Commissioning and Programming of a Bulk Bag Robot

Sunrob Robotics Oy

LAB University of Applied Sciences

Bachelor of Engineering, Mechanical Engineering and Production Technology

2021

Berke Türen

Abstract

Author(s)	Publication type	Completion year
Türen Berke	Thesis, UAS	2021
	Number of pages 36	

Title of the thesis

Commissioning and Programming of a Bulk Bag Robot

Sunrob Robotics Oy

Degree

Bachelor of Engineering

Name, title and organisation of the client

Jouni Könönen, Senior Lecturer, LAB University of Applied Sciences

Sami Pörsti, Founder of Sunrob Robotics Oy, Sunrob Robotics Oy

Abstract

The thesis work talked about the process of the robot commissioning and programming project and determined the way of solving a factory operation by semi-autonomous system replacement. Generally, thesis work content was based on working process with Kuka Workvisual 6.0 computer software and Kuka manuals for robot installations, operating instructions, programming, and safety configurations. The thesis focused on teaching the Kuka system & principles and completing a work task simultaneously. The project proceeded in optimal manner and yet, the robot and external units were able to operate conveniently. The master program of the robot was completed without errors.

Keywords

robotics, robot programming, robot commissioning, autonomous

Contents

1 Introduction

This is the thesis work that covers the knowledge regarding robot commissioning and programming as a guideline. Robot commissioning project is held by Sunrob Robotics in the spot of the client company. In the report, there will be an order of the progression steps that will be followed up and this work will primarily cover the steps after the installation of the Robot.

The subjects that would be detailly explained are background information of Sunrob Robotics Oy, general definition and purposes of Kuka robots, overview of the project & tasks, robot commissioning, start-up steps of Kuka robot, safety configurations and programming on Kuka robot.

For the computer aid aspects of the work, which is mainly related to thesis work, Kuka Workvisual 6.0, Visual Studio Code are used during the commissioning process. Kuka Workvisual 6.0 is the one main program for Kuka robots moreover, most of the settings for the robot are being handled by that software. Visual Studio Code is used as a support program due to its benefits with Kuka Robot Language editor. Kuka manuals are used as essential sources.

Kuka robot model that on the used for the work is KR 210 R3100-2 F. Some of the Kuka robots below as Figure 1.

Figure 1. A representative picture of Kuka robots (Techvitas)

2 Background information of Sunrob Robotics Oy

2.1 General definition

Sunrob Robotics Oy is a Lappeenranta based company that provides autonomous and robotic industrial solutions for the clients' competitiveness and innovative methods. Mainly their solutions contain the highest efficiency of repetitive work with minimum human factor in the work where it brings to the purpose of usage of robots in their products.

2.2 Solutions of Sunrob Robotics Oy

There are several types of services which are being presented by Sunrob Robotics Oy for the clients. Autonomous system simulations, mechanical design and manufacturing parts & tools, robot installation and commissioning, safety considerations of robot and robot programming are the primarily services. Obviously, projects have big number of diversities depending on the client's needs and usually all combinations of services, which are mentioned above, are used together for the final product. Here is an operation below as Figure 2.

Figure 2. Sunrob Robotics operations (Sunrob Robotics)

2.3 Reason of Kuka robot

Kuka robot is a familiar brand for Sunrob Robotics’ background and previous works that are experienced with it. Thus, it stands out as the first preference for Sunrob Robotics Oy.

There are also some explanations why Kuka brand is preferred. Apparently, according to (Sale 2015), Kuka robots have long lifecycle of worktime that there is no requirement of changing of the parts in the robot therefore, there would be lowered maintenance needs. Although there is an issue with a part, it could be replaced easily by the clever design type of Kuka robots.

Concerning the thesis report, Kuka robot language is a proprietary programming language which has connections with another programming language called Pascal (Braumann & Brell-Cockan 2011, 243).

3 General definition and purposes of Kuka robots

3.1 Overview of Kuka

Kuka is a German based company which provides intelligence automation solutions for the respective industries. Kuka is involved in various of fields of applications such as robot systems, production machines, production systems, automated and guided vehicle systems, mobility and process technologies. The field that is more related to the thesis topic and the used product from Kuka is in robot systems. Kuka robot can be differed by their application purposes and 6-axis-robot will be the target robot for the thesis topic. As well as industrial robot production, there are also Kuka software for commissioning and Kuka education platform for the advance experience of Kuka robots (KUKA 2021).

3.2 General mechanism and essentials of Kuka robot

Like most of the automation systems, industrial robots serve usually for repetitive tasks in factory areas. Kuka’s 6-axis-robots are one of them, and they can be used for more flexibility needed tasks. With a wide range of movement of 6-axis-robots variety of tasks can be increased such as insert loading automation and packing automation (Robotic automation system). As it can be seen in Figure 3, Kuka industrial robot consists of four essential elements and these are manipulator (1), robot controller (2), teach pendant (3) and connecting cables (4) (Kuka Roboter GMBH 2015a, 15). Teach pendant is the touch panel module that manipulator is directly positioned, and positions are taught via robot controller and cables. Robot motion can be activated only by the teach pendant.

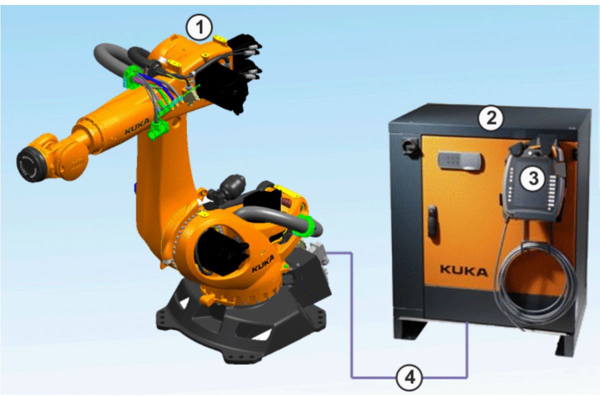

Figure 3. Kuka industrial robot (Kuka Roboter GMBH 2015a)

Kuka robots are moving depending on the motion type that is selected on the teach pendant. Joint coordinate motions and cartesian coordinate system motions are the two motion types, and they can be operated in different base systems (Kuka roboter GMBH 2000, 59). In fact, human might have an issue to figure out motions on joint coordinate motions for instance, a linear motion, therefore cartesian coordinate system is the preference for that purpose and robot converts cartesian coordinate motions to joint coordinate systems right before the motion is executed as Figure 4 (Kuka roboter GMBH 2000, 60).

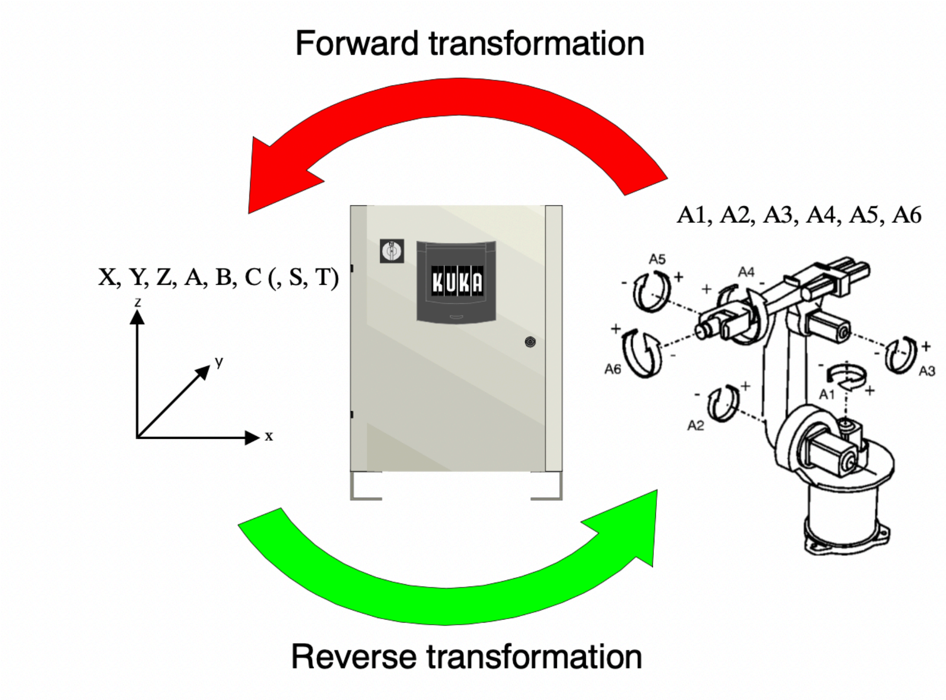

Figure 4. Coordinate systems transformations (Kuka Roboter GMBH 2000)

Joint coordinate motions

As it is illustrated in Figure 5, joint coordinate motions use the six different axes of robot joints (A1, A2, A3, A4, A5, A6). There are restrictions of all these axes thus, the robot that is used for thesis work cannot reach 360° motion in a single axis. Axes motions in joint coordinate system can move only in point-to-point motion commands because an axis motion cannot follow a cartesian linear motion by itself (Kuka roboter GMBH 2000, 60).

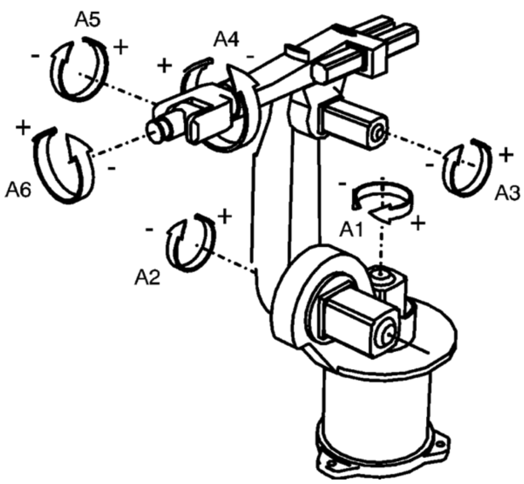

Figure 5. 6 axes motion (Kuka Roboter GMBH 2000)

Cartesian coordinate motions

Cartesian coordinate motion is using 3 axes which are perpendicular to each other in a space and named as X, Y, Z coordinates (Kuka roboter GMBH 2000, 61). A motion in cartesian coordinate motion is the combination of the impact of X, Y, Z vectors. Therefore, robot takes the shortest distance of this combination of the vectors and that motion is called translational motion as Figure 6 (Kuka roboter GMBH 2000, 61). Despite translational motion, there are three rotational motions (A, B, C) where they rotate in X, Y, Z coordinates individually as it is shown in Figure 7. (Kuka roboter GMBH 2000, 62). In other words, every rotational motion belongs to one single coordinate.

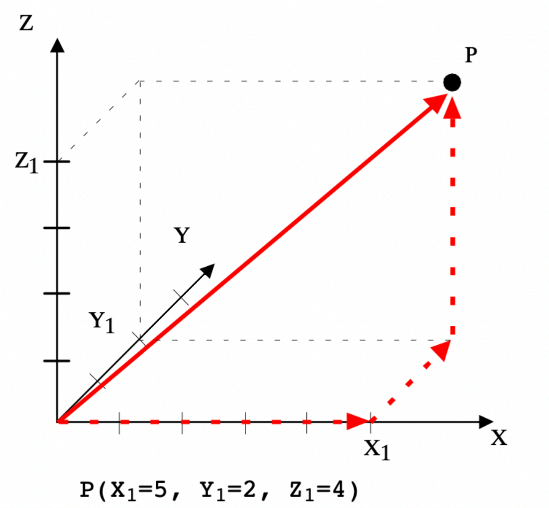

Figure 6. Cartesian motion with the combination of all coordinates (Kuka Roboter GMBH 2000)

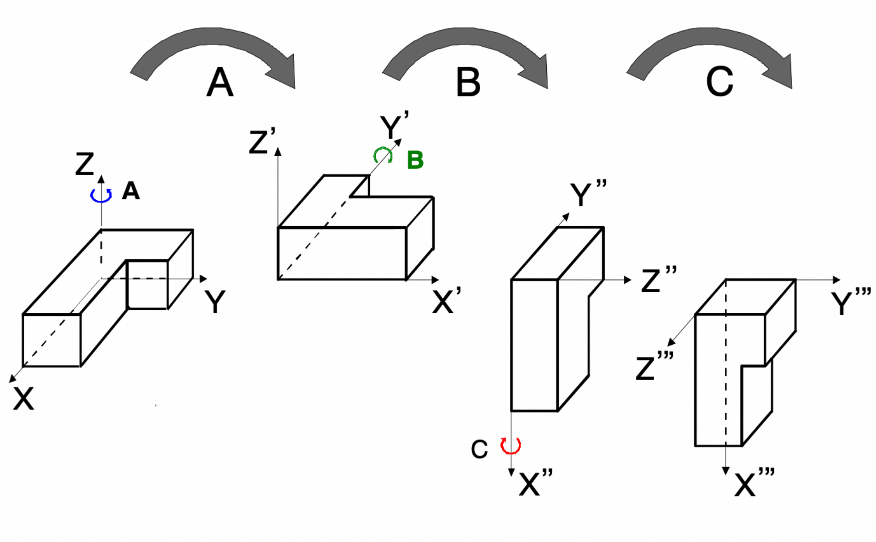

Figure 7. Rotational motion examples in all axes (Kuka Roboter GMBH 2000)

Coordinate systems

In Kuka robot, there are four cartesian coordinate systems which are mentioned as world coordinate system, robot coordinate system, tool coordinate system and base coordinate system as shown in Figure 8 (Kuka roboter GMBH 2000, 63). World coordinate system is the one that is fixed on the robot program and it is the reference for the other coordinate systems. Robot coordinate system is situated below the main body of the robot. It is generated from world coordinate system so that, robot follows the same orientation with that. Tool coordinate system is located on the centre of the tool surface as Figure 9. In the first set-up of tool coordinate system, the Z axis is aligned with axis 6 (Kuka roboter GMBH 2000, 64). In addition, if there should be a new tool that is required to be changed, either tool coordinate system can be edited or a new tool coordinate system can be created. Base coordinate system works in a strategy that it could be assigned as an any point and be set for the programming purposes.

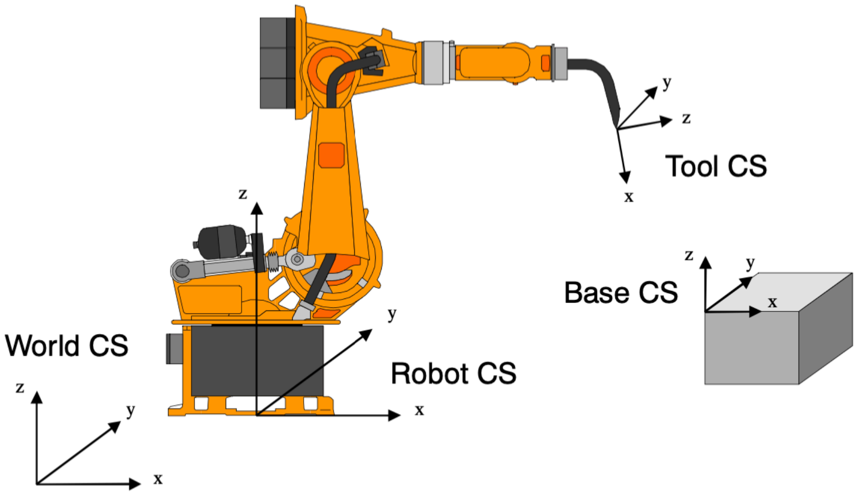

Figure 8. Cartesian coordinate systems (Kuka Roboter GMBH 2000)

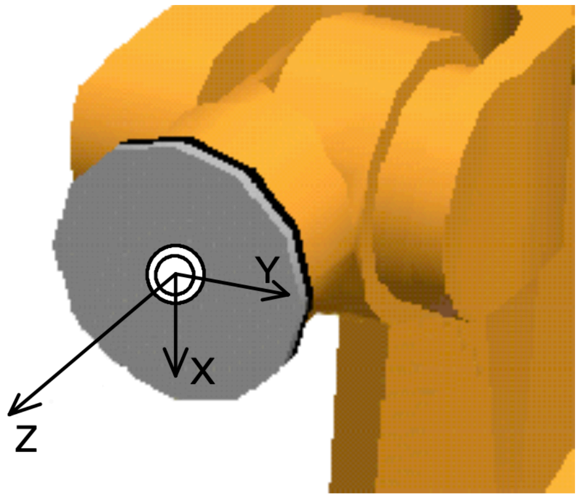

Figure 9. Default version of tool coordinate system (Kuka Roboter GMBH 2000)

3.3 A procedure to create an automation solution via Kuka robot

Considering the needs for the system and the solution, first step is creating a mind-mapping how the system should work and what parts for the system are needed. A simulation could be prepared on Kuka simulation software hence, the simulated system could be observed more and in detail. When the system design and robot actions are ready, then Workvisual software project can start. The function of Workvisual is that Kuka robot projects are created, and all the robot commissioning and programming tools are accessible on here. Workvisual provides assigning of terminals (digital input, digital output, analog input, analog output) into system and addressing inputs & outputs to addresses on Kuka robot. Field bus must be established to assign terminals. Simultaneously work visual procedures, these procedures could be followed. First, wiring the cables in Kuka controller to make necessary connections for robot motion and external devices. Afterwards, depending on the external inputs and outputs (valve terminals, pistons, pneumatic pistons, emergency buttons, turn on/off buttons), pneumatic and other physical connections can be done. Safety configurations and calibration of the robot can be completed. After testing the functionality of the inputs and outputs on Kuka robot, process can go to programming phase.

This procedure is prepared as a typical case of robot solutions and steps might be different or complex related to requested project.

3.4 Accessible services for Kuka robot

Kuka is offering couple of support options which are the solutions for various cases. There are calling, e-mail, on-site, e-documents, download service, spare parts, global standards, and Kuka college services available for Kuka customers (KUKA 2021). When reporting the problem, model and serial number of the used tool, definition of the problem and repetitiveness of the fault within a compiled overall information of the system should be the essential for requesting support (Kuka Roboter GMBH 2015a, 207). Spare parts can be shown as shown in Figure 10.

Figure 10. Representative support image (Kuka.com)

4 Overview of the project and tasks

The main idea of the project is to grab bulk bags from a cart and insert the bags under the feeder pipe. When the feeding process is completed, robot tool uses cable tie to close the bag. Components of the system are KRC4 type Kuka robot, custom design tool and external devices to complete the main task autonomously.

First task is related to take the bags one by one (starting from the first position) and leave the cart from the front side of the cart as shown in Figure 11. In order to start that task robot tool must go on the cart and bend over the hooks to grab the bag from its loops. First, robot goes to the right hook to grab the bag hook as shown in Figure 12 and moves to the left hook for the same operation as shown in Figure 13. It is assumed that the point of the view is from the cart through the black fence door. Thus, directions which are used as commentaries always follow that rule.

Figure 11. Bulk bags placed in a cart

Figure 12. Right hooks

Figure 13. Left hooks

After successfully grabbing a bag, robot tool moves under the spherical device as Figure 14 which is responsible for opening up the top the bag and pull the up sleeve with that. That spherical device is called as “Kurpitsa” in the programming section. When the robot comes under there, kurpitsa starts moving three times (only up & down) to grab the sleeve of the bag and at the third round, kurpitsa expands and starts using the vacuum holes on it to pull the sleeve of the bag up.

Figure 14. Spherical device “kurpitsa”

After the kurpitsa operation, robot tool grabs the bag and goes through the feeder pipe as demonstrated in Figure 15. Robot tool gets under the feeder pipe and places the bulk bag loops on the feeder pipe hooks and leaves the operation zone to get the second bulk bag robot.

Figure 15. Feeder pipe

After following the same root for the second bulk bag, robot completes the previous steps that are before the robot tool goes under the feeder pipe motion and robot waits for the feeding process of the first bag. Subsequently, the completion of the feeding process, the robot tool uses cable tie device, where it is located on the robot tool, and the sleeve of the bag is sealed by that method.

As it is planned, there will be maximum 20 bulk bags that will be operated in one run of the program. There are 2 carts, and each cart can have 10 bulk bags. In the first run, 7 bulk bags from one cart will be able to operate due to collusion possibilities and inaccessibility to some positions of the robot tool.

5 Robot commissioning

5.1 General process and Kuka controller

After physically installing the robot in the workplace, robot commissioning comes as a 2nd target for the general process of robot set-up services. In robot commissioning, tasks are divided as wiring cables, installing the necessary bus devices on the Kuka controller, adding on the external devices (valve terminals, proximity sensors, inductive sensors, ultrasonic sensors, pistons) in the task area and assigning all the devices on Workvisual software.

Kuka controller is the intersection area of commissioning task moreover, it acts as a brain for the entire system. All wires connected to there, and robot could be switched on only by there.

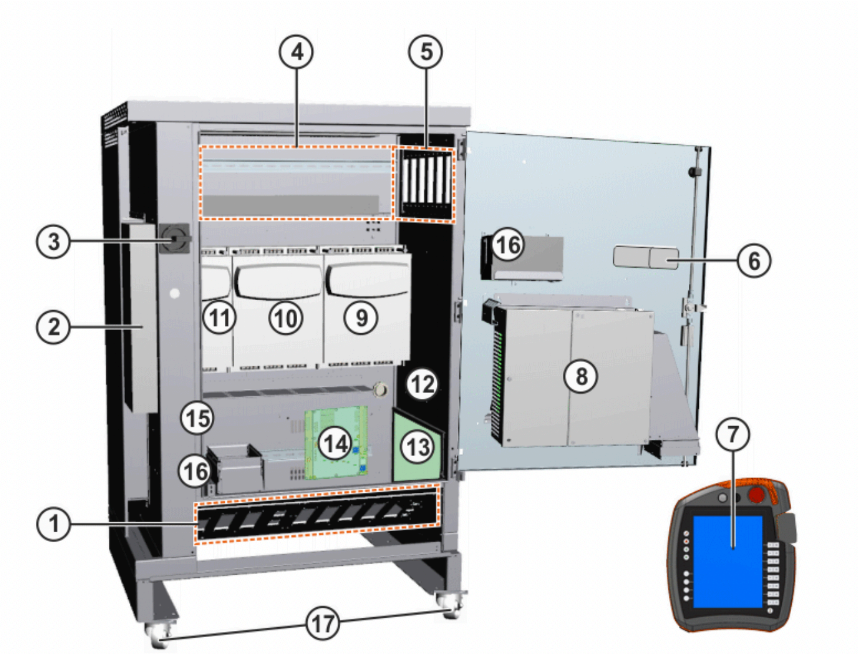

Figure 16. Kuka controller overview (Kuka Deutschland GMBH 2018)

As it can be seen in Figure 16, Kuka robot controller components are lower connection panel (1), mains filter (2), main switch (3), space for integration operations (4), side connection panel (5), controller system panel (6), teach pendant (7), control PC (8), optional drive controller axis 7 to 9 (9), drive controller axes 4 to 6(10), drive controller axes 1 to 3 (11), brake filters (12), cabinet control unit (13), safety interface board (14), fuse elements (15), batteries (16), and optional set of rollers (17) (Kuka Deutschland GMBH 2018,16).

There is Kuka power pack, and it provides the necessary voltage for the internal drive controllers and external drives. Its duty is to connect or disconnect the voltage for the drives (Kuka Deutschland GMBH 2018,17).

Kuka servo pack includes three drive controllers individually and it controls the axes motion and torque level of the robot by servomotors (Kuka Deutschland GMBH 2018,18). For the project, there are only two drive controllers in use for the objective (6 axes).

Control PC unit is the fundamental part of the system and it assists to graphical user interface, program creation & maintenance, sequence control, path planning, control of the drive circuit, safety equipment and communication with other computers (Kuka Deutschland GMBH 2018,18). So that a robot program or a new update on the system can be also altered.

Cabinet control unit is the main communication part of the Kuka controller, and it distributes the power for the other units. Any case of main voltage failure, cabinet control unit has extra batteries connected to it thus, system can be easily manipulated and taken under control (Kuka Deutschland GMBH 2018,19).

Safety interface board is the fundamental aspect of the safety interface and configurations (Kuka Deutschland GMBH 2018,19). There are two types of safety interface boards which are the standard and the extended version, and they consist of safety inputs and outputs in them (Kuka Deutschland GMBH 2018,20). Extended safety interface board must follow the standard one consequently, it can be operated (Kuka Deutschland GMBH 2018,20). Main task of the standard version is regarding safety interface of the robot controller additionally, extended version carries out range selection and range monitoring (Kuka Deutschland GMBH 2018,20).

Since there are 6 axes that could be moved, two drive controllers (10), (11), as shown in Figure 16, are in the system for the project, and they are distinctively responsible for the robot motion.

Lower connection panel is the place where most of the cables are going through. Power cable, motor cables, data cables, teach pendant cable, polyethylene cables and peripheral cables are the respective cables (Kuka Deutschland GMBH 2018,25). Figure 17 shows the number of the cable connection sockets in an order which are X1 power supply connection (1), motor connector interfaces (2), X11 safety interface (5), X19 teach pendant connection (8), X21 resolver digital converter (10), X66 Ethernet safety interface (11), Ground conductors (12 & 13) and optional sockets (3, 4, 6, 7, 9) (Kuka Deutschland GMBH 2018,26).

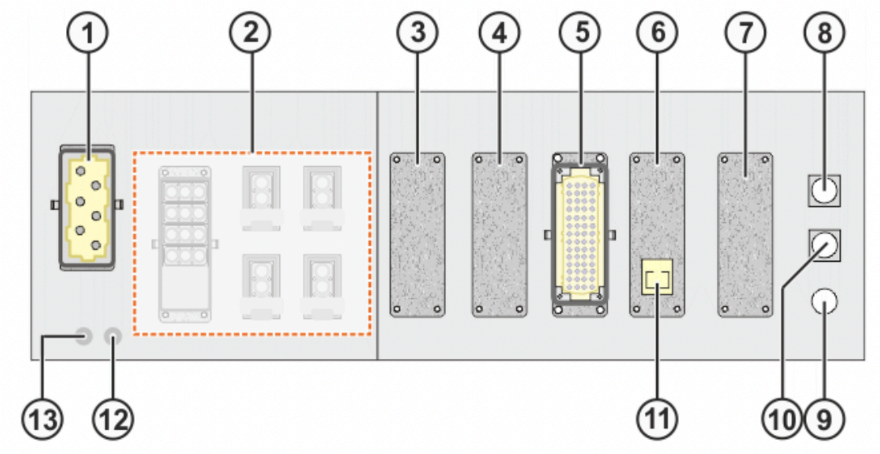

Figure 17. Lower connection panel (Kuka Deutschland GMBH 2018)

5.2 Commissioning on Workvisual

On Workvisual, everything starts at selecting the robot type and the controller model. KR 210 R3100-2 F is the model type of 6 axes robot and KRC-4 is the Kuka controller unit for the thesis work. Selected robot on Workvisual is shown in Figure 18. Next step is searching for the import data in Workvisual catalogues on the internet for the devices that would be connected on Kuka controller.

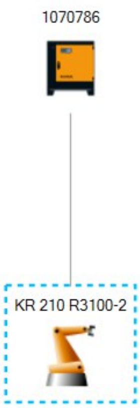

Figure 18. Kuka controller and robot model on Workvisual

When the robot and the Kuka controller is assigned correctly there will be Kuka controller bus (KCB) and Kuka system bus (SYS-X48) as default. Kuka extension bus (SYS-X44) will be filled by the components that are going to be used for the project and in Figure 19, this information can be followed.

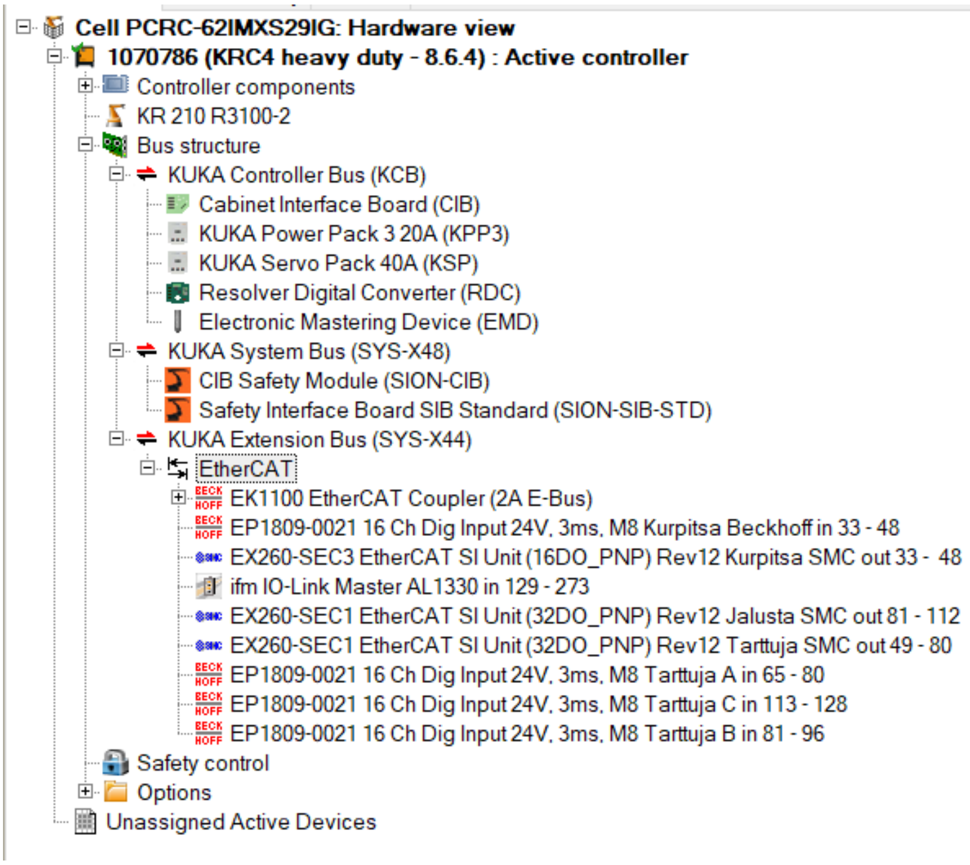

Figure 19. Kuka controller bus structure

In Figure 19, after a double-click to Kuka extension bus (SYS-X44), there can be a topology view found. It shows how the fieldbuses are related to each other. An example screenshot can be followed in Figure 20. If the physical connections are made right, topology view might show the right from, even though it is possible edit the hierarchy on Workvisual.

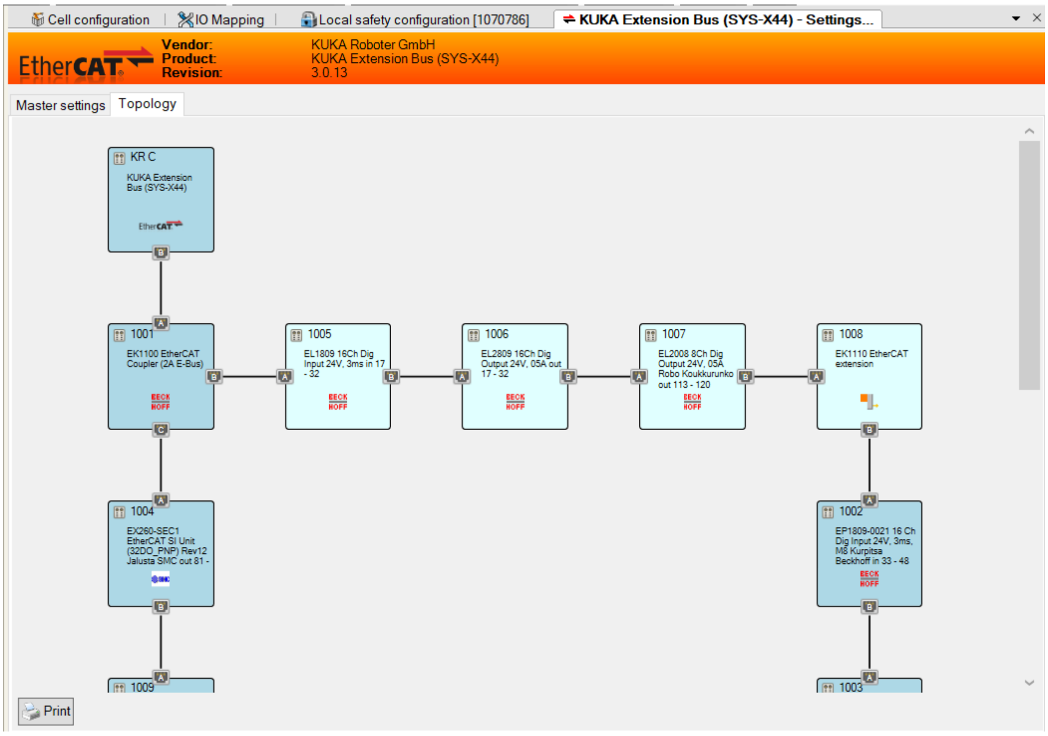

Figure 20. Topology view

After the previous step devices can be matched up by assigning the input and output values in “IO Mapping” section. As it can be seen in Figure 21, field bus addresses are added into empty slots in Kuka controller I/O.

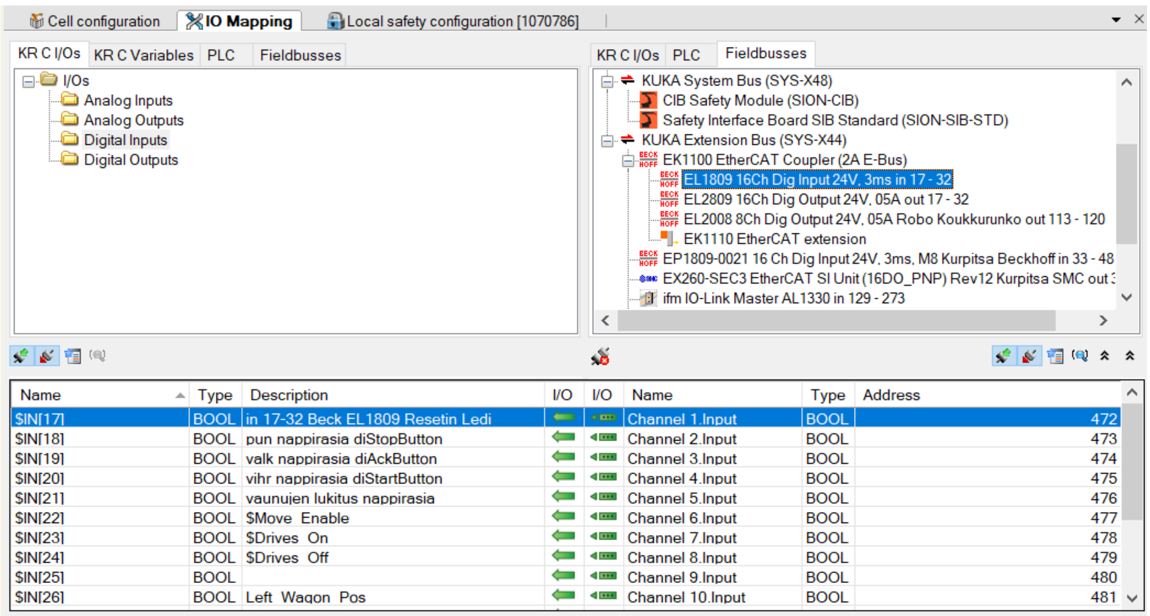

Figure 21. Kuka controller bus structure

6 Start-up steps of Kuka robot

6.1 Automatic external

In case of external control is needed such as start-stop-wait buttons, Automatic external interface could be used for that purpose (Kuka Roboter GMBH 2015b,189). Automatic external interface receives the information from the robot controller and depending on the reason of usage, it can manipulate the robot controller (Kuka Roboter GMBH 2015b,189). For the thesis work this method is preferred for start and stop button so that in the automatic running mode of the robot, program can start up by the sequence which includes the input and output signals. These input and output signals, in Figure 22, must follow the sequence below.

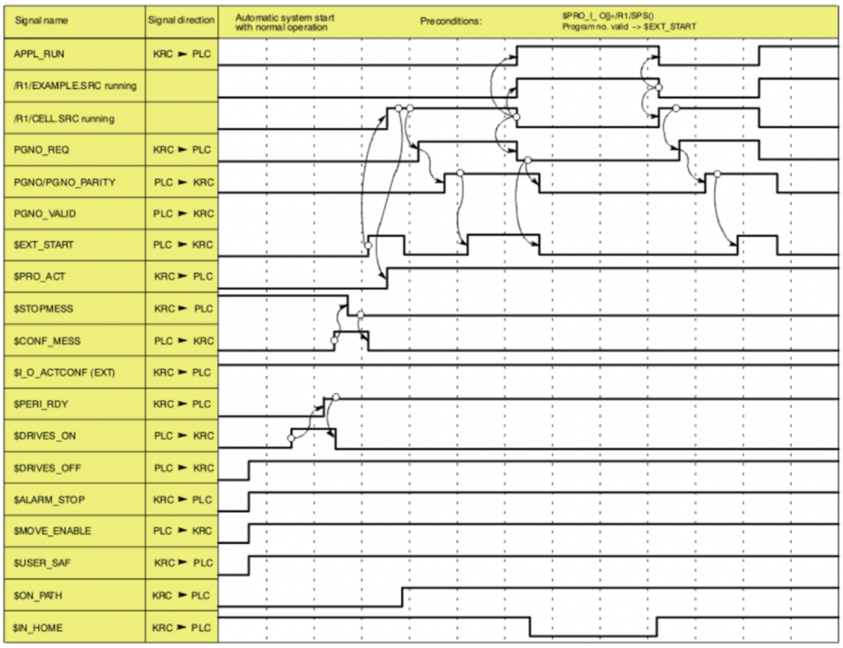

Figure 22. Start-up sequence (Kuka Roboter GMBH 2015b)

"PGNO_PARITY" checks whether the sent program to robot is pairable (Kuka Roboter GMBH 2015b,193). "PGNO_VALID" examines the program number that is sent for the robot (Kuka Roboter GMBH 2015b,193). "$EXT_START" can be activated while input and output signals are set (Kuka Roboter GMBH 2015b,194). "$MOVE_ENABLE" is the input that while it is true, robot motion and program execution can happen (Kuka Roboter GMBH 2015b,194). "$CONF_MESS" gets activated when there are errors displayed and it stays on as long as the errors are still displayed (Kuka Roboter GMBH 2015b,194). "$DRIVES_OFF" and "$DRIVES_ON" are the inputs that low-level pulse and high-level pulse play the important role for these inputs thus, robot drives switch off with low-level pulse and switches on with high-level pulse (Kuka Roboter GMBH 2015b,194). “$ALARM_STOP” is triggered when emergency stop button either on teach pendant or externally pressed (Kuka Roboter GMBH 2015b,195). “$USER_SAF” is in the reset when the safety is not possible in automation mode or when the input is released in the program teaching statement (Kuka Roboter GMBH 2015b,196). “$PERI_RDY” is running when the robot drivers are on and it connects with higher-level controller (Kuka Roboter GMBH 2015b,196). “$STOP_MESS” is responsible for showing the messages while the robot is not moving (Kuka Roboter GMBH 2015b,196). “$PRO_ACT” is an output that runs only when a program is running and it represents the process (Kuka Roboter GMBH 2015b,196). “PGNO_REQ” is an output that gets a program number from high-level controller (Kuka Roboter GMBH 2015b,197). “$APPL_RUN” is an output and high-level controller acquires the information from it that a program is executed (Kuka Roboter GMBH 2015b,197). “$IN_HOME” gives an information whether the robot in home position or not (Kuka Roboter GMBH 2015b,197). “$ON_PATH” is the checking whether the robot following the path in a program (Kuka Roboter GMBH 2015b,197).

6.2 Assigning new coordinate systems (tool calibration)

Usage of new coordinate systems is handy when the robot needs to move carefully in a limited area. Although there is a base coordinate for the robot that is being used, two more coordinates are added into the robot for the project which control the complex tool motions. Those tool coordinates help tool to bend over two to sides of the tool to grab loops of the bags from the hooks of the cart during the program running. This method is applied by tool calibration option. Advantages of the tool calibrations are tool motion in a straight line and moving robot while tool coordinate point is not moving (Kuka Roboter GMBH 2015b,124). Therefore, a relative coordinate system with flange coordinate system occurs as it can be seen in Figure 23 (Kuka Roboter GMBH 2015b,125).

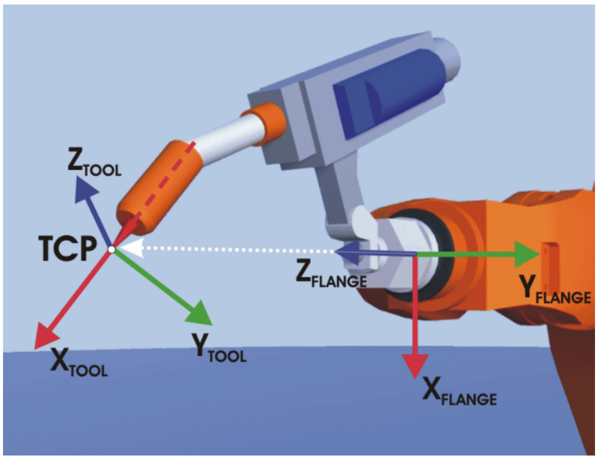

Figure 23. Tool calibration point coordinates and flange coordinates (Kuka Roboter GMBH 2015b)

From various of tool calibrations procedures, “XYZ Reference method” is used for the project. By that, X, Y, Z coordinates of the flange (or a previous tool) will be offset to the required tool calibration point of a new tool (Kuka Roboter GMBH 2015b,128). In order to do that, “Start-up > Calibrate > Tool > XYZ Reference” must be followed on the Kuka teach pendant and distance values can be inserted there in the respective gaps (Kuka Roboter GMBH 2015b,128).

The defined tool calibrations which are done on teach pendant can be followed in Figure 24.

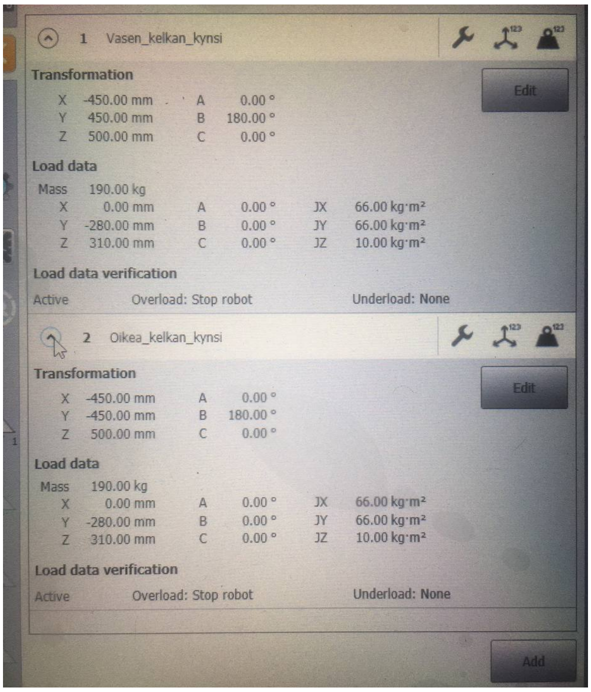

Figure 24. Tool base management for left-rear-claw and right-rear-claw

6.3 Mastering

In a start-up of a robot mastering is done to calibrate axes individually so that robot motion and the data that can be read are matching. There are two tools that can be used for mastering which are standard electronic mastering device (SEMD) and micro-electronic mastering device (MEMD) (Kuka Roboter GMBH 2015b,102).

Figure 25. Tool kit for mastering (Kuka Roboter GMBH 2015b)

As it can be seen in Figure 25, there are mastering box (1), screwdriver for MEMD (2), MEMD (3), SEMD (4) and cables (5) (Kuka Roboter GMBH 2015b,102). SEMD is used in the project and it is connected onto gauge cartridge. The cables are connected to SEMD from the other side where it goes to X32 slot as in Figure 26,27 and 28 (Kuka Roboter GMBH 2015b,102).

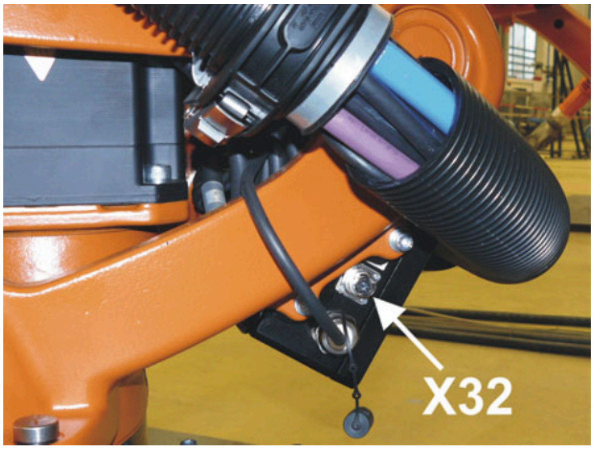

Figure 26. X32 Slot (Kuka Roboter GMBH 2015b)

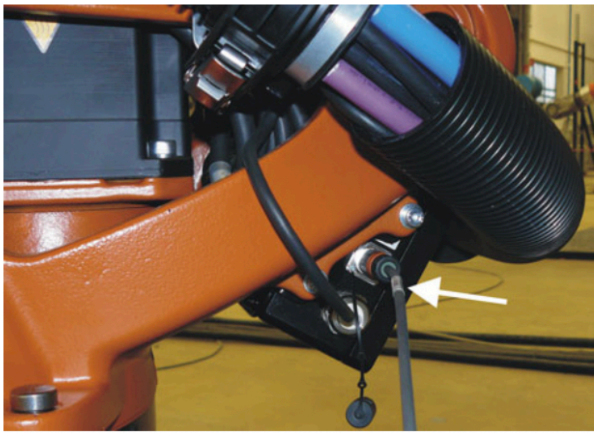

Figure 27. Cable connected to X32 Slot (Kuka Roboter GMBH 2015b)

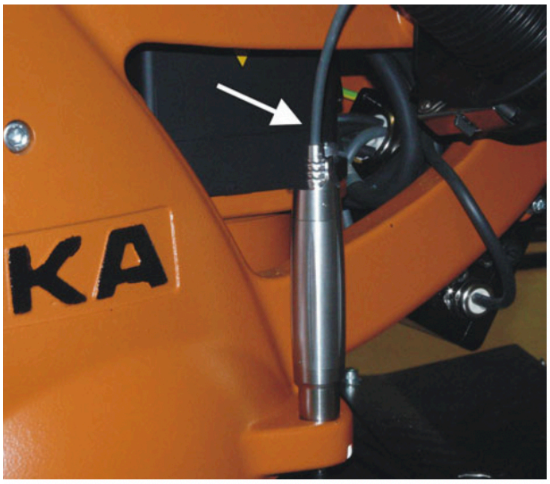

Figure 28. SEMD and signal cable are connected (Kuka Roboter GMBH 2015b)

Before mastering starts, axes are moved through positions where white marks are located as in Figure 29. That state is called as pre-mastering (Kuka Roboter GMBH 2015b,104). Procedure is going in such that when the axes positions are closed to the white marks, master key on teach pendant is pressed and procedure is completed when all the axes are verified (Kuka Roboter GMBH 2015b,109).

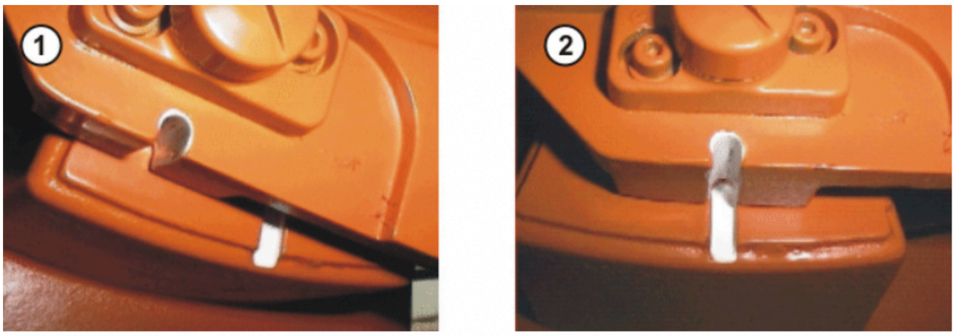

Figure 29. White marks for mastering (Kuka Roboter GMBH 2015b)

7 Safety configurations

7.1 General safety rules

Safety in Kuka robots follows the general safety rules with a high technology design and it might still cause serious hazards and permanent damages to the operator and the robot itself (Kuka Deutschland GMBH 2019,41). Industrial robots should be used only by the experts of the system and robot should be powered off in any possible of danger (Kuka Deutschland GMBH 2019,41). Industrial robot usage should be combined with the manuals that provide detailed information about the robot (Kuka Deutschland GMBH 2019,41). Kuka robots have “partly completed machinery”, means that a robot can perform if the system, which contains the robot and other machineries, is approved by EC machinery directive in “completed machinery” status (Kuka Deutschland GMBH 2019,42).

Personnel for the industrial robot can be divided into two groups which are system integrator and operators (Kuka Deutschland GMBH 2019,45). System integrator is responsible for installing the robot, completing the commissioning, checking the possible risks on operation mode and applying the safety functions (Kuka Deutschland GMBH 2019,46). Operators should learn how the entire system proceeds and should be qualified to operate the system (Kuka Deutschland GMBH 2019,46). In case of hazardous possibilities operators should be prepared (Kuka Deutschland GMBH 2019,46).

7.2 Cartesian workspaces

Cartesian workplace is the place that defined by the user which effects the possible motion of the robot and robot tool cannot pass through that limited area (Kuka Deutschland GMBH 2019,21). In order to detect the borders of that limited area, there should be spheres that are configured on the robot flange thus, the centre of these spheres can be offset to different points (Kuka Deutschland GMBH 2019,21). Cartesian workspace can work in a way that, it can be configured either the area that robot tool operates or the protected area that robot tool cannot go through (Kuka Deutschland GMBH 2019,23). The method of configuring a workspace is selecting the +- X, Y, Z length from a particular coordinate system point.

Four spheres and two cartesian workspaces are created for the task. Spheres are placed on the corner of the robot and the possible furthest points. The operation cartesian workspace is called as “fenced area” in Figure 30 and the protected area is called as “big pillar” in Figure 31. Configuration of the workspaces can be observed in the following figures. Fenced area refers the area that robot arm can reach at maximum level and big pillar refers the colon right next to the right wagon.

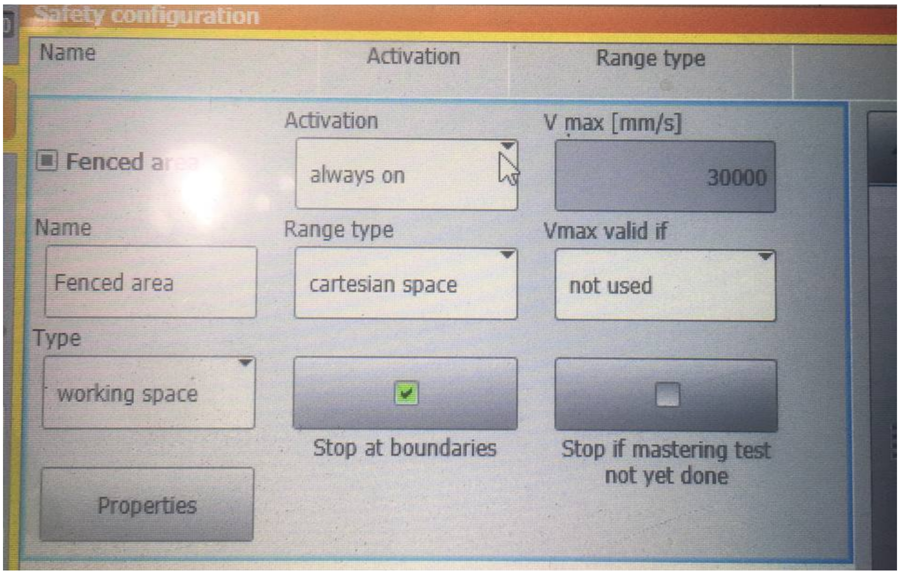

Figure 30. Fenced area

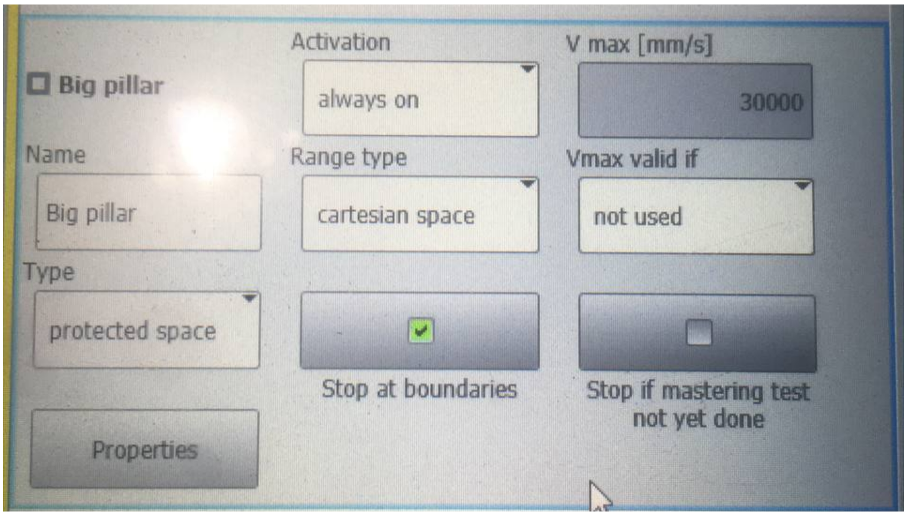

Figure 31. Big pillar

7.3 Mastering test

Mastering test is a checking step of the robot position via one reference point moreover, when the robot arm moves to stated reference point, mechanical point of the robot and the reference point should match (Kuka Deutschland GMBH 2019,125). Mastering test can be operated in several ways such as via Kuka reference switch, external system test, and a tracker that follows the point and encode it constantly (Kuka Deutschland GMBH 2019,125). In the project, Kuka reference switch is selected as a mastering test method. Robot tool reboot, remastering the robot and reconfiguration in I/O drivers set off mastering request test (Kuka Deutschland GMBH 2019,125). Robot can operate for two hours after the mastering test is completed and it is a useful idea to call mastering subprogram in a main program (Kuka Deutschland GMBH 2019,126). “masref_main.src” is the name of the main program that executes the master and for the reference positions “masref_user.src” is used (Kuka Deutschland GMBH 2019,128).

Reference switch can be placed a fixed place and in Figure 32, there is representative illustration regarding that. There are tool (1), actuating plate (2), reference switch (3), mechanical mounting fixture for the reference switch (4), and actuated reference switch (5) (Kuka Deutschland GMBH 2019,129). Figure 33 demonstrates the reference switch in the project.

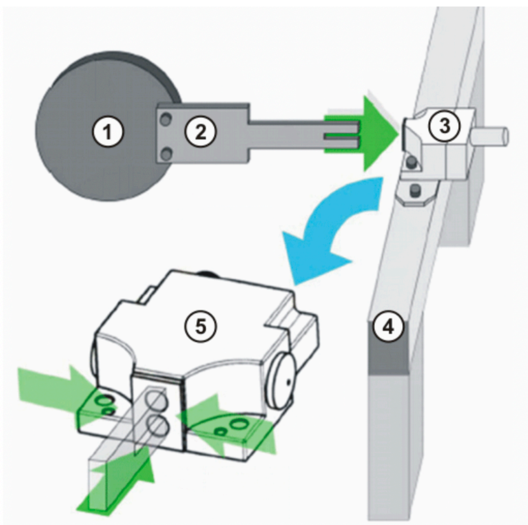

Figure 32. Reference switch (Kuka Deutschland GMBH 2019)

Figure 33. Reference switch for the project

At the time placing the reference switch, awareness of an optimal place would be the goal. Thus, the selection criteria for the place is about the regular motion of the robot during a program executes and the linear distance ($\pm$15mm) plus the rotational distance ($\pm 5^{\circ}$) in between actuating plate and reference switch (Kuka Deutschland GMBH 2019,130).

7.4 Brake test

For the axes in the robot, there are brakes where they are set individually therefore brake test checks out the braking torque of each brake whether they work conveniently (Kuka Deutschland GMBH 2019,139). In case of robot controller generates the message “break test is required”, test can be completed, and the robot can operate after two more hours (Kuka Deutschland GMBH 2019,140). Sequence of the brake test works such that, axes of the robot start moving maximum of 5° in both rotational directions and test the brakes when it comes back to the starting point (Kuka Deutschland GMBH 2019,141). There are starting point, ending point and parking point for the brake test however, only the parking point must be taught for the program Kuka Deutschland GMBH 2019,149). Starting point and ending point are the optional for the test and test can be handled from the last current position in which robot locates Kuka Deutschland GMBH 2019,140). In case of a brake test error, robot goes to the parking position (Kuka Deutschland GMBH 2019,141). Brake test can run via “BrakeTestReq.src” is the program that used for the brake test and for the project case it is used as a subprogram in the main program (Kuka Deutschland GMBH 2019,142).

8 Programming on Kuka robot

8.1 General definitions and essentials of programming

Programs can be created in “program” folder and a folder can be created under “R1” folder which also contains “program” folder in it (Kuka Roboter GMBH 2015b, 231). Programs can be reached either from teach pendant or Kuka Workvisual software via computer and be sent to Kuka controller if any edit needs to be done.

There are two types of files on Kuka programming which are named as source “SRC” file and data “DAT” file (Kuka Roboter GMBH 2015b, 359). SRC file keeps the programming codes in which motion, control definition sub-program commands are located, and an editor makes changes in this section (Kuka Roboter GMBH 2015b, 359). DAT file covers the locations and point coordinates of a program (Kuka Roboter GMBH 2015b, 359). Obviously, SRC and DAT files are working together that whenever a motion command is added in SRC, DAT file saves the location of the point. In addition, in case of creating a new program file, SRC and DAT files appear at the same time with a same name. Thus, self-created programs can work in that function.

Some of the data could be made globally (i.e., a signal), when it should be executed in multiple programs without assigning it in each program. “$CONFIG.DAT” file, which can be found in “system” folder, can store data without using the keyword global (Kuka Roboter GMBH 2015b, 363). It is an actively used file in the current project and it is always running while the system is active. An example content can be seen in Figure 34.

If an external program, which controls the motions such as external devices, is necessary for the entire system, “SPS.SUB” can be used for that reason. SPS.SUB activates automatically when the robot is turned on however, it can be manually controlled therefore, pausing and stopping SPS.SUB is available (Kuka Roboter GMBH 2015b, 453). Basic functions, if statements, while cases, switch cases are actively used in the current project in SPS.SUB. It is also used during the program creation path so that external devices can be manipulated for the corresponding position without typing codes in a program. Figure 35 demonstrates an example from the project.

```
;INPUT SIGNALS FOR ACCESSORY DEVICE POSITION SENSING

SIGNAL inKURPITSA_1000_UP $IN[33] ;kurpitsa 1000 cylinder up sensor
SIGNAL inKURPITSA_600_UP $IN[35] ;kurpitsa 600 cylinder upper position
SIGNAL inKURPITSA_600_DOWN $IN[36] ;kurpitsa 600 cylinder down position
SIGNAL inKURPITSA_YLA1 $IN[39] ;1st upper kurpitsa laser sensor for bag sleeve edge
SIGNAL inKURPITSA_YLA2 $IN[40] ;2nd upper kurpitsa laser sensor for bag sleeve edge
SIGNAL inKURPITSA_ALA1 $IN[47] ;1st lower kurpitsa laser sensor for bag sleeve edge
SIGNAL inKURPITSA_ALA2 $IN[48] ;2nd lower kurpitsa laser sensor for bag sleeve edge
SIGNAL Left_Wagon_Pos $IN[26] ;left wagon positioned correctly
SIGNAL Right_Wagon_Pos $IN[27] ;right wagon positioned correctly

INT LEV_VAS=1
INT LEV_OIK=1
INT LEV_VASM=0
INT LEV_OIKM=0

INT KURPITSA=1
BOOL KURPITSA_UP=TRUE
BOOL KURPITSA_MID=FALSE
BOOL KURPITSA_DOWN=FALSE

;VARIABLE FOR PAUSING ROBOT PROGRAM
BOOL Pause_Program=FALSE
INT Run_Program=1

;Variable for Wagon Lock Button

INT WAGONLOCK=1
INT WAGONSTATE=0
```

Figure 34. $CONFIG.DAT example from current project

```
IF KYNNET_VAS THEN
   $OUT[68]=TRUE
ELSE
   $OUT[68]=FALSE
ENDIF

IF KYNNET_OIK THEN
   $OUT[67]=TRUE
ELSE
   $OUT[67]=FALSE
ENDIF

SWITCH LEV_VAS
   CASE 0
      VAS_LEV_0=TRUE
      VAS_LEV_60=FALSE
      VAS_LEV_200=FALSE
      LEV_VAS=1
      LEV_VASM=0
   CASE 60
      VAS_LEV_0=FALSE
      VAS_LEV_60=TRUE
      VAS_LEV_200=FALSE
      LEV_VAS=1
      LEV_VASM=60
   CASE 200
      VAS_LEV_0=FALSE
      VAS_LEV_60=FALSE
      VAS_LEV_200=TRUE
      LEV_VAS=1
```

Figure 35. SPS.SUB example from current project

8.2 Tool commands on programming

Robot tool motions are mainly divided into three sections and these are point to point “PTP”, linear “LIN” and circular “CIRC” (Kuka Roboter GMBH 2015b, 277).

PTP is the method that robot tool acts to move in a shortest time but not the shortest distance motion because usually for the tool motion, it is not commonly a straight line (Kuka Roboter GMBH 2015b, 277). Non-straight paths can result in a shorter period than straight paths (Kuka Roboter GMBH 2015b, 277). Motion illustration can be observed In Figure 36.

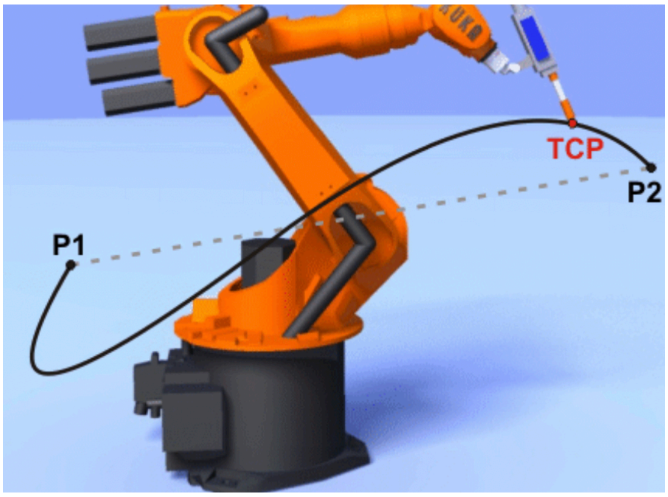

Figure 36. Point to motion representation (Kuka Roboter GMBH 2015b)

LIN motion is the motion that robot tool follows a complete straight motion between two points (Kuka Roboter GMBH 2015b, 278). Figure 37 shows an example.

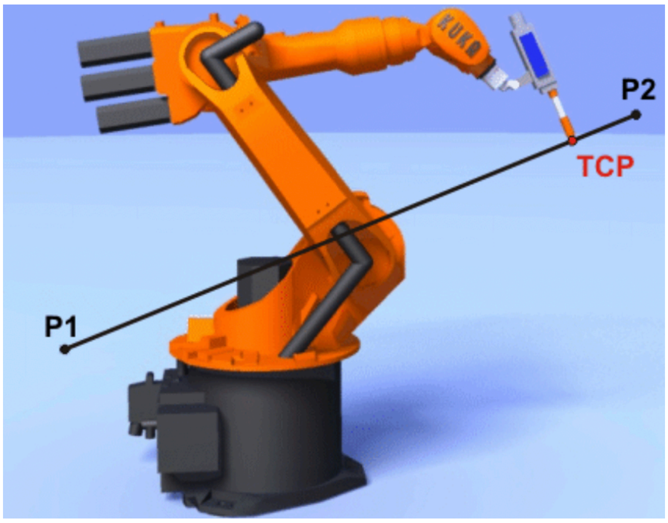

Figure 37. Linear motion representation (Kuka Roboter GMBH 2015b)

CIRC motion happens by a circular path of the robot tool (Kuka Roboter GMBH 2015b, 278). Circular motion can be indicated as a start point, an end point and auxiliary point (Kuka Roboter GMBH 2015b, 278). Figure 38 shows an example motion.

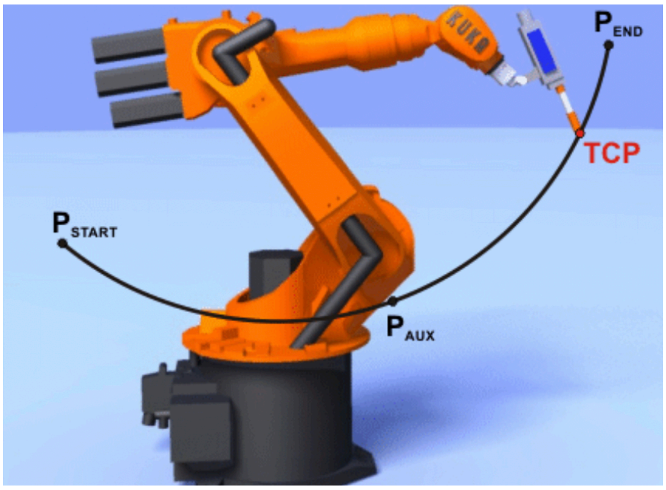

Figure 38. Circular motion representation (Kuka Roboter GMBH 2015b)

There are also robot tool motions which are generated from the three essential motion types (Kuka Roboter GMBH 2015b, 290). These generated motions are using spline block method (Kuka Roboter GMBH 2015b, 291). Name of these motions are SLIN, SPL, SCIRC (Kuka Roboter GMBH 2015b, 291). Instead of sharp motions of a robot tool in a path, Path acts like it is following a curvy path so motions become smooth as it is experienced during the robot programming for the project.

8.3 The master program

Master program works in such a way that, it includes all the processes on it. There are (Figure 39) interrupt, loop, if, halt functions, counters and subprograms are used to manipulate the program in an optimal way.

Interrupt gets activated when the “Pause_Program” Boolean is triggered (It is assigned in $CONFIG.DAT and it is in if-statement in SPS.SUB) and it makes “Pause()” subprogram run. It is been placed here since for the cautions of any collisions or damages. Program does not execute until “Pause_Program” is inactive.

“SPTP HOME” is the robot motion command that robot moves to home position before anything starts in a program and it can be also used at the end of a program (Kuka Roboter GMBH 2015b, 240). 2 wagons and 20 different bulk bag positions will be in the program so that three counters solve the problem just by calling the “Rightwagon ()” and “Leftwagon ( )” subprograms. Main actions are in the loop function and after a bag grabbing is completed, WagonCounter value increments

(“RightCounter” and “LeftCounter” increment in their respective wagon subprograms) below if-statement.

Program continues with “Kurpitsa ()”, “Putki ()” processes therefore, bulk bag comes under the feed pipe. “Suljenta ()” is activated when the condition is provided (37, 38 and 39th lines are in the commentary mode because they are not actively running in the current situation of the project), and loop gets back onto 27th line. When the wagon counter exceeds 20, program leaves the loop, and the master program ends.

```
1   &ACCESS RVP
2   &REL 3
3   DEF  MasterProgram ( )
4   INI
15
16  INTERRUPT DECL 26 WHEN Pause_Program DO Pause()
17  SPTP HOME VEL=100 % DEFAULT
23  LeftCounter = 1
24  RightCounter = 1
25  WagonCounter = 1
26
27  LOOP
28  IF WagonCounter < 11 THEN
29      Rightwagon()
30  ELSE
31      Leftwagon()
32  ENDIF
33
34  WagonCounter = WagonCounter + 1
35  Kurpitsa()
36  Putki()
37  ;IF PipeFeeding == TRUE THEN
38  ;Suljenta()
39  ;ENDIF
40
41  IF WagonCounter > 20 THEN
42     GOTO JUMPHERE
43  ENDIF
44
45  ENDLOOP
46  JUMPHERE:
47  END

48
49  DEF Pause()
50  BRAKE
51  WAIT FOR NOT Pause_Program
52  END
```

Figure 39. The master program

9 Summary

In this project, the procedure to achieve robot commissioning and installation, overview of the Kuka robots and the current project with that, start-up methods of the robot, safety rules which are needed follow and the company that runs the project are determined.

With the steps that are explained in this project shows the progression of this project. Recently, robot is able to do one complete run up to some level and the external devices are working precisely.

This work is also concentrated for the people who want to learn more about the robot commissioning and programming before they actually start working on a project or a career-orientated interests.

It can be clearly stated that the solution of Sunrob Robotics Oy, comparing to a non-automatic solution, brings an efficiency for the general task and innovation in bulk bag grabbing and feeding processes.

References

Sale, G. 2015. Why KUKA Robots? I'll give you my answer. Linkedin. Retrieved on 16 February 2021. Available at https://www.linkedin.com/pulse/why-kuka-robots-ill-give-you-my-answer-greg-sale/

Braumann, J., Brell-Cokcan, S. 2011. Parametric Robot Control. ACADIA 2011: Integration Through Computation: Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). Retrieved on 16 February 2021

The history of Kuka. Kuka. Retrieved on 23 February 2021. Available at https://www.kuka.com/en-se/about-kuka/history

6-axis Robots Articulated Robots. Robotic Automation Systems. Retrieved on 25 February 21. Available at https://www.roboticautomationsystems.com/6-axis-robots.html

Kuka Roboter GmbH. 2000. KR C… Expert Programming. Germany. KUKA

Kuka Roboter GmbH. 2015a. KR C4; KR C4 CK Operating Instructions. Germany. KUKA

Kuka Roboter GmbH. 2015b. KUKA System Software 8.3. Germany. KUKA

Kuka Deutschland GmbH. 2018. KR C4 midsize; KR C4 midsize CK Operating Instructions. Germany. KUKA

Kuka Deutschland GmbH. 2019. KUKA.SafeOperation 3.5. Germany. KUKA

Kuka. Referenced on 16 February 2021. Available at http://www.techvitas.ee/en/industrial-robots/

Sunrob Robotics. Referenced on 16 February 2021. Available at https://www.sunrob.com/en/

Kuka Roboter GmbH. 2015a. Example of an Industrial robot. Referenced on 23 February 2021.

Kuka Roboter GmbH. 2000. Coordinate transformation. Referenced on 23 February 2021

Kuka Roboter GmbH. 2000. Joint coordinate system for a jointed-arm robot with 6 axes. Referenced on 24 February 2021

Kuka Roboter GmbH. 2000.Translational description of the position of a point. Referenced on 24 February 2021

Kuka Roboter GmbH. 2000. Rotational description of the orientation of a point. Referenced on 24 February 2021

Kuka Roboter GmbH. 2000.Cartesian coordinate systems for robots. Referenced on 25 February 2021

Kuka Roboter GmbH. 2000. Robot flange Referenced on 25 February 2021

Kuka. Referenced on 3 March 2021. Available at https://www.kuka.com/zh-my/售后服务/my-kuka/kuka-marketplace

Kuka Deutschland GmbH. 2018. Overview of the robot controller. Referenced on 15 March 2021

Kuka Deutschland GmbH. 2018. Lower connection panel. Referenced on 15 March 2021

Kuka Roboter GmbH. 2015b. TCP calibration principle. Referenced on 31 March 2021

Kuka Roboter GmbH. 2015b. Mastering kit with SEMD and MEMD. Referenced on 7 April 2021

Kuka Roboter GmbH. 2015b. Moving an axis to the pre-mastering position. Referenced on 7 April 2021

Kuka Roboter GmbH. 2015b. Removing cover from X32. Referenced on 7 April 2021

Kuka Roboter GmbH. 2015b. Connecting the EtherCAT cable to X32. Referenced on 7 April 2021

Kuka Roboter GmbH. 2015b. Attaching signal cable to SEMD. Referenced on 7 April 2021

Kuka Deutschland GmbH. 2019. Example position of reference switch actuating plate. Referenced on 11 April 2021

Kuka Roboter GmbH. 2015b. PTP motion. Referenced on 16 April 2021

Kuka Roboter GmbH. 2015b. LIN motion. Referenced on 16 April 2021

Kuka Roboter GmbH. 2015b. CIRC motion. Referenced on 16 April 2021