep the stopper retracted for some time.

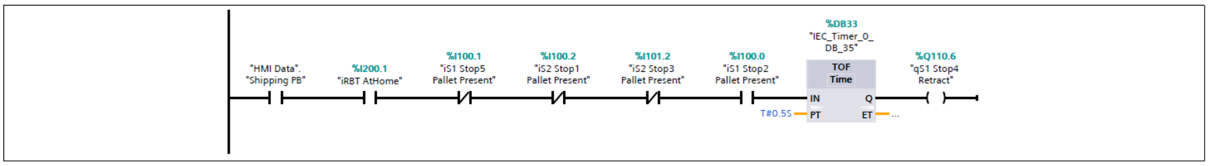


Network 19: Conveyor 1 stop 5

This is a continuation of the previous rung. Once the timers are done and there is no part at conveyor 2 stop 3 the stopper at conveyor 1 stop 5 is retracted. This check is needed for the Robot Arm.

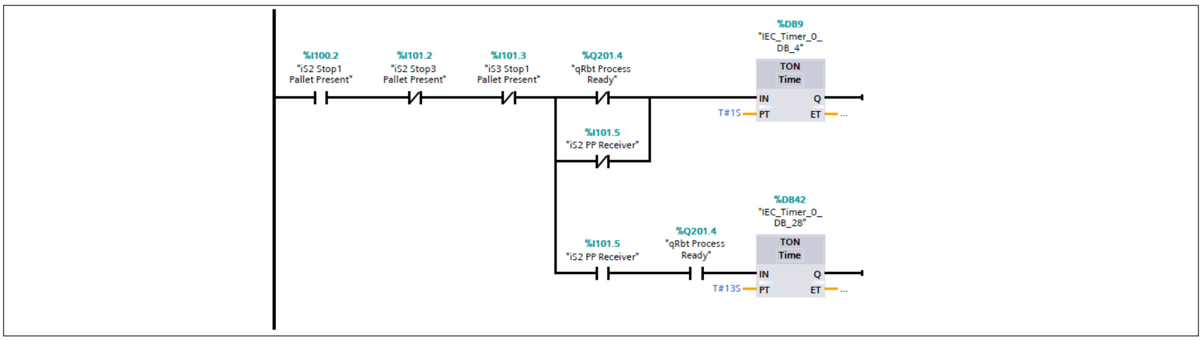


Network 20: Feed Part from Shipping



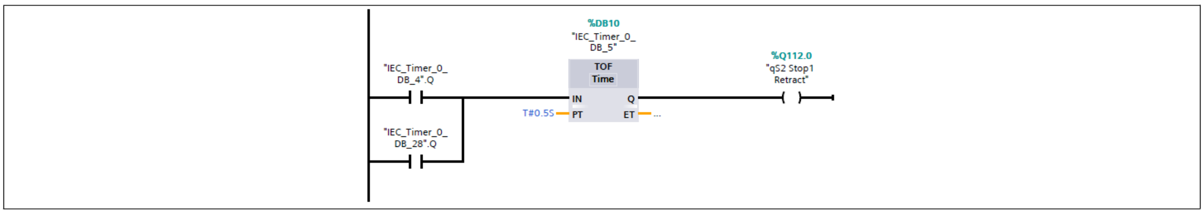
Network 21: Conveyor 2 stop 1 - Robot Arm station 1

Once there is a part on conveyor 2 stop 1 and no part on conveyor 2 stop 1 or 3 we start a timer to allow some time to pass before it releases the part and keep the stopper retracted for some time. This also checks that the Robot Arm is not ready and there is no part detected at PP reciever prox so that parts don't pile behind each other. If there is a part at PP receiver prox and the Robot Arm is ready then a new timer is started for 11.5s to allow for the Robot Arm to work.



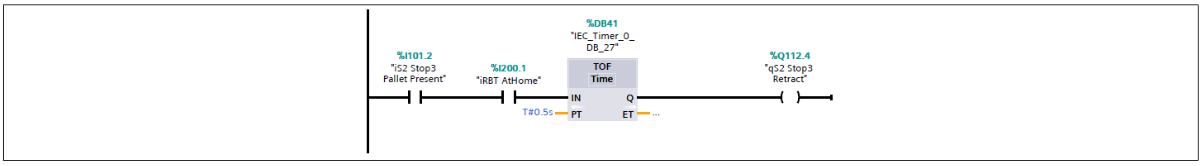
Network 22: Conveyor 2 stop 1 - Robot Arm station 1

This is a continuation of the previous rung. Once the timers are done the stopper at conveyor 2 stop 1 is retracted.

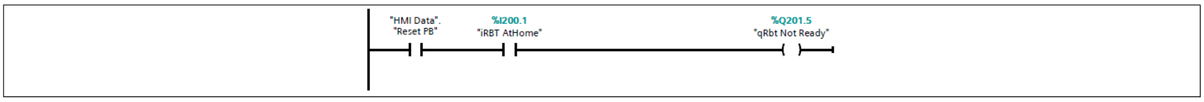


Network 23: Conveyor 2 stop 3 - Robot Arm station 2

Once there is a part on conveyor 2 stop 3 and the Robot Arm is back at home position we start a timer to allow some time to pass before it releases the part and keep the stopper retracted for some time.

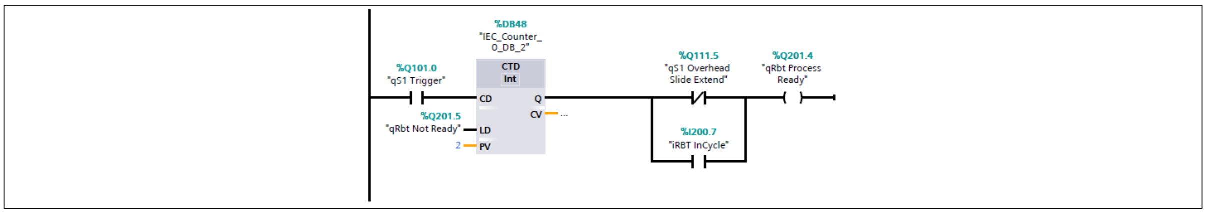


Network 24: Reset Robot Ready Bit



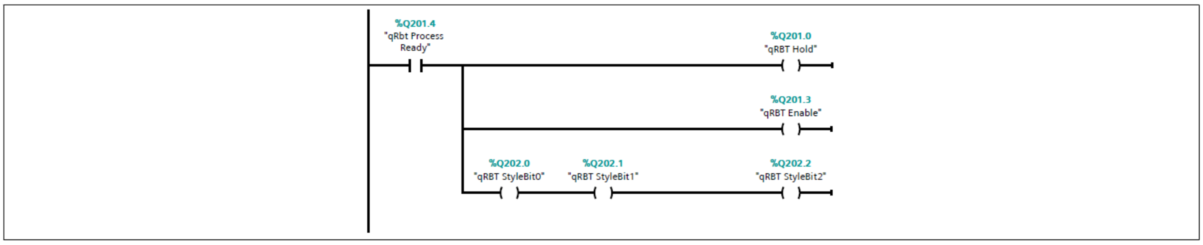
Network 25: Robot Arm ready

When the camera Trigger has performed twice for the same part we know for sure if the part is passed or fail, then we set the bit for the Robot ready bit. If shipping is full and passed parts are cycling on the conveyor then the Robot will not perform its operation.



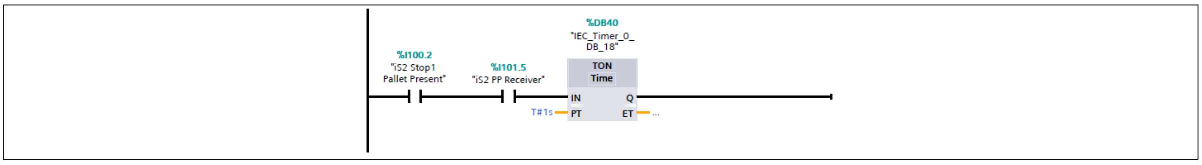
Network 26: Robot Arm hold, enable, and read the program

If Robot is process ready then this rung will turn off hold, turn on enable, and read the program from the memory.



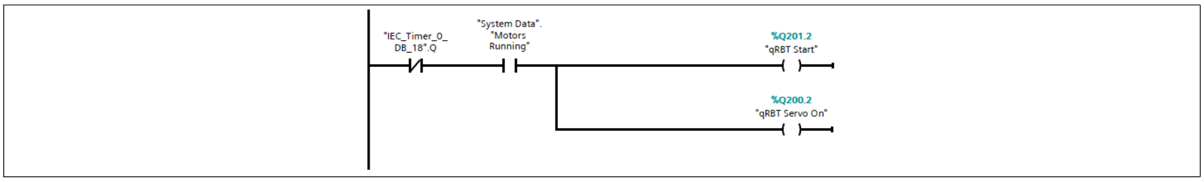
Network 27: Start Robot Arm

If there is part present at conveyor 2 stop 1 and prox detects a part at PP reciever, a timer is started to start the Robot.



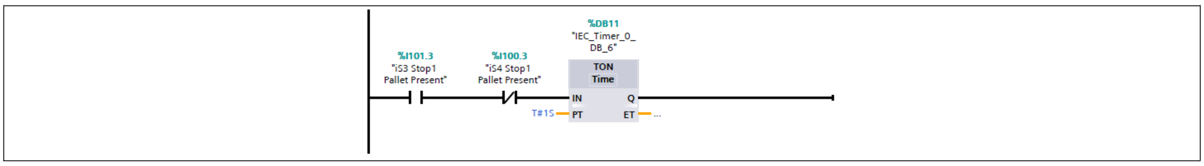
Network 28: Initialize Robot Arm start

Once the timer is done, we reset any faults set from before and start the Robot



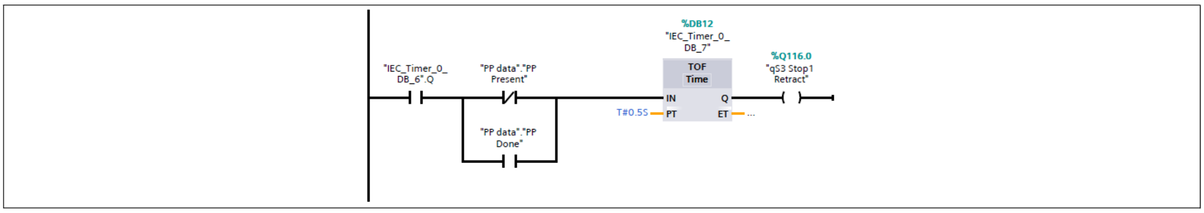
Network 29: Conveyor 3 stop 1 - Gripper Robot station

Once there is a part on conveyor 3 stop 1 and no part on conveyor 4 stop 1 we start a timer to allow some time to pass before it releases the part and keep the stopper retracted for some time.



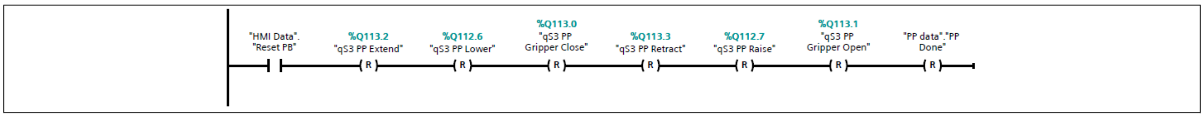
Network 30: Conveyor 3 stop 1 - Gripper Robot station

Once the "PP present" is on, meaning there was a tall part detected at the previous station, it prevents the stopper from retracting. This keeps the part stationary and allows the Gripper Robot to work on it. Once "PP done" is set the stop is retracted to let the part pass.



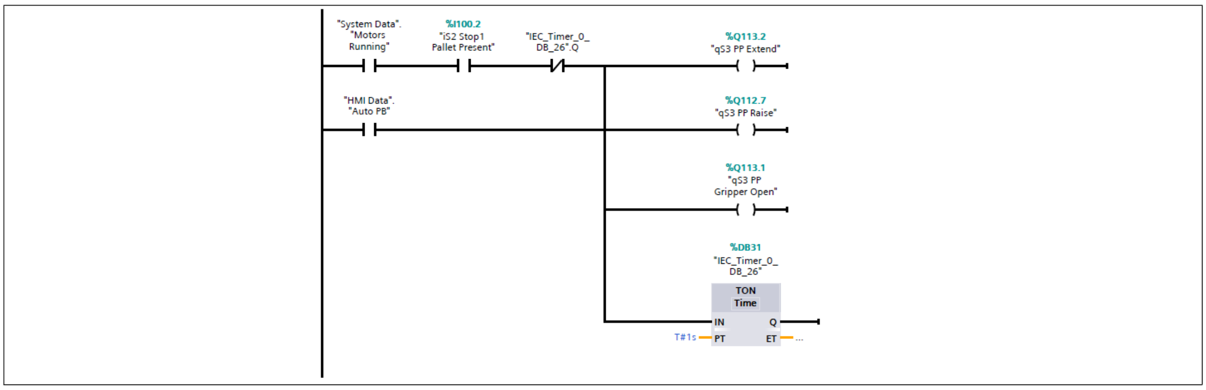
Network 31: Reset Gripper Robot bits

When reset button on HMI is pressed all the bits of the Gripper Arm is reset, including the Gripper done dummy bit.



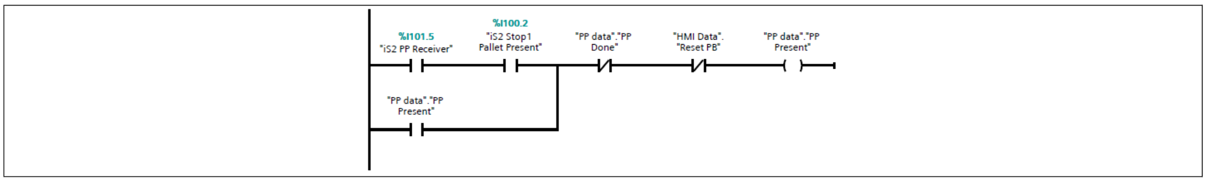
Network 32: Gripper Robot home position

This rung allows the Gripper Robot to go to home position at the beggining when auto button is pressed on the HMI, or the conveyor motos are running and there is a part at conveyor 2 stop 1. Home position is defined at fully extended, raised, gripper open. We have a timer to allow the operations to complete before turning the rung off.



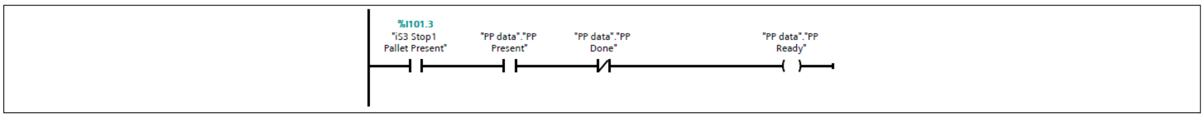
Network 33: Initialize Gripper Robot

Once the "iS2 PP Receiver" prox and conveyor 2 stop 1 detects a part, and dummy bit "PP done" and reset button on HMI is are not set, we latch a dummy bit "PP present." This bit allows us to start the Gripper Robot operation only when these input conditions are on.



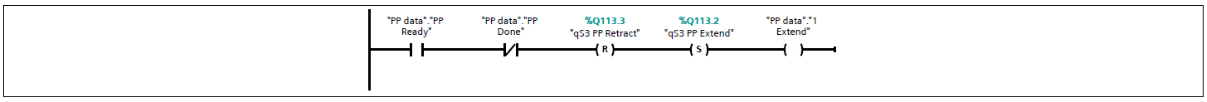
Network 34: Gripper Robot Ready

When there is a part at conveyor 3 stop 1, "PP Present" is set from before, and "PP done" is not set (Gripper Robot operation is not complete), we set dummy bit "PP Ready" as a pre-condition for the Gripper Robot to start working.



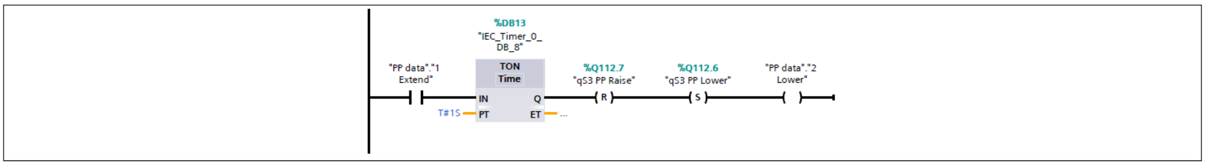
Network 35: Gripper step 1 - Extend

When "PP Ready" is set and "PP done" is not set then we reset the retract bit on the Gripper Robot and set the extend bit. This allows the Gripper Robot to extend. We also set a dummy bit "1 Extend" to transition to the next step for the full process for the Gripper.



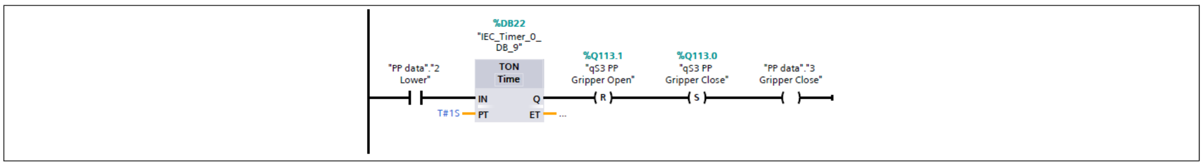
Network 36: Gripper step 2 - Lower

When "1 Extend" is set we give it 1s to perform the extend operation. Then we reset the raise bit on the and set the lower bit. This allows the Gripper Robot to lower. We also set a dummy bit "2 Lower" to transition to the next step.



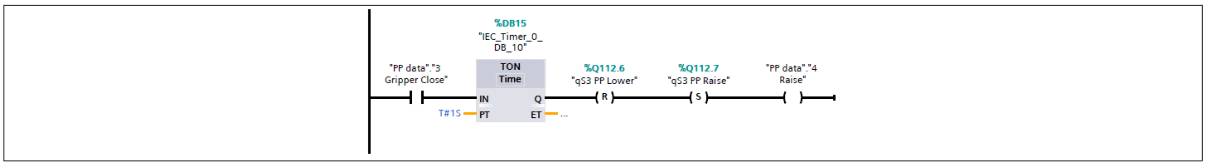
Network 37: Gripper step 3 - Close

When "2 Lower" is set we give it 1s to perform the lower operation. Then we reset the gripper open bit and set the gripper close bit. This allows the Gripper Robot to close its gripper. We also set a dummy bit "3 Gripper Close" to transition to the next step.



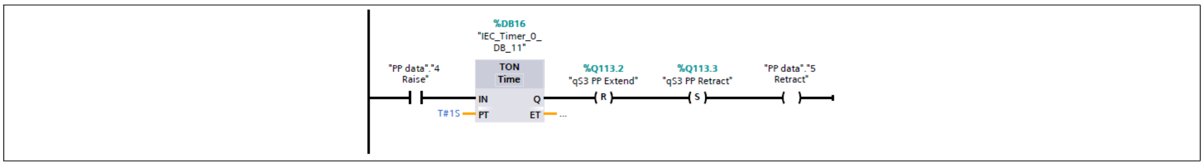
Network 38: Gripper step 4 - Raise

When "3 Gripper Close" is set we give it 1s to perform the gripper close operation. Then we reset the lower bit and set the raise bit. This allows the Gripper Robot to raise. We also set a dummy bit "4 Raise" to transition to the next step.



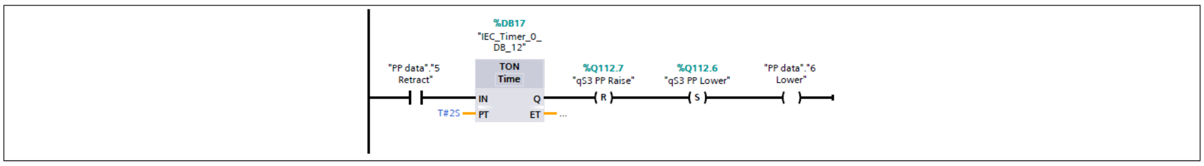
Network 39: Gripper step 5 - Retract

When "4 Raise" is set we give it 1s to perform the raise operation. Then we reset the extend bit and set the retract bit. This allows the Gripper Robot to retract. We also set a dummy bit "5 Retract" to transition to the next step.



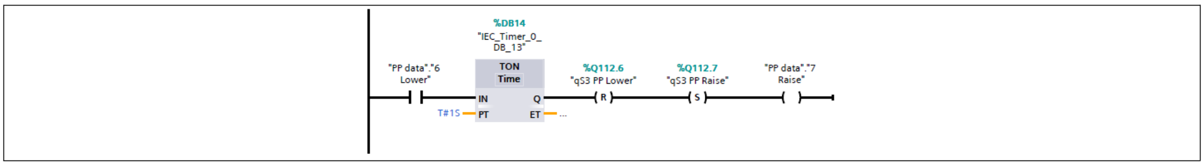
Network 40: Gripper step 6 - Lower

When "5 Retract" is set we give it 2s to perform the retract operation. Then we reset the raise bit on the and set the lower bit. This allows the Gripper Robot to lower. We also set a dummy bit "6 Lower" to transition to the next step.



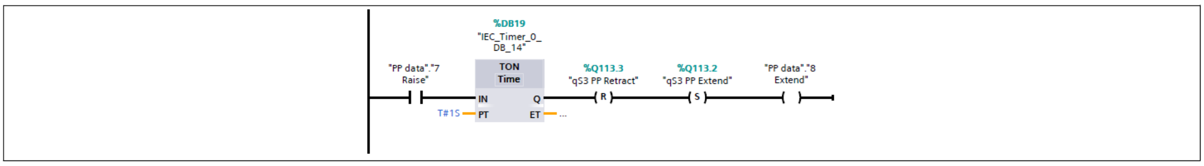
Network 41: Gripper step 7 - Raise

When "6 Lower" is set we give it 1s to perform the lower operation. Then we reset the lower bit and set the raise bit. This allows the Gripper Robot to raise. We also set a dummy bit "7 Raise" to transition to the next step.



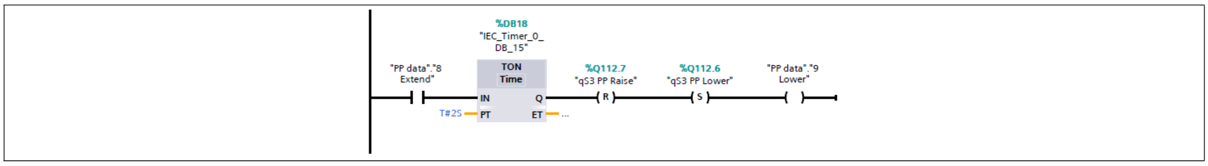
Network 42: Gripper step 8 - Extend

When "7 Raise" is set we give it 1s to perform the raise operation. Then we reset the retract bit and set the extend bit. This allows the Gripper Robot to extend. We also set a dummy bit "8 Extend" to transition to the next step.



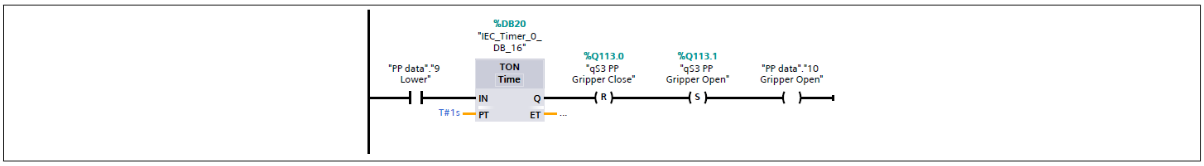
Network 43: Gripper step 9 - Lower

When "8 Extend" is set we give it 2s to perform the extend operation. Then we reset the raise bit on the and set the lower bit. This allows the Gripper Robot to lower. We also set a dummy bit "9 Lower" to transition to the next step.



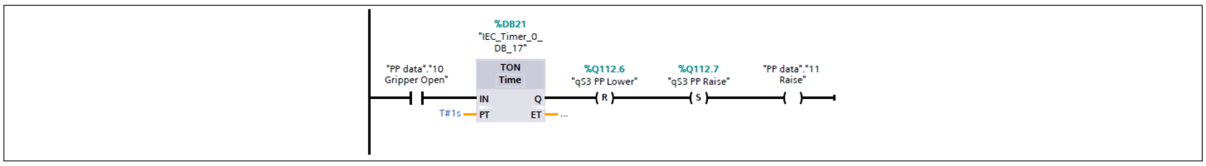
Network 44: Gripper step 10 - Open

When "9 Lower" is set we give it 1s to perform the lower operation. Then we reset the gripper close bit on the and set the gripper open bit. This allows the Gripper Robot to open its gripper. We also set a dummy bit "10 Gripper Open" to transition to the next step.



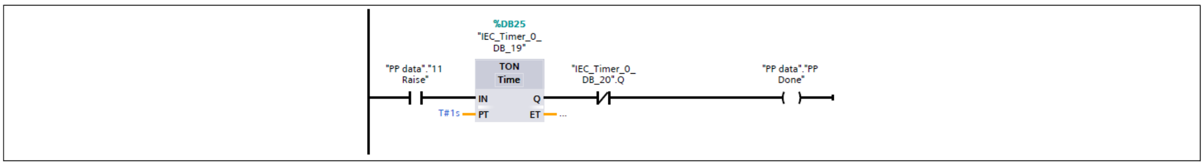
Network 45: Gripper step 11 - Raise

When "10 Gripper Open" is set we give it 1s to perform the gripper open operation. Then we reset the lower bit on the and set the raise bit. This allows the Gripper Robot to raise. We also set a dummy bit "11 Raise" to transition to the next step.



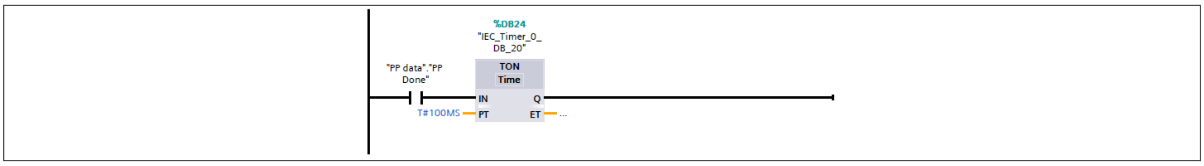
Network 46: Gripper done

When "11 Raise" is set we give it 1s to perform the raise operation. This is the last step so we set the dummy bit "PP done."



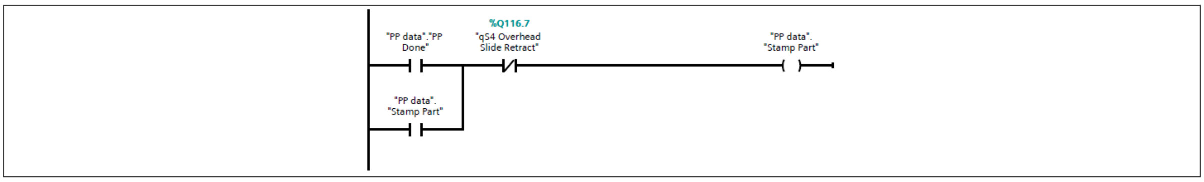
Network 47: Reset Gripper done bit

After the "PP Done" bit is set, a timer of 100ms is started to allow the full completion of the Gripper Robot operation. Once the timer is done the "PP Done" bit is turned off and reset in the rung above.



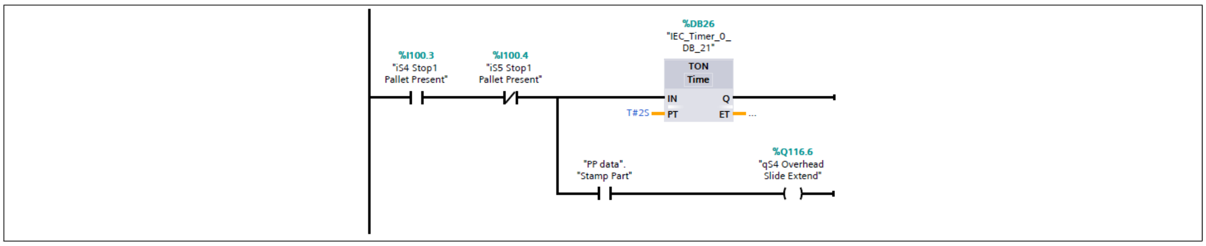
Network 48: Initialize Stamp

If "PP done" is set then we set a dummy bit "Stamp Part." This allows us to add an extra quality check to stamp the part after the dimensional check at the Gripper is done.



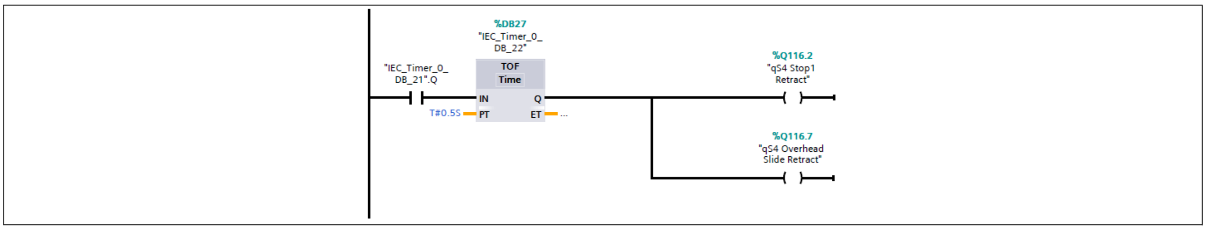
Network 49: Conveyor 4 stop 1 - Stamp station

Once there is a part on conveyor 4 stop 1 and no part on conveyor 5 stop 1 we start a timer to allow some time to pass before it releases the part and keep the stopper retracted for some time.
If the bit "Stamp Part" is set then we extend the overhead slide to perform the stamp operation.



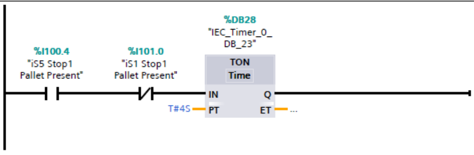
Network 50: Conveyor 4 stop 1 - Stamp station

This is a continuation of the previous rung. Once the timers are done the stopper at conveyor 4 stop 1 is retracted. This also allows the overhead slide to retract.



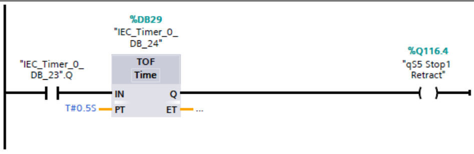
Network 51: Conveyor 5 stop 1

Once there is a part on conveyor 5 stop 1 and no part on conveyor 1 stop 1 we start a timer to allow some time to pass before it releases the part and keep the stopper retracted for some time.



Network 52: Conveyor 5 stop 1

This is a continuation of the previous rung. Once the timers are done the stopper at conveyor 5 stop 1 is retracted.



Chapter 3

Software platform

3.1 Introduction

This chapter briefly describes basic terms relating to the Siemens' digital factory environment, which is called Tecnomatix and the other tools used to upgrade the manufacturing process. Tecnomatix Process Simulate standalone [9, 11], Tecnomatix Plant Simulation [10, 12], Siemens NX 12.0 [13], Tia Portal V14 [14] and MATLAB enabled the work's process and are discussed upfront to let the reader understand how the linkages between one another emerged.
Tecnomatix is a comprehensive portfolio of digital manufacturing solutions that deliver innovations by linking all manufacturing disciplines together with product engineering – from process layout and design, process simulation and validation, to manufacturing execution. Tecnomatix is built upon the open Product Lifecycle Management (PLM).
The Product Lifecycle Management [15] provides access to product and process knowledge in the frame of the whole life cycle of a product (conception, design, manufacture, transportation, utilization, disposal and recycling). The PLM is originally based on Computer-aided Design (CAD), Computer-aided Manufacturing (CAM) and Product Data Management (PDM).
Tecnomatix is a part of the Siemens PLM Platform and it is categorized into groups such as:

- Robotics and automation planning – it is exploited in Process Simulate robotics (robotic production process, ect.).
- Plant design and optimization – it is used in Plant Simulation (optimization of product processes, ect.).

Figure 0-1 Sotware platform layout

3.2 Process Simulate

Process Simulate standalone provides an integration of the eMServer and its planning tools with the 3D environment. Main components of Process Simulate are the all the data of the Object tree (Resources, Parts, Appearances, ect.) and Operation tree (Figure 3-2). Object tree contains a hierarchy of objects in the study. Operation tree contains only operation structure and operations of objects in the study. Operations are activated in Sequence Editor or Path Editor. Sequence Editor serves for an overall simulation, whereas Path Editor is used to tune up a given operation, especially robotic operations and programs.
Process Simulate has many tools and settings which are used during examination of this work. Here it has been made an overview of the most important methods and tools from the point of view of creating this work. These tools are presented in the main menu bar of Process Simulate.

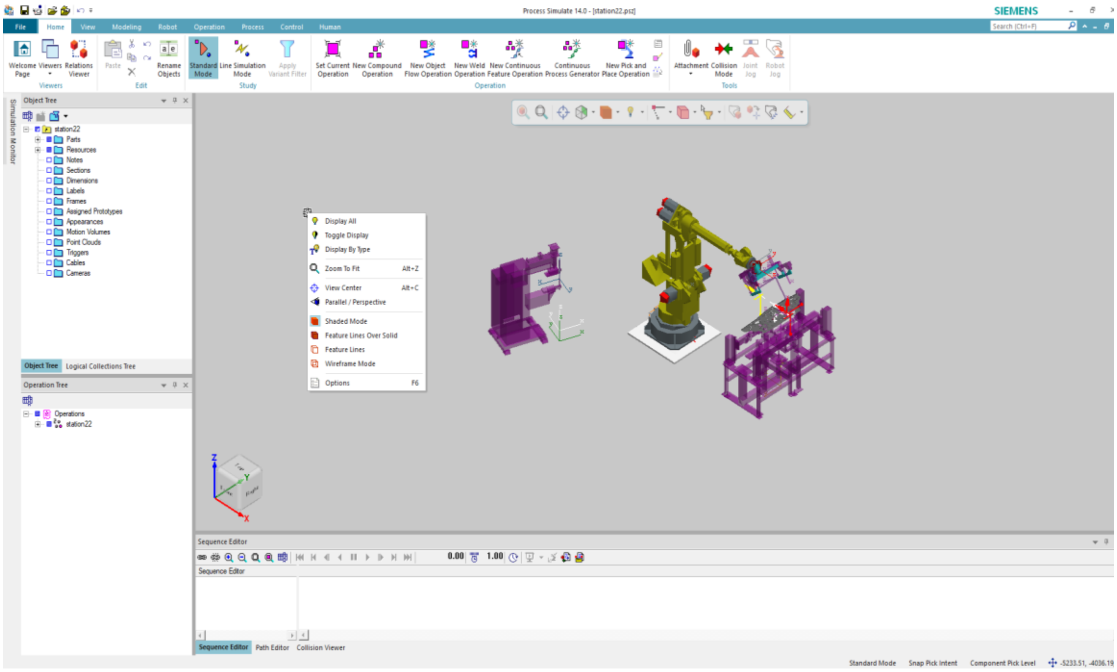


Figure 0-2 Basic Process Simulate environment from Learning Advantage

When building a model in Process Simulate there are several ways to do it. There are two main concepts regarding a simulation model; event-based and time-based. Event-based simulation is when the model only considers which signals that are true and false before performing an operation. Time-based simulation on the other hand does not consider signals; instead, it considers which operations that have happened. As can be imagined these two methods can be combined in many ways. To create an event-based simulation in a more time-based way you can have the operations setting signals when performed which then can trigger the next operation to happen. Process Simulate can handle both concepts and the fact that they can be combined like this makes the possibility for different solutions; almost infinite. This implies that there has to be a well-structured working order to be able to work with the model.

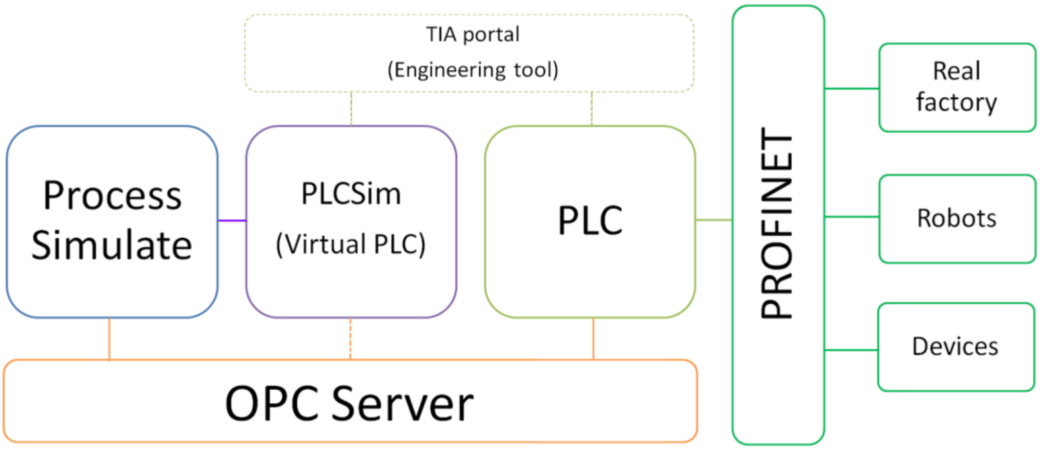


Figure 0-3

Process Simulate standalone is the main core of the work as far as it concerns the software platforms. Therefore, I have illustrated the basic knowledge of simulations I learnt form the Siemens' self-learning tool, named from now on Learning Advantage (LA).
Tecnomatix offers a suite of engineering study tools known collectively as Process Simulate. This suite includes Process Simulate Human, Process Simulate Robotics, Process Simulate Assembler (Flow Paths), and more.
Process Simulate is a dynamic environment that facilities concept verification, as well as assembly and serviceability studies, by enabling to conveniently perform these tasks:

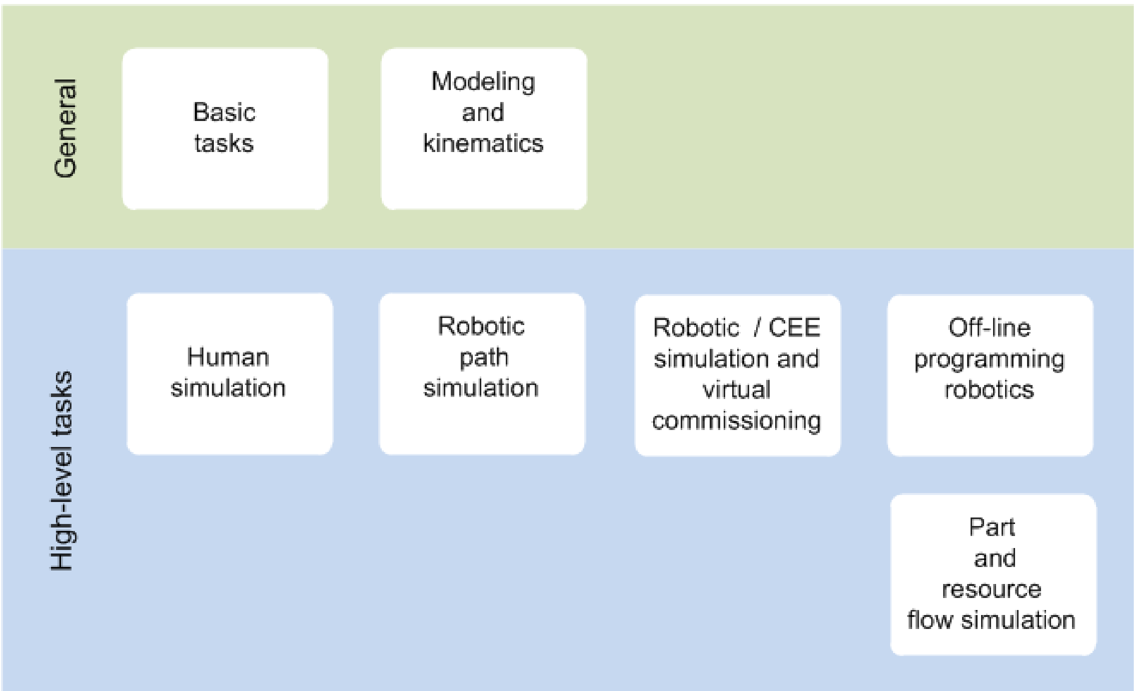


Figure 0-4 Process Simulate general course's tasks

Here are a few examples why you may want to use these parts of Process Simulate:

Basic tasks including modeling and kinematics

- Intuitive and native Windows environment tool
- Navigate your data and play simulations
- Layout objects in a study and setup how you visualize them
- Create or modify the components needed to perform a study (for example 3D modeling of kinematics for tooling and robots)
- Check collisions and clearances dynamically between simulated objects
- Create videos, Figures, and labels

Human simulation

- Perform human reach checks and ergonomic studies to desired situations.
- Develop human simulations
- Check collisions and clearances dynamically for tools, robot arms and the human hand.
- Perform serviceability studies of the assembly.
- Determine how to service a specified part of the assembly.

Robotic path simulation, PLC simulation, and off-line programming

- Dynamic 3D robot simulation
- Perform robotic reach checks to desired locations.
- Develop and download robotic processes and paths (including logic)
- Model PLC logic and signals as well as connect to existing PLCs
- Support for industrial robots from a vast array of vendors (such as ABB, Comau, Duerr, Fanuc, IGM, Kawasaki, Kuka, Nachi, NC, Reis, Staubli, Trallfa, and Yaskawa)
- Accurate cycle time calculation using realistic robot simulation (RRS)
- Upload robot programs from the shop floor
- Customizable robot specific abilities

Part and resource flow simulation

- Verify the feasibility of a product assembly.
- Develop a path for assembly and the disassembly of parts.

What is a simulation? Simulation is a very general term that can mean something different to different people, depending on your background.

Figure 0-5 Wooden mechanical horse simulator from World War I

In general, Simulation is the imitation of some real thing or process, it usually entails representing certain key characteristics or behaviors of a system. Instead, a Computer Simulation is an attempt to model a real-life or hypothetical situation on a computer so that you can study it and see how it works. Just about anything can be simulated on a computer, but usually there is a question that needs to be answered (or results verified). In Process Simulate standalone you can **create assembly process verification simulations**. Many other types of simulation are not performed (for example rain, corrosion, crane force loading, ect.).

Typical Process Simulate inputs:

1. Product

Parts are linked to a process in the plant BOP (bill of process)

Figure 0-6 Idea of product in Process Simulate course

2. Process

The plant BOP contains the sequence of process areas. Also, each process contains a sequence of operations needed to make the product in the specified plant with the specified resources.

Figure 0-7 Idea of process in Process Simulate course

3. Plant

Resources and work areas are linked to a process in the plant.

Figure 0-8 ideas of plants in Process Simulate courses

Typical Process Simulate outputs:

1. Sequencing product assembly/disassembly
2. Robotic reachability, cycle time, and controls logic
3. Human reachability, ergonomics, and standard time

And more:

- An entire product process structure properly sequenced for collision free assembly or disassembly.
- Ergonomic analysis and reports
- Time analysis

- Robotic path analysis
- Collision analysis
- Robotic programs
- Reach envelopes
- Robot controller interaction

3.3 Plant Simulation

Plant Simulation is software for integrated, graphic and *object-oriented* modelling, discrete-event time simulation, and animation. Many complex systems may be modelled and displayed in great detail closely resembling reality. The current Conveyor System's was a Plant Simulation's study and it has been imported into Process Simulate standalone for our project.

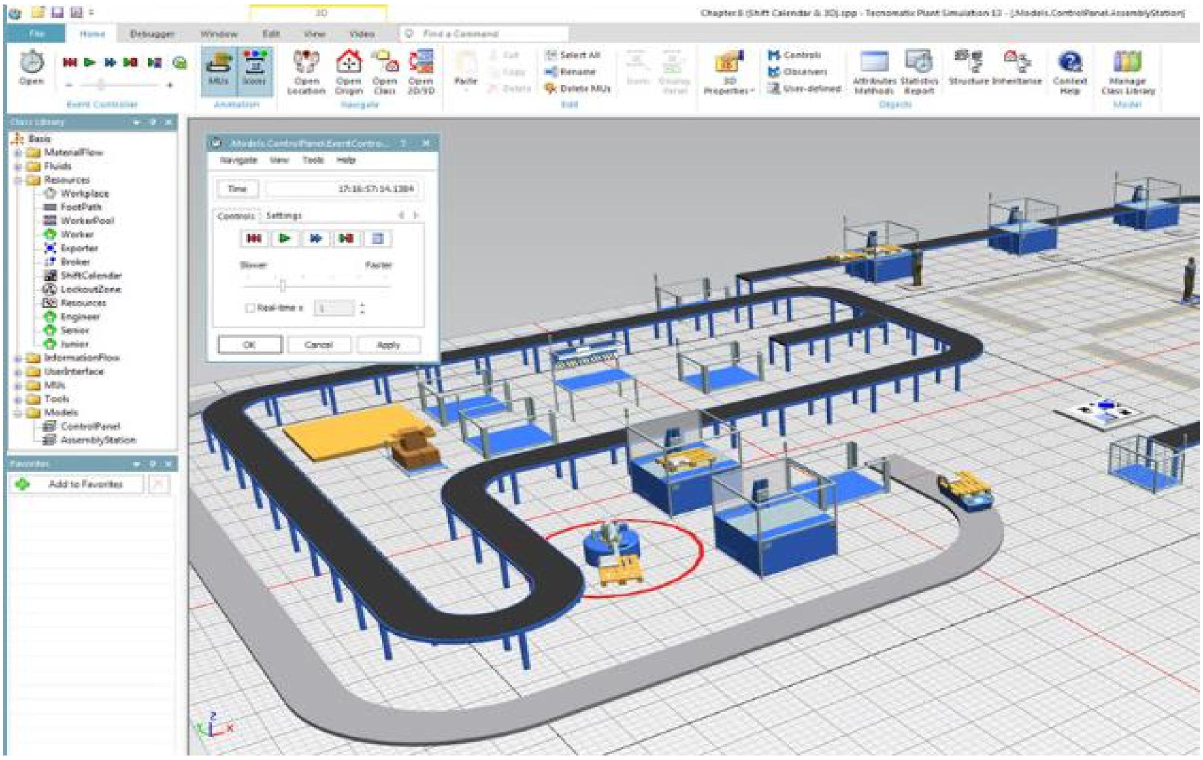


Figure 0-9 Plant Simulation characteristic environment

In the real world, time passes continuously. For instance, when watching a part move along a conveyor system, you will detect no leaps in time. The time the part takes to cover the system is continuous, such that the curve for the distance covered is a straight line.
A discrete event simulation (DES) program on the other hand only takes into consideration those points in time (events) that are of importance to the further course of the simulation. Such events may, for example, be a part entering a station, leaving it, or moving on to another machine. Any movements in between those events are of little interest to the simulation.
Plant Simulation uses DES. One major advantage of DES over time-oriented simulation (continuous or time-step simulation as it is in Process Simulate) is performance. Since the program can simply skip all the moments in time that are not of interest, it is possible to simulate years of factory operation in just minutes. That is particularly useful when you want to simulate different configurations of the same system. and make several replications for each configuration. Plant Simulation has built-in functionalities for exactly that purpose. DES tracks the state changes in the model components at the time the changes occur. Unlike continual simulation where the clock runs in a continuous manner, the clock in discrete-event simulation jumps from one event to the next

scheduled event. Events can schedule other events such as a part entering a machine, which schedules an event for the same part to leave the machine.
Plant Simulation is completely object-oriented. Understanding the basic principles of object orientated programming enables you to model large, complex systems in an organized and maintainable way.

Classes, Attributes, and Instances

As an example, suppose we want to model the patients in a system. The relevant properties of a patient are the patient's age, appointment time, and gender. We do not care about the exact age, appointment time and gender yet, but we only recognize that the model of a patient should have these properties. In *object-oriented* design terms, we have the *class Patient*, which has the *attributes age*, *appointmentTime*, and *gender*.
The class *Patient* does not represent individual patients yet, but it rather describes the properties of all patients. To get individual patients, we *instantiate* from the class *Patient*. Individual patients are now called *instances* of the class *Patient*. Each instance will have the same attributes (*age*, *appointmentTime,* and *gender*), but the values of those attributes can differ from instance to instance.

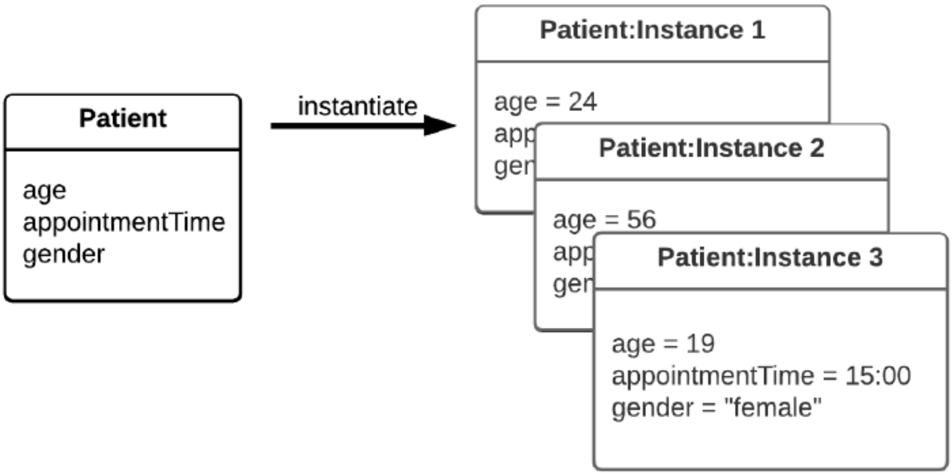


Figure 0-10 Plant Simulation tasks' idea

For example, you can use Plant Simulation to simulate certain aspects of:

- A call center
- A hospital
- An airport
- A shipyard
- A manufacturing plant
- Etc.

To plan a new facility
- Determine and optimize the times and the throughput
- Determine the dimensioning
- Determine the limits of performance
- Investigate the influence of failures
- Determine manpower requirements
- Gain knowledge about the facility behavior

- Determine suitable control strategies
- Evaluate different alternatives

To optimize an existing facility

- Optimize control strategies
- Optimize the sequence of orders
- Test the daily proceedings

To execute a plan

- Provide a template for creating the control strategies
- Test different scenarios during the facility warm-up phase
- Train the machine operators in the different facility states

3.4 Siemens NX 12.0

NX 12.0 is Siemens' CAD product to make design changes on the system before implementing the robotics setups into the Process Simulate environment. Siemens NX 12 software is a flexible and powerful integrated solution that helps you deliver better products faster and more efficiently. NX delivers the next generation of design, simulation, and manufacturing solutions that enable companies to realize the value of the digital twin.
Supporting every aspect of product development, from concept design known as "UG", is used among other tasks for: through engineering and manufacturing, NX gives you an integrated toolset that coordinates disciplines, preserves data integrity and design intent, and streamlines the entire process. NX, formerly known as "UG", is used to:

- Design: parametric and direct solid/surface modelling
- Engineering analysis: static; dynamic; electro-magnetic; thermal, using the finite element method; and fluid, using the finite volume method
- Manufacturing finished design by using included machining modules

3.5 TIA Portal v14

The Totally Integrated Portal enables complete access to the entire digitalized automation, from digital planning and integrated engineering to transparent operation. As part of the Digital Enterprise Software Suite, it joins PLM and MES in rounding out the comprehensive offering from Siemens for companies on the path to Industry 4.0, making it the perfect access to automation in the Digital Enterprise.
With TIA Portal and the possibilities offered by the new Version V14, machine manufactures and system integrators as well as plants operators benefit from:

- Shorter time to market thanks to innovative simulation tools, among other things
- Increased plant productivity thanks to additional diagnostic and energy management functions
- Greater flexibility thanks to coordinated teamwork

As it has already shown in the previous chapter, we worked on TIA Portal v14 to update the Ladder logic program that has been used for the new application

.

Chapter 4

System's robots

In this chapter, the robots that have been picked for the system's update are presented. During a brainstorming phase, it has been useful the assistance of AMT (Applied Manufacturing Technologies) which is a spread brand in the manufacturing field in the United States of America. By analyzing the Conveyor System's physical model and choosing an Industry 4.0's application that barely requires no product to be chosen for the process, the engineers' team from AMT we had got in touch with suggested the following 6-axis robots' models and located the most suitable space for the robots' operations in the system. It was my duty to contact ABB and KUKA, the selected robots' companies. Both companies were extremely hands-on and provided the Process Simulate models in behalf of our research. Therefore, a brief description for each model is presented as follows and all of them have been inserted into the Process Simulate environment when I planned the simulations.

4.1 Introduction to 6-axis robot

Industrial robots have various axis configurations. Most articulated robots, however, feature six axes, also called six degrees of freedom. Six axis robots allow for greater flexibility and can perform a wider variety of applications than robots with fewer axes [16].

<u>Axis 1</u>
This axis, located at the robot base, allows the robot to rotate from left to right. This sweeping motion extends the work area to include the area on either side and behind the arm. This axis allows the robot to spin up to a full 180 degree range from the center point.
<u>Axis 2</u>
This axis allows the lower arm of the robot to extend forward and backward. It is the axis powering the movement of the entire lower arm.
<u>Axis 3</u>
The axis extends the robot's vertical reach. It allows the upper arm to raise and lower. On some articulated models, it allows the upper arm to reach behind the body, further expanding the work envelope. This axis gives the upper arm the better part access.
<u>Axis 4</u>
Working in conjunction with the axis 5, this axis aids in the positioning of the end effector and manipulation of the part. Known as the wrist roll, it rotates the upper arm in a circular motion moving parts between horizontal to vertical orientations.
<u>Axis 5</u>
This axis allows the wrist of the robot arm to tilt up and down. This axis is responsible for the pitch and yaw motion. The pitch, or bend, motion is up and down, much like opening and closing a box lid. Yaw moves left and right, like a door on hinges.

Axis 6
This is the wrist of the robot arm. It is responsible for a twisting motion, allowing it to rotate freely in a circular motion, both to position end effectors and to manipulate parts. It is usually capable of more than a 360-degree rotation in either a clockwise or counterclockwise direction.

4.2 IRB 1200 5/0.9

Figure 0-1 IRB 1200

The IRB 1200 is one of ABB Robotics latest generation of 6-axis, with payload of 5 to 7 kg, designed specifically for manufacturing industries that use flexible robot-based automation, e.g. 3C industry. The robot has an open structure that is especially adapted for flexible use and can communicate extensively with external systems [17, 18].

The robot is equipped with the IRC5 Compact (IRC5C) or IRC5 (Single cabinet) controller and robot control software, RobotWare. RobotWare supports every aspect of the robot system, such as motion control, development and execution of application programs and communication.

The IRB 1200 is available in two versions and both can be mounted on floor, inverted or on wall in any angle (around X-axis or Y-axis).

The chosen robot type is the IRB 1200 5/0.9 and its most common characteristics are:

- Handling capacity (kg) = 5 kg
- Reach (m) = 0.9 m

Other technical data:

- Weight (kg) = 54 kg
- Airborne noise level (dB) < 70 dB
- Power consumption (kW) = 0.45 kW
- Pose repeatability (mm) = ± 0.025 mm

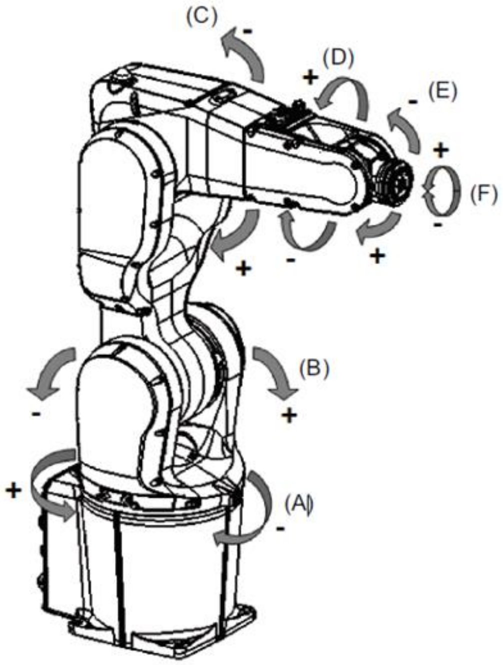


Figure 0-2 Manipulator axes

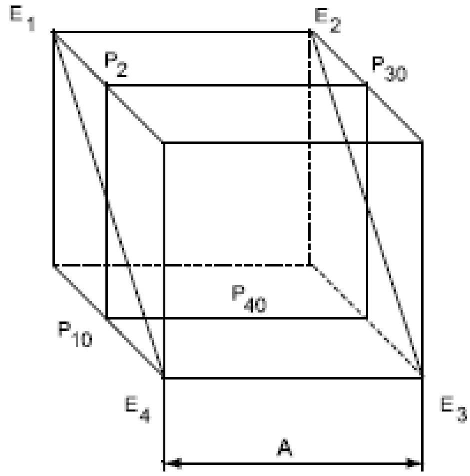


Figure 0-3 Working space with A = 250 mm

Robot motion:

Location of motion	Type of motion	Range of movement	Axis max speed
Axis 1	Rotation motion	+170° to -170°	288°/s
Axis 2	Arm motion	+130° to -100°	240°/s
Axis 3	Arm motion	+70° to -200°	300°/s
Axis 4	Wrist motion	+270° to -270°	400°/s
Axis 5	Bend motion	+130° to -130°	405°/s
Axis 6	Turn motion	Default: +400° to -400° Maximum revolution: ±242	600°/s

Table 0-1 Robot motion with resolution approximately 0.01° on each axis

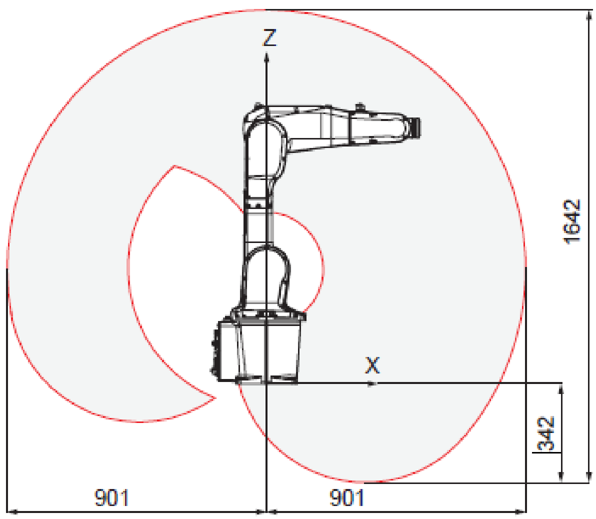


Figure 0-4 IRB 1200 work volume

4.3 IRB 1600 10/1.2

It literally is a robot family with five version and new possibilities open up with ABB's 1600 robot. The IRB family is deal for Arc Welding, Machine Tending, Material Handling, Gluing and During/Grinding applications [19, 20].

Figure 0-5 IRB 1600

The robot is equipped with the IRC5 controller and robot control software, RobotWare. RobotWare supports every aspect of the robot system, such as motion control, development and execution of application programs and communication.

The IRB 1600 is available in two versions and both can be mounted on floor, inverted or on wall in any angle (around X-axis or Y-axis).

The chosen robot type is the IRB 1600 10/1.2 and its most common characteristics are:

- Handling capacity (kg) = 10 kg
- Reach (m) = 1.2 m

Other technical data:

- Weight (kg) = 250 kg
- Airborne noise level (dB) < 70 dB
- Power consumption (kW) = 0.58 kW
- Pose repeatability (mm) = ± 0.02 mm

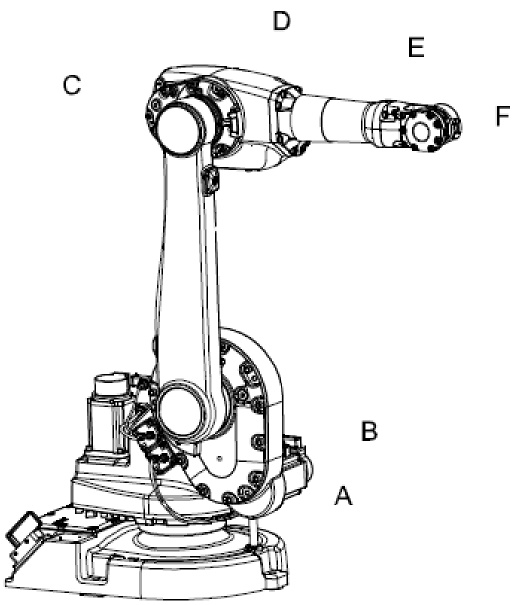


Figure 0-6 Manipulator axes

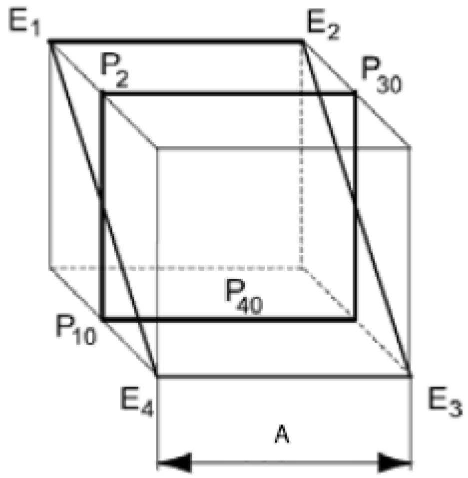


Figure 0-7 Working space with A = 400 mm

Location of motion	Type of motion	Range of movement	Axis max speed
Axis 1	Rotation motion	+180° to -180°	180°/s
Axis 2	Arm motion	+136° to -63°	180°/s
Axis 3	Arm motion	+55° to -235°	185°/s
Axis 4	Rotation motion	Default: +200° to -200° Max. rev: +190° to -190°	385°/s
Axis 5	Bend motion	+115° to -115°	400°/s
Axis 6	Turn motion	Default: +400° to -400° Maximum revolution: +288 to - 288	460°/s

Table 0-2 Robot motion

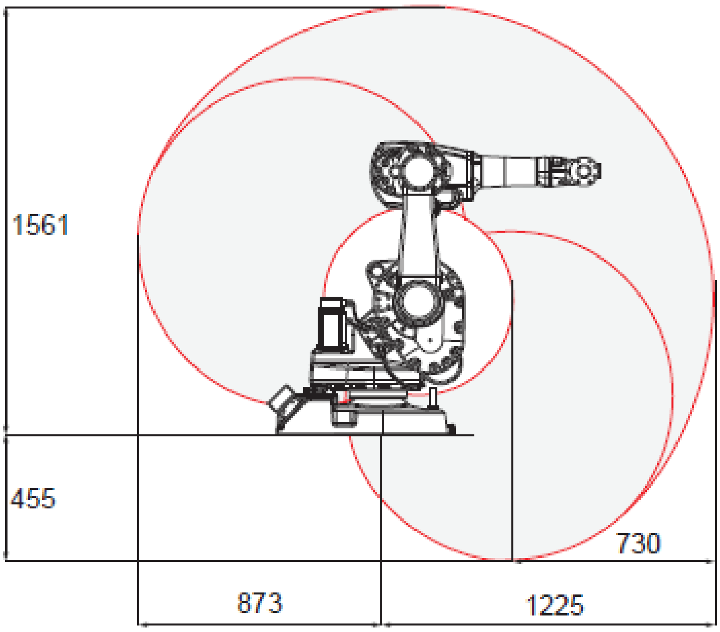


Figure 0-8 IRB 1600 work volume

4.4 KR 6 R700

KUKA robot systems comprise all the assemblies of an industrial robot, including the manipulator (mechanical system and electrical installations), control cabinet, connecting cable, end effector (tool) and other equipment. The KR AGIULS product family consists of the following types:

- KR 6 R700 (in this project)
- KR 6 R900
- KR 10 R900
- KR 10 R1100 (in this project)

Figure 0-9 KR 6

An industrial robot of this type comprises as components: manipulator, Robot controller, smartPAD teach pendant, connecting cables, software and accessories [21, 23].

The chosen robot type is the KR 6 R700 and its most common characteristics are:

- Handling capacity (kg) = 6 kg
- Reach (m) = 0.7067 m

Other technical data:

- Weight (kg) = 50 kg
- Airborne noise level (dB) < 70 dB (A) outside the working envelope
- Pose repeatability (mm) = ± 0.03 mm

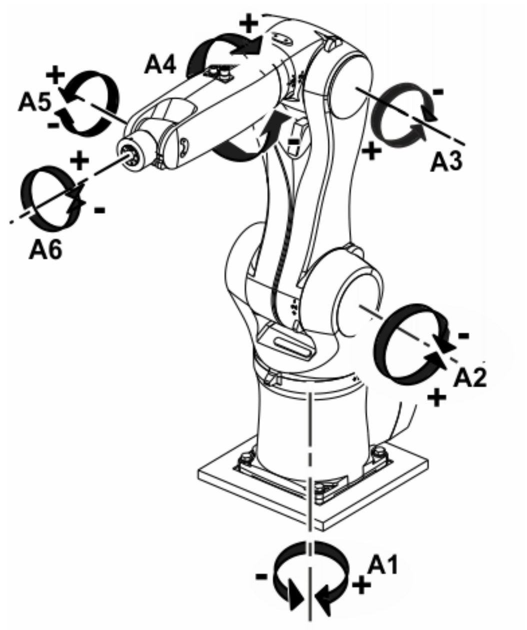


Figure 0-10 Manipulator axes

Location of motion	Type of motion	Range of movement	Axis max speed
Axis 1	Rotation motion	+170° to -170°	360°/s
Axis 2	Arm motion	+45° to -190°	300°/s
Axis 3	Arm motion	+156° to -120°	360°/s
Axis 4	Wrist motion	+185° to -185°	381°/s
Axis 5	Bend motion	+120° to -120°	388°/s
Axis 6	Turn motion	+350° to -350°	615°/s

Table 0-3 Robot motion

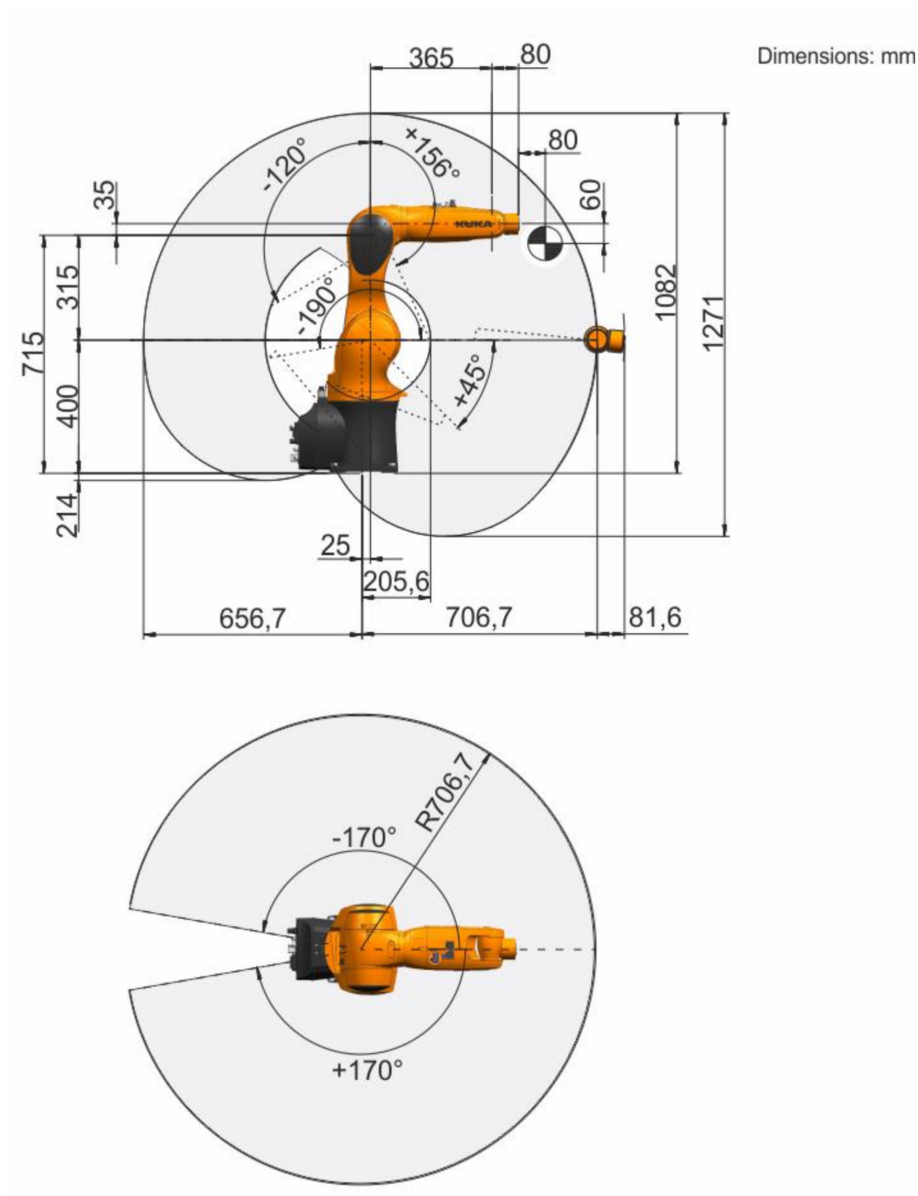


Figure 0-11 KR 6 work volume (two views)

4.5 KR 10 R1100

Figure 0-12 KR 10

KUKA robot systems comprise all the assemblies of an industrial robot, including the manipulator (mechanical system and electrical installations), control cabinet, connecting cable, end effector (tool) and other equipment. The KR AGIULS product family consists of the following types:

- KR 6 R700 (in this project)
- KR 6 R900
- KR 10 R900
- KR 10 R1100 (in this project)

An industrial robot of this type comprises as components: manipulator, Robot controller, smartPAD teach pendant, connecting cables, software and accessories [22, 23].

The chosen robot type is the KR 10 R1100 and its most common characteristics are:

- Handling capacity (kg) = 10 kg
- Reach (m) = 0.1101 m

Other technical data:

- Weight (kg) = 55 kg
- Airborne noise level (dB) < 70 dB (A) outside the working envelope
- Pose repeatability (mm) = ± 0.03 mm

Location of motion	Type of motion	Range of movement	Axis max speed
Axis 1	Rotation motion	+170° to -170°	300°/s
Axis 2	Arm motion	+45° to -190°	225°/s
Axis 3	Arm motion	+156° to -120°	225°/s
Axis 4	Wrist motion	+185° to -185°	381°/s
Axis 5	Bend motion	+120° to -120°	311°/s
Axis 6	Turn motion	+350° to -350°	492°/s

Table 0-4 Robot motion

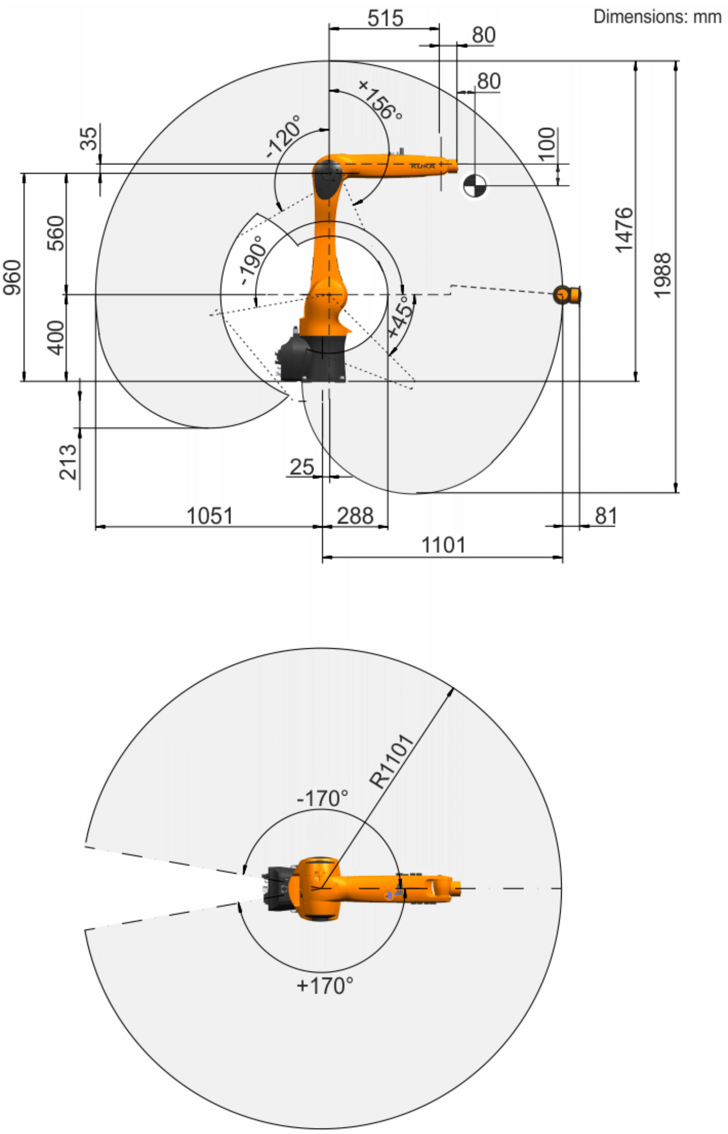


Figure 0-13 KR 10 work volume (two views)

Chapter 5

The Model

5.1 Introduction

The upgrade Conveyor System's model is made by the following main parts:

- The ABB and Kuka Process Simulate robot models
- The Conveyor System model
- The flowing parts models

All of them are described in depth in the chapter's paragraphs.

5.2 The robot models

The robots used are the ones shown in the previous chapter (IRB 1200, IRB 1600, KR 6 R700 and KR 10 R1100). Their specifications have been extensively listed and the simulation models have been defined by describing their kinematics properties. Although, there has been found no tool to analyze the models' dynamics, simply thinking about the lack of information regarding mass' distribution we were not able to compute recursive Newton-Euler robotics approach (balance of forces/torques).

5.2.1 The robot models' kinematics properties

The kinematics is one of the fundamental disciplines in robotics [24]. It does include the geometry of motion but does not include the forces causing the motion. Kinematics properties are needed for the design and control of robot mechanisms, they provide tools for describing:

- Structure and behavior of the robot mechanisms
- Response to actuator movements
- How to coordinate individual actuators to obtain desired robot motion

A robot manipulator is composed of a set of links connected by joints. Each joint can be considered as a single degree-of-freedom (DOF): revolute joint means the DOF is the angle of rotation, while prismatic (sliding) joints equals that DOF is the displacement. Robotics kinematics lead to two main kinematics problems:

- **Forward (direct) kinematics**:

Given the joint position, velocity, acceleration, compute the corresponding variables of the end-effector in a given reference frame (e.g. a Cartesian frame).

Joint Space ⟶ **Work Space**

- **Inverse kinematics:**

Given the joint position, velocity and acceleration of the end-effector, compute the corresponding variables in the joint space.

Work Space ⟶ **Joint Space**

5.2.2 Planar kinematics example

Forward kinematic model

Given: θ_1, θ_2
Compute: *x, y*

$$\varphi = \theta_1 + \theta_2$$

$$x = l_1 \cos\theta_1 + l_2\cos(\theta_1 + \theta_2)$$
$$y = l_1 \sin\theta_1 + l_2\sin(\theta_1 + \theta_2)$$

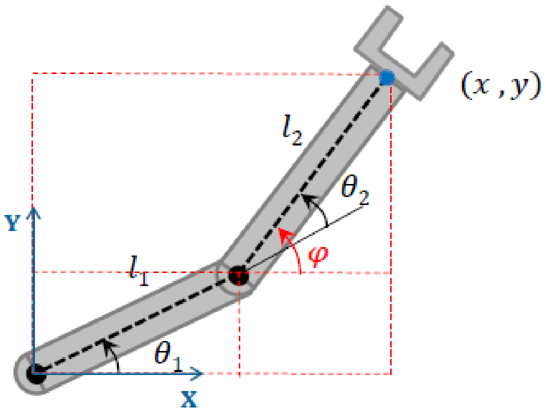


Figure 0-1 Planar kinematics example

This is an easy problem to compute forward kinematic.

However, the solution in a robotics problem is not often so simple, that is why we usually solve forward kinematics models using Homogenous Transforms that comprises the following steps:

- Assign a reference frame to each link
- Describe the relative position/orientation of these frames using homogenous transform matrices:

$^0_1\mathbf{T}, ^1_2\mathbf{T}, ^2_3\mathbf{T}$

- Compute transformation from reference frame to end-effector:

$$^0_3\mathbf{T} = {^0_1\mathbf{T}}\ {^1_2\mathbf{T}}\ {^2_3\mathbf{T}}$$

The homogenous matrices represent the rotation of angles or translation of segments along the axes.

Although, it seems clear that dealing with complicated mechanism means having a systematic method and a unique definition of kinematic model. In literature it is plenty of several conventions for robotics in order to assign frames to links. Within the project, I have briefly described one of them, known as *Denavit-Hartenberg (DH) Convetion,* as it follows:

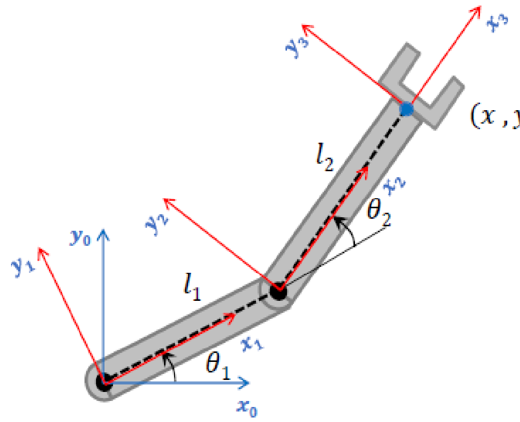


Figure 0-2 Forward kinematics model

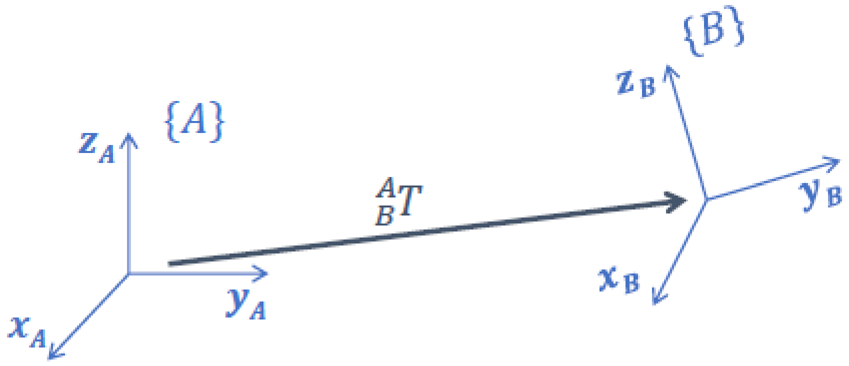


Figure 0-3 Assigning frames to links idea

1. Arbitrary assignment of frames to links
- 6 parameters are needed to describe the position and the orientation of a rigid body in the 3D space
2. Using DH convention
- Only 4 parameters are required, and a unique assignment of frames is made.

DH parameters procedure:

1. Translate {0} along $\boldsymbol{z_0}$ a distance d (intersection point of $\boldsymbol{z_0}$ with the common normal) to obtain {1}

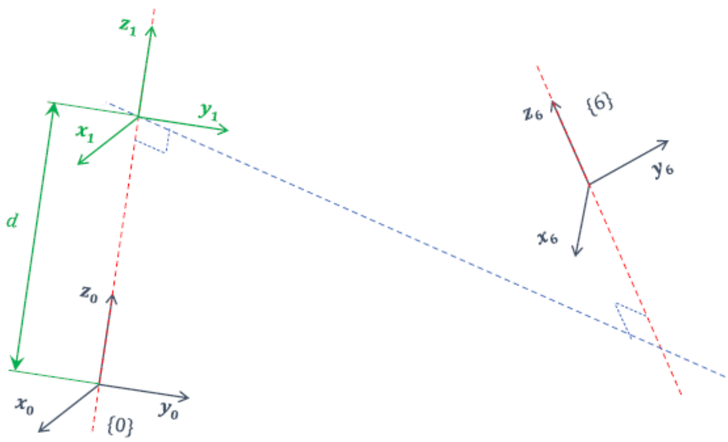


Figure 0-4 DH convetion – step 1

2. Rotate {1} about $\boldsymbol{z_1}$ by an angle θ until $\boldsymbol{x_1}$ is aligned with the common normal to obtain {2}

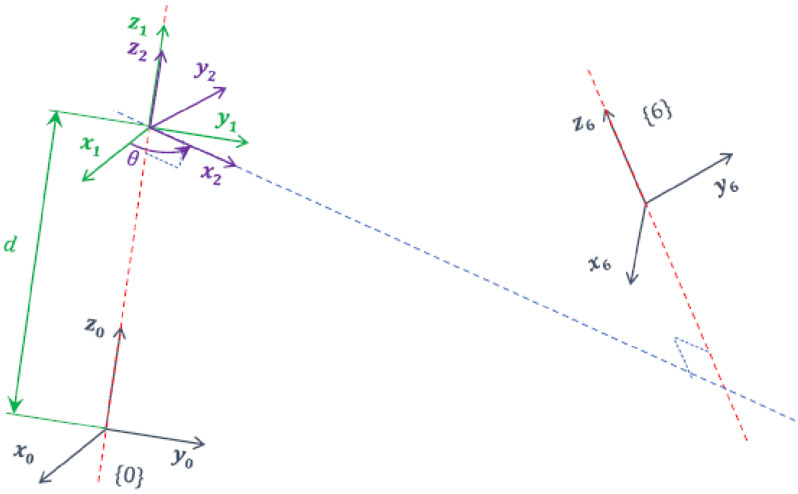


Figure 0-5 DH convetion – step 2

3. Translate {2} along $\boldsymbol{x_2}$ a distance a (length of common normal) to obtain {3}

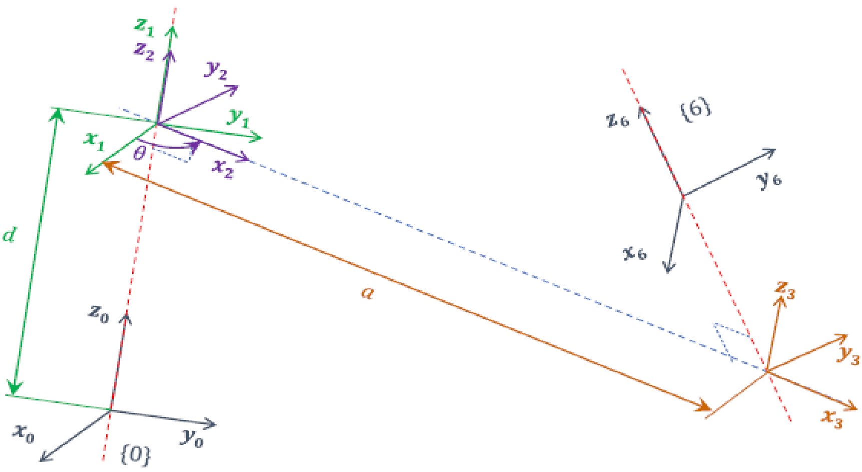


Figure 0-6 DH convetion – step 3

4. Rotate {3} about $\boldsymbol{x_3}$ by an angle α until $\boldsymbol{z_3}$ is aligned with $\boldsymbol{z_6}$ to obtain {4}

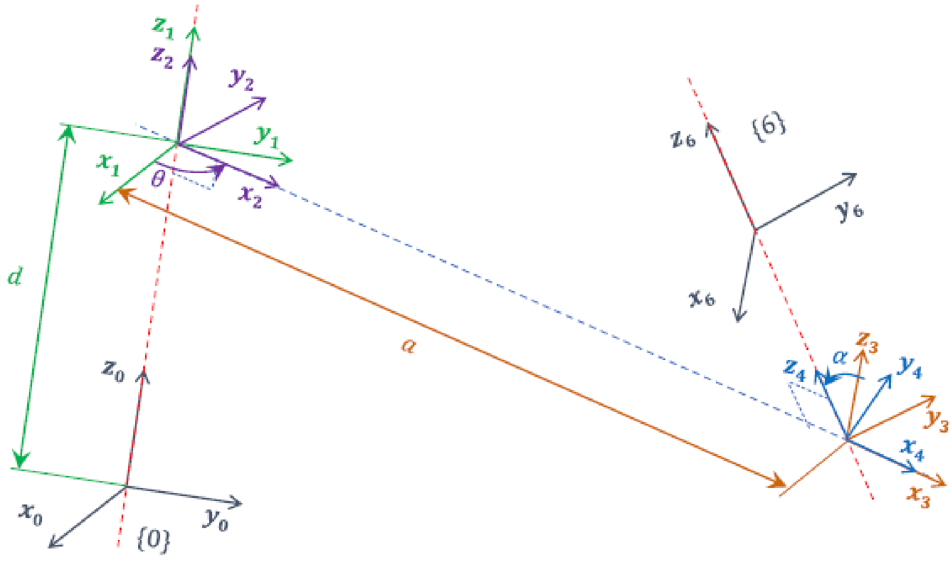


Figure 0-7 DH convetion – step 4

5. Translate {4} along $\boldsymbol{z_4}$ a distance b until {6} is reached to obtain {5}

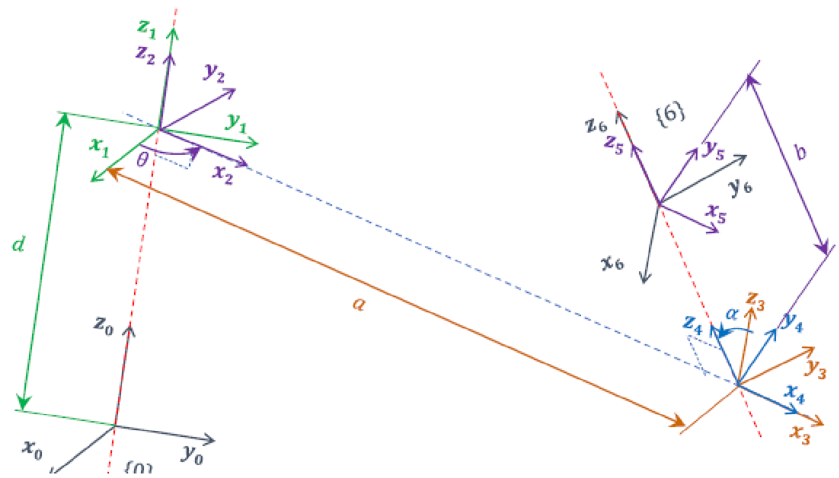


Figure 0-8 DH convetion – step 5

6. Rotate {5} about $\boldsymbol{z_5}$ by an angle φ to obtain {6}

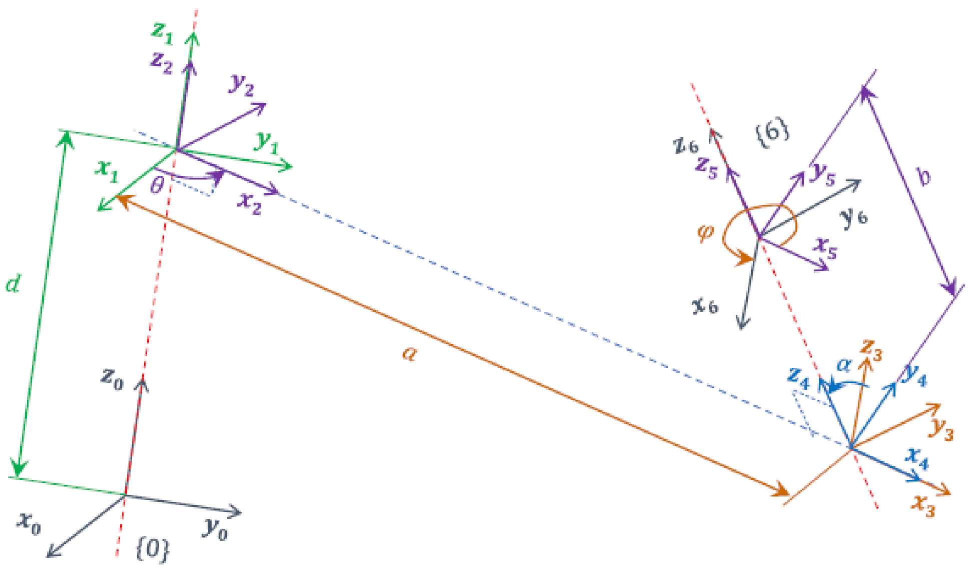


Figure 0-9 DH convetion – step 6

Six transformations (3 translations and 3 rotations) have been employed to move from frame {0} to {6} and six parameters are involved by using the homogenous transforms in the most general case with skew axes (d, θ, a, α, b, φ). Since position and orientation of frame {6} for second axis is not constrained by other considerations, we can choose its location to be coincident with frame {4}, then b, φ are not necessary. The parameters (d, θ, a, α) are known as Denavit-Hartenberg or DH parameters.

$$ {}^{0}_{4}\mathbf{T} = {}^{0}_{1}\mathbf{T}\, {}^{1}_{2}\mathbf{T}\, {}^{2}_{3}\mathbf{T}\, {}^{3}_{4}\mathbf{T} = T(z_0, d)\; R(z_1, \theta)\; T(x_2, a)\; R(x_3, \alpha) $$

It is needed to assign values to the DH parameters as they define each joint angle, joint offset, twist angle and links length. In literature, it is easy to find tables or MALAB codes that automatically compute the homogenous transforms and fully define the link frames in direct kinematics problem by choosing the DH parameters.

In this project, the ABB and KUKA robot models act forward kinematics and into the Process Simulate environment by planning different operations' paths it is possible to optimize the model's kinematics in terms of smoothness and the end-effector final motions in terms of responsiveness of the entire system.

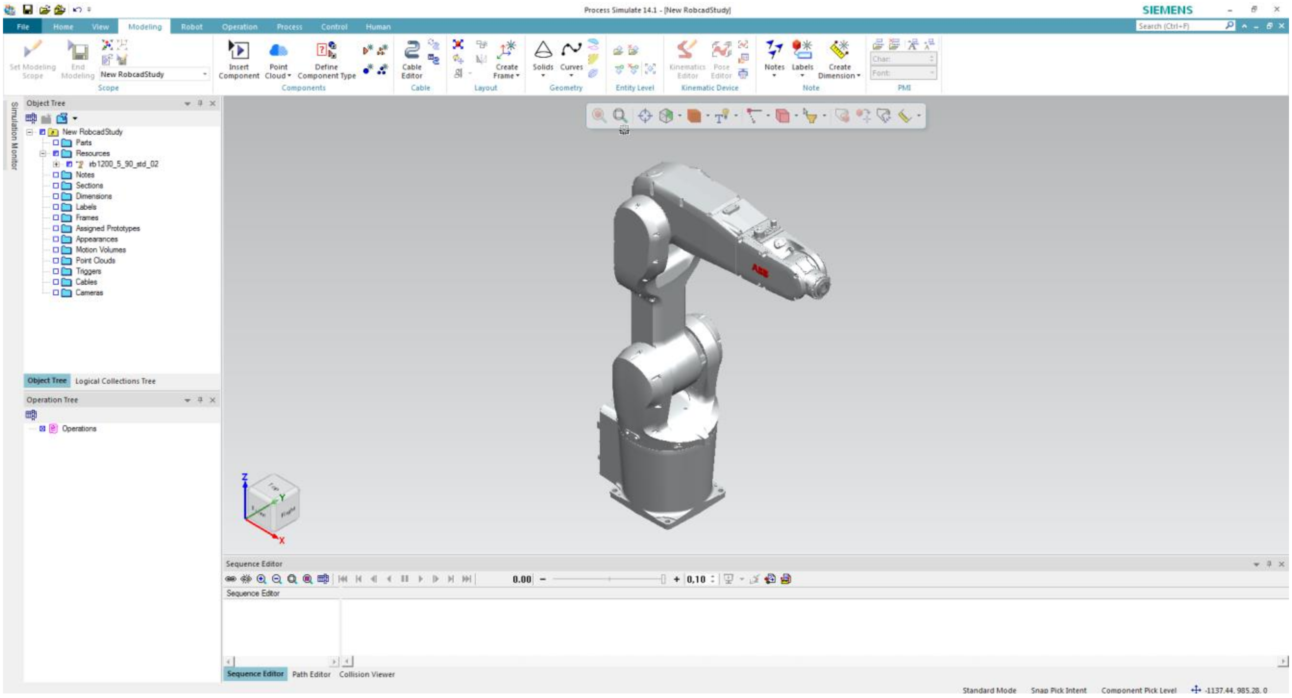


Figure 0-10 IRB 1200 Process Simulate model

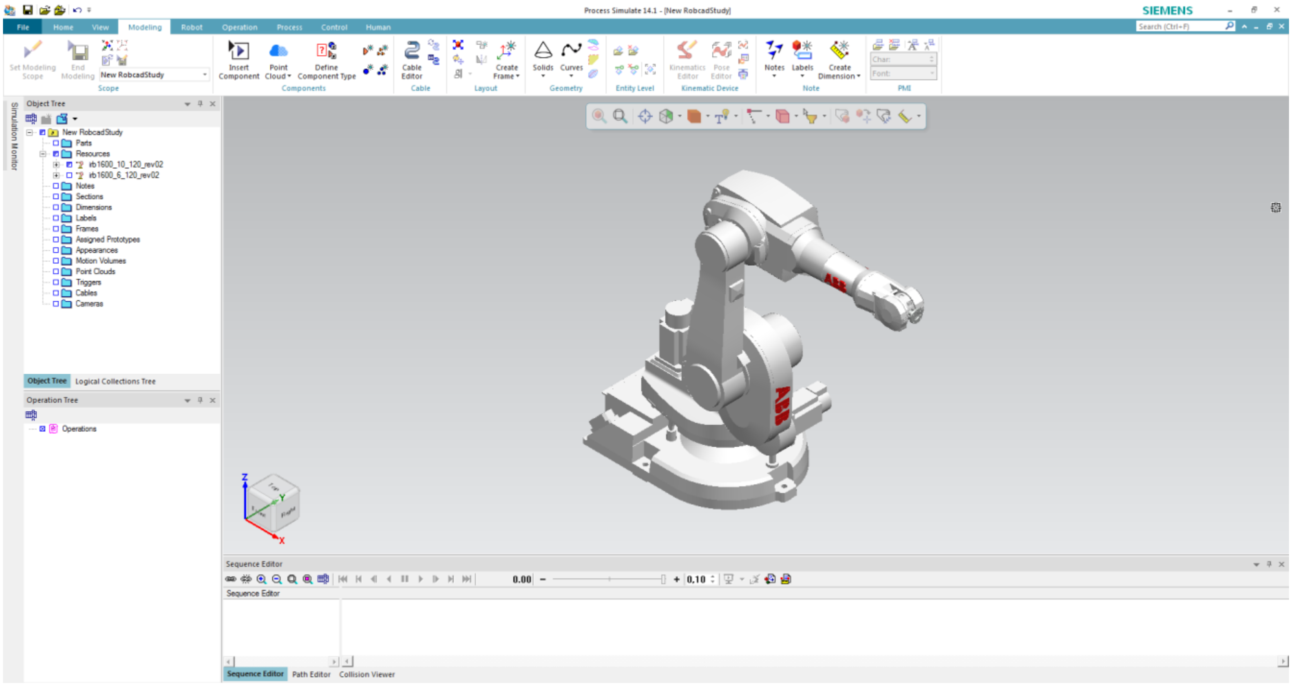


Figure 0-11 IRB 1600 Process Simulate model

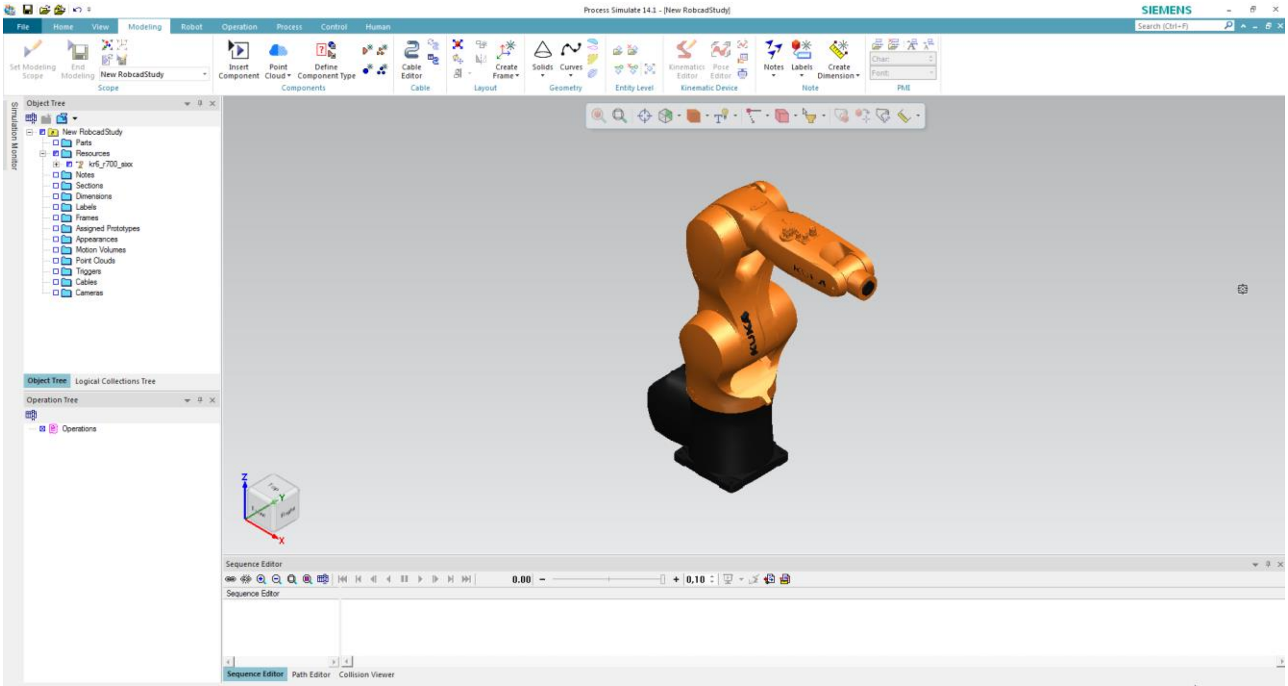


Figure 0-12 KR 6 R700 Process Simulate model

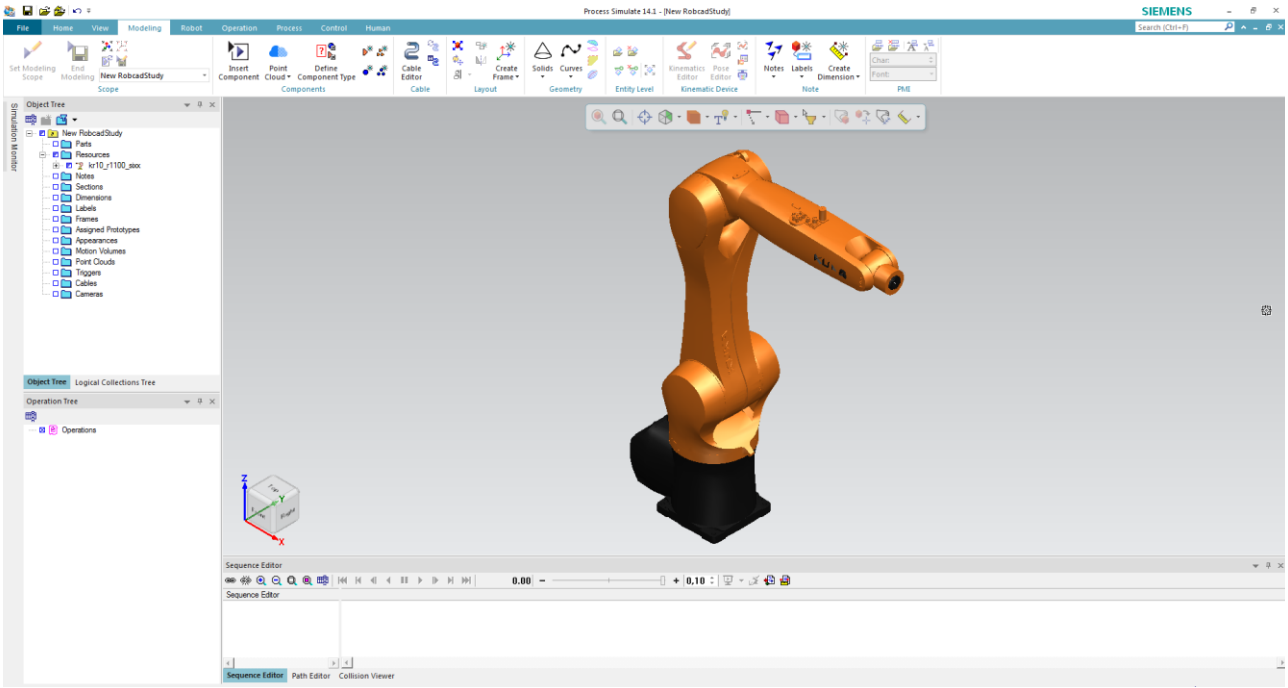


Figure 0-13 KR 10 R1100 Process Simulate model

5.3 The Conveyor System model

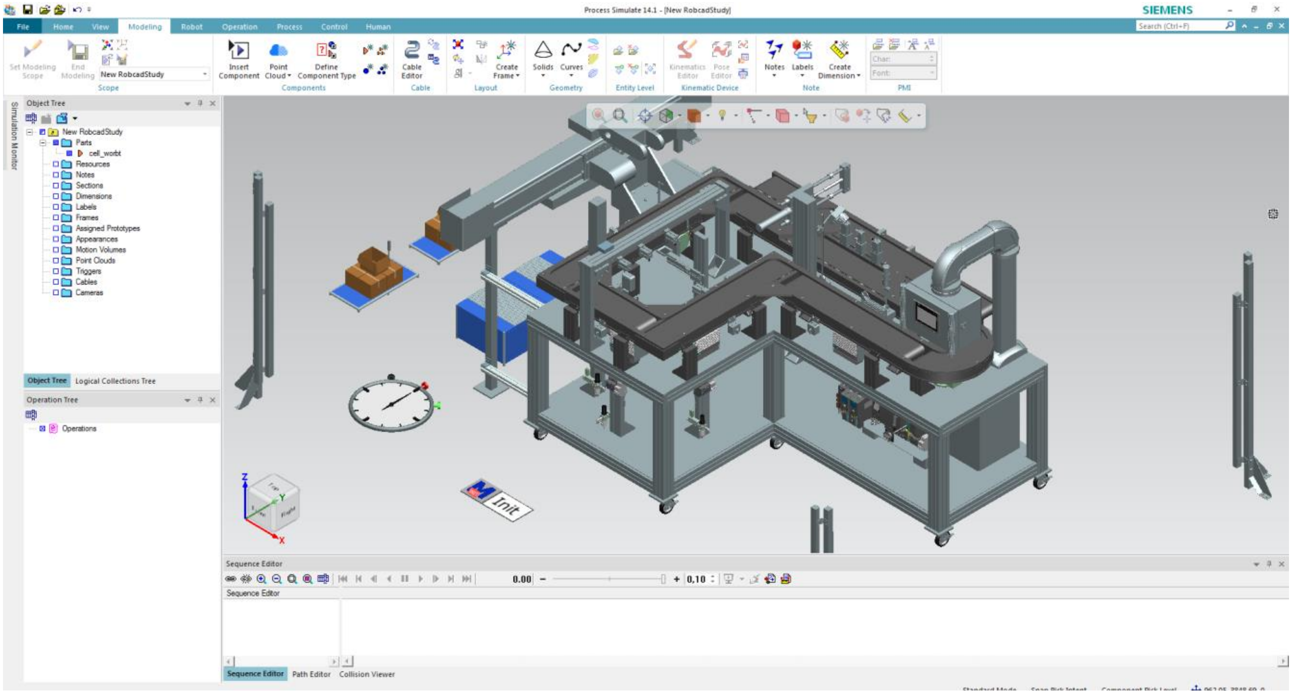


Figure 0-14 Present Conveyor System Process Simulate model once imported

The present Conveyor System model was designed by Siemens PLM in Siemens NX 12 and converted into Plant Simulation for discrete-event simulation. Within this project, it is the only application of the system into the Process Simulate environment so far. After finding the way to import it into Process Simulate and require the new version of NX in Oakland University, the model has been re-modelled in order to create a safe working space for the robots without occurring collisions, creating a "light" version of the system with no sliding mechanisms and pick and place mechanism since the Material Handling have been assigned to the robots as it is said in the next chapter.

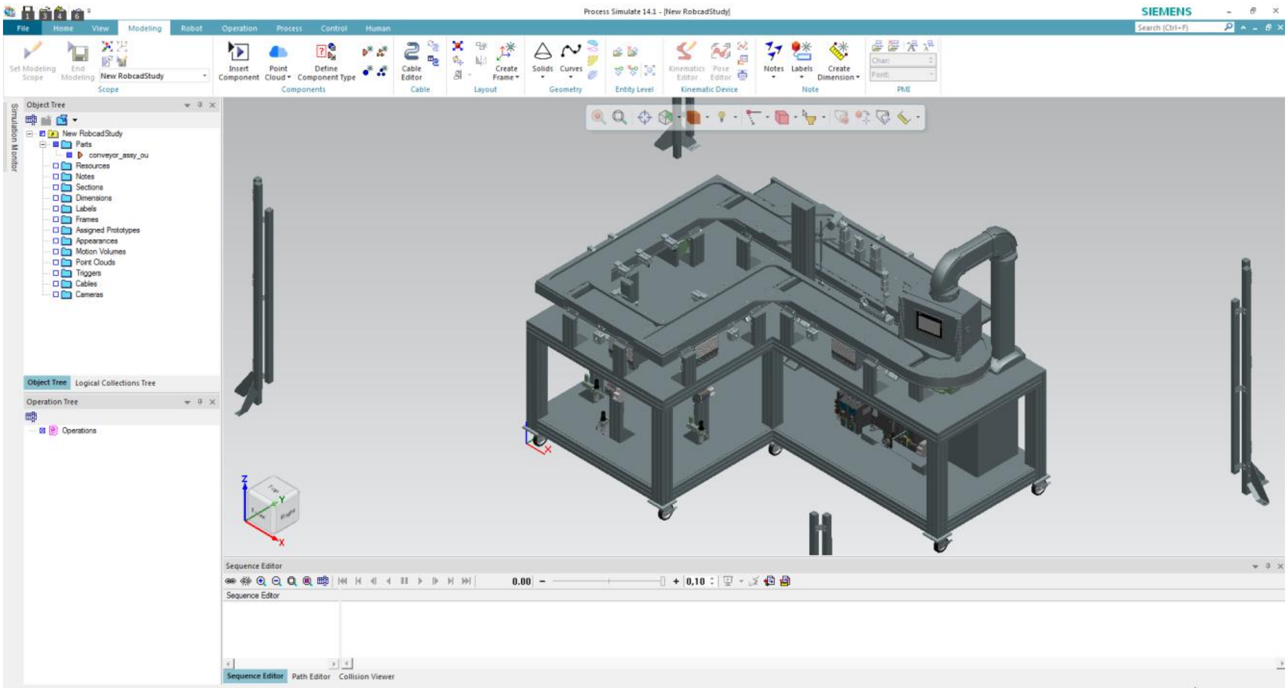


Figure 0-15 Present Conveyor System Process Simulate model after modelling

The new application was thought for a new activities class in Oakland University for the *Industrial Robots and Automation* summer semester's course. In the upgrade model more industrial instruments can be matched together: robot and cell safety, PLCs and cell control, simulation and offline programming, process control, cycle time and vision-based robot control.

5.4 The parts models

As it is shown later, the control logic's idea does not include a fixed assembly process because an Industry 4.0 focus is to lead into a flexible assembly/disassembly thinking of engineering processes. As a result, the parts that flow into the conveyor are ideally supposed to be any, assuming that the robots can recognize whether the next part out of their reaches needs to be handled in that particular moment or not.

AMT engineers supported us by picking a few assemblies the robots could deal with and according to our simulation both ABB and KUKA setups' robots mount the same gear into a gear bottom that are shown in the next figures:

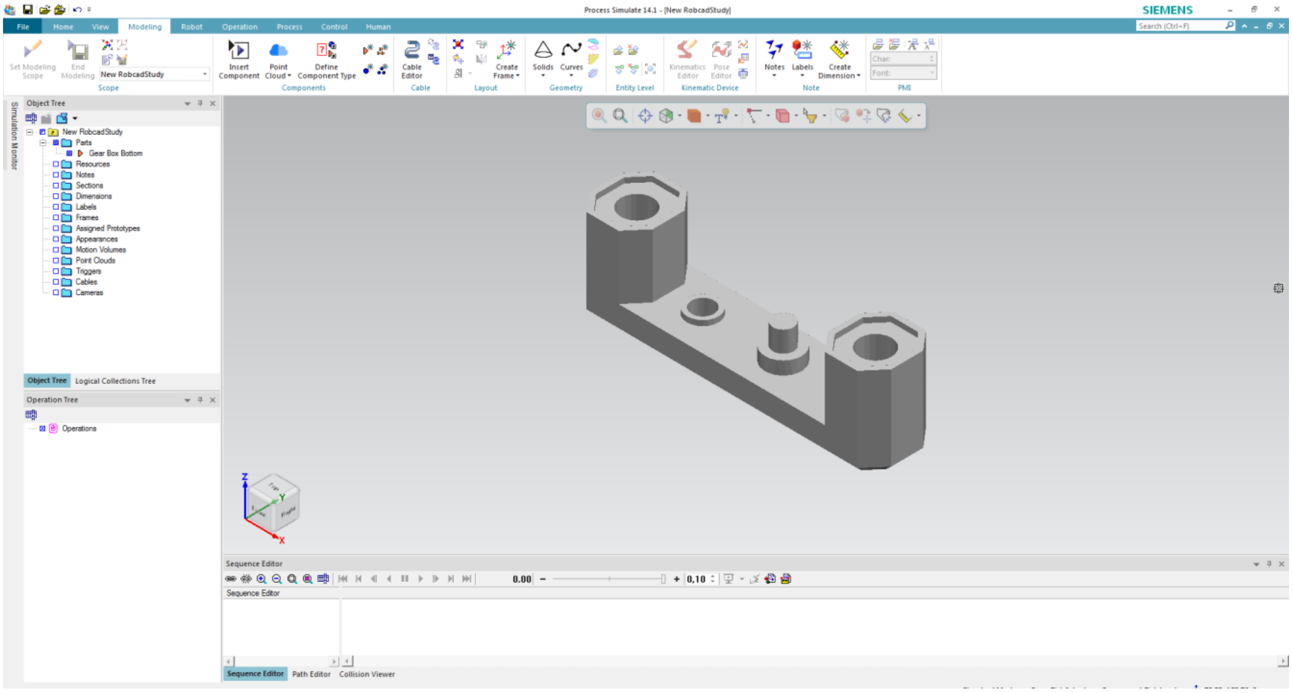


Figure 0-16 The gear bottom Process Simulate model

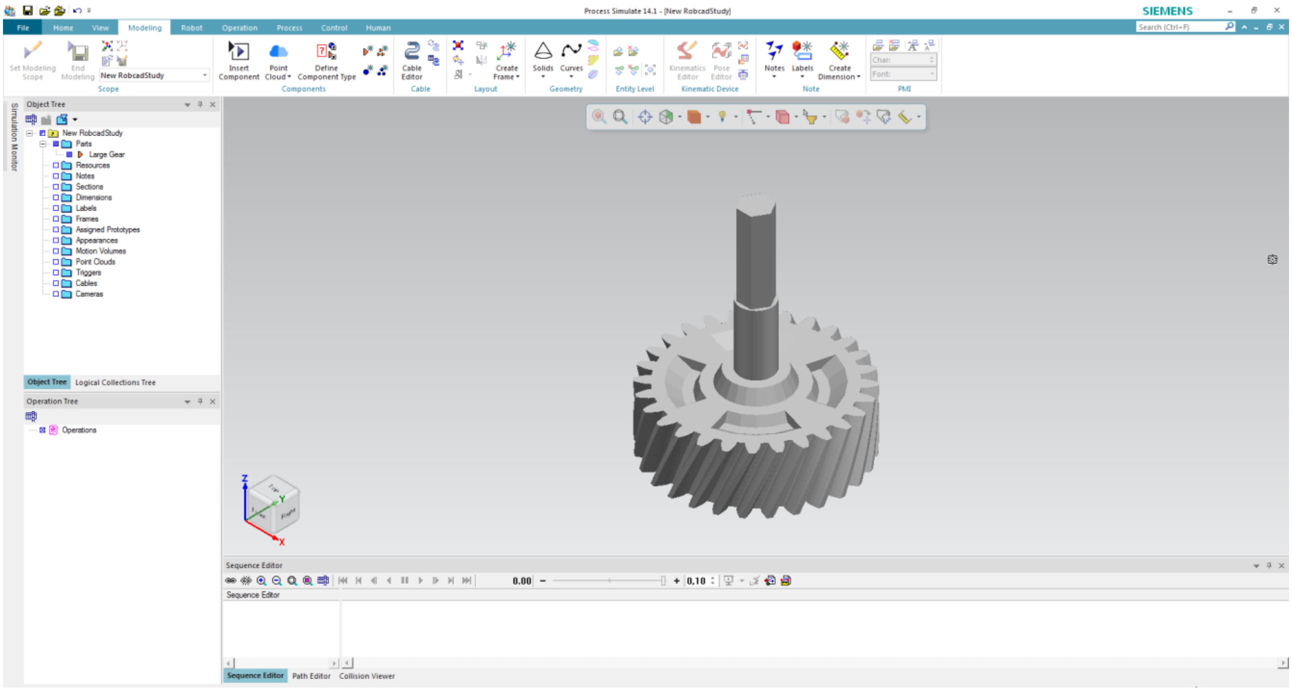


Figure 0-17 The large gear Process Simulate model

Chapter 6

Simulation

6.1 Introduction

The whole process has been simulated in order to perform virtual results before experimenting a real application on the Conveyor System. According to Siemens USA supporting team, it is unnecessary to make any kind of investment before testing appealing projects into a virtual world. It is extremely important to design an efficient virtual model that makes clear whether it fits with expectations or not in terms or reliability, efficiency and cost. Process Simulate platform ran a enough simulations that made us aware of the potential of the application both academically and industrially, the robots have been planned to always compute the same trajectories and a few further optimizations has been made to adjust some smoothness in the motion, speed and acceleration profiles by modifying some via points through the robots' trajectories. During the Robots' path planning task, it was not possible to completely plan a trajectory in the joint space because the start, end and via points configurations does not tell the whole kinematics story of joints' motion. We do not know a priori the joints configuration in most of the points of our paths indeed. Traditionally, it is not an issue because the operator can use the teach-pendant to register all the via-points, but as the process has to be entirely automatic, there is no real teach-pendant to guide the robotics operations and into the Process Simulate environment robots' trajectories only keeps track of the fastest motion as possible (computing triangles speed profiles as it is said later).
However, the points that has been decided in the path are arbitrary, as the wait pose, the via points and so on. The joint space path planning can be used to reach that positions, which are a priori known in the robots' configurations, with any available motion equation.

6.2 Trajectory planning

First, it is used to specify the criteria to properly understand what a trajectory is and in which ways trajectory planning can be designed. While a trajectory is a time history or position, velocity and acceleration for each DDF, trajectory planning shows how a trajectory or path needs to be specified through space [25].
Designing a path description for robots considers motions of the tool frame relative to the station frame. However, moving the tool frame from its initial position to a desired final position always involves both specifying via points with their time elapsed and considering the relative smooth motion's basic issue. It is desirable to have a smooth motion for the manipulator, analytically taking into account that smooth functions have a continuous first derivative and/or a second derivative. Robot applications do require smooth motion when holding parts or using a tool and it is very important to reduce wear and tear on the motors, gears and other mechanical components. Many choices are possible for path planning and the trajectory can be specified and planned in many different ways, as long as they do not exceed needed constraints on the kinematics requirements.

In this project, we always refer to a joint space trajectory planning domain, in which path shapes (in space and time) are described in terms of functions of joint angles:

$$\theta(t) = [\theta_1(t), \theta_2(t), \dots \theta_n(t)]^T \qquad for\ all\ t \in [t_o, t_f]$$

When a simulation runs and joint motion starts, it aims to move point to point from θ_A to θ_B in the fastest possible way. It means it moves each joint at its maximum speed and a few assumptions can be pointed out:

1. All joints begin motion at the same time
2. Each joint ends motion at different times, based on:
    a. Maximum joint speed
    b. Distance to travel between θ_A and θ_B
3. Motion is not smooth

What helps a path planning in simulations is that it is based on a joint interpolated motion. It means that before the simulation rans, it is required to determine which robot joints will take the longest time to complete the move from θ_A to θ_B and new pre-assumptions are possible then:

1. Run that joint at its maximum speed
2. Run all other joints at a speed slower than their maximum so that all joints complete the move at the same amount of time
    a. All joints start and finish move at the same time
    b. Motion is not smooth at start and end of move

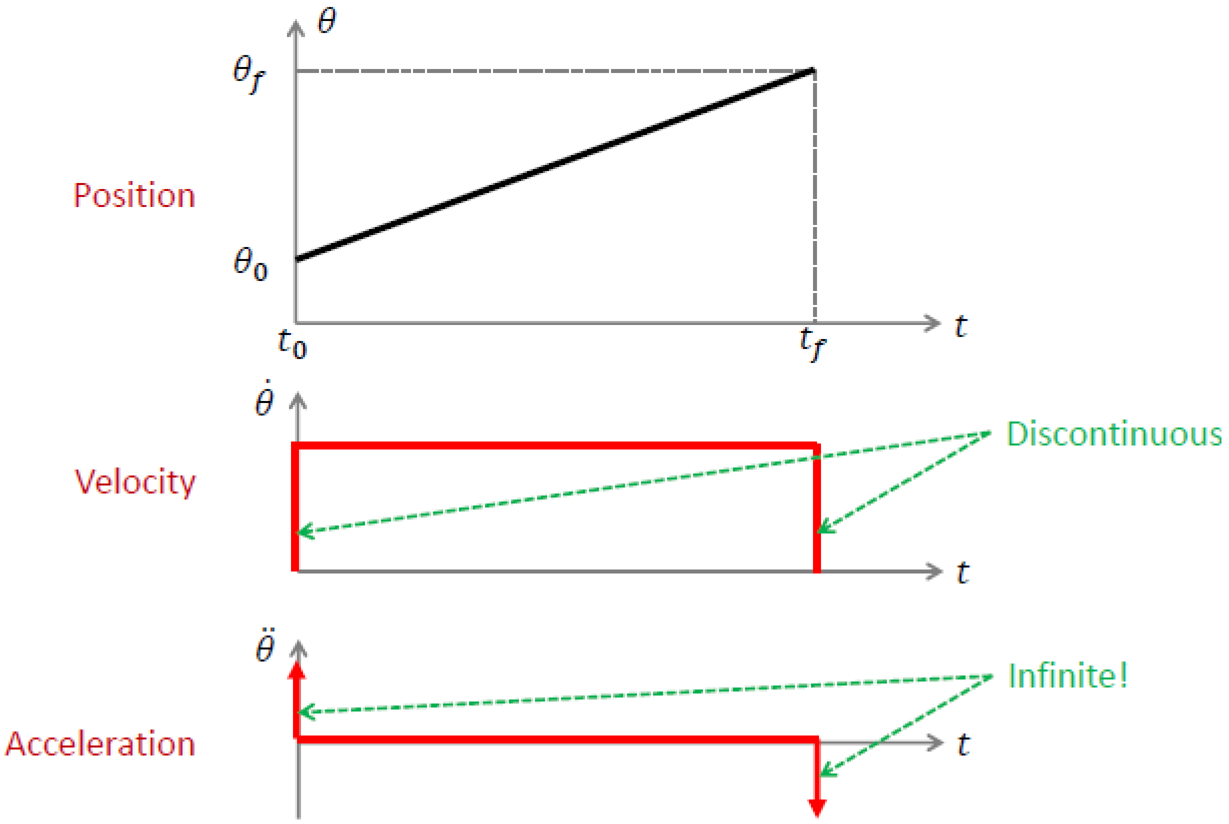


Figure 0-1 Linear joint interpolated motion

In the previous graphs the velocity profile of the joint achieves the desired speed during the motion, but in order to make smooth the start and stop of the motion other profiles are required for robots' trajectory planning, known as: cubic polynomials, higher-order polynomials and linear functions with parabolic blends. It is not on the matter of this project to seek an exact joint interpolated motion profile for the system's robots because the simulation environment does not take any of them as inputs to reach different via points, but we can recognize which joints profiles need to be optimized in order to find reliable results from simulation.

6.3 Sequence editor overview

The sequence editor is one of the viewers in Process Simulate environment [26] and it used to view and edit the process structure, it gives an overview of the main process' operations. Both ABB and KUKA setup's sequence editors are shown in this paragraph and it is useful to briefly introduce the system's simulations:

1. Robot 1:
    - Part flow simulation was planned to lead the tray from the starting station to the first robot stop
    - The first robot (IRB1200 for ABB and KR6 for KUKA) picks up the gear box bottom
    - The first robot places it into the tray
    - Another part flow simulation leads the tray to the second robot stop

2. Robot 2:
    - The second robot (IRB1600 for ABB and KR10 for KUKA) picks up the large gear
    - The second robot places it into the gear box bottom (fixing operations would be required in the future progress of the simulation)
    - Another part flow simulation is made to flow into the conveyor until the tray reaches the diverter where the parts would be ideally picked or read a new robot 1's operation

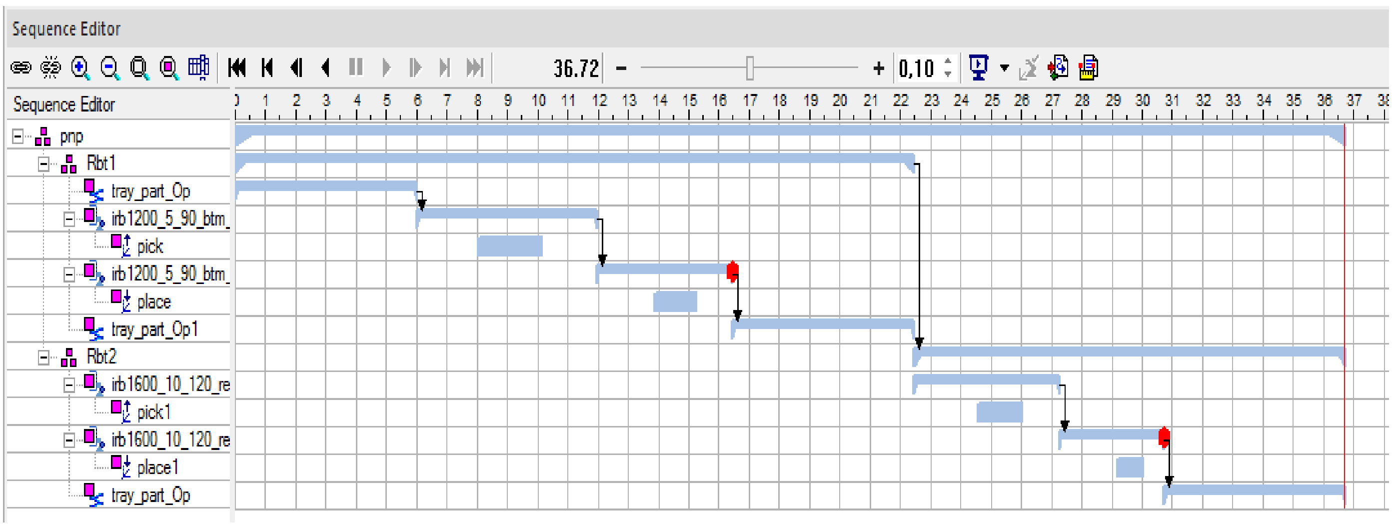


Figure 0-2 ABB sequence editor after simulation

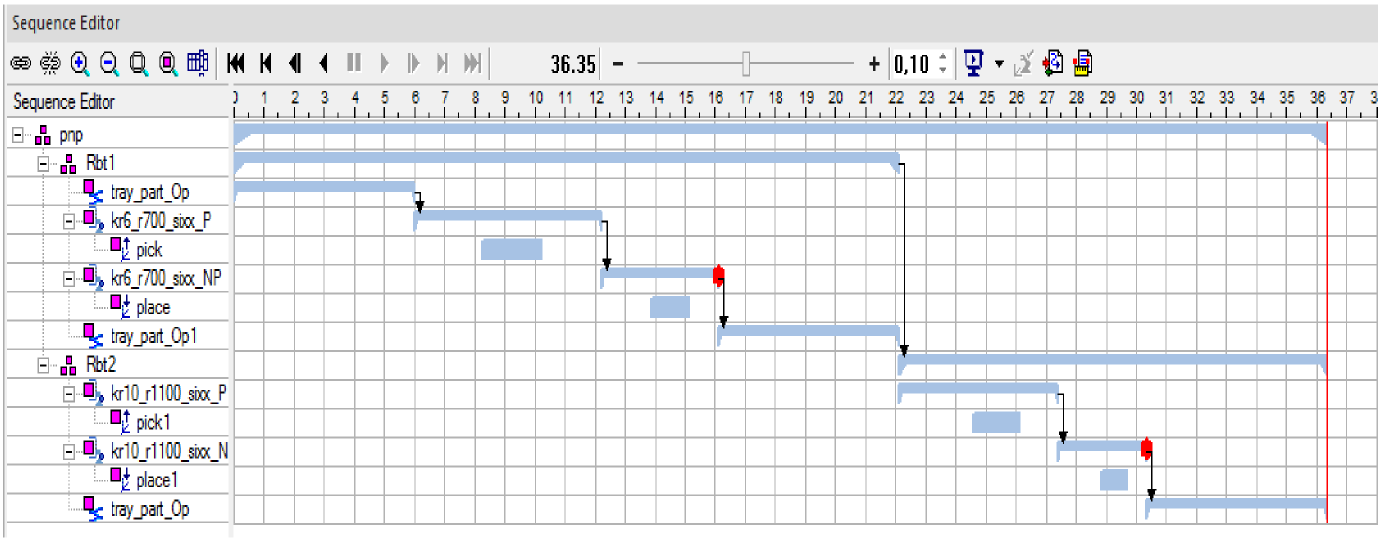


Figure 0-3 KUKA sequence editor after simulation

6.4 Path planning setup

Whereas the sequence editor concerns about the process structure, the path editor in Process Simulate environment [26] is used to view and edit operation/simulation paths. It is the simulative operations instrument that assists the simulators to achieve the dictates that have been mentioned in the paragraph 6.2. The operations have been already introduced, but here we describe in depth the Process Simulate operation paths.

6.4.1 Object flow paths planning

The first step in Process Simulate is always to load an assembly and create a preliminary path. After that the path is continuously refined. The path is simulated, and errors are detected and removed. If any errors remain, the path is simulated and corrected again. To create a path, you need to move the assembly part to the desired position using the placement commands provided by the platform. As a part is moved, locations can be marked and added to the path. Hence, a flow path is a sequence of locations, each of which constitutes a position and orientation for the assembly part. The movement of the assembly part along from the first to the last location in path is represented as a dotted line in the graphic viewer. Paths are typically associated to an operation in the sequence editor.

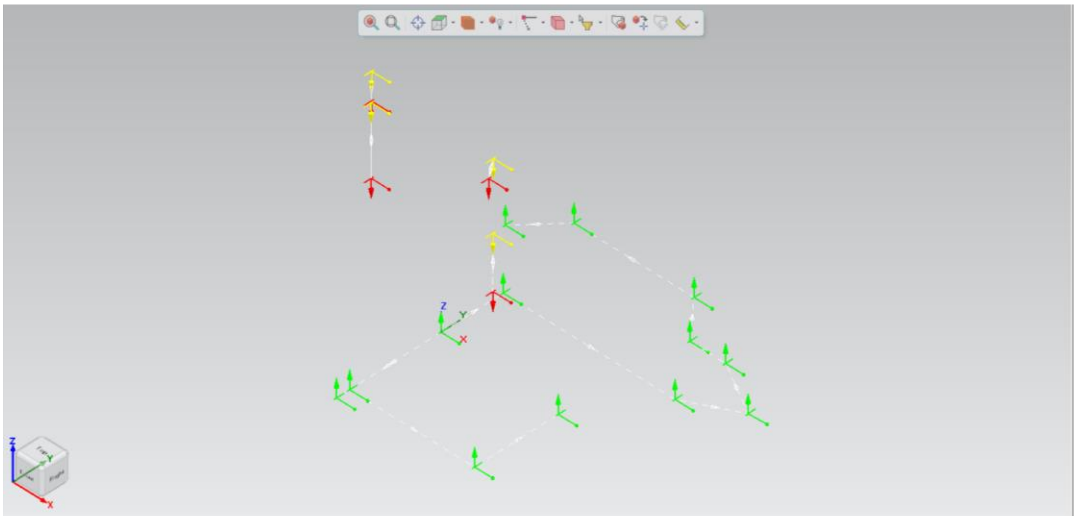


Figure 0-4 Tray part planned locations

6.4.2 Material handling paths planning

The system’s robots aim to do material handling operations on the parts and specifically pick and place operation as it was mentioned before. To create pick and place operations, we needed to add approach and retract locations on the part and define the sequence of operations.
First, the grasping tool or gripper must be mounted to the robot, the new pick and place (pnp) operation is created and a robot is assigned to the operation. Approach and retract locations are chosen relative to the part to add clearance moves before simulating. In the Process Simulate environment we can simply choose to add locations before and after one another or use the Robot Jog to freely position the location by pressing add current location. This task has to be done with great cure and it is the first change when it is clear the joints’ motion does not smooth out sufficiently, it could occur collisions or it does not respect safety ranges in terms of speed or maximum acceleration. After a few optimizations the pnp paths eventually have been defined. As an example, the first stop Robots’ paths have been proposed below. It is important to notice that the whole pnp operation has been divided into the two pick (figure 6.5 and 6.7) and place (figure 6.6 and 6.8) sub-operation and that the red locations represent the actual pick and place poses, while the yellow locations represent the via locations, which constitutes the reach and release poses. There is no difference between the second stop robots’ paths and the ones proposed here. The former paths meddle with the object flow paths though.

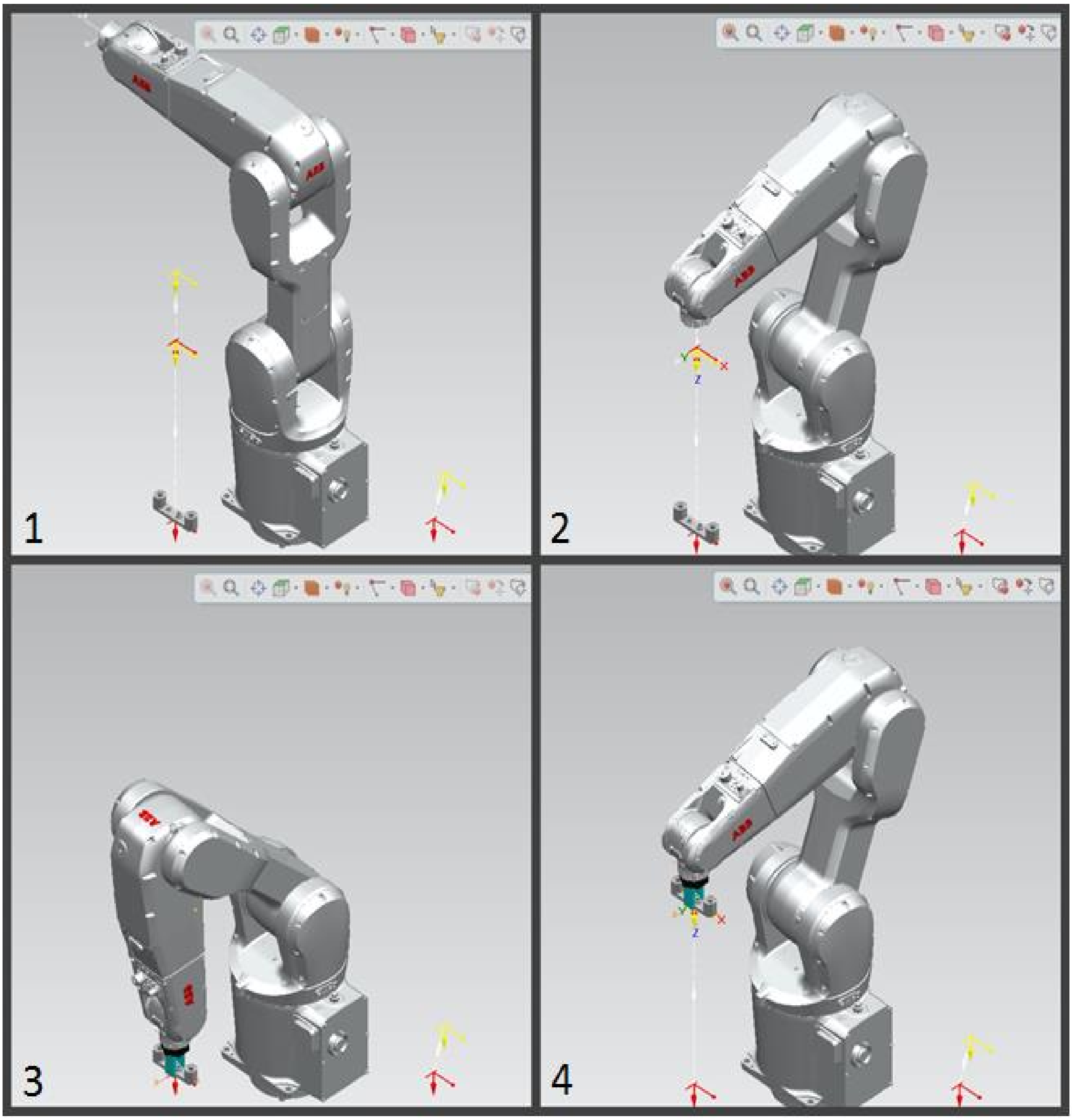


Figure 0-5 IRB 1200 pick poses: 1 home, 2 via (approach) location, 3 pick location, 4 via (retract) location

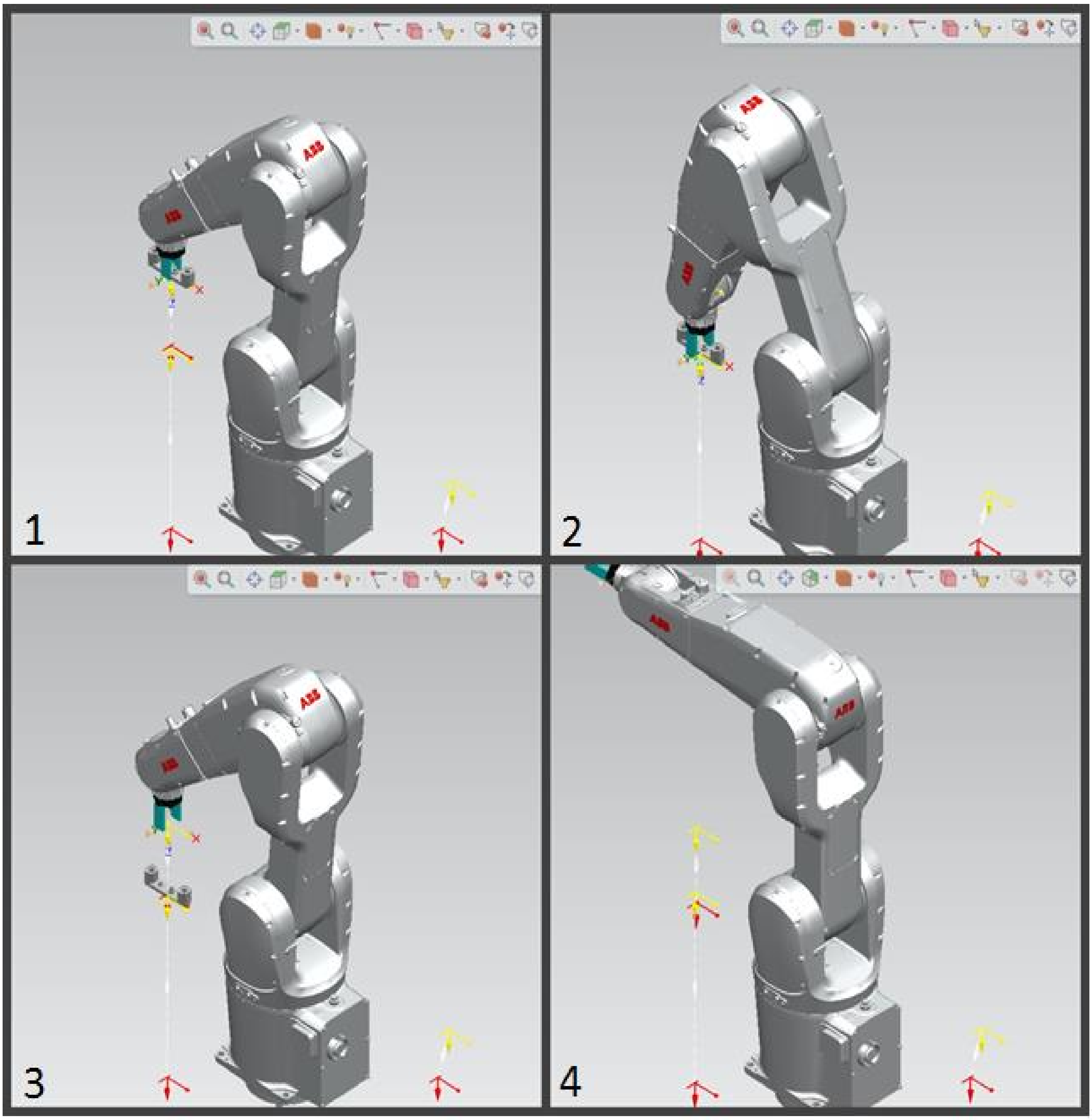


Figure 0-6 IRB 1600 place poses: 1 via (approach) location, 2 place location, 3 via (retract) location, 4 home

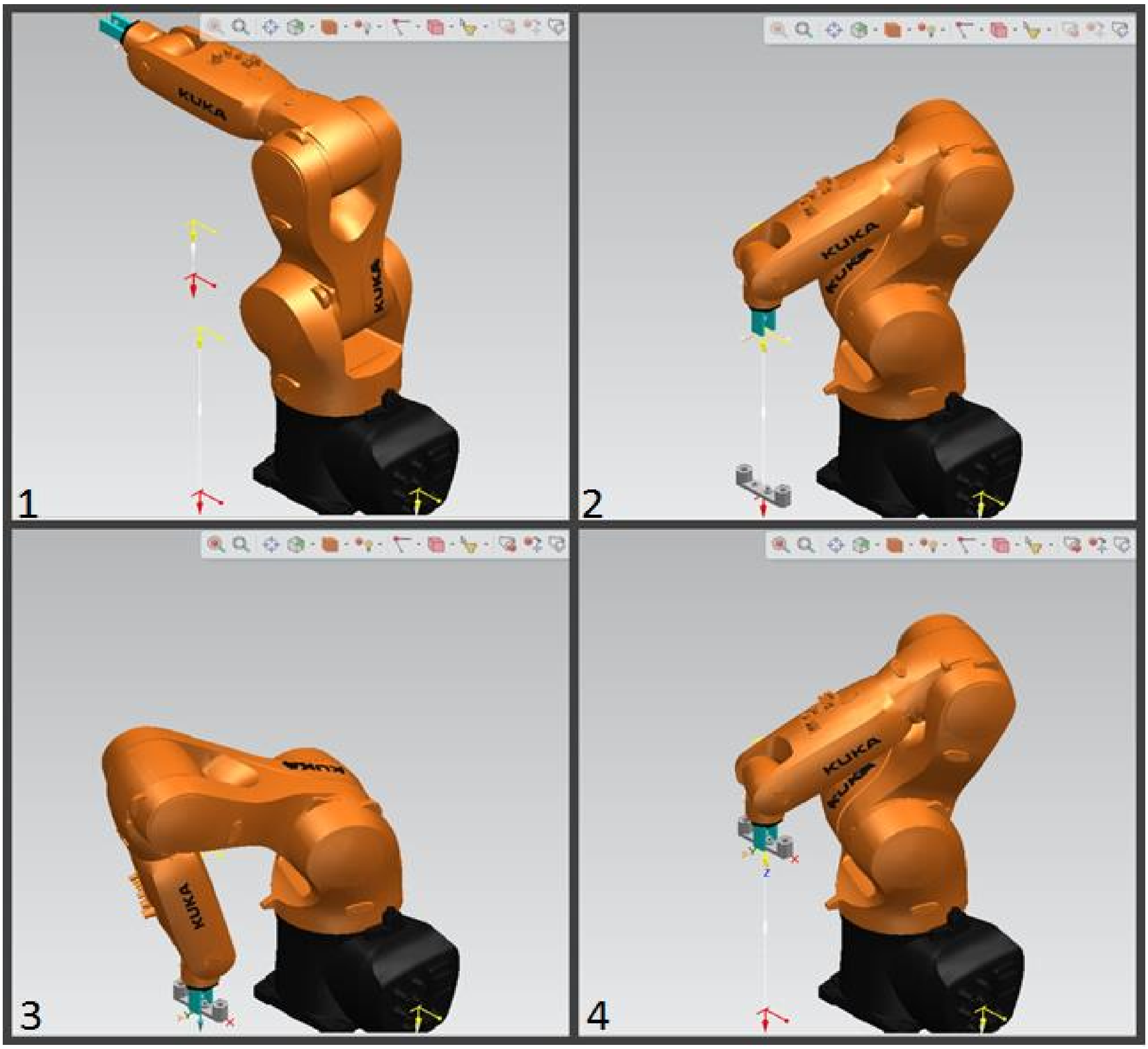


Figure 0-7 KR 6 pick poses: 1 home, 2 via (approach) location, 3 pick location, 4 via (retract) location

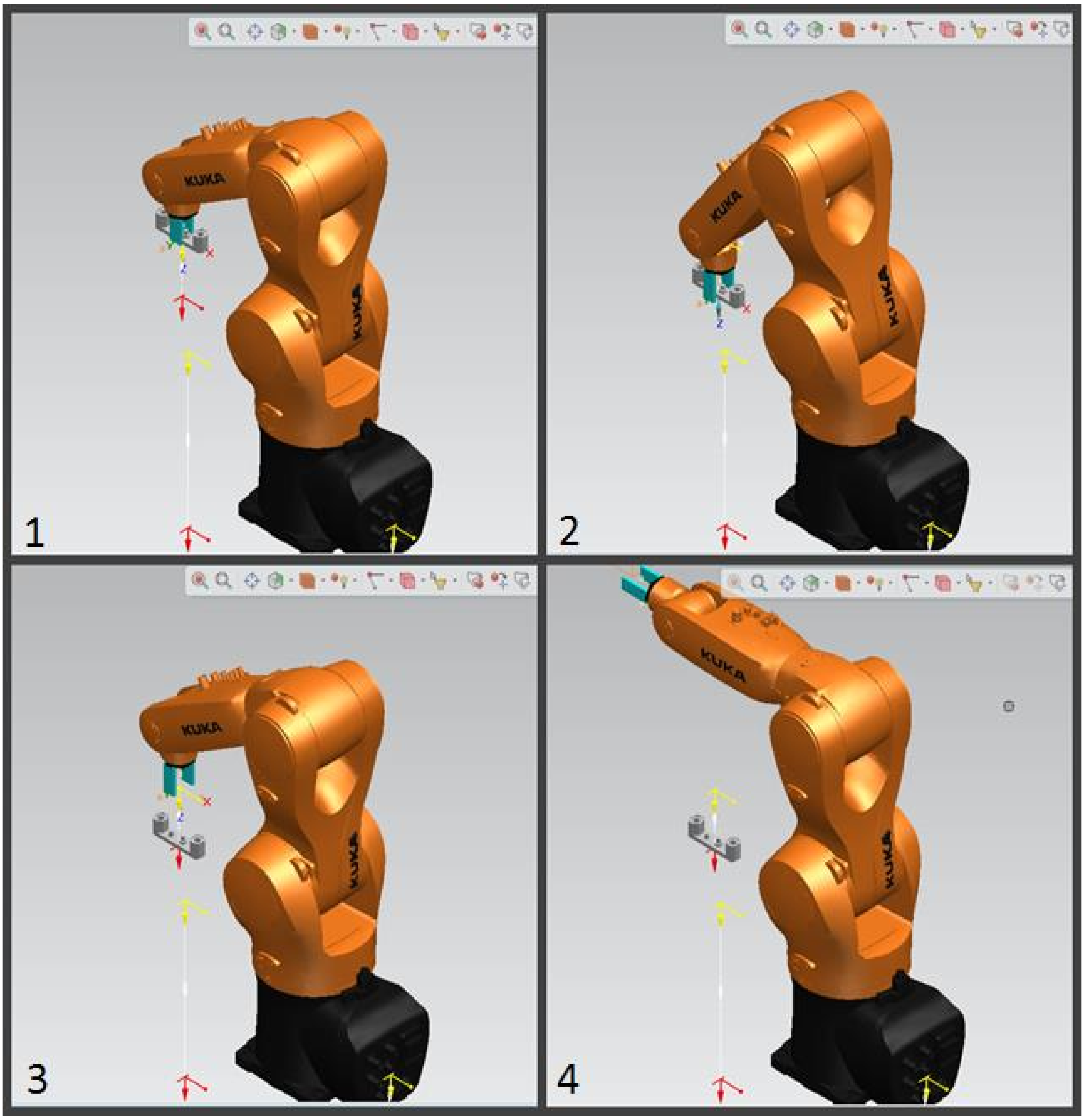


Figure 0-8 KR 10 place poses: 1 via (approach) location, 2 place location, 3 via (retract) location, 4 home

Chapter 7

Results

7.1 Kinematics reliability evaluation

In the previous chapter, the process simulations and the kinematics paths' algorithm have been presented, the latter is a sort of control for the robots that sends the motion inputs to them. Whereas the path editor and the sequence editor are two of the Process Simulate viewers that let us correct the trajectories, it is also important to mention the Robot viewer that receives the motion inputs and send back the robots' feedbacks that eventually are used to edit the trajectories.
Within the Robot viewer the Joint status has been continuously checked, the robots' position, speed, acceleration and TCPF speed have been evaluated, leading to path planning reconsiderations and optimizations. During the data test work, the smoothness of position and speed profiles have been guaranteed for the great part, while most of the acceleration profiles that presented spikes have been retested, lowering joints maximum speed and adding locations in order to create a point-to-point motion with continuous acceleration profiles.

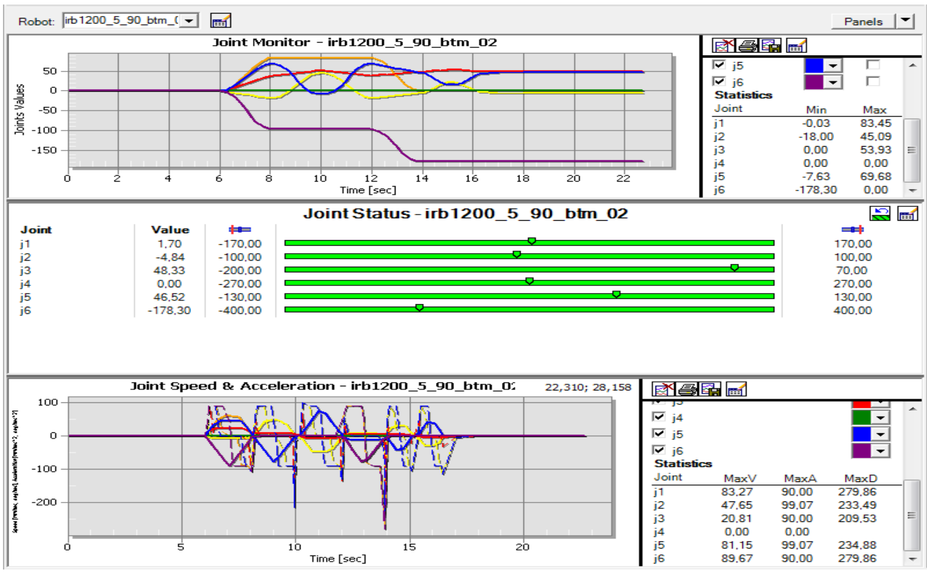


Figure 0-1 Robot viewer – Joint monitors

In the first paragraph of this chapter, those robotics feedbacks that have been collected in the Robot viewer are proposed. It has been made a long collection of kinematics graphs, representing the joints motions and guaranteeing the reachability and the robots companies speed limits that have been listed in chapter 4.
In the second paragraph a collection of manufactories factors has been emphasized, highlighting the importance of scheduling, production planning and the differences that have been found while working both with ABB's and KUKA's robots.

7.1.1 IRB1200 joints kinematics

In this paragraph the joint kinematics profiles of the ABB's robot at the first stop have been proposed. In figure 7.2 the joint 1 presents smooth position motion and a trapezoidal and triangle speed profile law. The acceleration is affected by it and changes sharply its profile during the beginning and the end of the joint robot base's rotation. The speed maximum value is far from its safety limit though. We assumed no further optimization is needed in that case, because the other joints might feel the effect of it. In figure 7.3 still the joint position looks smooth and the speed profile has slight disruptions and it is under the safety limit. The acceleration looks like the previous joint, plus some weak spikes. In figure 7.4 the joint position is close to the maximum reachability upper limit, but it is inside the range. The speed profile is quite steady, and the acceleration is weakly affected. In figure 7.5 the joint 4 does not rotate at all during the simulations. It links the upper arm to the end effector, rotating the former in a circular motion. It is used to move the part between horizontal to vertical orientations. In figure 7.6 and figure 7.7 the joint 5 and joint 6 profiles do not differ very much from the previous ones.

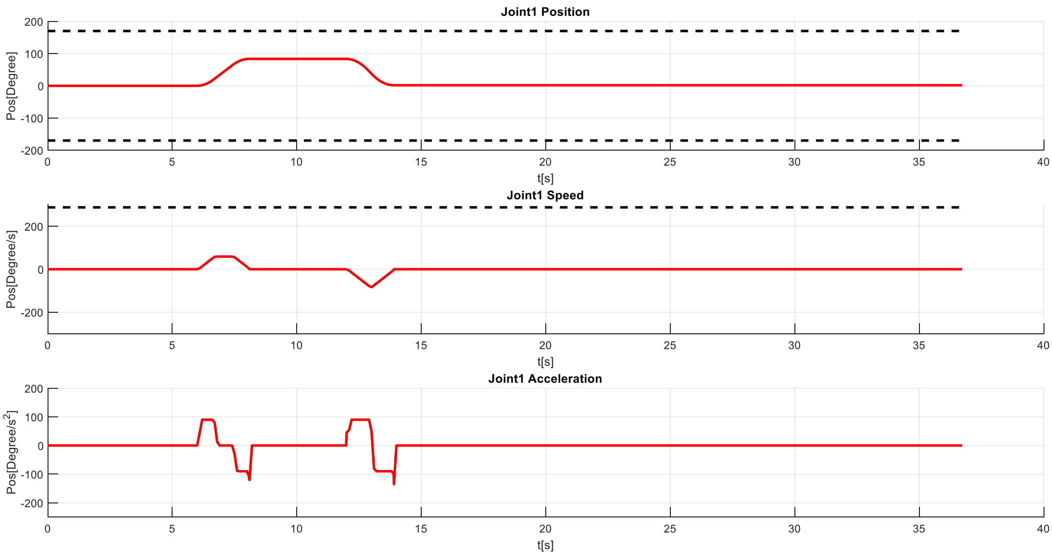


Figure 0-2 IRB 1200: Joint 1 – position, speed and acceleration profiles

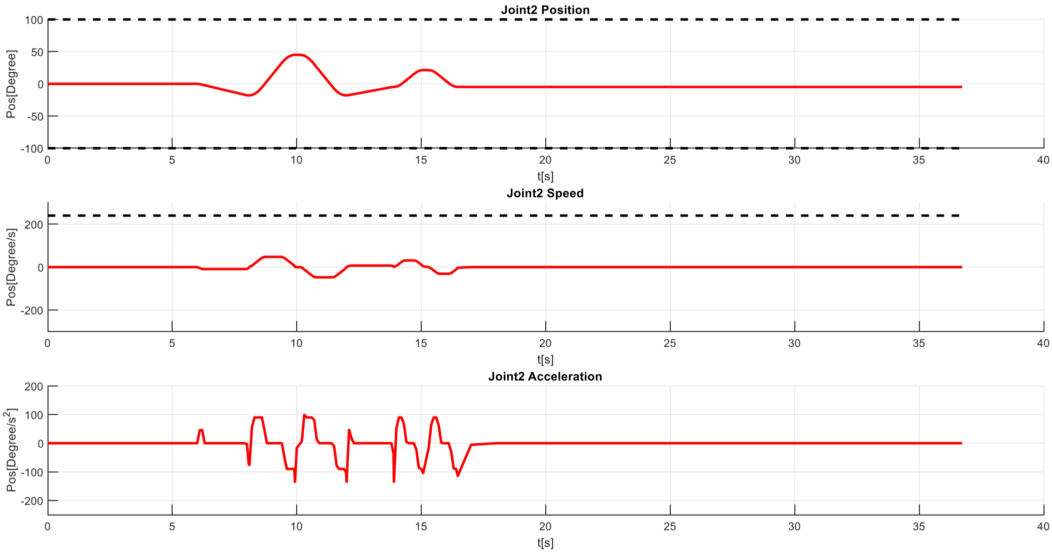


Figure 0-3 IRB 1200: Joint 2 – position, speed and acceleration profiles

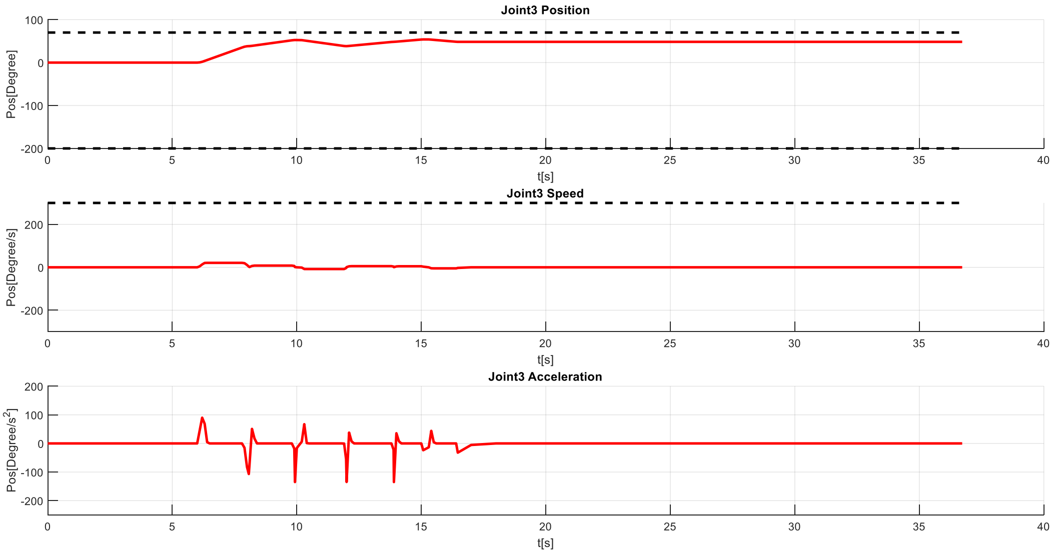


Figure 0-4 IRB 1200: Joint 3 – position, speed and acceleration profiles

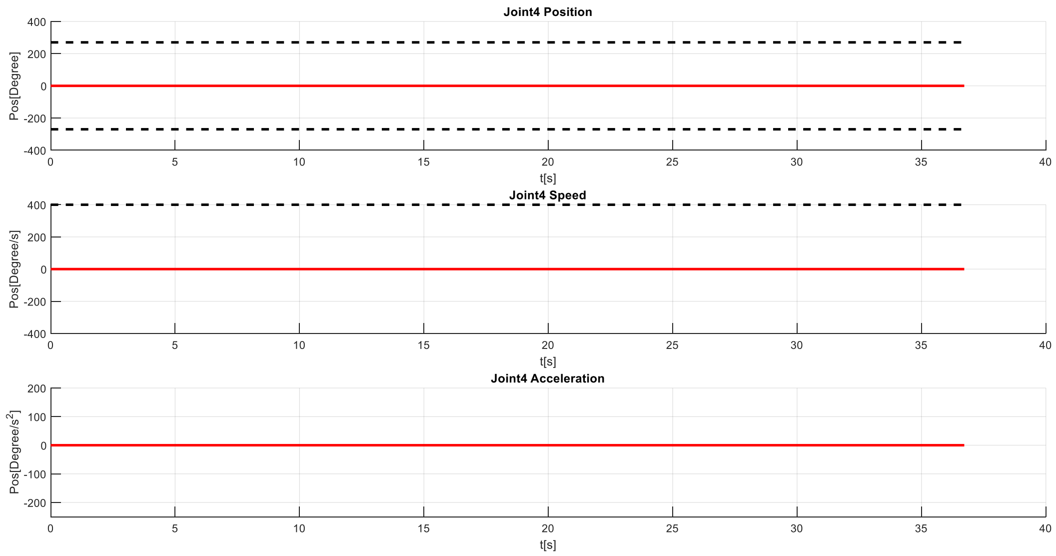


Figure 0-5 IRB 1200: Joint 4 – position, speed and acceleration profiles

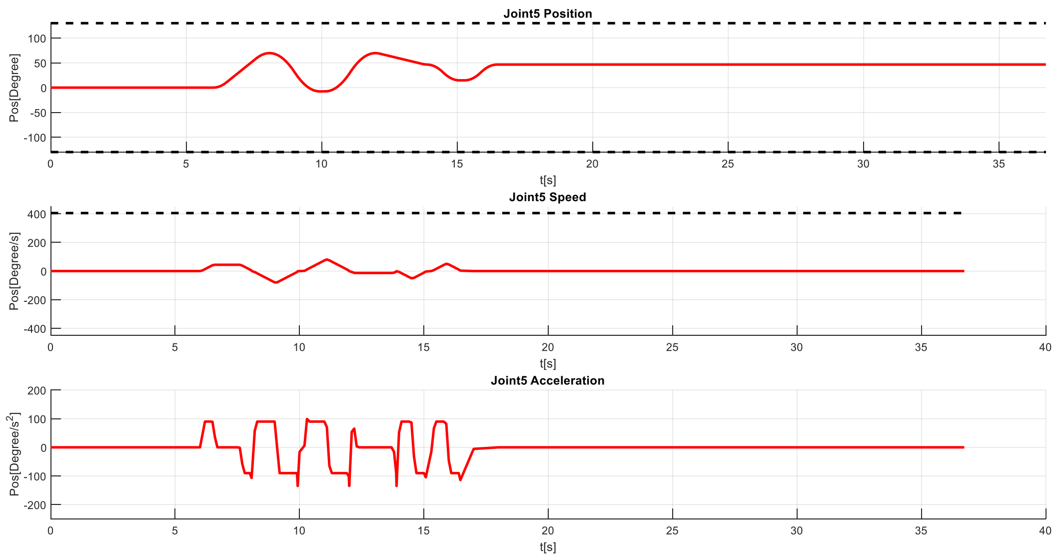


Figure 0-6 IRB 1200: Joint 5 – position, speed and acceleration profiles

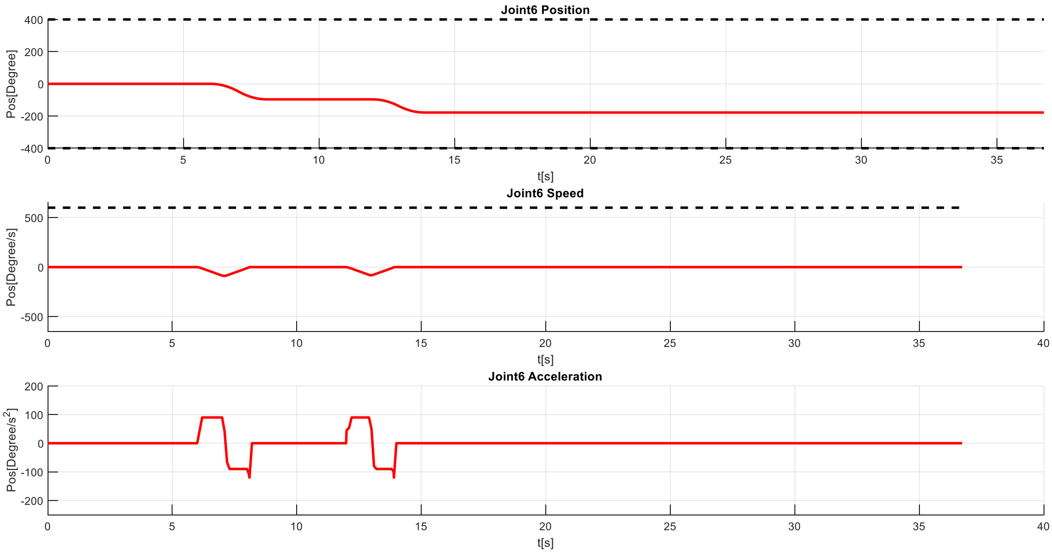


Figure 0-7 IRB 1200: Joint 6 – position, speed and acceleration profiles

7.1.2 IRB1600 joints kinematics

In this paragraph the joint kinematics profiles of the ABB's robot at the second stop have been proposed. In figure 7.8 the joint 1 presents smooth position motion and a trapezoidal and triangle speed profile law. The acceleration is affected by it and changes sharply its profile during the beginning and the end of the joint robot base's rotation. The speed maximum value is far from its safety limit though. We assumed no further optimization is needed in that case, because the other joints might feel the effect of it. In figure 7.9 still the joint position looks smooth and the speed profile has slight disruptions and it is under the safety limit. The acceleration looks like the previous joint, plus some weak spikes. In figure 7.10 the joint position is close to the maximum reachability upper limit, but it is inside the range. The speed profile is quite steady, and the acceleration is weakly affected. In figure 7.11 the joint 4 does not rotate at all during the simulations. It links the upper arm to the end effector, rotating the former in a circular motion. It is used to move the part between horizontal to vertical orientations. In figure 7.12 and figure 7.13 the joint 5 and joint 6 profiles do not differ very much from the previous ones.

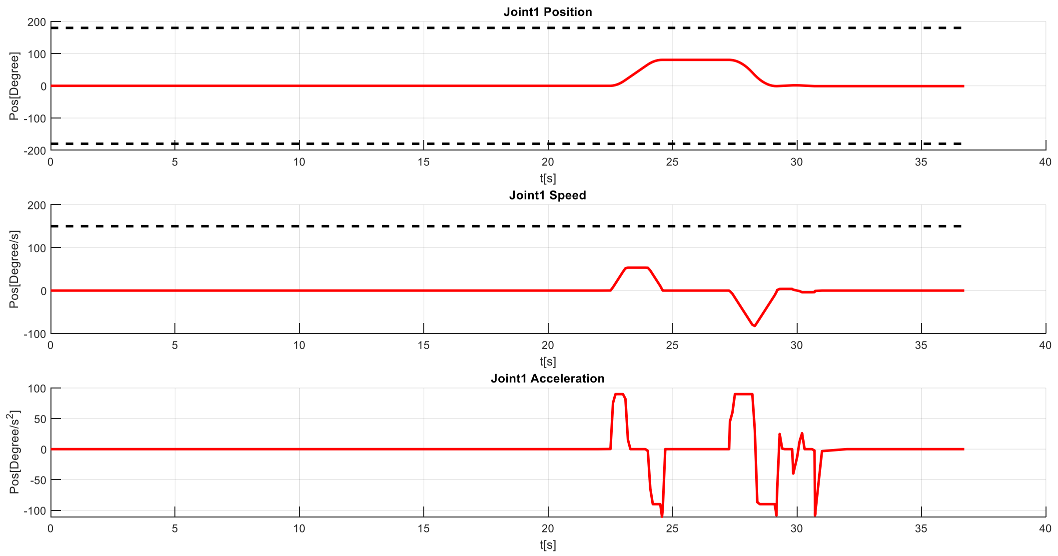


Figure 0-8 IRB 1600: Joint 1 – position, speed and acceleration profiles

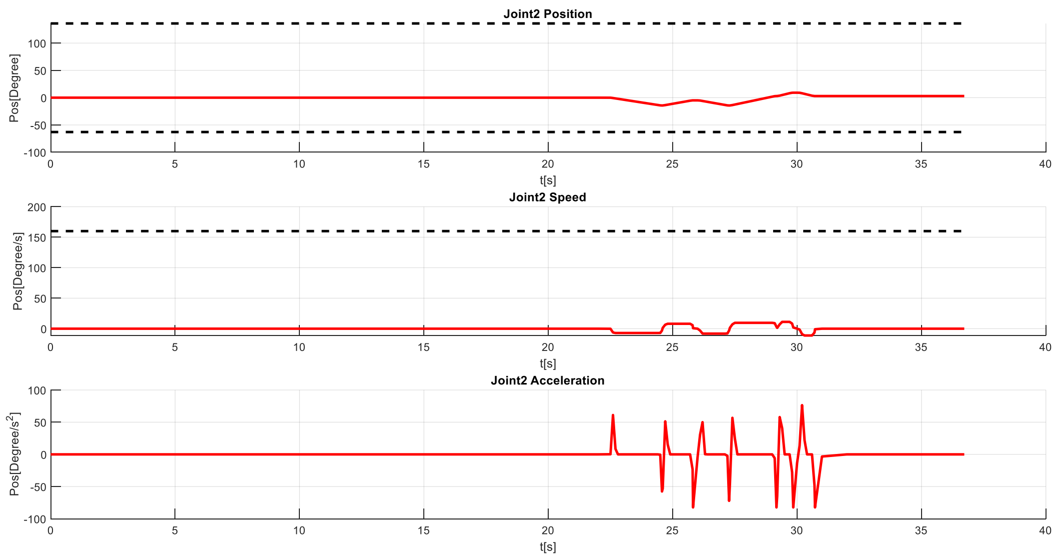


Figure 0-9 IRB 1600: Joint 2 – position, speed and acceleration profiles

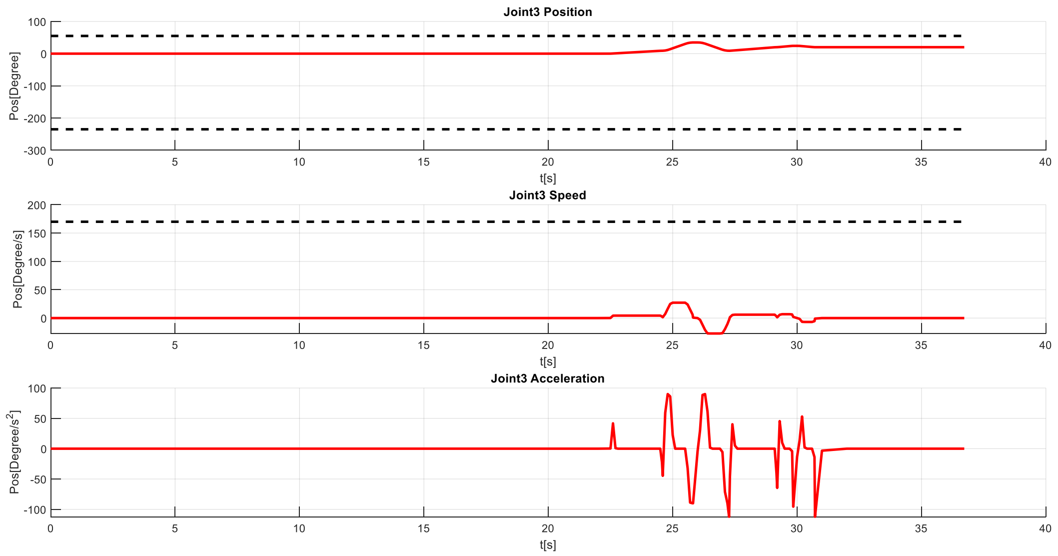


Figure 0-10 IRB 1600: Joint 3 – position, speed and acceleration profiles

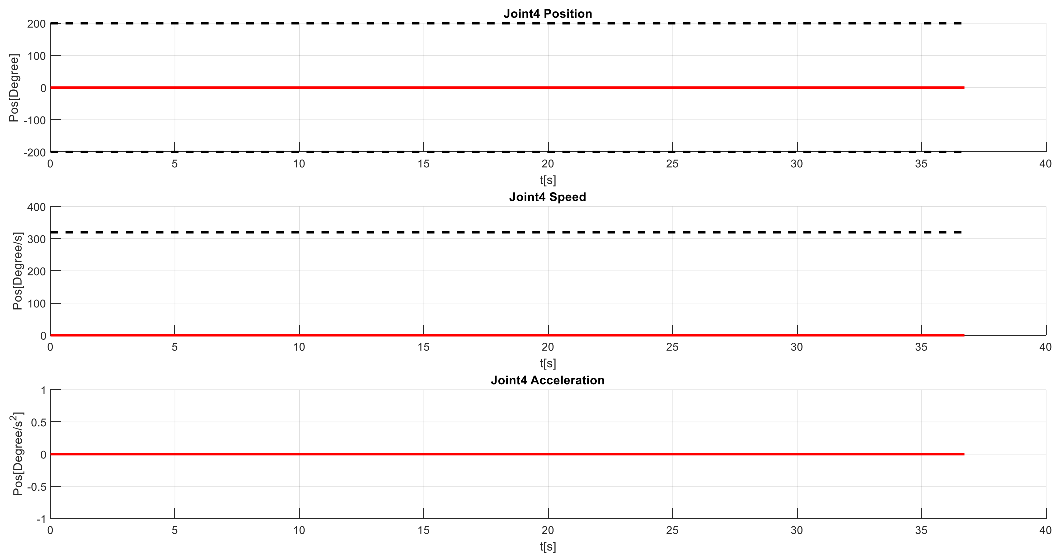


Figure 0-11 IRB 1600: Joint 4 – position, speed and acceleration profiles

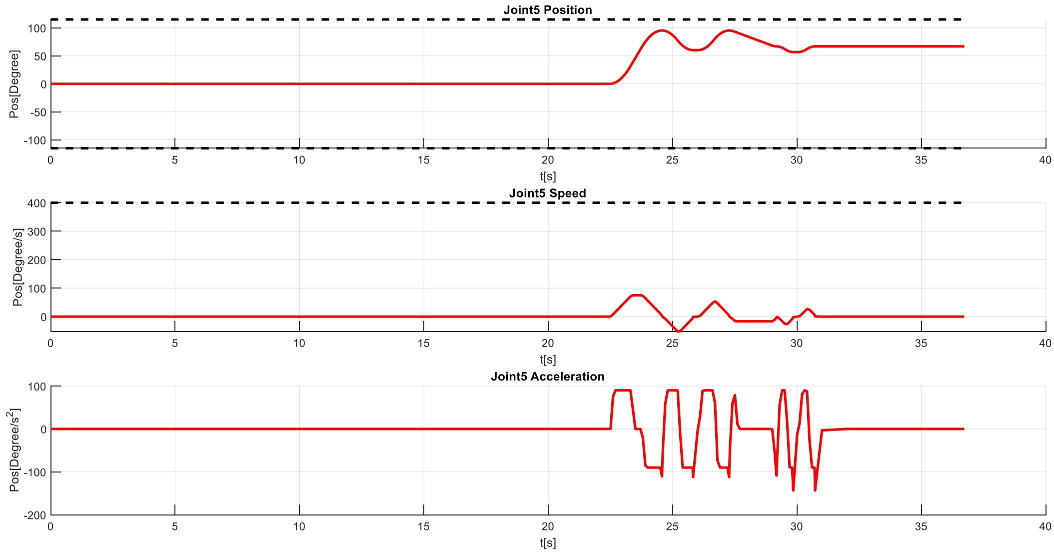


Figure 0-12 IRB 1600: Joint 5 – position, speed and acceleration profiles

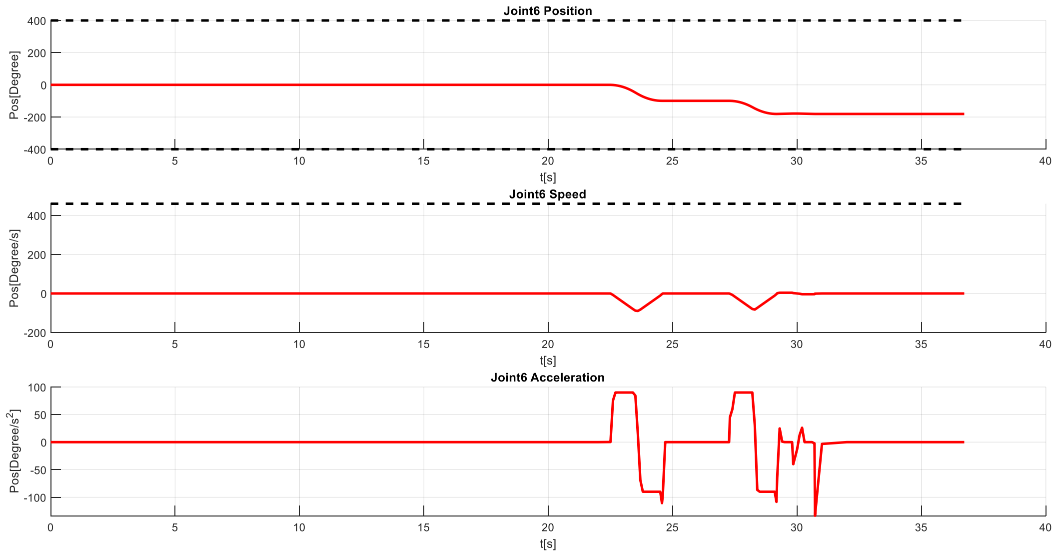


Figure 0-13 IRB 1600: Joint 6 – position, speed and acceleration profiles

7.1.3 KR6 R700 joints kinematics

In this paragraph the joint kinematics profiles of the KUKA's robot at the first stop have been proposed. In figure 7.14 the joint 1 presents smooth position motion and a trapezoidal and triangle speed profile law. The acceleration is affected by it and changes sharply its profile during the beginning and the end of the joint robot base's rotation. The speed maximum value is far from its safety limit though. We assumed no further optimization is needed in that case, because the other joints might feel the effect of it. In figure 7.15 still the joint position looks smooth and the speed profile has slight disruptions and it is under the safety limit. The acceleration looks like the previous joint, plus some weak spikes. In figure 7.16 the joint position is close to the maximum reachability upper limit, but it is inside the range. The speed profile is quite steady, and the acceleration is weakly affected. In figure 7.17 the joint 4 does not rotate at all during the simulations. It links the upper arm to the end effector, rotating the former in a circular motion. It is used to move the part between horizontal to vertical orientations. In figure 7.18 and figure 7.19 the joint 5 and joint 6 profiles do not differ very much from the previous ones.

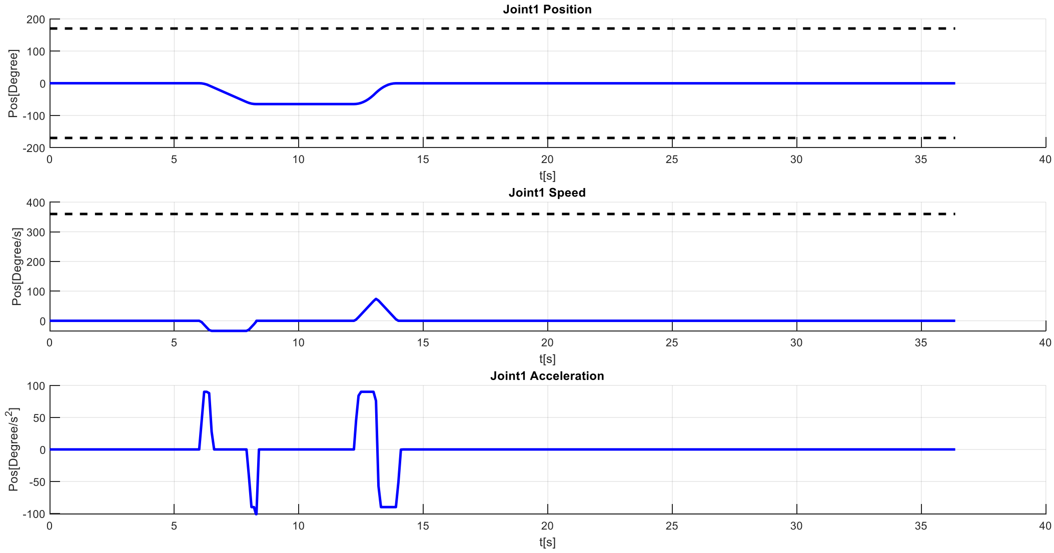


Figure 0-14 KR6 r700: Joint 1 – position, speed and acceleration profiles

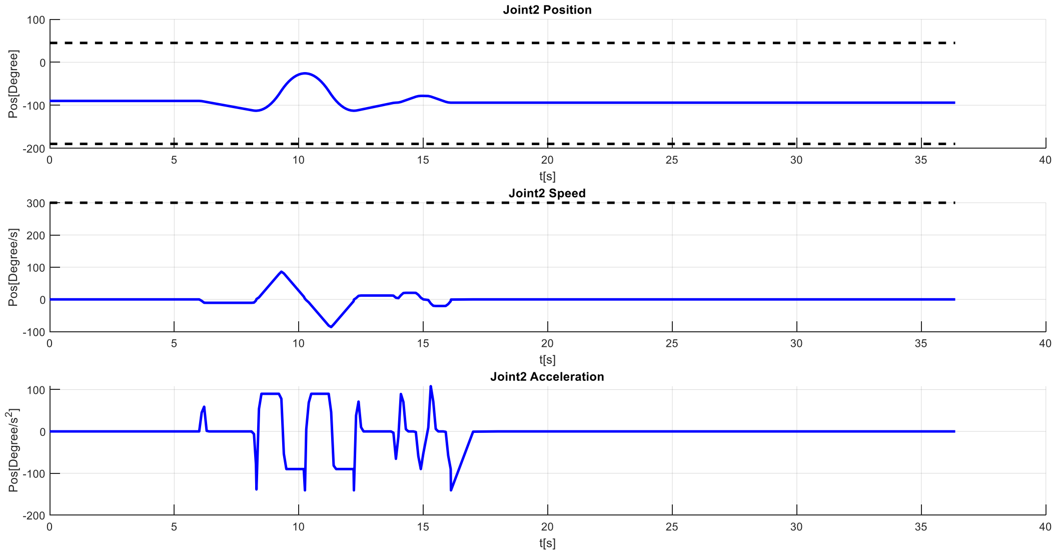


Figure 0-15 KR6 r700: Joint 2 – position, speed and acceleration profiles

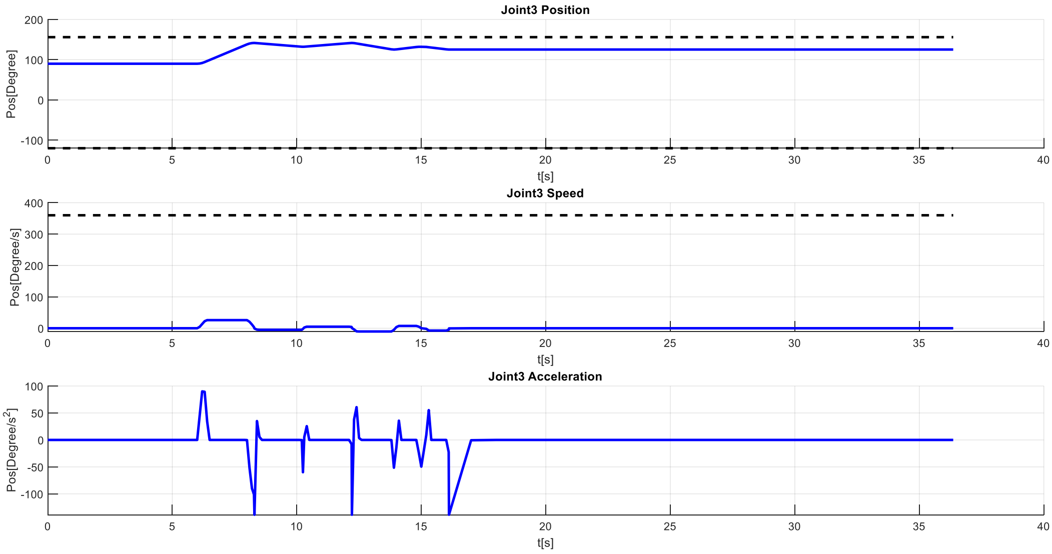


Figure 0-16 KR6 r700: Joint 3 – position, speed and acceleration profiles

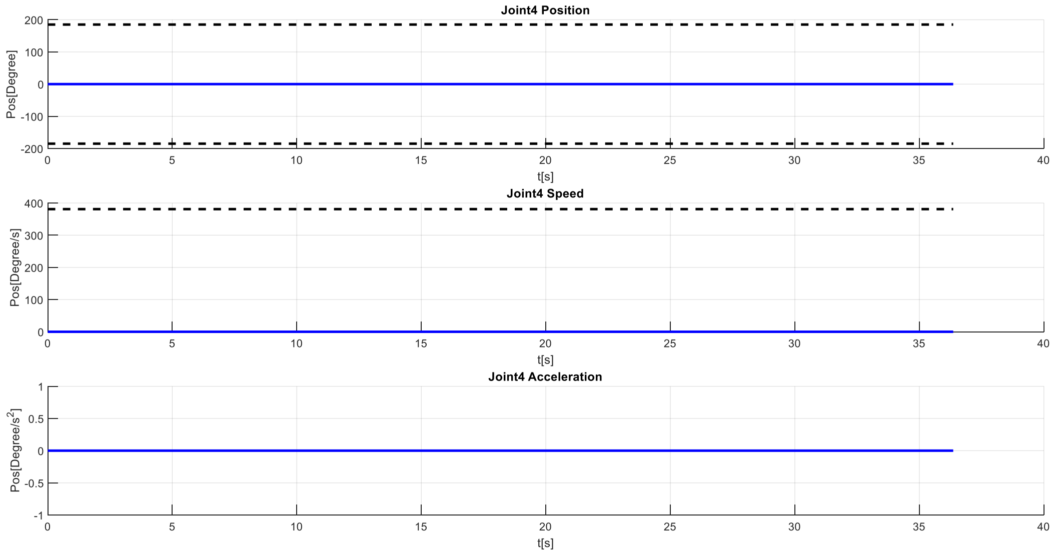


Figure 0-17 KR6 r700: Joint 4 – position, speed and acceleration profiles

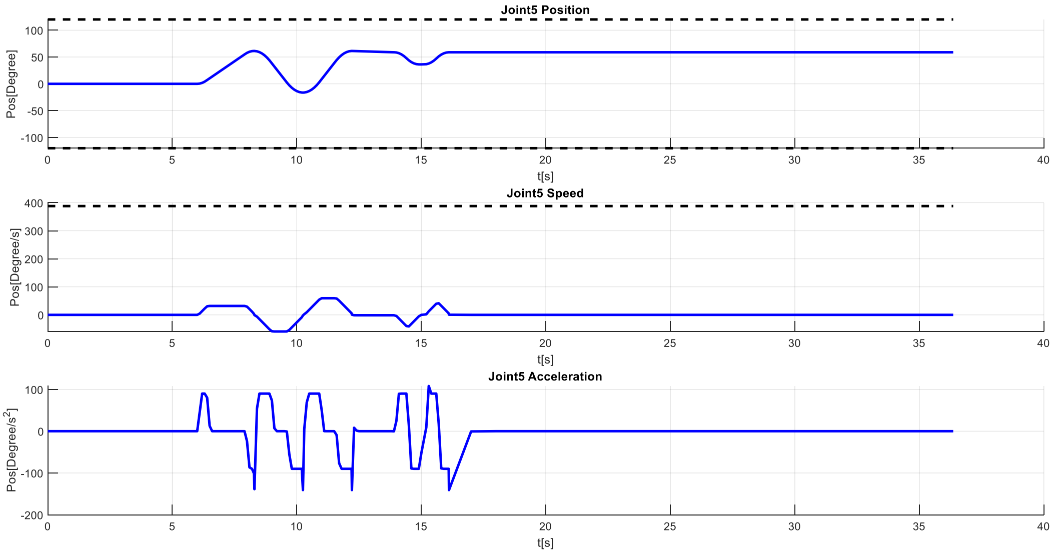


Figure 0-18 KR6 r700: Joint 5 – position, speed and acceleration profiles

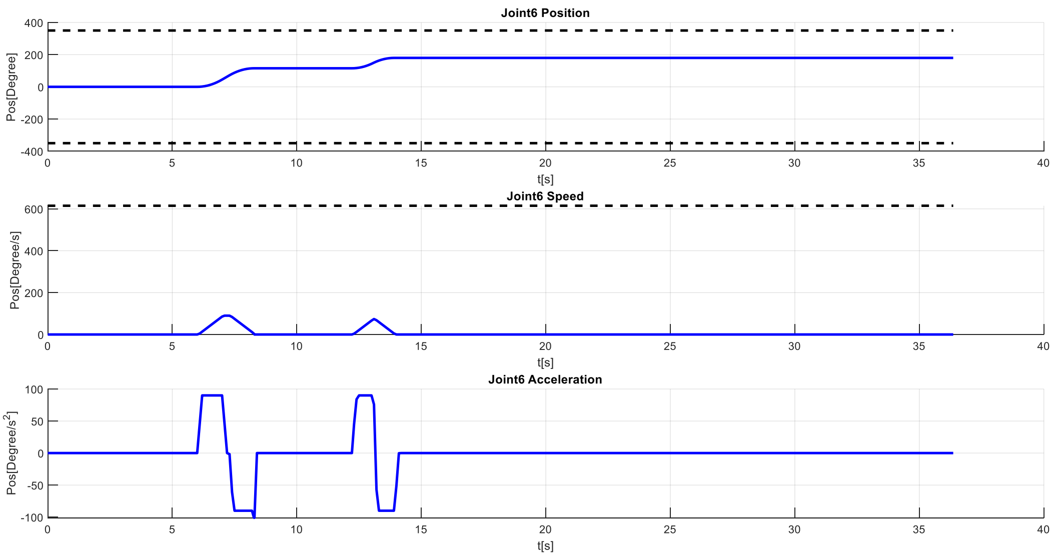


Figure 0-19 KR6 r700: Joint 6 – position, speed and acceleration profiles

7.1.4 KR10 R1100 joints kinematics

In this paragraph the joint kinematics profiles of the KUKA's robot at the first stop have been proposed. In figure 7.20 the joint 1 presents smooth position motion and a trapezoidal and triangle speed profile law. The acceleration is affected by it and changes sharply its profile during the beginning and the end of the joint robot base's rotation. The speed maximum value is far from its safety limit though. We assumed no further optimization is needed in that case, because the other joints might feel the effect of it. In figure 7.21 still the joint position looks smooth and the speed profile has slight disruptions and it is under the safety limit. The acceleration looks like the previous joint, plus some weak spikes. In figure 7.22 the joint position is close to the maximum reachability upper limit, but it is inside the range. The speed profile is quite steady, and the acceleration is weakly affected. In figure 7.23 the joint 4 does not rotate at all during the simulations. It links the upper arm to the end effector, rotating the former in a circular motion. It is used to move the part between horizontal to vertical orientations. In figure 7.24 and figure 7.25 the joint 5 and joint 6 profiles do not differ very much from the previous ones.

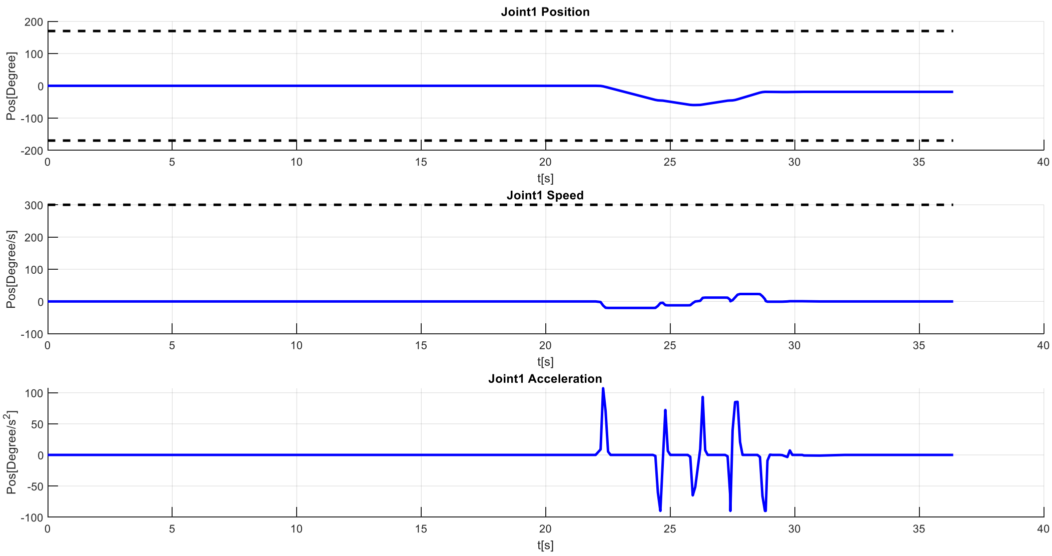


Figure 0-20 KR10 r1100: Joint 1 – position, speed and acceleration profiles

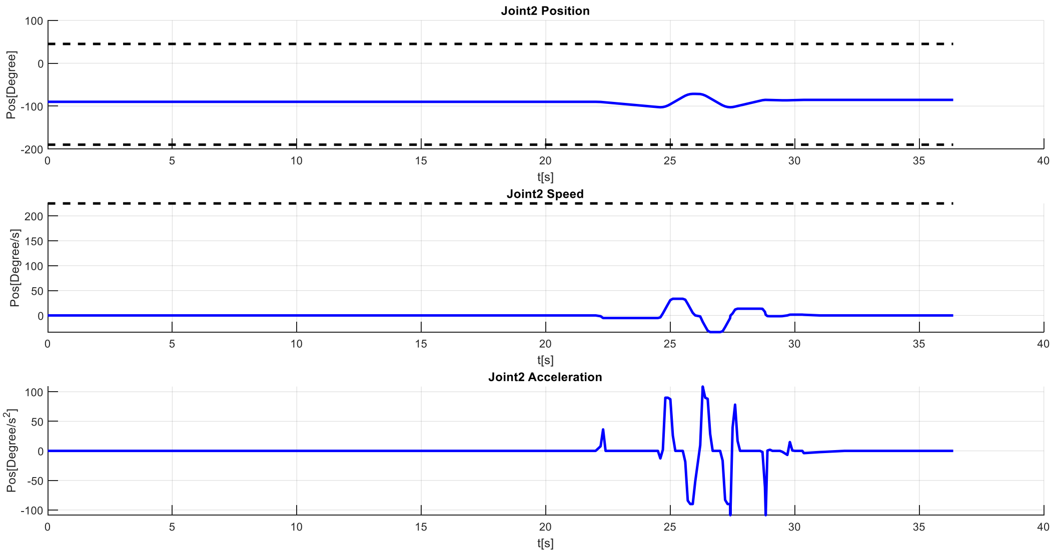


Figure 0-21 KR10 r1100: Joint 2 – position, speed and acceleration profiles

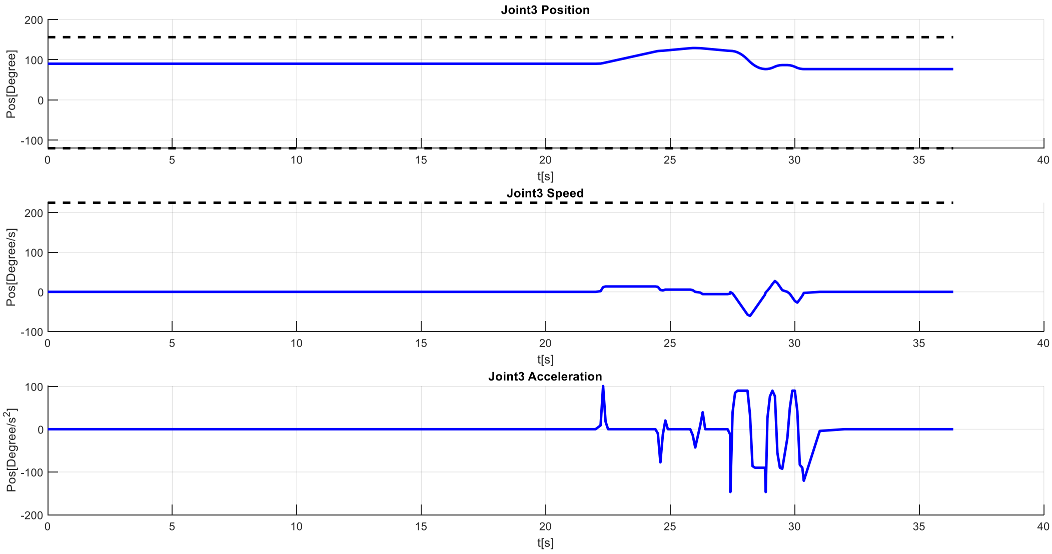


Figure 0-22 KR10 r1100: Joint 3 – position, speed and acceleration profiles

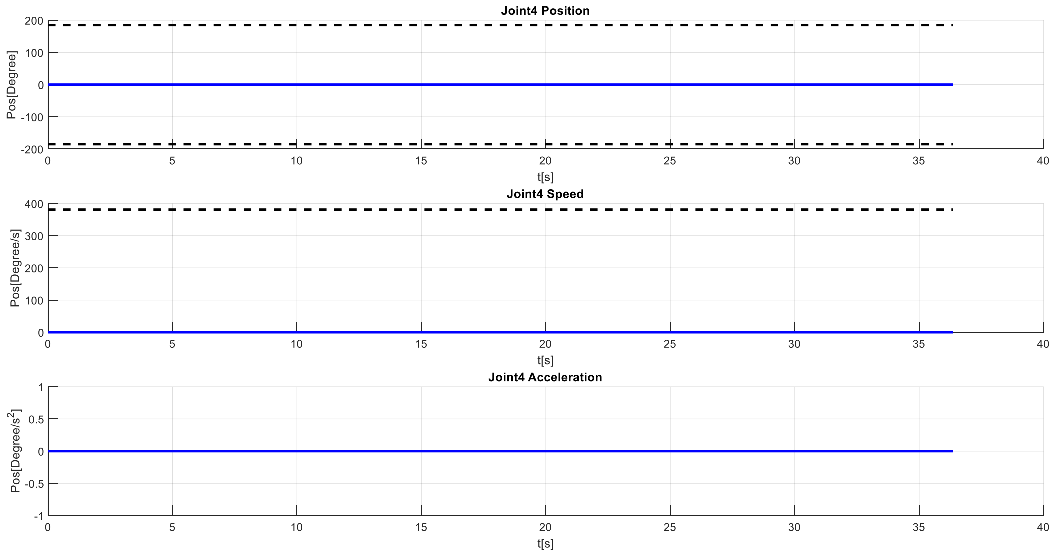


Figure 0-23 KR10 r1100: Joint 4 – position, speed and acceleration profiles

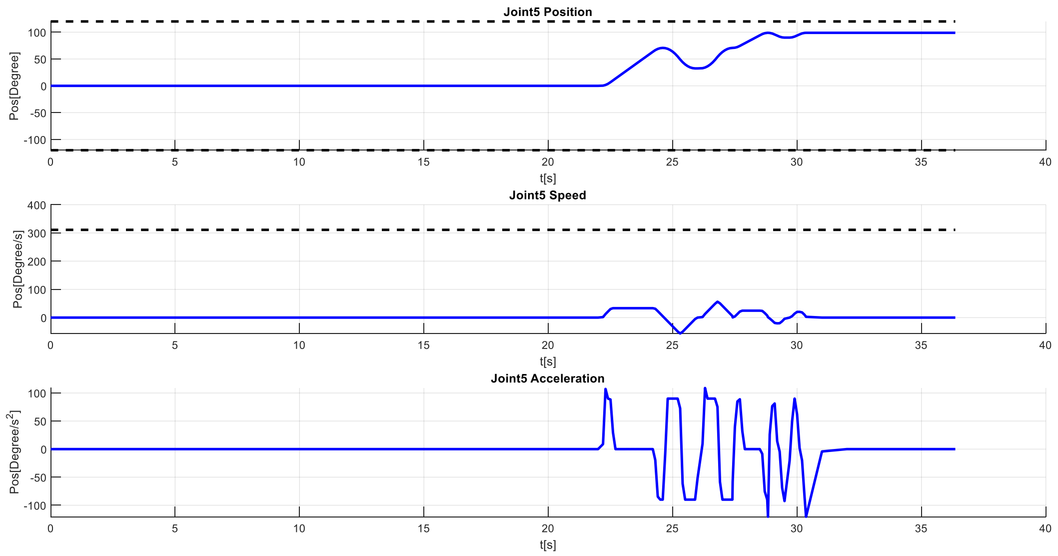


Figure 0-24 KR10 r1100: Joint 5 – position, speed and acceleration profiles

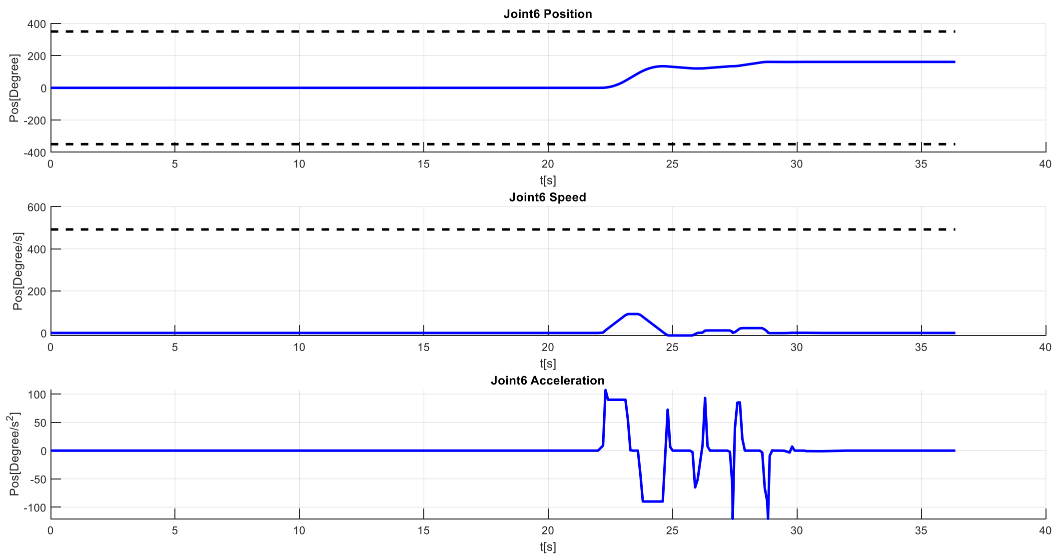


Figure 0-25 KR10 r1100: Joint 6 – position, speed and acceleration profiles

7.1.5 ABB IRB1200 vs KUKA KR6 R700 kinematic comparison

In this paragraph the two robots' joints at the setup's first stop have been kinematically compared in order to understand which one exploits at best its characteristics onto the process. Greater range of manipulation occurs for small six-axis robots a greater probability to rust and spikes in the speed profile occurs sharp changes in the acceleration. It is indeed important to analyze the two robots in the same chart and see which one are more safely designed, guaranteeing less joints corrosion and reliability. In figure 7.26 both ABB and KUKA have smooth position profile with more or less the same manipulability range. KUKA's joint 1 rotates less than ABB's and its speed triangle profile law reaches a safer maximum. In figure 7.27 joints 2 have a position offset because KUKA's joint starts motion at -90°.However, the profiles are extremely similar though. Whereas, in the speed graphs there are two triangles in KUKA's profile replaced by the trapezoids in the ABB's one, providing slighter changes and smaller ranges in the acceleration. In figure 7.28 still the two joints' position cover similar ranges and have a 90° offset. The speed profiles present both trapezoidal law and the accelerations feel similar effects. In figure 7.29 we have the joints 4 that do not rotate at all. In figure 7.30 the position profiles still are similar, but ABB has a sharper motion that anticipates the KUKA's in time. It causes a triangle speed profile law again, but the differences with the KUKA's trapezoids are very small in terms of values and repeatability, making the two acceleration very similar. In figure 7.31 the position and speed profiles are extremely symmetrical and the sharpness in both accelerations are very similar.

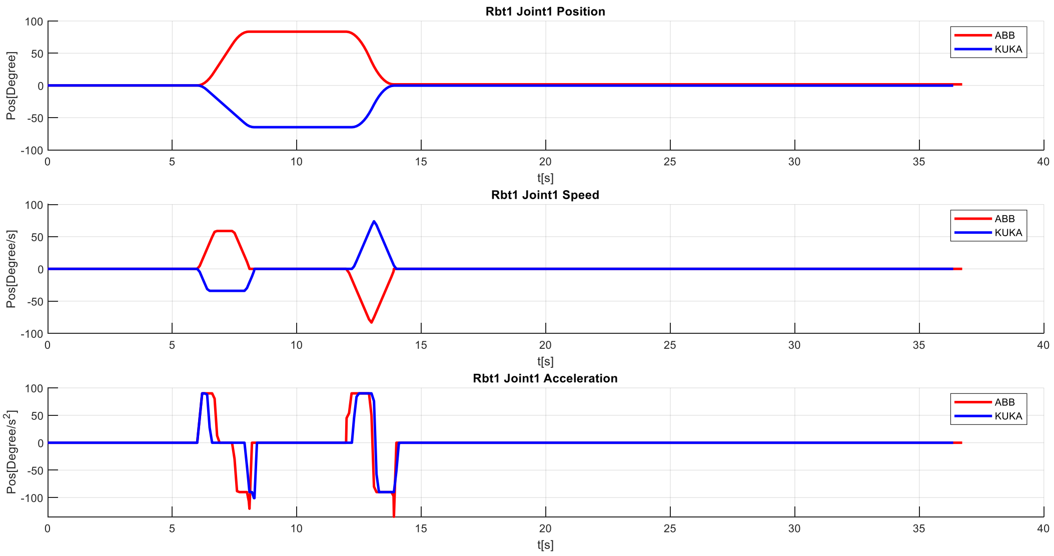


Figure 0-26 IRB1200 vs KR6 r700: Joint 1 – position, speed and acceleration profiles

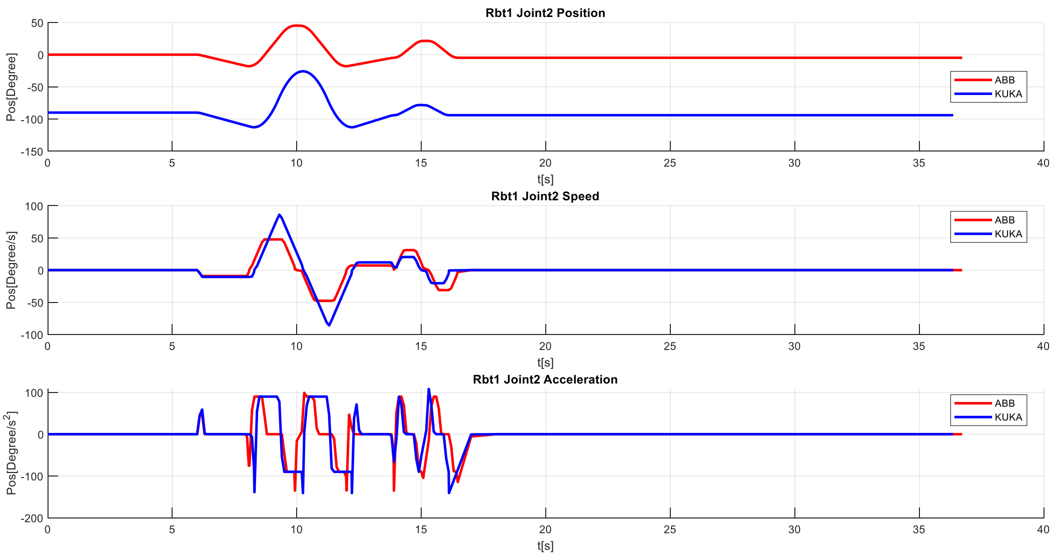


Figure 0-27 IRB1200 vs KR6 r700: Joint 2 – position, speed and acceleration profiles

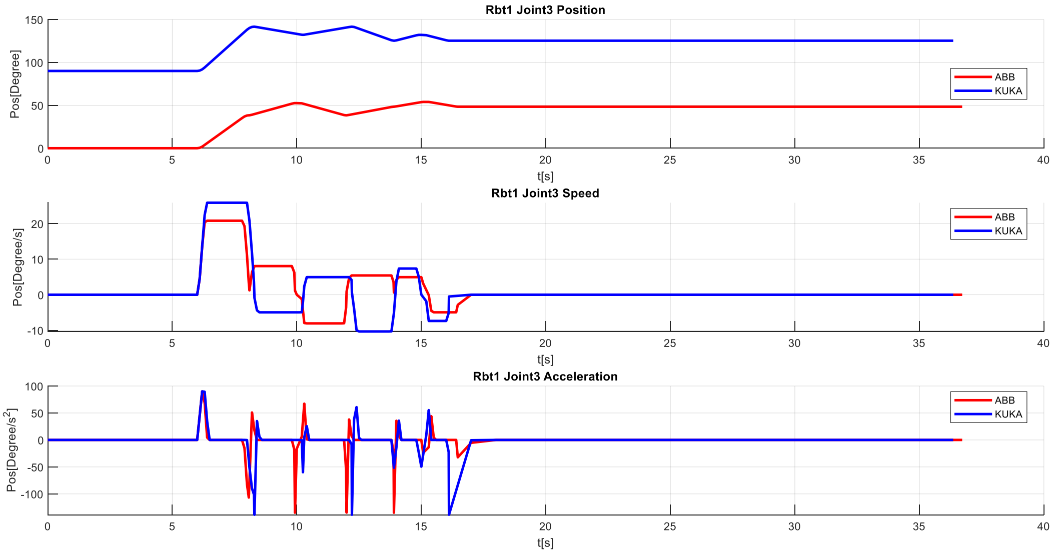


Figure 0-28 IRB1200 vs KR6 r700: Joint 3 – position, speed and acceleration profiles

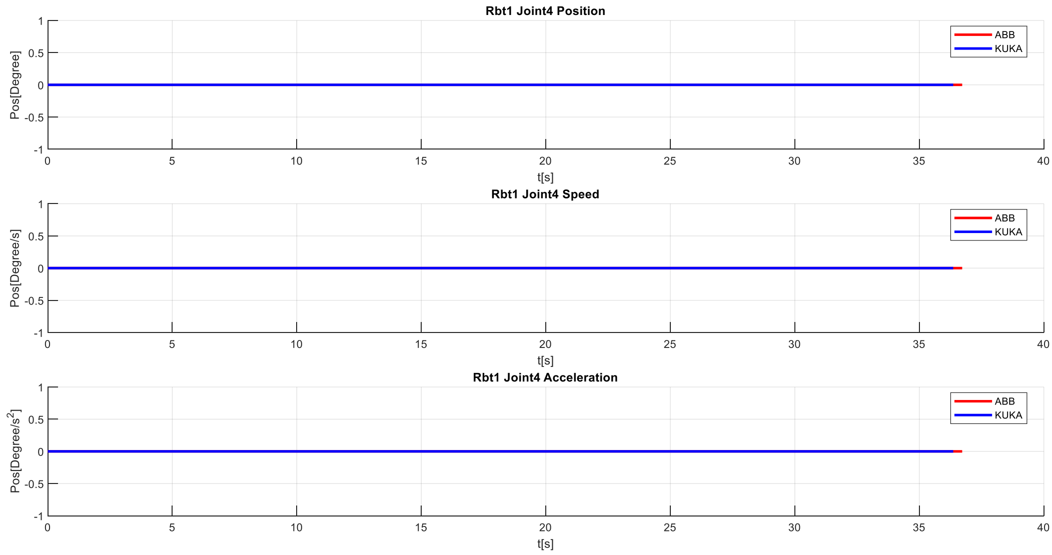


Figure 0-29 IRB1200 vs KR6 r700: Joint 4 – position, speed and acceleration profiles

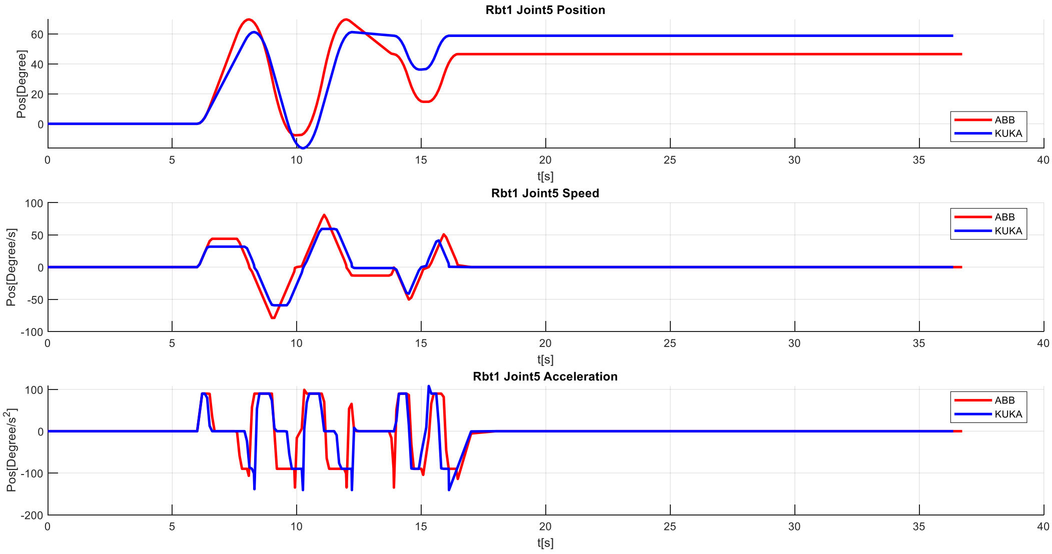


Figure 0-30 IRB1200 vs KR6 r700: Joint 5 – position, speed and acceleration profiles

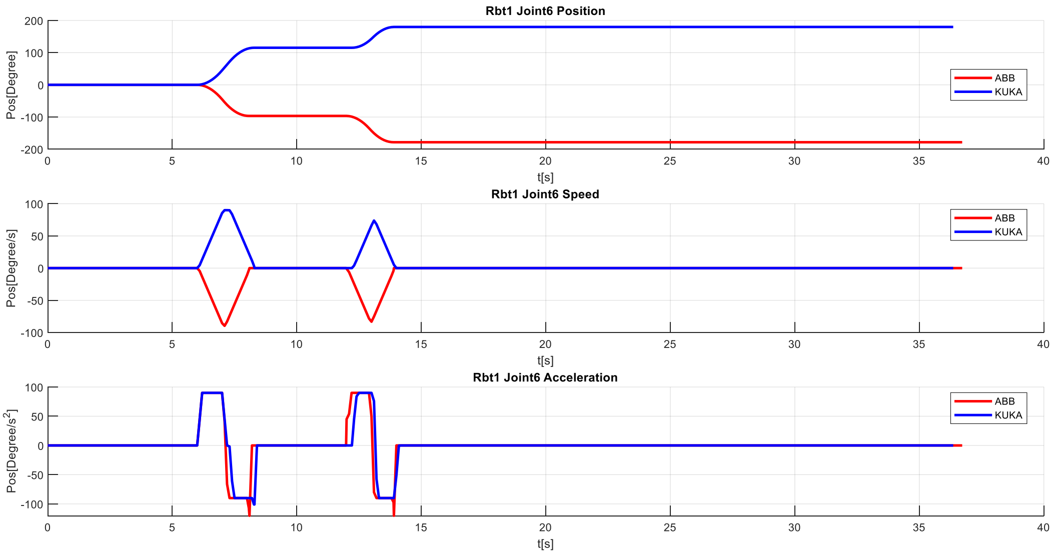


Figure 0-31 IRB1200 vs KR6 r700: Joint 6 – position, speed and acceleration profiles

7.1.6 ABB IRB1600 vs KUKA KR10 R1100 kinematic comparison

In this paragraph the two robots' joints at the setup's second stop have been kinematically compared in order to understand which one exploits at best its characteristics onto the process. Greater range of manipulation occurs for small six-axis robots a greater probability to rust and spikes in the speed profile occurs sharp changes in the acceleration. It is indeed important to analyze the two robots in the same chart and see which one are more safely designed, guaranteeing less joints corrosion and reliability. In figure 7.32 both ABB and KUKA have smooth position profile with a similar manipulability range. KUKA's joint 1 rotates very slightly while the ABB's one has a quite defined trapezoidal profile. It causes for the ABB's joint 1 sharper speed triangle and trapezoidal profiles law reaching higher values in the speed profiles and more fluctuations in the acceleration. In figure 7.33 joints 2 have a position offset because KUKA's joint starts motion at

-90°, it also has a greater range, it causes, in the speed graphs, two trapezoids with wide range and a few more fluctuations in the acceleration profile. In figure 7.34 still KUKA's position and speed cover wider ranges and have an initial 90° offset. KUKA's speed profiles present triangle law, while ABB's joint has only trapezoidal profiles law. Indeed, the KUKA's acceleration feels the effect more than ABB's one. In figure 7.35 we have the joints 4 that do not rotate at all. In figure 7.36 the position profiles still are similar, but ABB has a sharper motion that anticipates the KUKA's in time. It causes a triangle speed profile law again and higher maximum speed, but the differences with the KUKA's trapezoids are very small in terms of values and repeatability, making the two acceleration very similar. In figure 7.37 the position and speed profiles are extremely symmetrical and the sharpness in both accelerations are very similar.

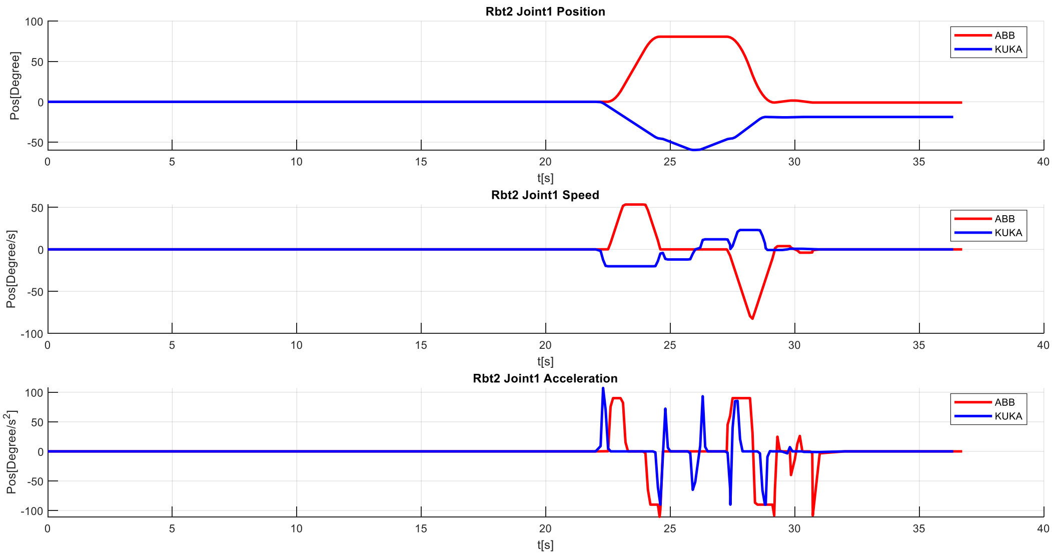


Figure 0-32 IRB1600 vs KR10 r1100: Joint 1 – position, speed and acceleration profiles

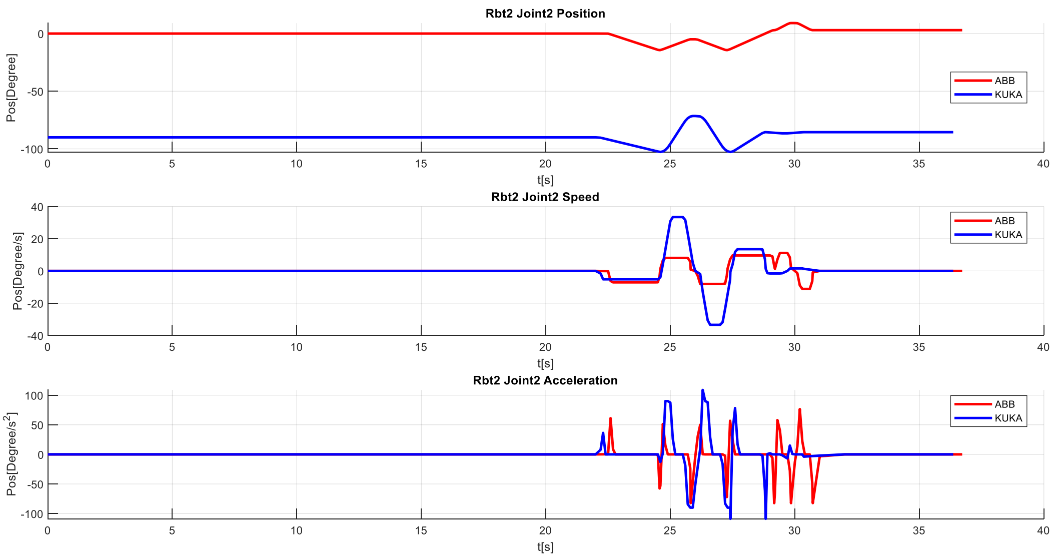


Figure 0-33 IRB1600 vs KR10 r1100: Joint 2 – position, speed and acceleration profiles

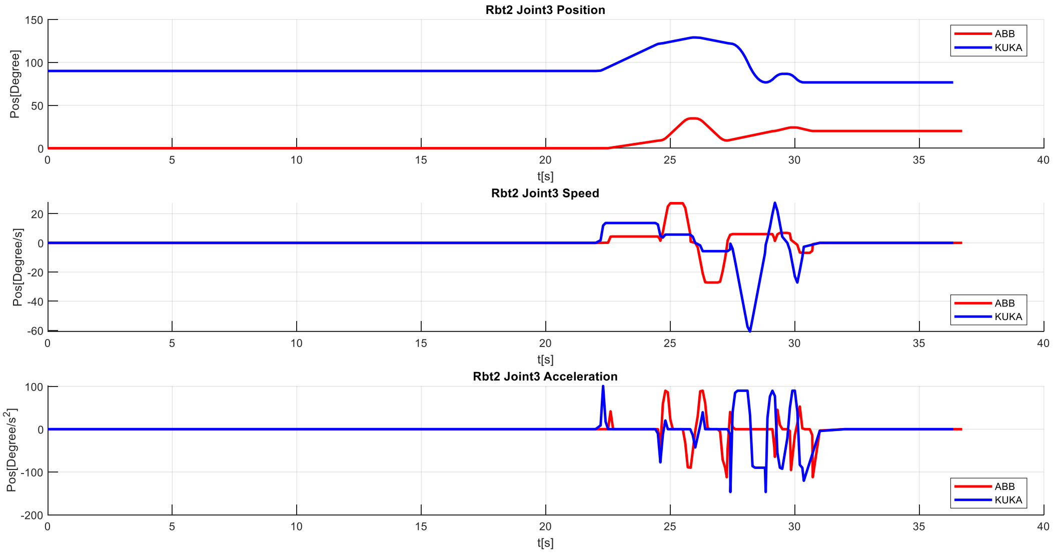


Figure 0-34 IRB1600 vs KR10 r1100: Joint 3 – position, speed and acceleration profiles

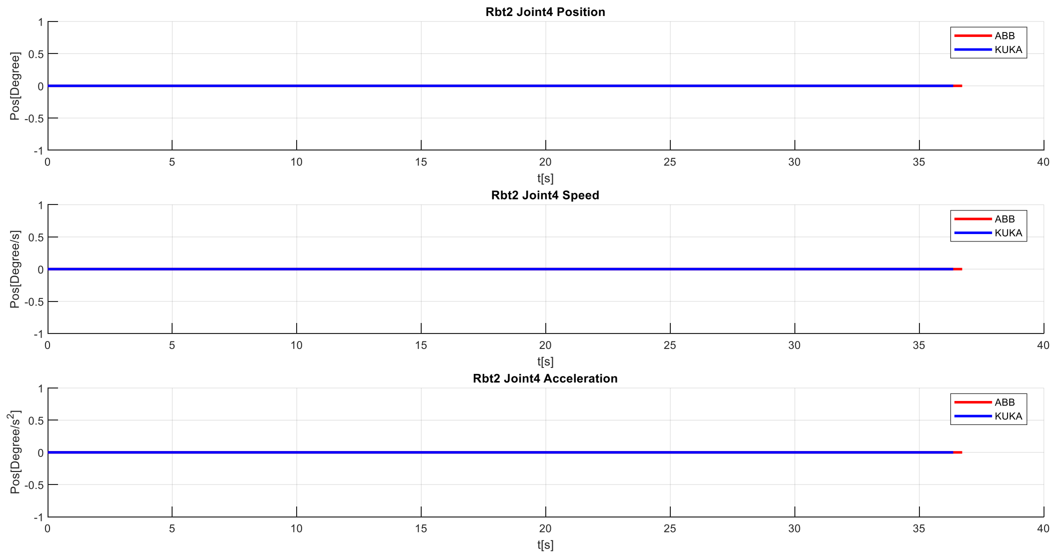


Figure 0-35 IRB1600 vs KR10 r1100: Joint 4 – position, speed and acceleration profiles

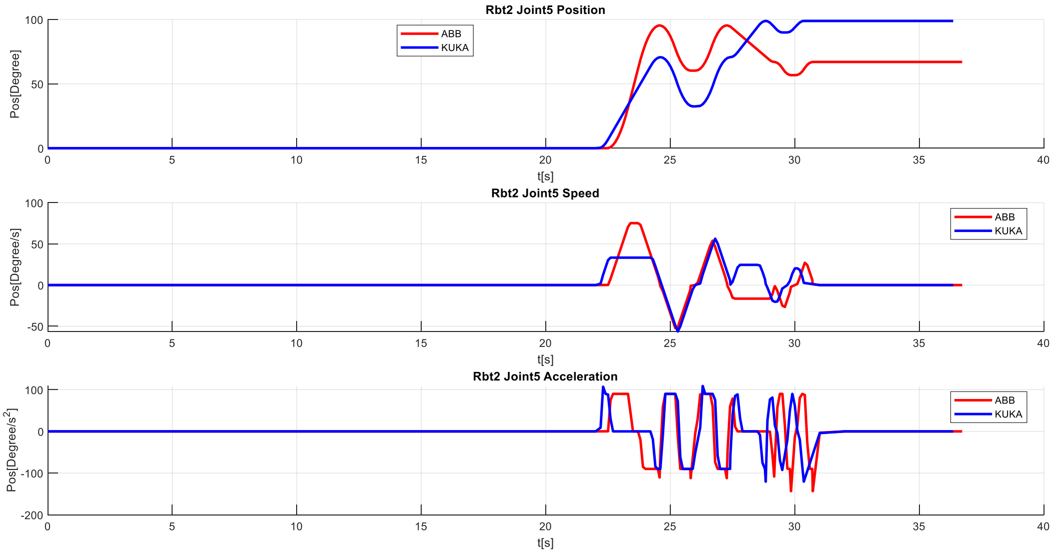


Figure 0-36 IRB1600 vs KR10 r1100: Joint 5 – position, speed and acceleration profiles

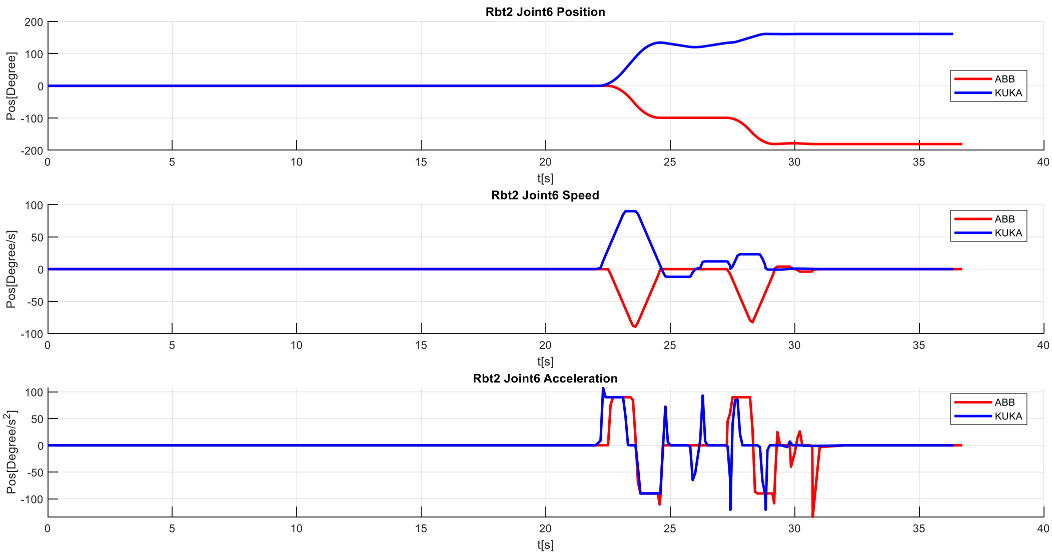


Figure 0-37 IRB1600 vs KR10 r1100: Joint 6 – position, speed and acceleration profiles

7.1.7 ABB vs KUKA TCPF speed comparison

In this paragraph the tool center point frame of ABB's and KUKA's robots both at first and second stops have been proposed. It appears that the ABB's TCPFs have higher speeds than KUKA's, especially while approaching the part. Several spikes can be noticed in here and it is meaningful for robots that manage to quickly react at inputs and to conclude the task in the shortest time possible.

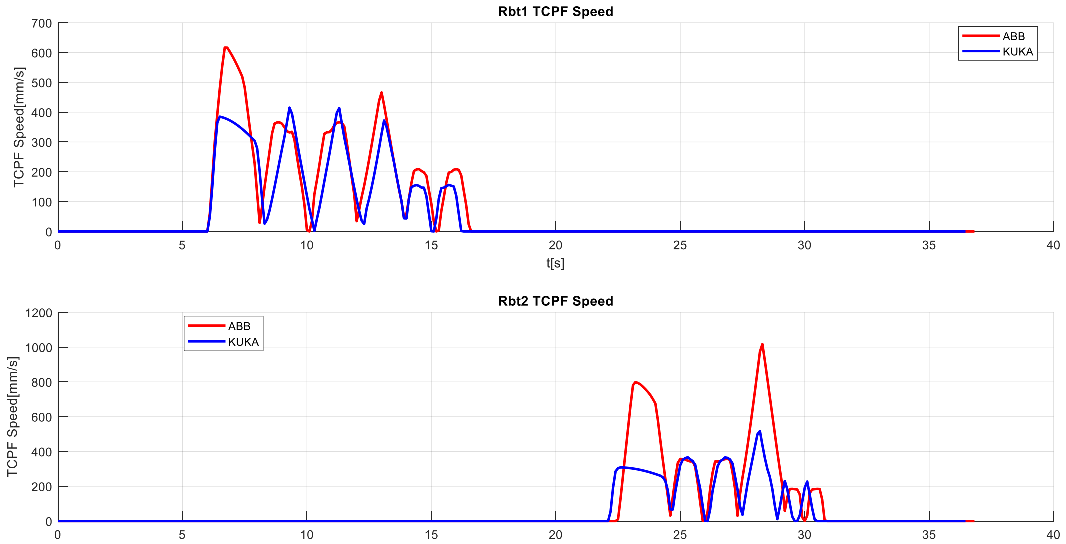


Figure 0-38 IRB1200 vs KR6 r700 and IRB1600 vs KR10 r1100 – TCPF speed profiles

7.2 Production planning

In this paragraph the manufacturing outputs have been collected. In order to understand how to schedule the process and fix the bottle necks, it has been proposed a comparison between ABB and KUKA in terms of production timings. The next steps aim to divide the whole process into sub-processes to analyze which part of each setup contains weaknesses in terms of time either considering the operations of the robots and the processes of the system.

7.2.1 Time study

The first table and chart describe each operation happening into the Process Simulate environment before and after each robot stop and it is the process time study. The part flows three times: to the first robot, to the second robot and to the destination before re-entering into the conveyor's process. They all only depends on the conveyor performance and there is no difference between ABB's and KUKA's setups. Plus, there are the two pick and place operations for each setup, so seven process operations in total. The lead time is the time between the start and the end of a production process.

Task	ABB	KUKA
Flow Operation1	6	6
Pick Operation1	5,98	6,21
Place Operation1	4,48	3,9
Flow Operation2	6	6
Pick Operation2	4,8	5,3
Place Operation2	3,45	2,95
Flow Operation3	6	6
Lead Time	36,71	36,36

Table 0-1 Process time study – setups comparison

As it shown in the chart, flow operations are all the same and have been included to complete the system's sub-processes. Whereas, both ABB's robots are faster in the pick operation, the KUKA's are in the place operation. All the differences are less than one second and the bottle neck results to be pick at first stop.

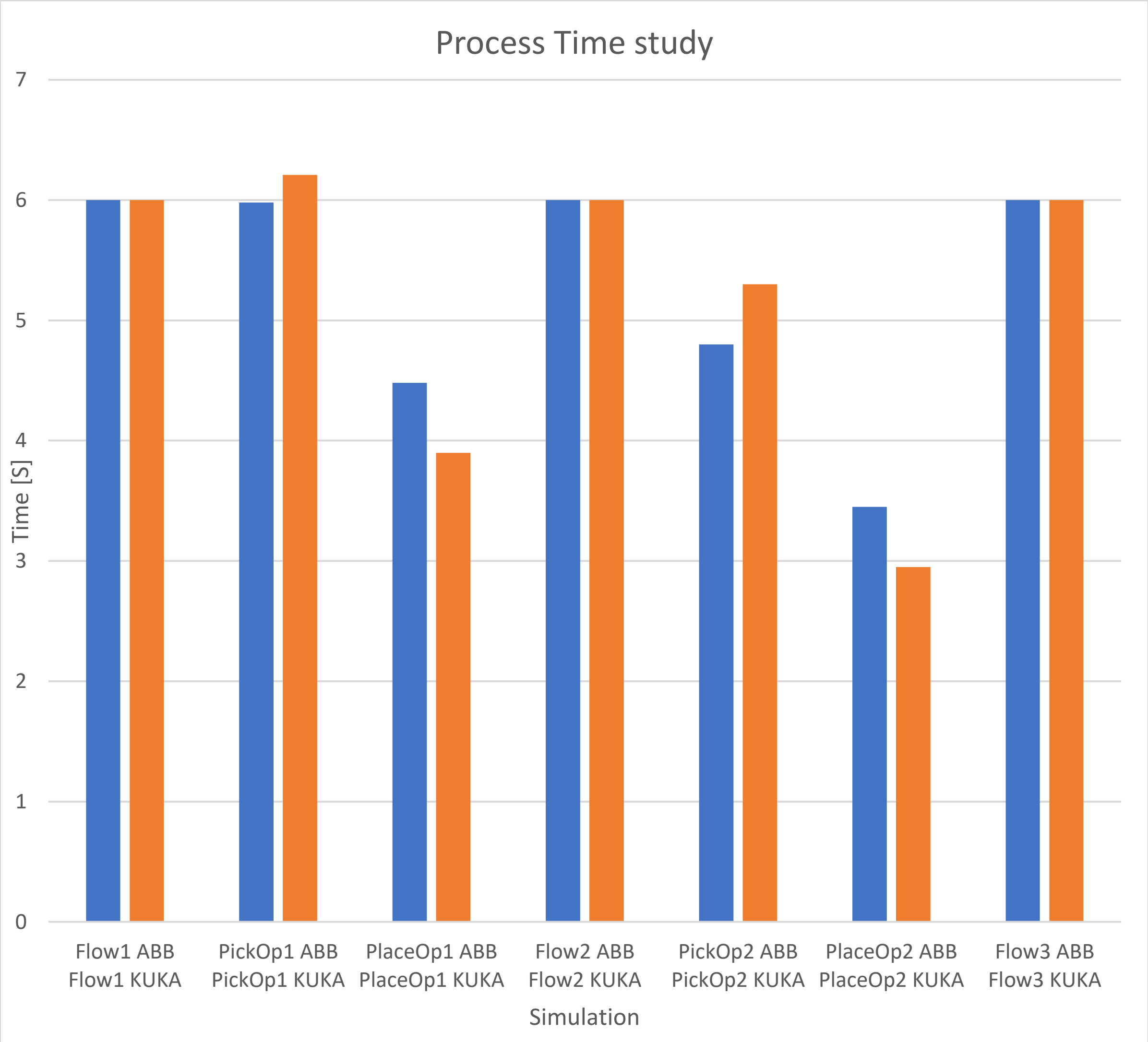


Figure 0-39 Process time study – setups comparison

The second table and chart describe each operation happening into the Process Simulate environment while each robot operatically picks and places its part. The robots' motion has been easily grouped as their location: there is a reaching motion of the pick operation, a grasping motion of the pick operation, a releasing motion of the pick operation, a reaching motion of the place operation, a grasping motion of the place operation and releasing motion of the place operation, so six sub-operations per the two robots at each stops makes twelve in total.

Location description	ABB - IRB1200	KUKA - KR6_700
Reach_Pick1	2,07	2,28
Grasp_Pick1	2,05	1,97
Release_Pick1	1,85	1,97
Reach_Place1	1,91	1,69
Grasp_Place1	1,39	1,2
Relase_Place1	1,19	1,00
	ABB - IRB1600	KUKA - KR10_1100
Reach_Pick2	2,10	2,49
Grasp_Pick2	1,45	1,5
Release_Pick2	1,25	1,3
Reach_Place2	1,90	1,42
Grasp_Place2	0,87	0,86
Relase_Place2	0,67	0,66

Table 0-2 Robots time study – setups comparison

As it can be viewed in the chart, ABB's robots are confirmed to be faster for the pick operation, while KUKA's are for the place operation. There is no great difference between IRB1200 and KR6 sub-motions at first stop (less than 0.3 s), it increases a bit more at the reaching motion of the IRB1600 and KR10. KUKA's robot gains 0.48 s at the reach place pose, while ABB's robot gains 0.39 s at the reach pick pose. As it was shown before because of the lead time, KUKA's robots are faster than ABB by working in sequence.

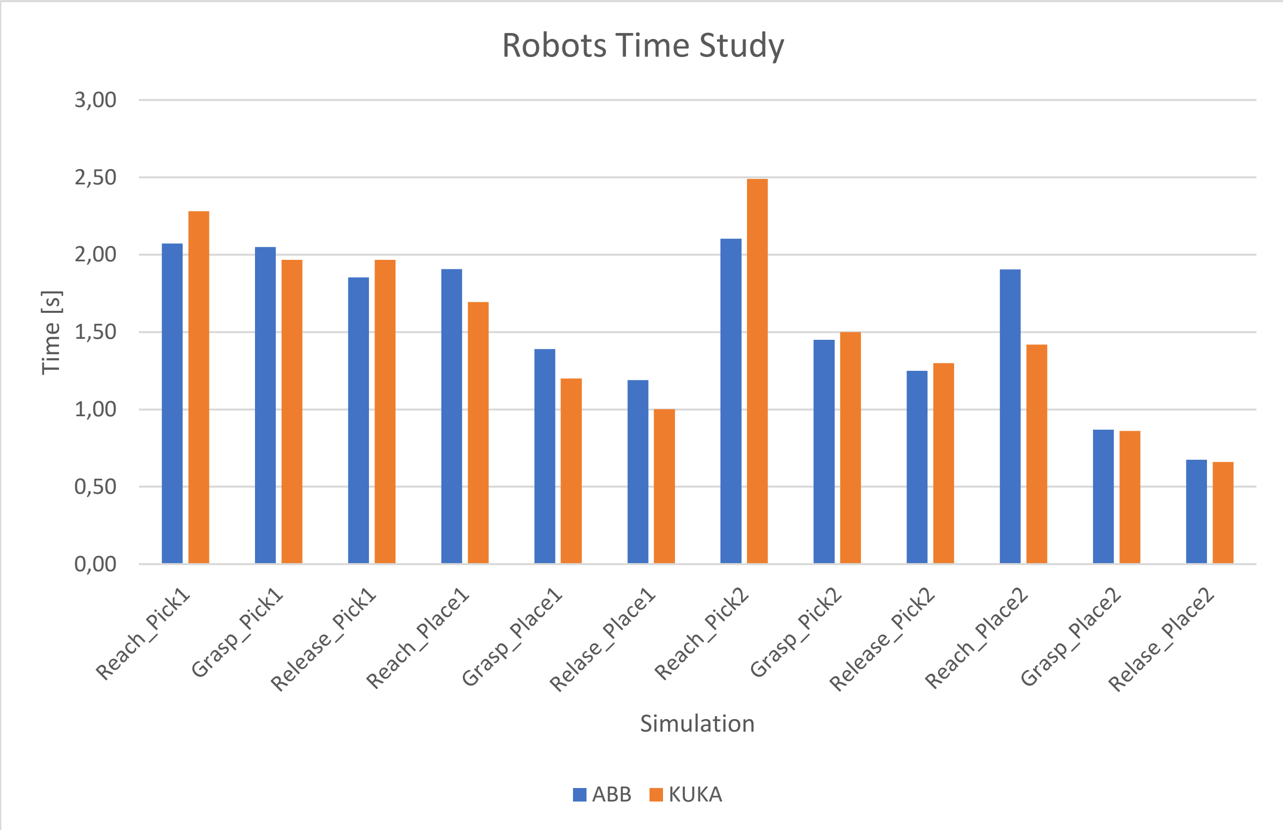


Figure 0-40 Robots time study – setups comparison

7.2.2 Cycle time and throughput

In this paragraph the already mentioned time differences have emerged by analyzing the robots' cycle time and their throughput. Cycle time is the total time from the beginning to the end of a specific process. It is a more mechanical measure of the process capability than the lead time.

Robots	Cycle time
ABB - IRB1200	10,46
ABB - IRB1600	8,26
KUKA - KR6_700	10,11
KUKA - KR10_1100	8,25

Table 0-3 Robots cycle time

While the second stops robots only have 0,01s cycle time's gap, the first stop ABB is 0,35s slower than its relative KUKA. In the chart it is clear to see how the first ABB's robot is the bottle neck of the setups' scheduling.

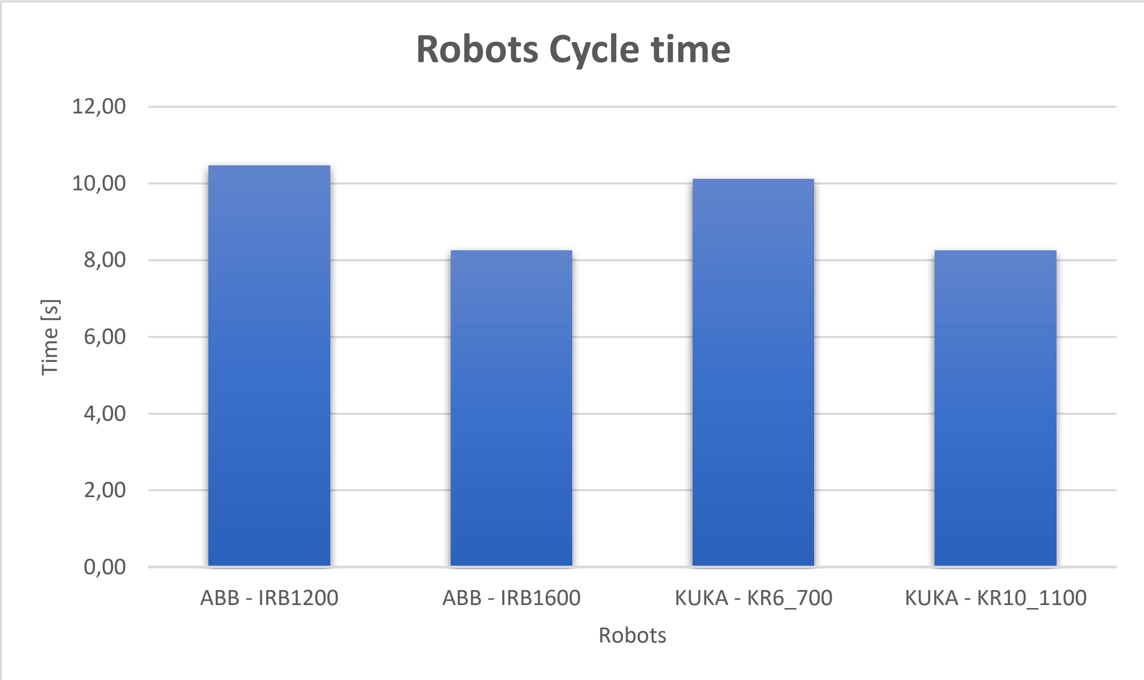


Figure 0-41 Robots cycle time

On the other hand, throughput is the maximum rate of production or the maximum rate at which something can be processed. In other words, it the amount of material passing through a system or process.

Robots	Processing time	Part in hour	Part in day
ABB - IRB1200	36,64	98,25	2358
ABB - IRB1600	36,64	98,25	2358
KUKA - KR6_700	36,12	99,67	2392
KUKA - KR10_1100	36,12	99,67	2392

Table 0-4 Robots throughput

It is easy to understand that when cycle time is higher, throughput is lower. Indeed, since the two robots work in sequence the meaningful time difference between IRB1200 and KR6 makes throughput better for KUKA setups as the amount of production hours increases. While only one sub-assembled part can virtually differ between the two robotics setups, in the next chart it is clear how this increasing gap of sub-products for an ideal 24-hours working day awards KUKA's setup as the first stop ABB's robot is the bottle neck of the process.

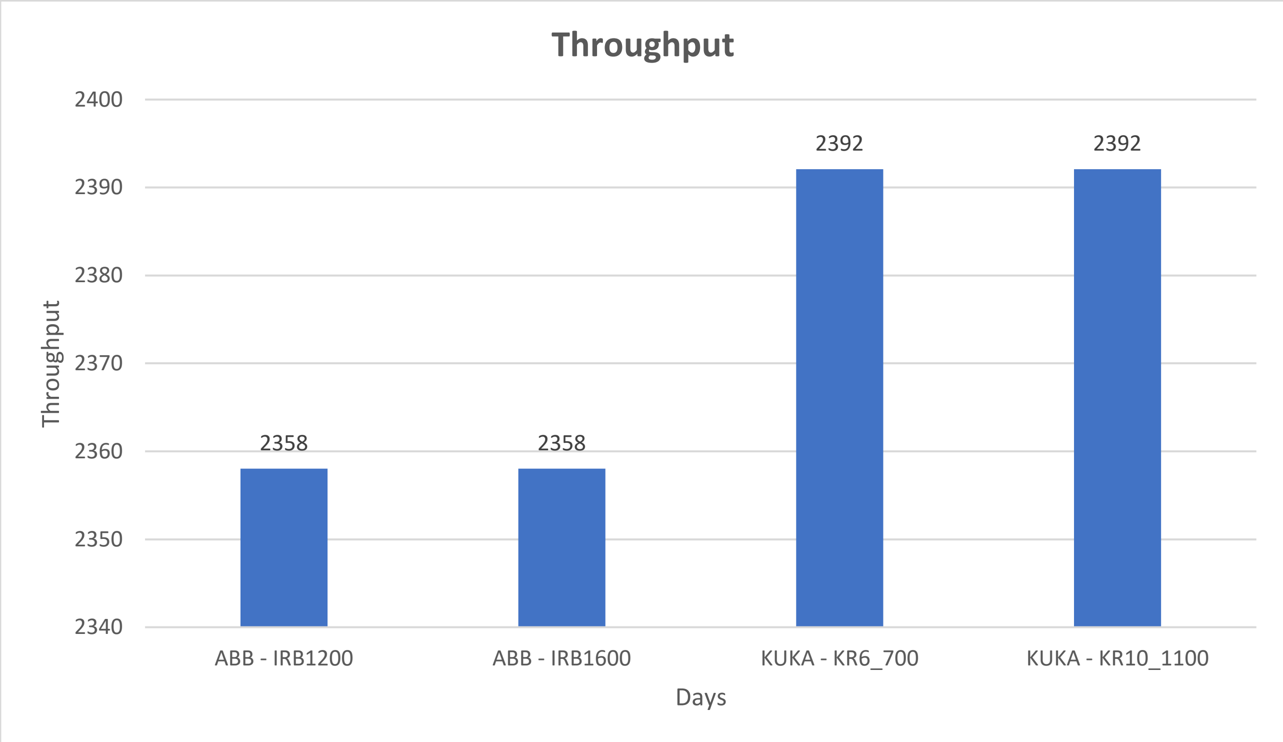


Figure 0-42 Robots throughput

Chapter 8

Conclusion

This project can be considered as a starting-point for a smarter and smarter system, it has been impressive to deal with the several parts that have been touched during the work: from the Conveyor System giving a true idea of how industrial processes are done, to robots' trajectory planning. The results are quite optimizable and future improvements are expected, the process idea is very simple indeed but once the new model has been created into the Process Simulate environment is easy to complicate and adapt to a more complete autonomous production. The autonomous tasks have been virtually proven though, and the simulations will be a solid background for future improvements. The process idea has been a valid example for the adapted method with high awareness of refereeing to a potential future project for the real process. The tools used in this project are multiple: Process Simulate, NX 12, Plant Simulation and TIA Portal from the Siemens software's world were used to give birth to the new robotics setup virtual model and their advantages and limits in order to predict the actual application have been understood. Especially the whole learning of Process Simulate standalone is a plus a project like this provided me and it can be exploited in new applications for both academical and industrial purposes. Furthermore, the collected data has been implemented into a thousand- eighty-four lines MATLAB code that will not be part of the M.Sc. thesis.

As it has been outlined in the introduction of this paper, the choice to implement a new starting-point to automatize the system was led by a strong desire to experiment new technological integration methods for modern factories. Industry 4.0 addresses connectivity and flexibility of such robotics applications in a factory setting. This can affect how both traditional and collaborative robots are used. Traditional robots, whether small or large, can work at very high speeds while carrying larger weights, making themselves suitable for many manufacturing, packaging and assembly applications, but they need to have protected spaces to work near humans. Collaborative robots are designed to work at low speeds that are safe near humans without the need of any additional protection and their applications tend to be totally integrated and direct replacement of human operations. Hence, it affects also how application cells are designed by making them more flexible using any type of robot for a great variety of applications. One of the main advantages is to be able to design the application cell that can easily handle rapid changes in product manufacturing or flow. It is, therefore, very attractive for small-to-medium factories that generally cannot afford custom setups for each product and quick changeover that any kind of robot. All these Industry 4.0 concepts will require inclusion of smart and cyber technology in traditional manufacturing environment.

To sum up the brainstorming ideas behind the project that have been collected during the work, some following brief commenting sections have been listed.

Future work

This project, being completely new, has several parts that can be improved, between software complementing operations to programming aspects. As far as it concerns the latter, it has been discussed the idea of implementing, trying to test in depth Industry 4.0's dictates, a part-centric robot program to transfer programming of applications to a higher part-level by describing what process need to be done to manufacture the part. It is a switch to the robot role's perspective, that means we are not directly writing application programs for specific robots or machines. The entire manufacturing process can be stored in a cloud or PLC and smart connected robots and machines in a flexible factory setting inquire the cloud if there is any application available that matches their capability and setup. In this way, parts are directed to proper machines or robots where manufacturing processes can be performed in a proper sequence to manufacture the part. One of the important steps to realize this way of programming industrial production is to have a smart flow control of parts to different machines in the factory and automatic conversion of part-centric programs to specific robot or machine programs. Both traditional and collaborative robots will need to have vision and other sensor interfaces to allow for handling of a variety of different parts and new software interfaces to allow for connection with the cloud and interpretation of high-level instructions to robot programs. This will eventually help realize a flexible manufacturing environment on the factory floor. All this improvement may be object of future works, both for a M.Sc. thesis or a Ph.D. research, not only in Mechanical Engineering.

Class activities

One of the aims of this research is to help design a virtual system that can be ultimately object of a classes project at Oakland University. Siemens will lend his product in behalf of a the Industrial Robots and Automation course in the summer semester, in which both in lectures and laboratory activities the new version of the Conveyor System can be analyzed, letting groups of students focus on several aspects of the application cell such as: robots and cell safety, PLCs and cell control as has been mentioned in the section above, simulation and offline programming, process control, cycle time, vision based robot control, pick and place mechanism, RFID and bar code readers. Groups would eventually coordinate to implement one working application process.

Software platform

The Siemens Tecnomatix platform that has been used is very good and let simulate many devices and environments very accurately. The Process Simulate standalone version that has been installed is enough to predict the robotics joints' behavior, even if the optimization of the path planning surely can be improved as it has been said in the previous chapter. Although, the academic version I have worked with has pointed out a restriction by Siemens while setting the instruction for the Learning Advantage's Process Simulate advanced courses. It has been caused by a missing license to set the "real" robots' controllers, used by the actual robotics companies indeed, as the RCS (realistic control simulation) mode could not be turned on. It would have allowed to physically test the robots, giving a chance to monitor the kinematics results beyond the simulations. It also would have let Process Simulate give us as output joints' power consumption profile, then a slight piece of information about the dynamic efficiency. Although, the torques would have been computed anyway because the provided models did not have enough details in order to set masses distribution to iterate the recursive Newton-Euler method.

Physical system

The present system has been shown in Chapter 2 and its new version has not been built yet. Il will be direct interest of Oakland University Industrial and System Engineering Dept. and Siemens PLM in Troy, MI to schedule the construction and implementation of robots. At first, the solution has been a virtual model because it would not be very cheap of course. However, the class project represents a solid investment for school to widen students' perspective that have taken complementary courses and for companies' trainees as we well.

Bibliography

[1] *How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective*. Malte Brettel, Niklas Friederichsen, Michael Keller, Marius Rosenberg.

[2] *Industry 4.0: A survey on technologies, applications and open research issues*. Yang Lu. University of Kentucky, Lexington, KY, USA, University of Manchester, Manchester, UK.

[3] *Human-Machine-Interaction in the Industry 4.0 Era*. Dominic Gorecky, Mathias Schmitt, Matthias Loskyll, Detlef Zühlke. Innovative Factory Systems, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany.

[4] *Cobots vs Robots*. URL: http://hmkdirect.com/news/cobot-vs-robot/.

[5] *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*. Michael Rüßmann, Markus Lorenz, Philipp Gerbert, Manuela Waldner, Jan Justus, Pascal Engel, and Michael Harnisch.

[6] *Wikipedia - Conveyor System*. URL: https://en.wikipedia.org/wiki/Conveyor_system.

[7] *Final Project One Page*. Leonardo Sawaya. Oakland University, Rochester Hills, MI, USA.

[8] *Automation of mechanical systems*. Vladimir Viktorov, Federico Colombo. Ed. by CLUT, 2013.

[9] *Visualize an event-based simulation model made in Process Simulate*. Andreas Larsson, Henirk Nilsson. Chalmers University, Goteborg, Sweden.

[10] *Simulation Modelling using Practical Examples: A Plant Simulation Tutorial.* Martijn R.K. Mes, University of Twente, Enschede, Netherlands.

[11] *Siemens PLM Software, Process Simulate standalone Learning Advantage course*. URL: https://training.plm.automation.siemens.com/Process Simulate standalone

[12] *Siemens PLM Software, Plant Simulation Learning Advantage course* URL: https://training.plm.automation.siemens.com/Plant Simulation

[13] *Siemens PLM Software, NX Learning Advantage course* URL: https://en.wikipedia.org/wiki/Siemens_NX

[14] *Tia Portal Openness*. Siemens AG Digital Factory.

[15] *Wikipedia - PLM.* URL: https://en.wikipedia.org/wiki/ Product lifecycle.

[16] *Robot Worx - What are six axis Robots?* URL: https://www.robots.com/faq/.

[17] *ABB 6 axis - IRB1200 data sheet*. URL: https://new.abb.com/products/irb-1200/.

[18] *ABB 6 axis IRB 1200 technical specification*. URL: https:// new.abb.com/products/irb-1200/.

[19] *ABB 6 axis - IRB1600 data sheet*. URL: https://new.abb.com/products/irb-1600/.

[20] *ABB 6 axis - IRB 1200 product specification*. URL: https:// new.abb.com/products/irb-1600/.

[21] *KUKA 6 axis - KR6 R700 data sheet.* URL: https://kuka.com/industrial-robots/KR6-R700-sixx/.

[22] *KUKA 6 axis - KR10 R1100 data sheet.* URL: https://kuka.com/industrial-robots/KR10-R1100/.

[23] *KR AGILUS - technical specifications*. URL: https:// kuka.com/industrial-robots/.

[24] *Forward Kinematics lectures*. Khalid Mirza. ECE4500-5500, Oakland University, Electrical and Computer Science Department, Rochester Hills, MI, USA.

[25] *Trajectory Planning lectures*. Khalid Mirza. ECE4500-5500, Oakland University, Electrical and Computer Science Department, Rochester Hills, MI, USA.

[26] *Siemens PLM Software, Process Simulate standalone Learning Advantage course, Basic concepts*. URL: https://training.plm.automation.siemens.com/Process Simulate standalone.