Tampere University

Siri Huhtala

IMPLEMENTATION OF PALLETIZING TASK WITH AN INDUSTRIAL ARTICULATED MANIPULATOR

Bachelor’s thesis
Faculty of Engineering and Natural Sciences (ENS)
Supervisor: Luis Enrique Gonzalez Moctezuma
April 2024

ABSTRACT

Siri Huhtala: Implementation of palletizing task with an industrial articulated manipulator
Bachelor's thesis
Tampere University
Bachelor of Technology
April 2024

Robotic applications are versatile, but most of them belong to industrial environments. One popular type of industrial articulated manipulators are palletizing robots. Robotic arm palletizers can be such as pick-and-place robots. Pick and place automation can be used, for picking up parts before assembly or for placing objects on a tray or a pallet for different reasons.

This thesis presents an implementation of a palletizing task with an industrial articulated manipulator. The aim of this thesis is to learn about palletizing and robotics in order to create and implement a practical pick-and-place program. This thesis uses an iterative test-driven development approach, focusing on task-level programming. The robots contemplated are the ABB robots IRB 14000 and IRB 120. The program is made and simulated with RobotStudio and Rapid with a virtual controller and in offline mode. The simulating and the program are made with IRB 14000, but the final implementation is finished with IRB 120.

The implementation phase included designing the program's structure and coding its functionalities. Challenges arose regarding pallet orientation and gripper programming, despite which, the program successfully operated with different ABB robot models, demonstrating its adaptability and effectiveness.

The results show that the dynamic palletizing strategy is proven efficient, requiring only two initial robtargets instead of individually teaching each object's position. The study also reveals the ease of transferring RobotStudio programs across different ABB robot models, underscoring the platform's flexibility.

The conclusion reflects on the lessons learned from the study, including the importance of planning, and understanding programming methodologies. It summarizes the achievements of this thesis in developing a practical palletizing program and implementing it with an ABB robot. Recommendations for future improvements are also provided. Overall, this thesis demonstrates the feasibility of using RobotStudio for palletizing applications and gives a practical method for creating a pick-and-place palletizing program.

Keywords: pick-and-place, palletizing, robotics, RobotStudio, IRB 14000, IRB 120

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Siri Huhtala: Lavan kuormaustehtävän toteutus teollisuusrobotilla
Kandidaatintyö
Tampereen yliopisto
Tekniikan kandidaatti
Huhtikuu 2023

Suurin osa nykyisistä robotiikan sovelluksista ovat teollisuuskäytössä. Eräs teollisuusrobotiikan sovellus on tavaran kuormalavalle siirtämiseen käytetyt robotit eli kuormaus- tai lavausrobotit. Lavojen täyttämistä suoritetaan myös ihmisen käsivartta muistuttavilla poimi- ja aseta -roboteilla, joita voidaan käyttää myös esimerkiksi osien poimimiseen ennen niiden yhteen kokoamista tai esineiden sijoittamiseen tarjottimelle tai kuormauslavalle eri syistä.

Tämä kandidaatintyö kertoo yksinkertaisen kuormaustehtävän toteutuksesta teollisuusrobotilla. Työn tavoitteena on perehtyä lavaukseen ja robotiikkaan käytännöllisen poimi- ja aseta -ohjelman luomiseksi ja toteuttamiseksi. Työssä vertaillaan ABB:n robotteja IRB 14000 ja IRB 120. Ohjelman toteutus ja simulointi tehdään virtuaalisella robottiohjaimella offline-tilassa RobotStudio-ohjelmalla ja rapid-ohjelmointikielellä. Simulointi ja ohjelma tuotetaan IRB 14000 robotilla ja lopullinen fyysinen testaus suoritetaan IRB 120 robotilla.

Lavauksen, robotiikan, ohjelmoinnin ja työssä käytettyjen välineiden ja työkalujen esittelyn jälkeen työ keskittyy toteutusvaiheeseen, johon kuuluu ohjelman rakenteen suunnittelu ja sen toiminnallisuuksien ohjelmointi. Haasteita syntyi lavasuuntautuneisuudesta ja tarttujien ohjelmoinnista. Ongelmista huolimatta ohjelma toimi menestyksekkäästi testatuilla ABB-roboteilla.

Tulokset osoittavat, että dynaaminen ohjelmointitapa on tehokas ja että RobotStudio-ohjelma on hyvin joustava, sillä ohjelman siirtäminen eri ABB-robottimallien välillä on erittäin yksinkertaista. Ohjelma toteuttaa sille asetetut käytännöllisyysvaatimukset, koska se tarvitsee vain kaksi alkuperäistä robtargetia sen sijaan, että jokaisen kohteen sijainti opetettaisiin robotille erikseen.

Lopuksi työssä kerrataan sen aikana opittuja asioita, kuten suunnittelun ja ohjelmointimenetelmien merkitys. Viimeinen luku tiivistää työn saavutukset ja kertoo ehdotuksista, joilla työtä voisi vielä jatkaa. Kaiken kaikkiaan tämä kandidaatintyö osoittaa RobotStudion käyttökelpoisuuden robottisovelluksissa ja antaa käytännöllisen menetelmän lavausohjelman toteuttamiseen.

Avainsanat: poimi- ja aseta, lavaus, robotiikka, RobotStudio, IRB 14000, IRB 120

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

3D space	Three-dimensional space
CAGR	Compound annual growth rate
NC robots	Numerical control robots
PC	Personal computer
PLA	Polylactic acid
kg	kilogram
USD	United states dollar

.

1. INTRODUCTION

Some of the first robots were made in the 1950s along with the first industrial robots created by George Devol [1]. During the years robots and robotics have grown to be inseparable from modern technology and industry. The different ways of using robots have multiplied and the growing will not stop any time soon. From all different possible ways to use a robot, they are traditionally mostly used in industrial environments.

In 2022 the value of the worldwide industrial robotics market was USD 26.52 billion, moreover from 2023 to 2030, projections indicate that the compound annual growth rate (CAGR) of the market is 10.5% [2]. Robots are a great help in handling, assembling, processing, and transporting among other usages. One common way of using industrial robots is palletizing.

This bachelor's thesis is made for the Future Automation Systems & Technologies Laboratory at Tampere University. The purpose of this thesis is to learn about palletizing, robotics and follow up with the implementation of the palletizing task with an ABB robot. The way of palletizing used in this thesis is pick-and-place-method with an industrial articulated manipulator. The research questions for this paper are the following:

- How to create a palletizing program to lessen the work of the ABB robot's operator?
- How to implement the program with an ABB robot?

In order to implement a palletizing task, it is necessary to know the basics of robotics and to learn how to program robots. The robotics theory in this paper concentrates on industrial robotics.

This paper has five chapters including this introduction. The second chapter consists of six subchapters. The first one explores some popular ways of using and implementing palletizing and its different techniques. The second one explains more about robotics, how to choose a robot and about end effectors of robots. The third one describes the hardware utilized, including some comparison between the robots IRM 14000 and IRB 120 and their grippers. The fourth subchapter illuminates more on the software side of robots and a bit about teach pendants. The fifth one depicts the programming application

RobotStudio and the programming language Rapid it uses. The sixth subchapter narrates lightly about additive manufacturing that is also used before the implementation.

In the third chapter of this thesis is the implementation of the palletizing program. First the chapter describes the designing process of the program and secondly it explicates more detailed explanation on the whole programming, testing, and implementing process. The final chapter concludes this thesis by summarising the results and errors and analyses them.

2. THEORETICAL BACKGROUND

This chapter introduces first different types of palletizers and palletizing solutions. Second, more about robots and industrial robotics and end effectors of robots. Third, the chapter explains about the two ABB robots that were contemplated and considered to be used in this thesis. Fourth, theory about programming methodologies and robot programming. It also explains about teach pendants. Fifth, this chapter introduces the used application RobotStudio and the programming language Rapid. Finaly, little about additive manufacturing, which is used in the implementation.

2.1 Palletizing

The procedure of stacking objects onto a pallet is called palletizing. Palletizing is applied in order to ship the items as economically and efficiently as possible. When palletizing boxes, it is typical to wrap the pallet in plastic or in the case of palletizing differently shaped items they can be strapped to pallets. [3]

Automated palletizers can be either inline palletizers or robotic arm palletizers. Inline palletisers are used for single product businesses. Whereas robot arm palletisers are suited for products of a range of different shape and weight. [3]

Other robotic palletizing solutions are layer palletizing and mixed case palletizing. In Layer depalletizing & palletizing the gripping system handles a full layer of products at the same time, instead of picking and placing each case of the layer separately. Layer palletizers can also be made to handle multiple layers, which would increase the profitability even more. [4]

In mixed case palletizing the cases that are to be stacked onto a pallet are not identical. For example, the pallets that can be seen in grocery stores are packed with layers of different products. Wurll [5] gives an overview of a few different companies and their mixed case palletizing solutions. He presents both inline and robotic arm palletizers that are used by the companies. The difficulties in mixed case palletizing are the software for calculating the optimized mixed pallet and using the perfect gripper for each wanted case. [5]

In addition to other palletizing robots there are also pick-and-place robots that are common in manufacturing environments. Pick-and-place robots free human workers and handle repetitive tasks faster and more accurately. These robots can be used for palletizing among other possibilities, such as picking up parts for assembly. [6]

Packaging automation market includes robotic pick and place automation, secondary packaging automation, tertiary and palletizing automation [7]. Secondary packaging means the outer packaging of products which is often made of cardboard and includes labels on it. Tertiary packaging helps the transporting of products, it includes palletizing and shrink wrapping. [8] From 2023 to 2030 the packaging automation market is expected to expand at a CAGR of 9.8%, by 2030 it would reach USD 108.2 billion. The growing need for efficient transporting and delivery holds the need for new solutions high. [7]

2.2 Industrial manipulators

A robot is a mechanism that is programmed and has the ability of working without constant involvement of humans. A robot contains a control system and can execute either locomotion, manipulation, or positioning. There are many types of robots, for example industrial robots, service robots and medical robots. [9]

Industrial robots have three or more axes, and they are manipulators that can be reprogrammed for multiple purpose. Industrial robots are used in industrial environment, and they can be immovable or fixed to a mobile platform. [9] For example, robots that transport parts between warehouses and workstations are mobile robots. Their passages are strictly limited and fenced so that people can not get hurt by their fast movements. Another example of industrial robots are assembly robots, they have various parts within their reach, and they put those parts together. Assembly robots are also practically inside closed cages, so that their quick and sudden movements do not endanger anyone or anything.

All industrial robots consist of mechanical unit, drive, control system and tooling. The mechanical unit refers to the base, the manipulative arm, and the physical tool of the robot. The drive system is the power supply of the robot, it may be hydraulic, pneumatic, or electric. Each of these are used for different needs. [10] Pneumatic robots have strong, and fast movements. Hydraulic robots are the most powerful and they are used for lifting heavy forces, their movements are a bit more easily controlled, but the power supply system needs more space than the other options. Electric robots are very small and accurate in their work. The demand for electric robots has risen a lot in recent years.

The control system of a robot controls the whole robot. The controller communicates and processes information according to which it initiates, terminates, and coordinates the robots' motions and sequences. Tooling of a robot refers to the functions the robot needs to perform for the application. It may mean using a single spray-painting tool or changing tools during a work cycle and using them. [10]

In industrial environments, another type of robot utilized is collaborative robots. Collaboration with robots means that within the same space there are people and robots working. Collaboration robots are robots that are specifically made to work side-by-side with humans and to not cause any danger to them. Collaborative robots are special because they do not need to be located behind barriers like common industrial robots and this one of the reasons for their popularity. This capability is enabled by integrated sensors that detect external forces; if these forces exceed a certain limit, the robot automatically halts its movement [11].

One important term in robotics is degrees of freedom which signify the count of independent motions that a robot can achieve in relation to its base. Typically, the number of axes corresponds to the robot's degrees of freedom, indicating the directions along which the robot can move independently. In simpler terms, each joint in the robot's structure contributes to its degrees of freedom, as shown in figure 1, enabling the robot to reach various positions and orientations in 3D (three-dimensional) space. [10]

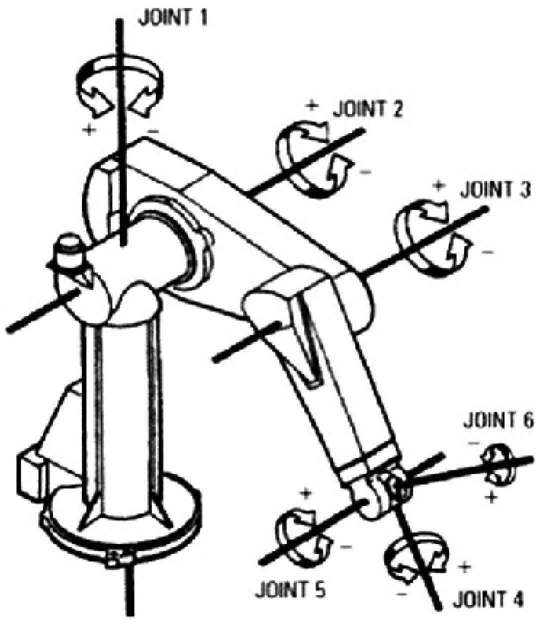

Figure 1: Robot with six degrees of freedom [10]

Five or six degrees of freedom are typical for robots. Some special applications have only four or possibly seven or more axes. [10] For example ABB’s IRB 14000 has seven axes and the palletizing robot IRB 660 has four axes [12].

2.2.1 End effectors

End effectors of a robot can be either tools or grippers. They are attached to the wrist of a robot and are used to perform the wanted task as in figure 2. Tools refer to devices that can be used to get a specific job done, for example drills, grinders, and paint sprays are tools. On the other hand, grippers are devices for gripping or holding an object. [10]

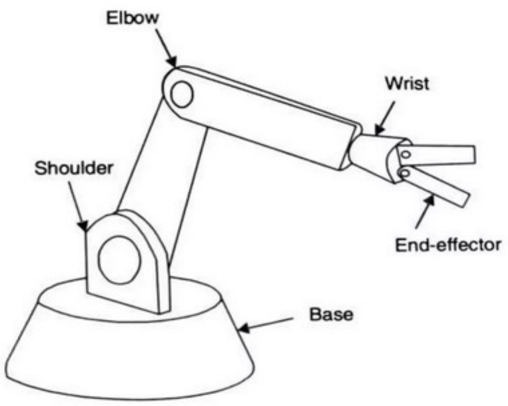

Figure 2: End effector attached to robot wrist [10]

Grippers can be mechanical hands, hooks, magnets, or suction devices. They may be used for applications of material handling, machine loading, or assembling. To select the perfect gripper for each use, there are many factors to consider, for instance source of power, gripping style, weight, sensor capabilities and number of jaws. [10]

The grippers could be pneumatic, hydraulic, or electrical. Pneumatic grippers are cost-efficient and require very little space. Hydraulic grippers are the most powerful, but they tend to be expensive and less accurate than the other options. Electrical grippers are very controllable and clean, but they have less force compared to pneumatic ones and are larger due to their motor. [10]

Gripping style can be categorized as parallel or angular, and either external or internal. Angular gripper rotates its jaws, making it useful for repositioning tooling, but they can be challenging to design and sometimes impractical. In contrast, parallel grippers require less space to work than angular ones and they are also simpler to make and use. External and internal grippers are either angular or parallel. With external ones grasping objects from the exterior surface and internal grippers use the internal surface. The number of jaws is also an important factor when selecting a gripper. Most often two jaws are enough. [10]

The acceptable weight of the gripper added to the weight of the gripped object varies on the robot utilized. Robots have different lifting capacities, for instance ABB's IRB 14000 can handle up to 0.5 kg, while the IRB 120 is capable of lifting 3 kg [13]. Grippers also differ in sensor capabilities, for example some grippers may have a proximity sensor or feedback mechanisms for sensing the griping force similarly machine vision and other sensor types are possible. [10]

One method of classifying grippers involves distinguishing between passive and active types. Passive grippers encompass nonprehensible end-effectors, wrap end-effectors, and pinch end-effectors. On the other hand, active grippers comprise active servo grippers and dexterous robot hands. [10]

ABB's Smart grippers for the robot IRB 14000 are electrical servo grippers that can also be equipped with one or two vacuum grippers, and it is also possible to add a machine vision to them. The gripper with servo fingers, vision and one vacuum weights 0.262 kg which implies that the robot can only lift 0.238 kg with the smart gripper. [14]

2.3 Description of the hardware

The robot used in this thesis was first supposed to be ABB's IRB 14000, also called YuMi. However, after testing the program with the robot, it was first realized that there is a problem with the robot's smart grippers and that it would be better to change the robot to another one. Another ABB robot was chosen. The smallest robot of ABB is IRB 120 in figure 4. The robot has 6 axes and weights 25 kg and can be mounted almost anywhere due to its portability and easy integration ability [13].

Figure 3: IRB 120 [15]

This robot was chosen because of practical reasons. Starting with the fact that the previous robot IRB 14000 was used for other purposes at the same time, so changing the grippers back to the necessary ones would have been very impractical. Additionally, time limits were getting closer, and the IRB 120 robot was found to be a lot more fitting for this implementation.

One difference about these two robots is that IRB 14000 has two robotic hands while IRB 120 has only one hand. The robot IRB 14000 is meant to be efficient and work with its both hands at the same time but in this implementation only its left hand would have been used. Which makes the robot choice impractical. The IRB 120 has only one hand, therefore it was a lot more practical and a more advantageous option.

Another problem with the first chosen robot IRB 14000 was the configurations of the arm. Degrees of freedom was easily lost when the robot arm tried to reach the calculated places. With IRB 120 this problem did not exist in same amounts, because this robot has only 6 axes.

The gripper used with IRB 120 is MHF2-12D2-X83A2 gripper [16]. It is a pneumatic and parallel, which means that the fingers slide open and closed in tracks. The weight of the gripper is 300 grams, and it has two jaws, therefore two fingers are needed. The sensor capabilities of the gripper are minimal and unnecessary for this project.

This gripper is classified as a pinch end-effector, making it a passive gripper designed for holding parts without controlling the grasp force or manipulating them. Pinch end-effectors require the addition of separate fingers as they are usually sold without them. [10]

2.4 Programming industrial robots and palletizers

Programming methodologies include analyzing problems by forming algorithms with contemporary programming methods, designing programs, and implementing them [17]. According to Harvey a software development methodology's goal is to enhance productivity, quality of code and collaboration. Some of the most popular development approaches are waterfall, agile, scrum, lean, and test-driven development. [18]

The waterfall model if a sequential approach to software development. First the developer knows the requirements for the project and makes the design according to them. Following the design phase comes the implementation, verification and at last maintenance. [18] In waterfall model all the steps are done only once, and the developer does not change the earlier parts after the step is done.

In contrast the agile software development is iterative and values individuals, interactions, working products, customer collaboration and adaptability over processes, tools, thorough documentation, contractual negotiations, and ensuing predefined plans. [18] The iterative methodologies are often described as loops because in them the developer does a bit of something and gets feedback about it and is ready to make changes if needed.

Scrum is a way to use agile software development. It separates projects into sprints that last from two to four weeks and highlights the significance of bravery, focus, dedication, respect, and transparency. The lean and test-driven development are also agile. The seven fundamental principles of lean software development are minimizing waste, improving feedback, postponing commitments, fast delivery, embedding integrity within the process, empowering the team, and sustaining a broad perspective. In test-driven development, the program is written to fulfill the requirements of a test for each new feature. Upon passing each of the tests the code is improved, this minimalizes the total amount of needed code. [18]

According to Gupta et al. industrial robots can be specified as manual manipulators, playbacks, NC (Numerical Control) robots or intelligent robots. These four levels explain different performance capabilities of robots. Manual manipulators are the simplest, they have fixed or preset task sequences to perform. The difference in playbacks is that they have a pre-programmed fixed instructions to repeat so their tasks and working patterns are a bit more complicated than the simple tasks of manual manipulators. NC robots also have their tasks ready-made, those tasks are given through numerically loaded information. The fourth level of industrial robots are intelligent robots and as their performance is controlled by their own recognition capabilities. [10]

The simplest way to make a robot program is to manually guide the robot through the wanted positions and record everything [19]. This robot guiding can be used with all robot types, but it is mainly used for manual manipulators, and playbacks. The programs for playbacks, NC robots, and intelligent robots can be modified and created using various programming languages. There are hundreds of different programming languages that are used for robot programming, but here are some popular robot programming languages: C++, Python, Pascal, Java, Scratch and Rapid [20].

All the programming methodologies can be used with any of the programming languages for creating programs for robots. According to Lozano-Perez in task-level programming the robots is taught positions of objects, instead of the whole movements [19].

Embedding data directly into the source code of an application is called hardcoding. It is used to make the code simpler, more efficient, portable, and more secure. Some disadvantages of hardcoding are inflexibility, lack of scalability, issues in version control and challenges in collaboration. [21]

Hardcoding in palletizing would mean teaching or programming every single object and movement to the robot separately. This method would take up a lot of time and would not benefit the robot operator at all. Instead, the software of the robot should automatically calculate the palletizing locations from some simple inputs from the operator, such as dimensions and properties of the palletizing pattern. [22]

2.4.1 Teach Pendants

The robot guiding explained in chapter 2.4 can be done in different ways. Collaboration robots can be guided by the operator by physically touching the robot with their hands and turning the axels of the robot's hand and wrist. However, industrial robots are located inside safety cages so the easiest way to guid them is with a teach pendant.

A teach pendant device is a tool for interacting with a physical robot remotely [23]. It can be used for example for guiding the robot, running programs with it and even for creating programs. According to McMahon teach pendants typically have a stop button for an instant shutdown, a display screen for looking and editing commands and a command input keyboard. Teach pendants allow the robot operator to have an unrestricted movement around the robot, instead of directly interfacing with the robot or relying on a fixed terminal. [23]

The ABB Teach Pendant 3HAC028357-001 has all the above-mentioned features. It is a wired, 3.06 kg handheld device and has a touch screen and a joystick instead of only keypads. [24] This pendant was used in this thesis for teaching the robot the necessary positions of the movable objects.

2.5 RobotStudio and Rapid

ABB's RobotStudio is the engineering tool used in this thesis for programming, configurating and simulating. RobotStudio is the PC (personal computer) application that ABB uses for modelling, programming, and simulating its robots and robot cells. It can be used either online or offline. In online mode, RobotStudio has a connection to a real robot controller, unlike in offline mode when the robot controller is only virtual. The virtual controller emulates a robot controller and works identical to the actual robot controllers although in a PC. [25]

In RobotStudio the operator can create 3D robot cells and visualize safety features, multi robot control and remote robot supervision. Also simulating and testing is easy through RobotStudio. To use RobotStudio the application needs to be downloaded and a license is also needed. [25]

The programming language in RobotStudio is Rapid. As a high-level programming language rapid utilizes English words instead of using only zeros and ones [26]. Rapid programs consist of program modules and system modules as in figure 3. Program modules may have different data and routines in them. In addition, a program can have multiple modules but only one entry procedure which is the Main procedure located in one of the program modules. Modules can be used for instance for defining interfaces or for storing geometrical data. [27]

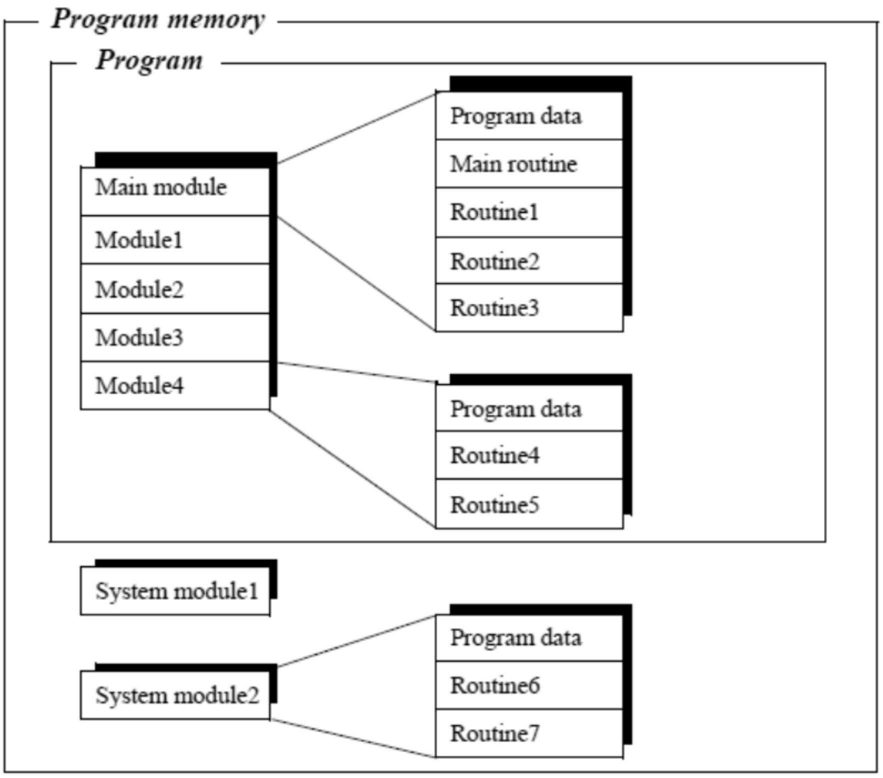

Figure 4: Outline of a rapid program [27]

As in figure 3, system modules however define system-specific data and routines, such as tools. Notably, these modules are excluded during program saving. Thus, any modifications made to a system module will impact all presently active programs or those loaded into the program memory at a later point. [27]

A Rapid program progresses line by line, beginning from the main procedure. To make the program simpler and easily readable it is necessary to use modules in where the wanted routines are written. [27]

Three categories of routines exist in Rapid; those are procedures, functions, and trap routines. Procedures function as subprograms, while a function is designed to return a value of a specific type and serves as an input for an instruction. In contrast, trap routines serve the purpose of responding to interrupts. These trap routines can be connected, with specific interrupts; for instance, when a particular input is activated, the corresponding trap routine is automatically executed. [27]

Each module begins with definition of data used in the module. In Rapid there are three categories for data: constants, variables, and persistents. A constant implies a fixed, unchanging value and is only modifiable manually. On the other hand, a variable is dynamic and can be assigned new values during the execution of a program. Finally, a persistent is akin to a "persistent" variable; upon saving a program, the initialization value of a persistent is also saved in its current value, so that the stored state persists across program sessions. [27]

One of Rapids more complex data types is a robtarget and it will be important for the program made in this thesis. Robtarget consists of position, orientation, and angles of each of the robot’s axes. [26] In other words, it is a point with place and a specific direction in which the robot reaches the point with a specific configuration.

Other necessary data for this thesis is numeric values, num, which in rapid can be integers or decimal numbers. Another fact about the language is that Rapid is not strict about upper or lower cases either, both can be used the same way. [27]

To make Rapid programs readable to all users the “RAPID development guidelines for handling applications” was made. It gives instructions on how to for example use program modules or name targets and signals universally. [28]

2.6 Additive manufacturing

Additive manufacturing creates objects from computer aided designs (CAD) by building the object layer by layer, this also called 3D-printing [29]. For this thesis the two pallets, the six test objects and the two fingers of the gripper are self-designed, and 3D printed. The movable objects, as in figure 5, are designed so that they can be picked and placed easily, additionally they can be piled on top of one another if chosen. The pallets have each six places for the movable objects to be placed at. This way it is easier to see the preciseness of the robot.

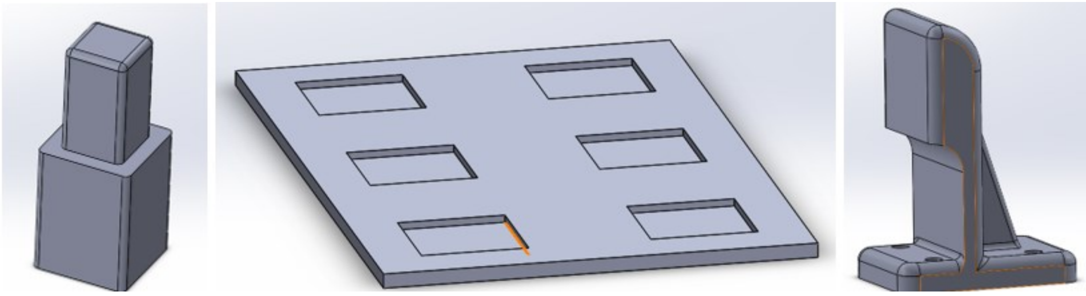

Figure 5: A movable object, a pallet, and a gripper finger

In figure 5 there is also the design for the gripper finger. As told in chapter 2.3 the MHF2-12D2-X83A2 gripper require separate gripper fingers. Creation of new gripper fingers was also necessary due to the small size of the movable objects.

The material for the 3D printing was PLA, polylactic acid, which is the most popular filament in 3D printing. It is popular because it is easy to print, and the needed temperature is quite low. It does not smell bad like some other filaments and there are a lot of different colour and special effect choices. PLA is also made of renewable resources such as sugar cane or corn starch. [30]

Printing time for the six movable objects was about 4 hours, printing the two pallets lasted also for 4 hours, but the two gripper fingers only took 1,5 hours.

3. IMPLEMENTATION

This chapter is about the implementation. First, about the designing process of the program and after that comes the programming phase and the program made for this thesis.

3.1 Designing

The program is made with an ABB robot so one requirement for the program is that it is made with RobotStudio and Rapid. The program should be clear and easy to understand. Overall, the functional requirements for the robot program are the following:

- find the correct place to go next,
- move the gripper above the object that is to be picked up,
- lower down to the correct position to pick the object without tipping over any objects,
- grab the object and rise it above the other objects, in a way that nothing tips over,
- move to the correct place to place the object,
- lower down to the correct height,
- release the object from the grip,
- go back up without tipping anything over,
- repeat the process.

The functional requirements include movements for the robot and for the gripper. Furthermore, the need to find the objects and the correct place for placing the objects appears. Additionally, the right amount of repetition is also to be included within the program.

The programming methodology used is the iterative test-driven development and task-level programming. The robot is taught positions of objects, instead of saving each of the needed movements. The more restrictive movements as lowering down and rising above the other objects are written in the program, but moving from the first pallet to the second pallet is not restricted. Instead, it is made easier for the robot by using the MoveJ -command, which tells the robot to use movements that are practical and simple from the view of the robot's joints instead of linear movements which might look simple to the operators's eye but can include many impractical turns and twists for the robot.

The program and testing are made in offline mode with RobotStudio by using a virtual controller and visualizing everything within RobotStudio. First the chosen robot and a virtual controller for it are added to the RobotStudio project. The positions that are necessary to be taught to the robot are visualized and saved. With having some positions saved the program can either be created step by step with the help of the visualization and converting the saved movements into code or by directly writing the program with Rapid. This program was made with both of the ways.

The first test for the program was to make the robot move above the first target and to make a linear down and up movement for picking the object. The second test was to move over to the second target and make a linear down and up movement for placing the object. Both of these were made with visual programming. After the larger movements were done the timings and gripper opening and closing was created by writing directly in Rapid. The location calculating was also written with Rapid, but only after the whole pick-and-place function was working correctly.

3.2 Programming

As the goal of the program is to be as practical as possible and lessen the work of the robot's operator as much as possible, the program will be divided to two different parts. The parts will be calculation and the movement procedure. In figure 6 is the flowchart of the program.

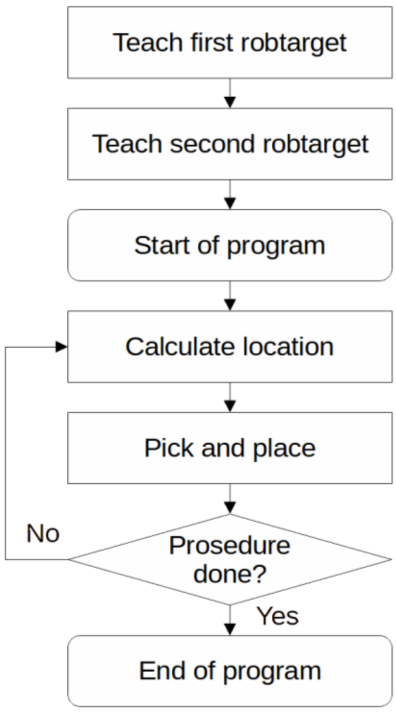

Figure 6: Flowchart of the program

As in the flowchart in figure 6, first the robot controller is taught the first two positions, orientations, and configurations as robtargets. The teaching is done with the robots Teach Pendant, which is a handheld tool for instructing and manipulating the robot's axes. Next the program can begin. First the program sends the taught robtargets to the pick-and-place procedure and performs the first movement round. Then it calculates the positions for the next object and continues through all six objects. Velocity of the robot will be faster when moving from a pallet to another than when lowering or lifting the end-effector of the robot.

Pseudo code is a way to represent algorithms and implementations to people who might not even understand the used programming language. It is made of annotations and informative text with no syntax thus computers would not be able to interpret it. [31] In program 1 is the pseudo code of the program made for this thesis.

```
/ This program has 2 parts first the counting of coordinates and
/ then the pick and place algorithm for the used robot.

/ The robot needs to be taught the first position of both pallets
CONST robtarget Pallet_1;
CONST robtarget Pallet_2;

PROCEDURE Count_coordinates()
    VAR num x := 0;
    VAR num y := 0;

    / Count the place for the next object
    FOR x_placement FROM 1 TO 3 DO
        y := 0;
        FOR y_placement FROM 1 TO 2 DO
            Pick_and_place x,y;
            y := y + 47;   // Add 47mm as it is the distance to
                           // the next place
        ENDFOR
        x := x + 47;
    ENDFOR
END PROCEDURE

/ Instruction to the used robot
/ Picks and places according to the given coordinates
PROCEDURE Pick_and_place(num x, num y)

    / The new coordinates are based on the first coordinates
    / of each pallet
    VAR robtarget pick is the given coordinates on first pallet;
    VAR robtarget place is the given coordinates on second pallet;
    VAR robtarget pick_point is 70mm upwards from first pallet;
    VAR robtarget place_point is 70mm upwards from second pallet;

    / Move to the given position and pick the object
    MoveJ to pick_point;
    MoveL to pick;
    Wait for 0.5 seconds;
    Close the gripper;
    Wait for 0.5 seconds;
    MoveL to pick_point;

    ! Move to the given position and place the object
    MoveJ to place_point;
```

```
            MoveL to place;
46          Wait for 0.5 seconds;
            Open the gripper;
48          Wait for 0.5 seconds;
            MoveL to place_point;
50
        END PROCEDURE
```

Program 1. *Pseudocode of the program*

In program 1, from lines 8 to 22 is the Count_coordinates procedure which fulfils the goal of the program. Which was to lessen the work of the robot's operator. Without the procedure the operator would have to teach each of the twelve places on each pallet separately to the robot. After that in the procedure Pick_and_place is the instructions, the controller gives to the robot hand for executing the wanted actions.

4. EXPERIMENTAL RESULTS

In this thesis the purpose was to palletize with the pick-and-place method, as the available robots in the laboratory are industrial robot arms and no other types of palletizers.

While I made, tested, and simulated the program within RobotStudio a concern about the used coordination rose. The concern was about placing the pallets in a different angle than straight in the robot's coordination. The pallets need to be straight in relation to the robot's own coordination system moreover the two pallets should be the same size or at least the first pallet should be smaller than the second so that all the objects have space. Though, the objects can be twisted around their z-axis if they keep the same distance from one another, this is because the robot finds the other objects according to the place and orientation of the first object. In the end this problem about the pallets' angles was left unsolved, and the pallets cannot be placed however wanted.

It was slightly difficult to find how to program the grippers in RobotStudio. First, I tried to get the virtual smart grippers to work with the help of some videos, but in the end the pneumatical MHF2-12D2-X83A2 gripper started to work as wanted with some help from the supervisor. The smart grippers would have been unnecessarily complex for this thesis as the only needed function would have been the servo fingers. Additionally, the problem with the laboratory's smart grippers was in the servos precisely. With the servos I would have had to give the grippers the exact width how much I want them to press the object to pick them up, but the laboratory's grippers tended to press too hard on objects.

On the day I was supposed to run my program with the laboratory's robot the smart grippers of the IRB 14000 were nowhere to be found. Other users of the laboratory had changed the end effectors of the robot. The solution to this drawback was to change the whole robot to another one that would have different grippers. Commissioning of the robot would have been faster with the smart grippers because the gripper of IRB 120 needed the fingers to be 3D printed before it could be used.

In the programming and simulating phase, I used the virtual IRB 14000 robot. Despite this when precisely the same program was used with IRB 120 it worked perfectly. This proved that the same RobotStudio program can be used with multiple ABB robots without changes or complicity.

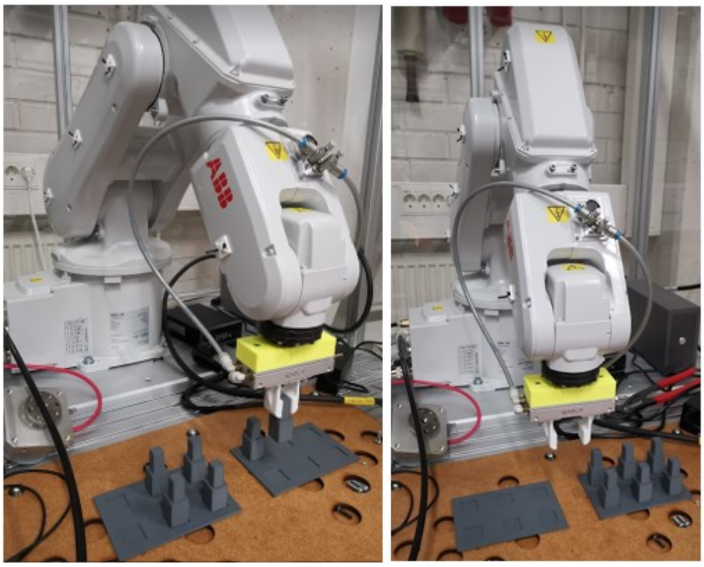

Figure 7: IRB 120 in action and with the gripper and objects

When running the program with IRB 120, the robot picked and placed each of the objects easily. In figure 7 is the final chosen robot IRB 120 with its gripper and all the 3D printed objects and pallets.

The palletizing strategy I used in this thesis is more on the dynamic side and it is not completely hardcoded palletizing. In hardcoded palletising each of the objects would have to had been taught for the robot separately, that would have meant six for the first pallet and six for the second pallet, so all together 12 robtargets. Instead, in my program the robot only needed two robtargets to get started.

In my program the number of movable objects and the distance between objects are fixed, but they could easily be changed by updating the numbers in the calculating procedure. The numbers could also be changed to be variables and the program could ask them from the operator as an input, but these changes were not seen necessary.

My program can be used for any small pick-and-place problem. The size limit for the usable pallets is the reach of the robotic arm, and the robot would stop moving if the program told it to reach too far.

In the end, the change of the robot was only good for my thesis. I got to compare the robots and see their similarities and differences. I also had read that the RobotStudio programs would work on other ABB robots as well and it was interesting to get the chance to prove that. If I were to further improve this thesis I would find even more about the similarities and difference of the robots and I would also find more about the reasons why does the program work with both robots.

One result achieved with this thesis was that using RobotStudio is not too difficult to learn, and it is a great and flexible tool for robot programming. However, as a result of the pallet positioning concern, the program is not completely versatile to other purposes than its current application.

Errors in this thesis might be caused by the one-sidedness of sources in the robotics chapter. Another error may be the way the program was designed. I could have searched for readymade pick-and-place algorithms and learn from them, whereas instead I created everything from the beginning myself.

5. CONCLUSIONS

Palletizing is a quickly growing and an important market. Robotics are developing and spreading to different professions. Thanks to collaboration robots the uses for robots are widening.

RobotStudio was found to be a great and versatile tool and with Rapid it makes robot programming, simulating, and operating easy. Robot programming does not differ much from other software creating, except for the robot instructing part. Additive manufacturing is a great and quick way to create parts. Designing and programming go hand in hand. I should have planned and learned about the programming methodologies before beginning the programming phase.

The first goal of this thesis was to learn about palletizing and robotics enough to create a practical palletizing program to lessen the work of the ABB robot's operator. RobotStudio was proved to be very practical and flexible tool to reach this aim.

The program could have been made in other ways as well, but the used way was found to be practical and sufficient. Though the first change I would make to the program would be changing the numeric values in the calculation procedure into variables, so that the program would be even easier to use for the robot operator.

The second aim of this thesis was to implement the program with an ABB robot. Though the program was first created and directed for IRB 14000 it also worked perfectly with the small one-handed IRB 120. The implementation included making the program, choosing the robot and gripper and 3D printing the necessary pieces.

REFERENCES

[1] Robotnik, History of robots and robotics, 2021 (Read 5.2.2024). Available: https://robotnik.eu/history-of-robots-and-robotics/

[2] Grand view research, Industrial Robotics Market Size, Share & Trends Analysis Report By Application (Handling, Assembling, Processing), By End-Use (Automotive, Electronics), By Region, And Segment Forecasts, 2023 – 2030. (Read 7.3.2024). Available: https://www.grandviewresearch.com/industry-analysis/industrial-robotics-market

[3] IPM, What Is Palletizing? Types & Products, 2023 (Read 5.10.2023). Available: https://callipm.com/blog/what-is-palletizing/

[4] MHI, Robotic Palletizing, 2023 (Read 10.11.2023). Available: https://www.mhi.org/solutions-community/solutions-guide/robotics

[5] C. Wurll, Mixed Case Palletizing with Industrial Robots, Proceedings of ISR 2016: 47st International Symposium on Robotics, Munich, Germany, 2016, pp. 1–6. Available: https://ieeexplore-ieee-org.libproxy.tuni.fi/document/7559186

[6] 6 River Systems, What is a pick and place robot? 2023 (Read 24.11.2023) Available: https://6river.com/what-is-a-pick-and-place-robot/

[7] Meticulous Research, Packaging Automation Market by Offering, Type (Robotic Pick & Place, Secondary Packaging, Tertiary & Palletizing), End-use Industry (Healthcare & Pharmaceuticals, E-Commerce & Logistics, Food & Beverage), and Geography - Global Forecast to 2030, 2023 (Read 11.3.2024) Available: https://www.meticulousresearch.com/product/packaging-automation-market-5324

[8] Wayne Automation Corporation, Primary, Secondary, and Tertiary Packaging: How Each One Supports Your Business, 2024 (Read 11.3.2024) Available: https://wayneautomation.com/news/primary-secondary-and-tertiary-packaging/

[9] Robotics – Vocabulary. The Finnish Association for Standardization, ISO 8373 Helsinki, 2021, pp. 22. Available: https://online.sfs.fi/fi/index/tuotteet/ISO/ISO/ID9998/8/1043935.html.stx#

[10] A. K. Gupta, S. K. Arora, J. R. Westcott, Industrial Automation and Robotics. Dulles, 2017, pp. 397–560.

[11] Robots and robotic devices — Collaborative robots. The Finnish Association for Standardization, ISO/TS 15066 Helsinki, 2016, pp. 33. Available: https://online.sfs.fi/fi/index/tuotteet/ISO/ISO/ID9998/1/402851.html.stx#

[12] Robo DK, ABB IRB 660-180/3.15, 2024 (Read 7.3.2024). Available: https://robodk.com/robot/ABB/IRB-660-180-3-15

[13] Robots.com ABB IRB 120, 2023 (Read 23.11.2023). Available: https://www.robots.com/industrial-robots/abb-irb-120

[14] ABB, Product manual IRB 14000 gripper, 2015 (Read 29.3.2014) Available: https://library.e.abb.com/public/6c35d74e5be34fcb93bf6810c449a2bd/3HAC054949%20PM%20IRB%2014000%20Gripper-en.pdf

[15] ABB, Product specification IRB 120, 2019 (Read 25.12.2023). Available: https://library.e.abb.com/public/6aed5e91083f4fceb358eea2fe4c1bab/3HAC035960%20PS%20IRB%20120-en.pdf?x-sign=edIex5StIjpgmJBJJ95tak9NHdyuuut6mzzJHESGKt5i1JG8dhL-RBdvggKptgBJn

[16] SMC, MHF2-12D2-X83A2 gripper, low-profile, MHF2 GRIPPERS. (Read 20.12.2023). Available: https://www.smcpneumatics.com/MHF2-12D2-X83A2.html

[17] V. Kowalenko, Programming methodology, The Partition Method for a Power Series Expansion, 2017. Available: https://www.sciencedirect.com/topics/mathematics/programming-methodology

[18] C. Harvey, Top 10 Programming methodologies, developer.com, 2022. Available: https://www.developer.com/project-management/10-top-programming-methodologies/

[19] T. Lozano-Perez, Robot Programming, Proceedings of the IEEE, vol. 71, no. 7, July 1983, pp 821–841. Available: https://homes.cs.washington.edu/~ztatlock/599z-17sp/papers/robot-programming-lozano-perez-83.pdf

[20] Nanit Robot, Programming and Robotics: Robot Coding Languages, 2023. (Read 18.3.2024) Available: https://nanitrobot.com/en/movy_programyvannia/

[21] J. Knell, What is Hard Coding, Bloom Institute of Technology, 2023. (Read 29.3.2024) Available: https://www.bloomtech.com/article/what-is-hard-coding

[22] A. Owen-Hill, The Essential Guide to Robotic Palletizing, RoboDK, 2020. (Read 29.3.2024) Available: https://robodk.com/blog/the-essential-guide-to-robotic-palletizing/

[23] M. McMahon, What is a Tech Pendant? EasyTechJunkie, 2011, modified 2024. (Read 28.3.2024) Available: https://www.easytechjunkie.com/what-is-a-teach-pendant.htm

[24] J. Kosmala, video: Robot teach pendant overview, ABB, 2010. Available: https://new.abb.com/products/3HAC028357-001/3hac028357-001 -> Downloads

[25] ABB, Operating manual RobotStudio, 2023 (Read 6.11.2023). Available: https://library.e.abb.com/public/ddc614e6eef84ab7ac619c164b0ef8b8/3HAC032104%20OM%20RobotStudio-en.pdf

[26] ABB, Operating manual Introduction to RAPID, 2007 (Read 13.11.2023) Available: https://moodle.tuni.fi/pluginfile.php/3545492/mod_resource/content/0/Introduction%20to%20Rapid%20programming.pdf

[27] ABB, Technical reference manual RAPID overview, 2004-2023 (Read 27.11.2023) Available: https://search.abb.com/library/Download.aspx?DocumentID=3HAC050947-001&LanguageCode=en&DocumentPartId=&Action=Launch

[28] ABB, Application manual RAPID development guidelines for handling applications, 2013 (Read 12.11.2023). Available: https://library.e.abb.com/public/52b9b59332b33318c1257b2d00413bfa/RWMT_AM_3HAC046417-en.pdf

[29] TWI, What is additive manufacturing? Definition, types and processes, 2019 (Read 11.3.2024) Available: https://www.twi-global.com/technical-knowledge/faqs/what-is-additive-manufacturing

[30] 3Dkauppa, PLA, (Read 20.12.2023) Available: https://www.3dkauppa.com/tietoa-filamenteista

[31] Geeks for Geeks, How to write a Pseudo Code? Updated 23.11.2023 (Read 27.11.2023) Available: https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/