UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Technology

Iryna Hurova

Kitting station of the learning factory

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisor(s):
Associate Professor of robotics engineering Karl Kruusamäe
Lecturer of robotics technology Veiko Vunder

Tartu 2023

Kitting station of the learning factory

Abstract:

Learning factories has become a new educational approach that allows students to get hands-on experience working in a simplified factory environment. The aim of this thesis is to assemble a multi-robot setup where two or more autonomous robots can communicate with each other to achieve the desired goal. Robot Operating System (ROS) is used to create software for the manipulator robot in the kitting station, which figures as one of the components of the learning factory.

Keywords:

Learning factory, ROS, xArm7, multi-robot system, autonomous robot.

CERCS: T120 Systems engineering, computer technology; T125 Automation, robotics, control engineering

Õppetehase komplekteerimisjaam

Lühikokkuvõte:

Õppetehas on hariduslik lähenemine, mis võimaldab õpilastel saada praktilisi kogemusi töötamisel lihtsustatud tootmiskeskkonnas. Selle lõputöö eesmärk on kokku panna mitme robotiga seadistus, kus kaks või enam autonoomset robotit saavad omavahel suhelda soovitud eesmärgi saavutamiseks. Robootika arendusplatvormi ROS kasutatakse tarkvara loomiseks komplekteerimisjaama manipulaatorrobotile, mis on üks õppetehase komponentidest.

Võtmesõnad:

Õppetehas, ROS, xArm7, mitme roboti süsteem, autonoomne robot

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T125 Automatiseerimine, robootika, juhtimistehnika

TABLE OF CONTENTS

ABBREVIATIONS

LF - Learning Factory

ROS - Robot Operating System

IP - Internet Protocol

PLC - Programmable Logic Controller

RFID - Radio Frequency Identification

DMC - Data Matrix Codes

VPS - Value-oriented Production System

IoT - Internet of Things

DHCP - Dynamic Host Configuration Protocol

YAML - YAML Ain't Markup Language

PC - Personal Computer

URDF - Unified Robotics Description Format

SDK - Software Development Kit

AR - Augmented Reality

1 INTRODUCTION

Robotics technology has the potential to bring many benefits to society, making our lives easier, safer, and more efficient. It is now the beginning of an era when robots become smarter and more capable of performing a wider range of tasks. The integration of robotics technology in manufacturing, customer services, agriculture, healthcare [1], and overall in our day-to-day life is happening in real-time worldwide. In order to realize the full potential of the robots in the areas mentioned before they need to be able to communicate effectively with each other as well as with humans. Therefore, many big tech companies are seeking to hire more experienced workers who have already acquired the necessary skills and knowledge needed for working with complex robotic systems [2].

Traditional educational methods provide learning through lectures, laboratory work, projects, and competitions. Students can gain theoretical knowledge from lectures and later apply it during the practical sessions, where they have an opportunity to program real or simulated robots. With this type of studying a problem may arise when applying obtained skills in a real-world environment [3].

Creating a simplified version of the working environment where robots need to cooperate may help to close the gap between industry and academia. By practicing with an already existing setup students may learn how to analyze and improve the pieces of software that were developed by other people. In addition, they can familiarize themselves with the solutions that were used to assemble all the components together, try to develop a better solution or adapt it for a particular purpose. This learning method will prepare future employees to be more effective and efficient when they start working in the industry.

The goal of this thesis is to develop a kitting station component of the learning factory and integrate it into the multi-robot system. The final result has the potential to serve as an introduction for students to the tools and technologies commonly utilized in modern robotics projects.

2 LITERATURE REVIEW

Robots have become an essential part of the modern world since they have a wide range of applications in almost all industries because of their precision and convenience [1]. In many cases, robots are better suited than humans for certain tasks for various reasons. For instance, in fields like manufacturing, and agriculture they can be useful for performing monotonous tasks that require high precision [1]. Robots can also be helpful in risky fields like underwater or space exploration, which are dangerous for humans. Replacing regular workers in industries such as customer service, food preparation, and entertainment can be cost-effective in the long run, and can also serve as an interesting attraction for customers [1].

A massive integration of robots in different fields of our lives requires a lot of qualified specialists who can program and maintain those robots. Consequently, a new challenge emerges on how to equip future workers with both theoretical knowledge and practical skills that are needed to catch up with a rapidly evolving industry landscape [4].

This chapter covers the possible improvement of traditional education methods that will help in bridging the gap between industry and academia (Section 2.1), the technologies that it should teach students (Section 2.2), and the tools that can be used for this purpose (Section 2.3).

2.1 Learning Factories

The high demand for improved and enhanced productivity in industrial applications requires the deployment of robots to automate tasks. As a result, an increasing number of companies are investing in industrial robots to improve their operational efficiency and overall output [3], [5]. To provide an overview of the real-world factory to individuals with limited experience in the field, and develop an understanding of the practical applications of robotic technology, while also sharpening their theoretical knowledge, new teaching techniques have been introduced. Among these methods, the Learning Factory (LF) has gained prominence [3]. LF is a work-based learning approach that has been adapted from the field of medicine to provide students with both practical and theoretical experience. Through the use of the LF, students can gain hands-on experience working with industrial robots in a simulated factory environment [3]. The makeup of an LF may differ [3], with the possibility of incorporating various types of robots such as manipulator or mobile robots, as well as the

potential presence of diverse sensing devices that can be attached to the robots or function independently.

If a setup of potential LF satisfies the characteristics outlined in Figure 1, it can be classified as a learning factory [3]. Namely, the setup, depending on its intended purpose, should provide education through teaching (providing information), training (developing abilities), or research. Authentic experience ensures that the learner will familiarize themselves with both technical and organizational aspects of factory work by testing out their theoretical knowledge across multiple stations [3]. With an adaptable setup, it is easier to construct a new combination of workstations that would meet specific educational objectives for a particular course. Similarly to the real-world factory, the simulated factory should produce a product or service as output. The didactical concept should incorporate formal and informal learning, allowing trainees to learn through their own actions, whether through on-site or remote learning. Lastly, an operational model is necessary to ensure the learning factory is functioning properly [3].

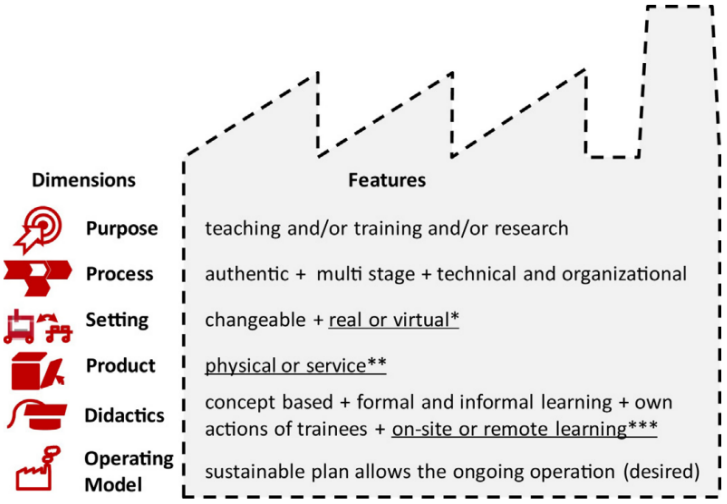

Fig. 1. Key characteristics of learning factories [3].

2.1.1 Classification of learning factories.

Numerous LFs have been effectively employed to facilitate student learning in robotics [6]–[8]. There are three main types of LFs [9]:

1) Automation LF
2) Process LF
3) Competition LF

Automation LF is designed to teach students the basics of automation, including Programmable Logic Controller (PLC) technology, sensors, and industrial networking [9]. An LFs create a simplified copy of a real automated production line, allowing students to study the function of the automation technology without being overwhelmed by the complexity of a real factory. The main focus is on presenting state-of-the-art automation technology.

The Industry 4.0 LF AutFab of the University of Applied Sciences Darmstadt [8] is a good representative of the automation LF (Fig. 2). It incorporates elements of Industry 4.0 and Smart Factory technologies, which involve the integration of control processes utilizing Radio Frequency Identification (RFID) technology, Ethernet-based networks, cyber-physical components, and the implementation of mobile robots. The facility includes a control room where operators can monitor and optimize production processes in real-time using advanced software applications.

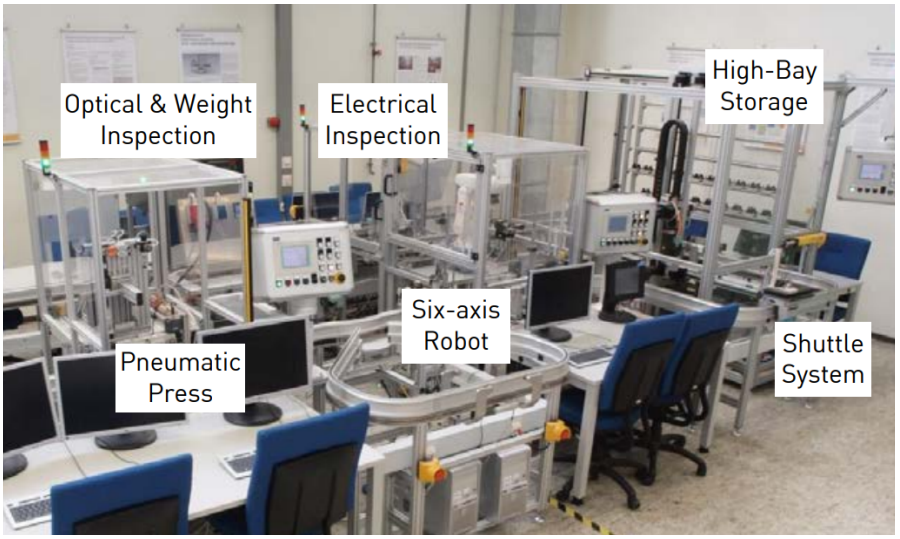

Fig. 2. *The fully automated Industry 4.0 LF AutFab of the University of Applied Sciences Darmstadt [8].*

The AutFab comprises multiple workstations, including a high bay storage, two assembly stations, and two inspection stations (Fig. 2) [8]. The high bay storage is operated by a three-axis motion-controlled robot, while the assembly stations utilize a six-axis industrial robot and a pneumatic press. The first inspection station uses an automatic optical

inspection system to scan all two-dimensional data matrix codes on the components and verify the correct assembly of all required components, including monitoring the weight of the final product. The second inspection station conducts an electrical test that automatically adapts to the specific product using one of the Data Matrix Codes (DMC). Intelligent shuttles are responsible for the transportation of components between the various workstations [8].

Process LF teaches students about the optimization of processes rather than technology [9]. This type of LF deals with lean methods and principles, like value stream analysis and design [10], just-in-time [11], line balancing [12], problem-solving, or job optimization. The setup is completely focused on modeling the process in question. The model must be accurate enough to use the same optimization methods used in industry. For example, at BMW LF (Fig. 3) [7] employees learn how to apply the principles of the Value-oriented Production System (VPS) to their work, including reducing lead times, improving quality, increasing flexibility, and optimizing productivity. The facility includes a simulated production line, where trainees can practice implementing VPS techniques and working as a team to identify and solve production problems.

Fig. 3. *BMW LF in Munich, Germany. VPS center training area [7].*

Competition/gamification LF involves a project-based learning approach [9]. The concept of such experiential learning is based on the idea that learning can take place without the presence of a teacher, as it relies on the meaning-making process of the learner's experience [3]. This approach involves going through several stages of transformation,

including a concrete experience, observation, and reflection to determine what is working or failing, abstract conceptualization to understand cause-and-effect relationships and analyze events, and active experimentation where the learner puts their ideas into practice and translates their new understanding into predictions [3]. Through this process, knowledge is gained and learners are empowered to take control of their own learning [3].

An example of this type of LF is the RoboCup Logistics League [6]. Teams compete to optimize material flow in a virtual factory by stacking colored bases and rings according to real-time orders (Fig. 4). Robots must plan and execute while considering uncertainty and minimizing travel distances. Each team has machines for specific purposes and scores points for successful product delivery. Two teams compete simultaneously, requiring collision avoidance and re-routing when necessary. The robots communicate wirelessly, but network availability and reliability can vary, requiring robust coordination.

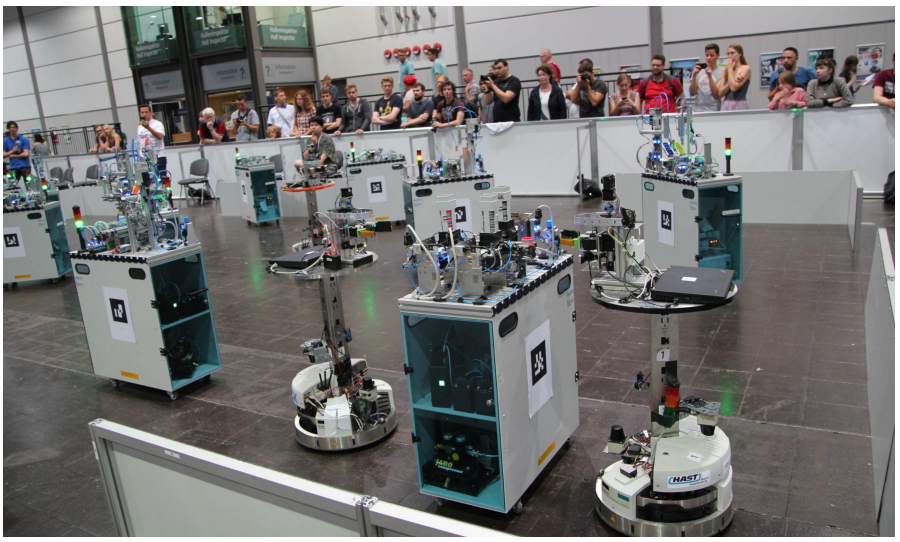

Fig. 4. *RoboCup Logistics League [6].*

2.2 Autonomous Robotics

It has been over 50 years now since humans started a big journey of robotics automation [13]. A great demand for autonomous robots has risen when people understood the contribution of autonomous robots to industrial progress. From performing simple monotonous tasks with little to no awareness of their surroundings, robots quickly transformed into smart autonomous robotics systems that have applications in various fields, such as manufacturing, transportation, agriculture, healthcare, and space exploration [14]. The benefits gained from the incorporation of autonomous robots include labor and utilization

stability, reduction of error rate, lower long-term costs, increase in human safety, and improved data collection [14].

2.2.1 Industrial Revolutions

Industrial revolutions drive advancements in the production industry by introducing new technologies, leading to a significant change in how manufacturing processes are carried out (Fig. 5). Most modern-day factories and production industries are currently at Industry 3.0 evolution level [15]. The invention of a PLC and various electronic devices such as integrated circuits and transistors brought autonomous robots to reality. Nevertheless, technological innovations are becoming even more rapid day by day, and now, we are in the midst of the fourth industrial revolution: Industry 4.0 [15]. The aim of Industry 4.0 is to introduce a better version of machines, that will be able to perform real-time data monitoring, tracking the status and positions of products as well as to hold the instructions to control production processes [16]. The key elements of the fourth industrial revolution are cloud computing and big data, cyber-physical systems, machine learning, artificial intelligence, and the Internet of Things (IoT) [15], [16]. In the close future, Industry 5.0 is going to complement the advancements made during Industry 4.0 by bringing humans back into the equation [17].

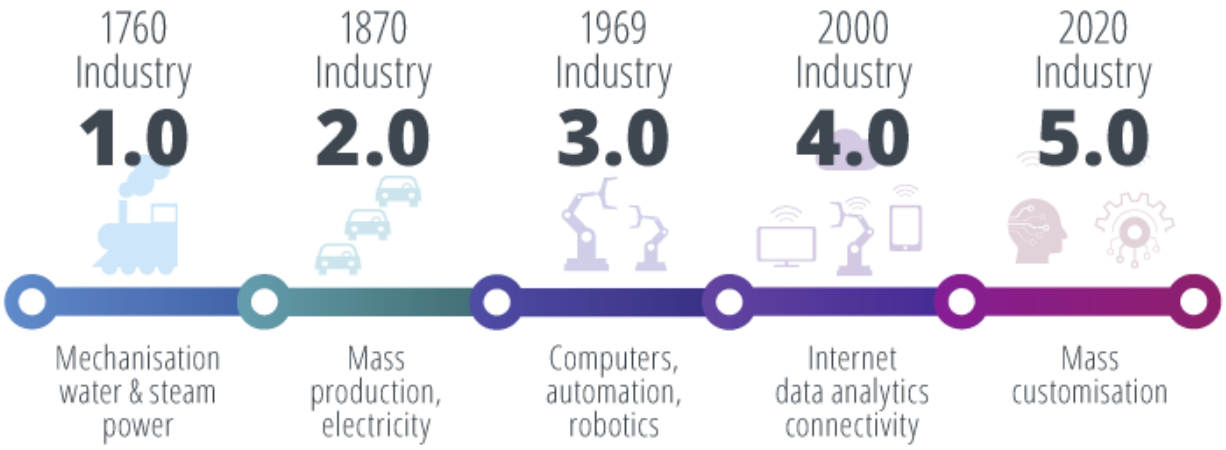

Fig. 5. *Five industrial revolutions [18].*

2.2.2 Industry 4.0 autonomous robots

Today the key component of automated systems highlighted in many articles is artificial intelligence and machine learning [13], [14], [19]. Industry 4.0 robots are expected to be fully aware of their surroundings, analyze and adapt to the changing environment, and

have the ability to make reasonable decisions based on the information gathered from various sensing devices and other robots, all of which comprise a large, automated system [20]. Sensors like cameras, lidars, and radars can accurately detect and recognize objects and obstacles in the industrial environment [21]. In order to navigate and avoid obstacles a robot must be capable of accurately determining its position in the industrial environment and building a map of its surroundings. It should be able to plan and execute its actions to achieve the objectives, such as assembling a product or transporting materials based on the information it gathers from the sensing devices [22]. Automated robots should be able to control their own movements and perform tasks through their actuators such as motors, grippers, and manipulators. These actuators should be robust and able to withstand the harsh conditions of the industrial environment [20]. Another crucial part is the ability to establish communication with other machines, sensors, and humans in the industrial environment. This communication may take place over wired or wireless networks and should be secure and reliable [20]. Finally, the design of the robot has to ensure the safety of humans and other living beings around it. This includes fail-safe mechanisms to prevent accidents and minimize the risk of injury [22].

2.2.3 Multi-robot autonomous systems

As mentioned in Figure 1 LFs are meant to be multi-stage. In order to fulfill this requirement different types of stations can be assembled into one complex functioning system. One way to create a multi-stage setup is to use multiple robots that communicate with each other to achieve the desired goal. Creating robotic systems that involve multiple robots presents additional challenges beyond those encountered when developing single-robot systems. There are two general categories of multi-robot systems: swarms, in which all robots demonstrate the same behavior, and teams, in which each robot behaves distinctly [19]. In either scenario, the system's behavioral requirements must be considered on the microscopic level (individual robots) and the macroscopic level (entire system). Multi-robot systems can present difficulties in ensuring safety and liveness properties due to the number of concurrent interacting agents and the changeability of the system’s environment [19]. Despite all the complications, that kind of system is necessary to create a fully automated factory environment.

2.3 Robot Operating System (ROS)

ROS is a powerful tool for building and testing robotics applications, and it has become an essential part of the robotics ecosystem [23]. The key features that make it so

convenient are publish-subscribe messaging system which enables communication between software components, various tools for visualizing robot data, such as sensor readings and robot movements, simulation environments, that allow developers to test and debug their software before deploying it on a physical robot, etc [23]. Hardware agnosticism is one of the reasons why companies may prefer to use ROS. A package management system makes it easy to install, update, and share software components that can be implemented on different devices with little to no modifications [24]. ROS is also well-suited for constructing heterogeneous systems, which makes it useful for establishing multi-robot communication (Section 3.1.2) [25].

ROS is becoming an increasingly important tool in industrial robotics. Its modular architecture and extensive number of tools and libraries make it an attractive choice for developers and companies looking to build advanced robotics systems [26].

2.3.1 Robots that support ROS

Some notable companies that use ROS in their products and services include Starship Technologies, Boston Dynamics, Yanu, 10Lines, and ABB Robotics [23]. Many companies have ROS-based software support for their robots. Popular manipulator robots with ROS support include Universal Robots [27], [28], Franka Emika Panda Arm [29], UFactory xArm [30], [31] KUKA Robots [32], [33], ABB Robots [34], [35] (Fig. 6). Universal Robots and Franka Emika Panda Arm are characterized as collaborative robots capable of working alongside human operators, while UFactory, KUKA, and ABB are industrial robots primarily used in manufacturing and automation. Finally, Kinova Robots are designed for research and educational purposes.

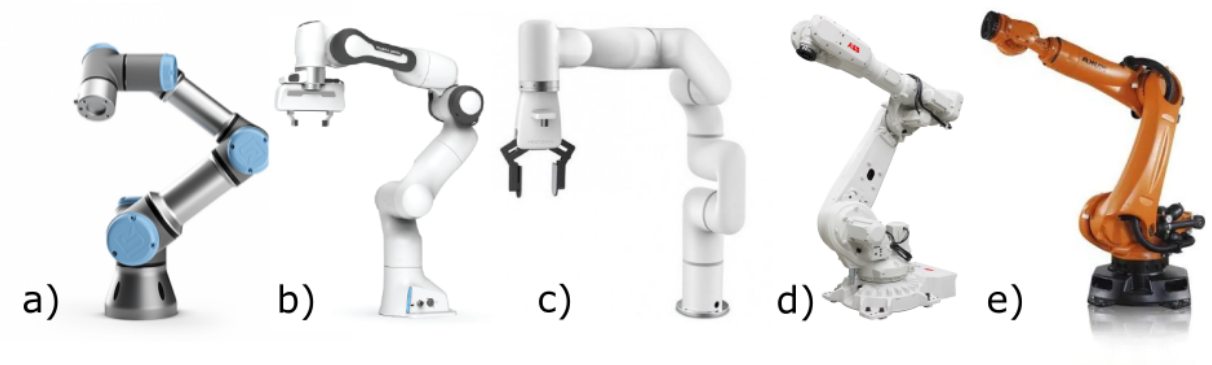

Fig. 6. *Manipulator robots that have ROS support a) Universal Robots, b) Franka Emika Panda Arm, c) UFactory xArm, d) ABB Robots, e) KUKA Robots [27], [34], [36]–[38]*

2.3.2 Learning ROS

Despite all the advantages of ROS, mastering it without a hands-on learning approach can turn out to be really hard. While official tutorials and books [39] offer valuable resources for attaining theoretical knowledge and practical skills with simulated robots, real-world experience with physical robots is essential for gaining proficiency in ROS. In this regard, ROS-based LFs have been shown to be a highly effective and dynamic approach to learning, offering trainees the opportunity to practice with actual robots in a controlled and safe environment [40].

The Tampere RoboLab (Fig. 7) is an example of the ROS-based learning environment at Tampere University, Finland [40], [41]. Even though the concept of a learning environment is somewhat different from an LF the example is still relevant since the overall purpose in both cases is to provide hands-on experience with robots. RoboLab setup includes the robots that have ROS support mentioned in Section 2.3.1. The resulting outcome of the learning environment showed that students who participated in the program were highly satisfied with their experience and that the program had a positive impact on their learning outcomes [41].

Fig. 7. *The Tampere RoboLab [41].*

3 OBJECTIVES AND REQUIREMENTS

This thesis is part of a project that aims to construct a learning factory that later can be used to teach students the automation of processes carried out by multiple robots. The LF includes mobile and manipulator robots that need to communicate to transfer a payload. The goal of this thesis is to develop a component for the LF which includes a manipulator robot.

3.1 Requirements for LF

- One or more mobile robots.
- One or more manipulator robots.
- Automated transfer of payload.
- Wireless communication between robots.

3.2 System requirements for LF

3.2.1 Software

- ROS Noetic
- Ubuntu 20.04 (Focal)

3.2.2 Soft requirements for hardware

- Manipulator robot with ROS MoveIt support
  - UFactory xArm
  - Franka Emika Panda Arm
  - Universal Robots
- Mobile robot with ROS support
  - Robotont
  - TurtleBot
- Camera with ROS driver
  - Intel RealSense

4 DESIGN

The development of the kitting station of the LF includes creating software for the autonomous robotic arm and connecting it to the main network that enables communication with other components of the LF. Kitting station itself is a transporting unit that consists of a working area where the process of kitting takes place and shelves, bins, or compartments to hold different objects that need to be stored. In automated kitting stations, manipulator robots are used to pick an object and store it in the preferred place. In this project, ROS was used to build software for the UFactory xArm 7 manipulator robot that will be storing books on the shelf. The interoperability of ROS makes it a perfect tool for developing an LF component that will be compatible with other components regardless of the hardware or software they use, as long as they support ROS.

Sections 4.1, 4.2, and 4.3 discuss the process of creating the manipulator robot aspect of the LF and the outcomes that were obtained. Namely, Section 4.1 describes the use case of the developed LF, Section 4.2 describes the network configuration and tools used to allow communication between mobile and manipulator robots, and Section 4.3 describes the approach used to program an autonomous manipulator robot.

4.1 The use case for the LF

In order to fulfill the requirements outlined in Chapter 3 the system depicted in Figure 8 was constructed. One of the scenarios where autonomous multi-robot systems could be beneficial is the library. To demonstrate the potential use of a developed LF the robotic pipeline was assembled where a mobile robot carries the box with the books from its initial position to the kitting station where the manipulator robot stores the books on the shelf.

Since LF requires to be changeable and multi-stage (Section 3.1) two possible setups were created: single-arm setup and dual-arm setup. In both cases, the system is required to achieve the stated goal - transfer the books from the mobile robot to the shelf. While in a single-arm setup, all work is done by one manipulator robot, a dual-arm setup includes an additional robotic arm, and the task is distributed between two robots. When the kitting station receives a message that the books are delivered one arm picks the book, and passes it to the second arm which then places it on the shelf. This process repeats until no books are left in the box and finally the message that signifies the finished work is sent back to the mobile robot.

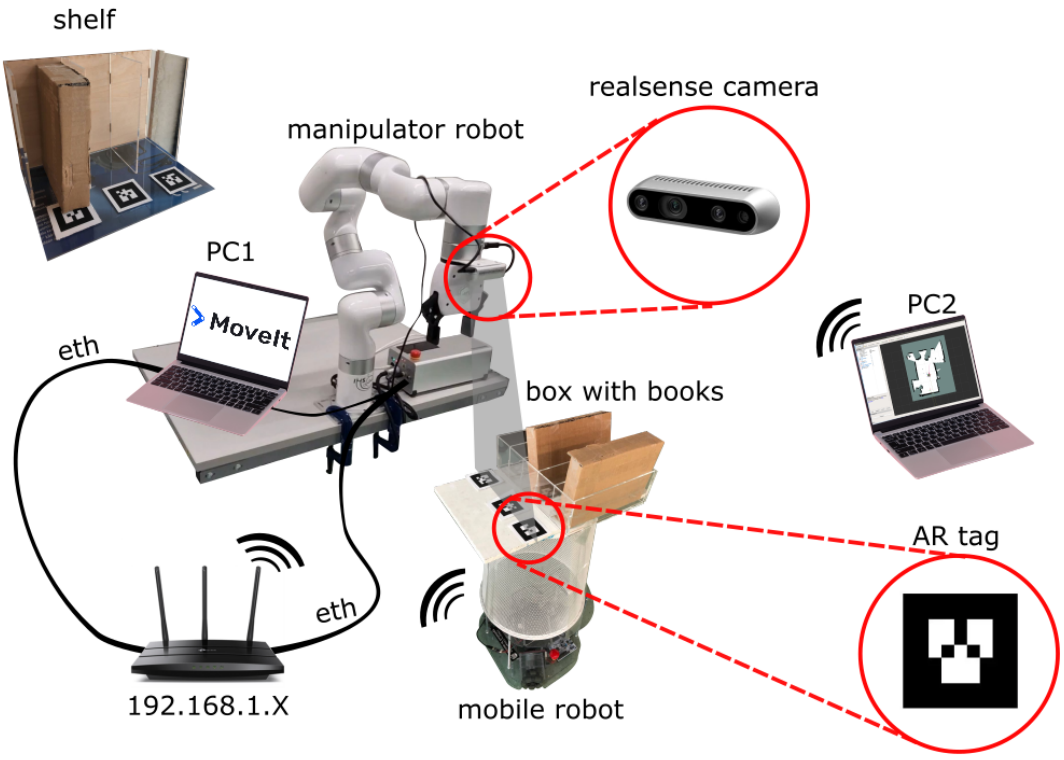

Fig. 8. *Full setup includes a manipulator robot that is connected to the router using an ethernet cable and controlled by one computer (PC1); a mobile robot that is wirelessly connected to the router and is controlled by a second computer (PC2).*

4.2 Network configuration and wireless communication between robots

In a multi-robot system one of the key requirements that need to be taken into account is communication between different components of the system. The importance of reliable communication cannot be overstated as it plays a critical role in the overall performance and success of the system. Specifically, lower latency enables robots to exchange information and coordinate their actions more effectively. Connecting robots to the same WiFi network is the easiest way to achieve a stable connection between multiple components of the LF. Functionally speaking, creating automated solutions is more straightforward when IP addresses are fixed, rather than dynamically assigned. In this project the IP addresses for every piece of hardware were reserved in the router's DHCP server so that it will always assign the same IP address to the same device (Fig. 9).

	ID	MAC Address	Reserved IP Address	Description	Status	Modify
☐	1	14-4F-8A-B2-B2-90	192.168.1.180	robotont1		
☐	2	18-1D-EA-57-EE-73	192.168.1.111	robotont2		
☐	3	34-60-F9-E7-C2-BB	192.168.1.112	PC1		
☐	4	38-22-E2-19-84-85	192.168.1.110	PC2		

Fig. 9. *Table of reserved IPs in the router's DHCP server. The reserved IPs include PC1 which controls the kitting station, PC2 which controls mobile robots, and two mobile robots - robotont1, and robotont2.*

In the case of xArm 7, there are three ways of network settings for the robotic arm (Fig. 10):

1. The control box is directly connected to the PC (Fig 10(a)).
2. The control box, PC, and router are connected by Ethernet cable (Fig 10(b)).
3. The PC and router are connected by wireless network, and the control box and router are connected by Ethernet cable (Fig 10(c)).

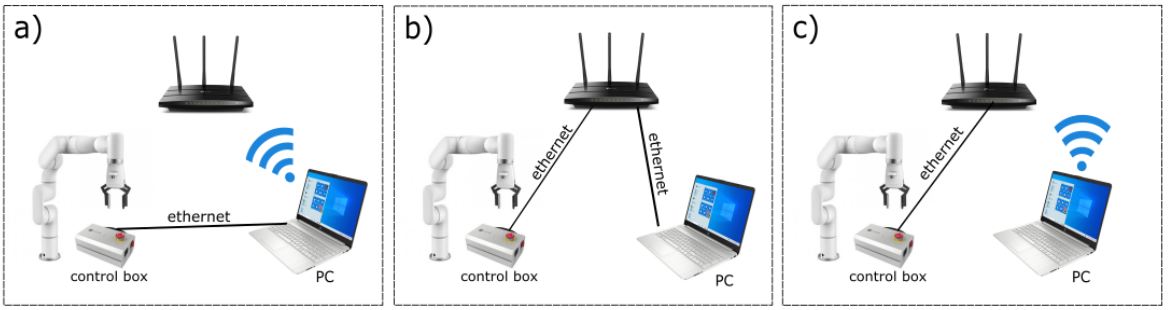

Fig. 10. *The Robotic Arm Network Setting options.*

The first option is optimal only for working with one manipulator robot since it requires a direct ethernet connection to PC. On the other hand, the third method seemed like a suitable choice since it allows the connection of multiple manipulator robots to one PC that is not restricted in its location by an ethernet cable. However, this way has a major drawback due to the delay and packet loss of wireless connection and the communication latency may have a bad impact on trajectory execution, unlike the second option which is chosen for further work with xArm 7.

In ROS there is a central point for managing the communication between all of the nodes which is called ROS Master. When working with multiple robots, there are two options for the design of the system:

1. Start one ROS Master and use it for all components of the system.
2. For every component of the system create its own ROS Master.

Both options have their own advantages and disadvantages. For instance, a single master does not require the installation of an additional package (*FKIE Multimaster*), however, it makes the system more vulnerable by having one major point of failure. On the other hand, a multimaster system has a decentralized architecture in which components can function together as a whole or separately if needed. For this project, the second option was chosen. With the use of the ROS package *FKIE Multimaster*, it is possible to connect multiple ROS Masters. This package allows for the synchronization of all nodes, topics, and services of different robots. In order to make PCs that control robots recognisable for one another, */etc/hosts* need to be modified by adding an IP address and a host name of the PC with which it wants to synchronize.

ROS provides four types of communication between nodes: Publisher/Subscriber, Service/Client, Action Server/Action Client, and Parameter Server. Even though, technically, all of those patterns are suitable for the current setup, the Service/Client type of information exchange was selected due to its simplicity in terms of one-to-one communication.

4.3 Creating software for autonomous manipulator robots

One of the reasons why ROS is so convenient is that it decomposes the complex software components into smaller and more manageable pieces called packages. This section will discuss the composition of a package created for the kitting station of LF, used open-source packages, and the implementation of the described use case with one and two manipulator robots in the setup.

4.3.1 Overview of utilized open-source ROS packages and tools

MoveIt

Manipulator robots are designed to perform tasks involving grasping, moving, and manipulating objects. Programming such robots to perform manipulation tasks can be a complex and time-consuming process. However, the development of an open-source software framework MoveIt [42] simplified this process by providing a set of libraries and tools that

could be used with a wide range of robots and sensors. It assists with complex robot manipulation tasks, such as object grasping, path planning, collision avoidance, and control. MoveIt Setup Assistant is a tool that helps to generate the necessary configuration files for MoveIt to work with a specific robot. It takes in the URDF model that describes the robot's physical structure, joints, and links and outputs the MoveIt configuration package with the specified robot's kinematics and defined planning groups. The obtained configuration package requires slight modifications before it will be ready to use. Usually, the robots that support ROS already have an open-source set of packages that provide everything that is needed to control a real or simulated robot including controllers, planners, visualization, robot description, and MoveIt configuration package. In the case of the manipulator robot that was used in this project (UFactory xArm 7) this set of packages is called *xarm_ros* [31].

Intel RealSense ROS1 Wrapper and librealsense2

In order to supply an autonomous robot with awareness of its surroundings the Intel RealSense camera was mounted on the robot's end effector. The communication with this camera is established with the help of the *realsense2_camera* ROS package from the Intel RealSense ROS1 Wrapper which provides the nodes for publishing camera data using *librealsense2* SDK.

AR Track Alvar

The only information about the environment that the manipulator robot in the developed kitting station needs to receive from the camera is the location of the box with the books on top of the mobile robot and the location of the shelf where those books will be transferred. To extract this information from the camera feed a lot of image processing and calculations need to be done. However, to simplify this process, ROS package *ar_track_alvar* was used. This package can identify and track the location of specific AR tags with the camera. By attaching those tags to the objects of interest (box and shelf) it is now possible to obtain their coordinates.

TF

To let the robot know where the camera is attached the coordinate frame of the camera needs to be linked to the coordinate frame of the robot. For that purpose, the *static_transform_publisher* from the *tf* ROS package that maintains the relationship between coordinate frames was used. The transformation is also required for the coordinates of the AR

tags obtained by *ar_track_alvar* since they are calculated relative to the camera coordinate frame and need to be transformed in relation to the robot coordinate frame.

RViz

RViz is a 3D visualization tool that allows the display of sensor data, robot models, and other types of data in a 3D environment.

FKIE Multimaster

To establish communication between multiple ROS masters the metapackage *multimaster_fkie* is used. It allows for the synchronization of all of the nodes, topics, services, and parameters across different robots connected to the same WiFi network.

4.3.2 ROS package for kitting station

To combine together all of the mentioned ROS packages and tools the ROS package for the kitting station of LF was created (Fig. 11). The package contains one main launch file which is called *director* that includes all other essential launch files:

- *xarm7* - launches everything that is needed to control xArm 7 including MoveIt and RViz.
- *vision* - starts the camera driver and AR tracking.
- *transform* - creates the transforms between robot, camera, and AR tags.
- *programs* - runs all the nodes that are in charge of the robot's actions.
- *multimaster* - allows for communication between multiple robots.

Besides launch files, the package contains a source folder where four nodes for the library scenario are located, a server for the message exchange between mobile and manipulator robots, and a YAML file for the storage of shelf location.

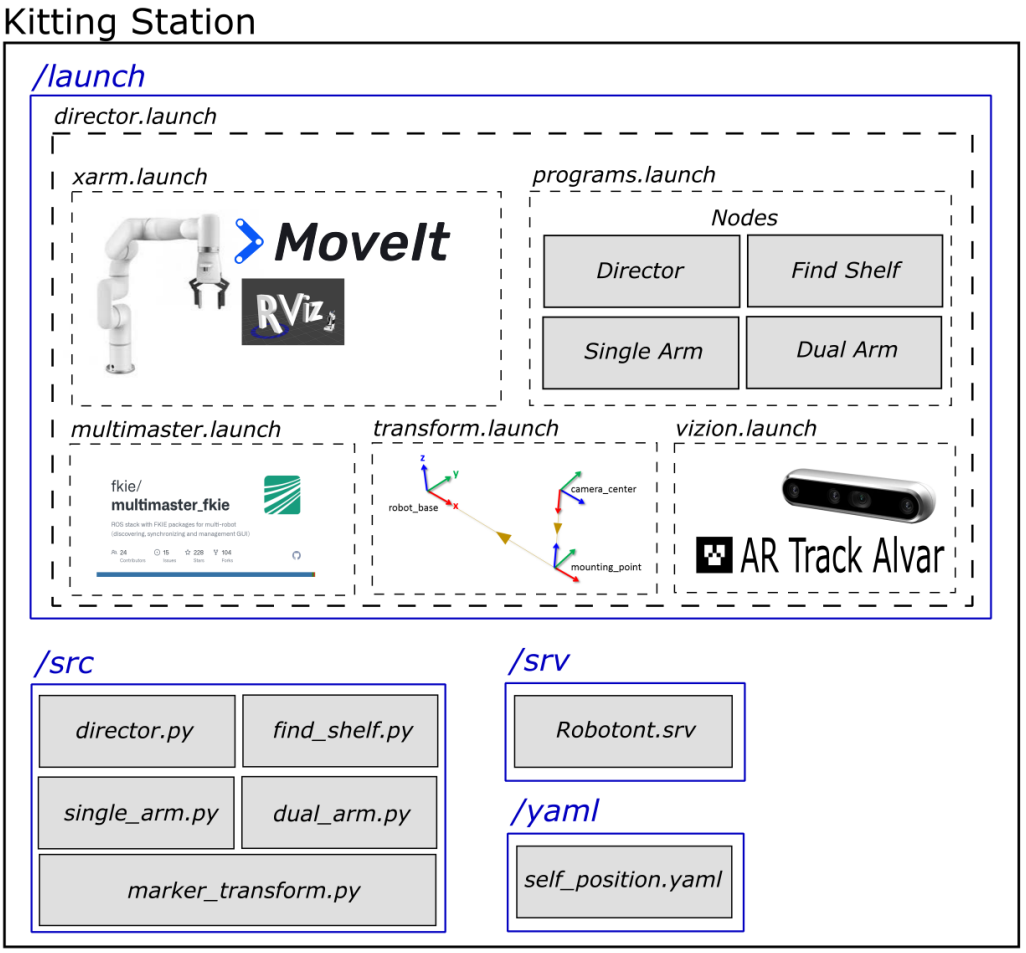

Fig. 11. *ROS package for the manipulator robot component of LF.*

4.3.3 Implementation of Single and Dual Arm Setups for the library use case

As mentioned in Section 4.3.1, the MoveIt configuration package is crucial for easy and safe communication with the manipulator robot. The *xarm_ros* repository provides a MoveIt configuration package for a single xArm 7 with gripper which is used for a Single Arm setup, however, there is no configuration package for dual xArm 7 with gripper. Hence, to create a Dual Arm setup a new configuration package for dual xArm 7 was generated with MoveIt Setup Assistant. Nevertheless, the generated package will not work right away. In order to make it fully functional the package was modified by taking the MoveIt configuration package for dual xArm 6 and xArm 7 with gripper as a reference. The exact changes that were made can be seen in the GitHub repository *dual_xarm7_moveit_config* (APPENDIX 1). In the current MoveIt configuration, two arms are located approximately 1 meter apart from each other and for simplicity are called Left Arm and Right Arm.

In *director.launch* it is possible to define which setup is currently used (with one manipulator robot or with two) by setting the argument '*dual_arm'* to '*true'* or to '*false'*. Depending on this argument the Single Arm or Dual Arm node will be launched together with Director and Find Shelf node in *programs.launch*. This argument also influences what MoveIt configuration package will be used and how many camera nodes should be launched (Table. 1).

	dual_arm = false	dual_arm = true
setup	Single Arm	Dual Arm
xarm.launch	xarm7_gripper_moveit_config	dual_xarm7_moveit_config
programs.launch	director.py find_shelf.py single_arm.py	director.py find_shelf.py dual_arm.py
vizion.launch	launched for one robot	launched for two robots
transform.launch	launched for one robot	launched for two robots

Table 1. *The difference in launched files when 'dual_arm' argument is set to 'true' or to 'false'.*

The Director is a central node that coordinates what program is going to run based on the input where the user states whether they want to run Find Shelf or Single/Dual Arm program (Fig. 12). After receiving the input, the Director sends the request to the corresponding node using the Server/Client type of communication (Fig. 14).

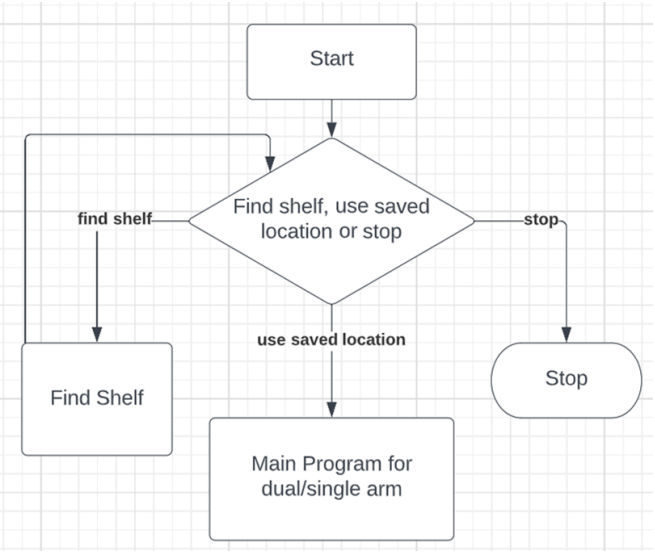

Fig. 12. *Logic diagram of Director node.*

When first setting up the environment or changing the location of the shelf, the Find Shelf program needs to be chosen to save new coordinates of the shelf to the YAML file (Fig. 14). After the request from the Director node is received, it starts scanning for the shelf which has specific AR tags attached (Fig. 13). When the shelf is found its coordinates are saved for later use.

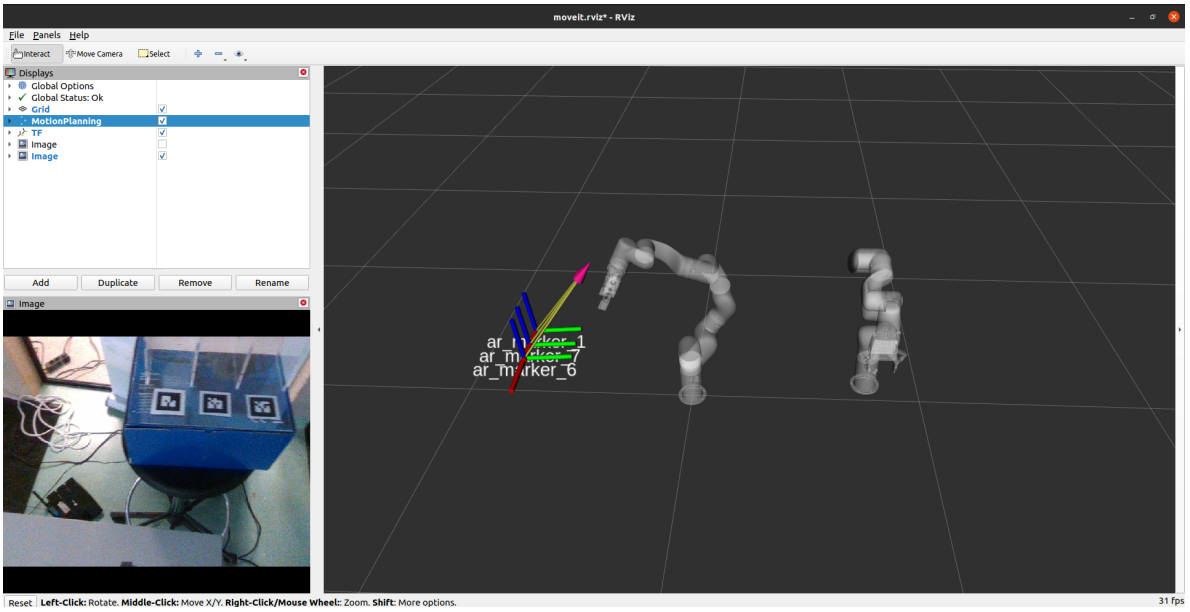

Fig. 13. *Visualization of two manipulator robots in RViz. The left robotic arm is looking for the AR tags attached to the shelf to obtain shelf coordinates and save them to the YAML file.*

When the location of the shelf is already known the Single or Dual Arm node can be initialized. In both cases, the initialization steps are the same:

1. Receive a request from the Director node.
2. Initialize MoveIt commander.
3. Read the shelf coordinates from the YAML file.
4. Move the arm(s) to the start position.
5. Wait for the message from the mobile robot to start transporting the books from the box on top of the mobile robot to the shelf.

The mobile robot is supposed to arrive at the predefined spot, however, small deviations from the expected location are not critical since the manipulator robot relies on the real-time data coming from the camera on top of the gripper that translates the coordinates of the AR tags attached to the box. When all of the books are placed on the shelf, the arm returns to the start position and the node sends the response to the mobile robot that signifies that it can move

back to the point of start (Fig. 14). In the case of a Dual Arm setup the intermediate action of passing the book from one robot to another takes place.

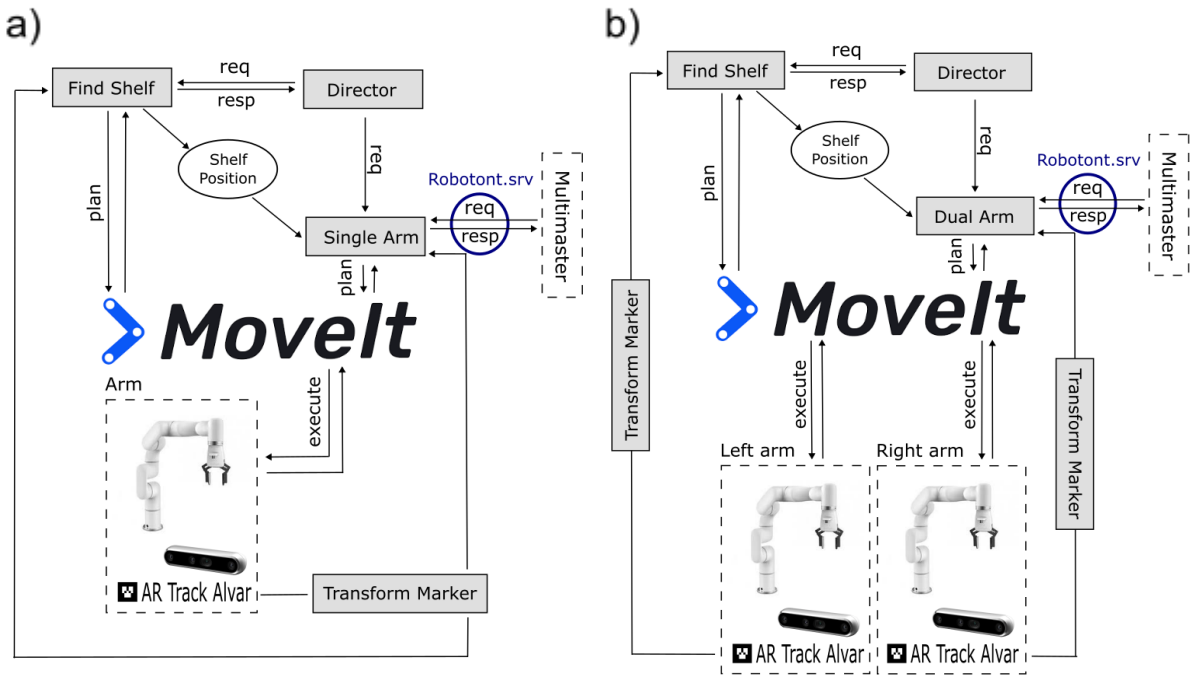

***Fig. 14**. Information flow diagram for a) Single Arm setup and b) Dual Arm setup.*

5 RESULTS AND DISCUSSION

This section summarizes the work that was done to construct the kitting station of the learning factory. Section 5.1 reveals the achieved results related to the work that was described in Chapter 4 and the overall performance of the manipulator robot component. Section 5.2 discusses the limitations of the assembled setup and Section 5.3 describes the future work to improve the achieved result.

5.1 Results

The constructed component of the learning factory fully satisfies the listed requirements (Section 3). As expected this project provides a learning environment where students can get experience with ROS, manipulator robots, and MoveIt, they can learn how to perform basic operations like picking and placing objects, coordinate the movements of robots in the shared space and use sensor devices to supply the robot with the awareness of its surroundings (Fig. 15).

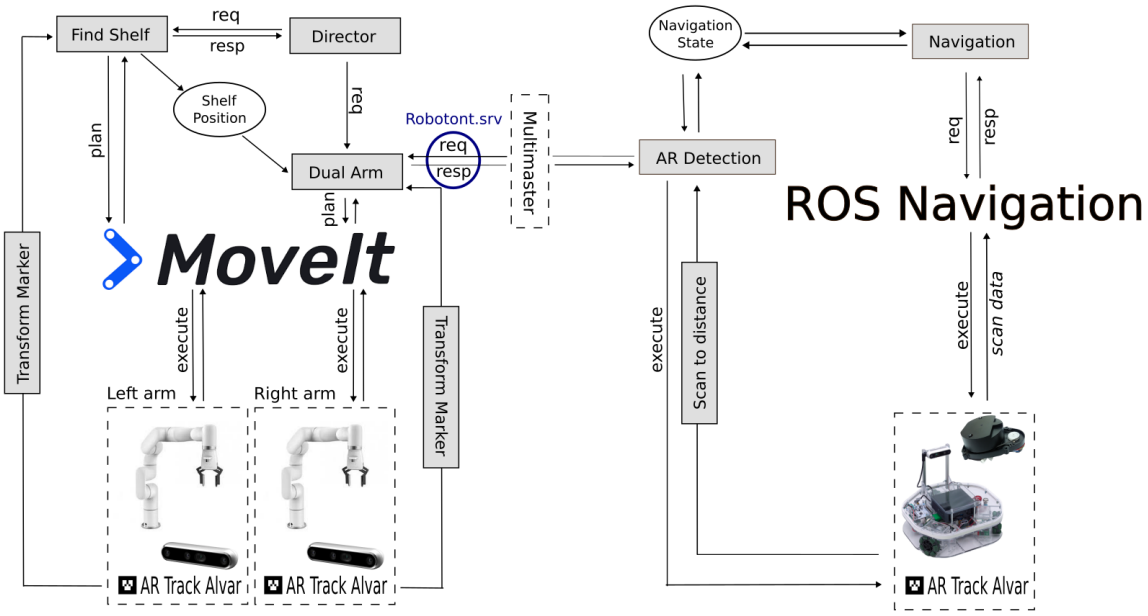

Fig. 15. *Information flow diagram for a full learning factory setup. This schematic includes kitting station with two manipulator robots and a mobile robot component that performs payload transportation.*

5.1.1 Mobile and manipulator robots communication

The two-sided wireless communication between mobile and manipulator robots was established by connecting them to the shared WiFi network. With the use of a *multimaster_fkie* ROS package, robots are able to exchange messages by synchronizing topics/services/parameters. The presence of a separate ROS Master for each component of LF makes the system more fail-proof by removing a single point of failure and distributing the centers of management across multiple machines.

5.1.2 Manipulator robot performance

Currently a robot can do 3 main tasks:

- Get the location of the objects by detecting AR tags.
- Perform pick and place action.
- Coordinate motion with collision avoidance when working with multiple manipulator robots

When setting up the learning factory there is one central node that directs what action the robot has to perform right now and three nodes that are in charge of different tasks that the robotic arm can do.

Director: The main program that requests feedback from the user about what program they want to run, then sends the request to the corresponding node using the Server/Client type of communication. By this time there are three possible options to choose from: Find Shelf, Single Arm, or Dual Arm.

Find Shelf: This program is used when setting up the environment. After the request to move from the director node is received, it starts scanning for the shelf which has specific AR tags attached. When the shelf is found its coordinates are saved to the YAML file for later use.

Single Arm: This node after receiving a request from the director node initializes the MoveIt commander, reads the shelf coordinates from the YAML file, moves the arm to the start position, and waits for the message from the mobile robot to start transporting the books from the box on top of the mobile robot to the shelf (Fig. 16).

Dual Arm: This program is made for the setup with two robotic arms and starts similarly to the Single Arm program. This kind of setup provides an opportunity to practice collision avoidance and motion coordination of two manipulator robots in one system. Continuing with the library scenario, the first arm is now picking the book from the box and passing it to the second arm which then stores it on the shelf (Fig. 17).

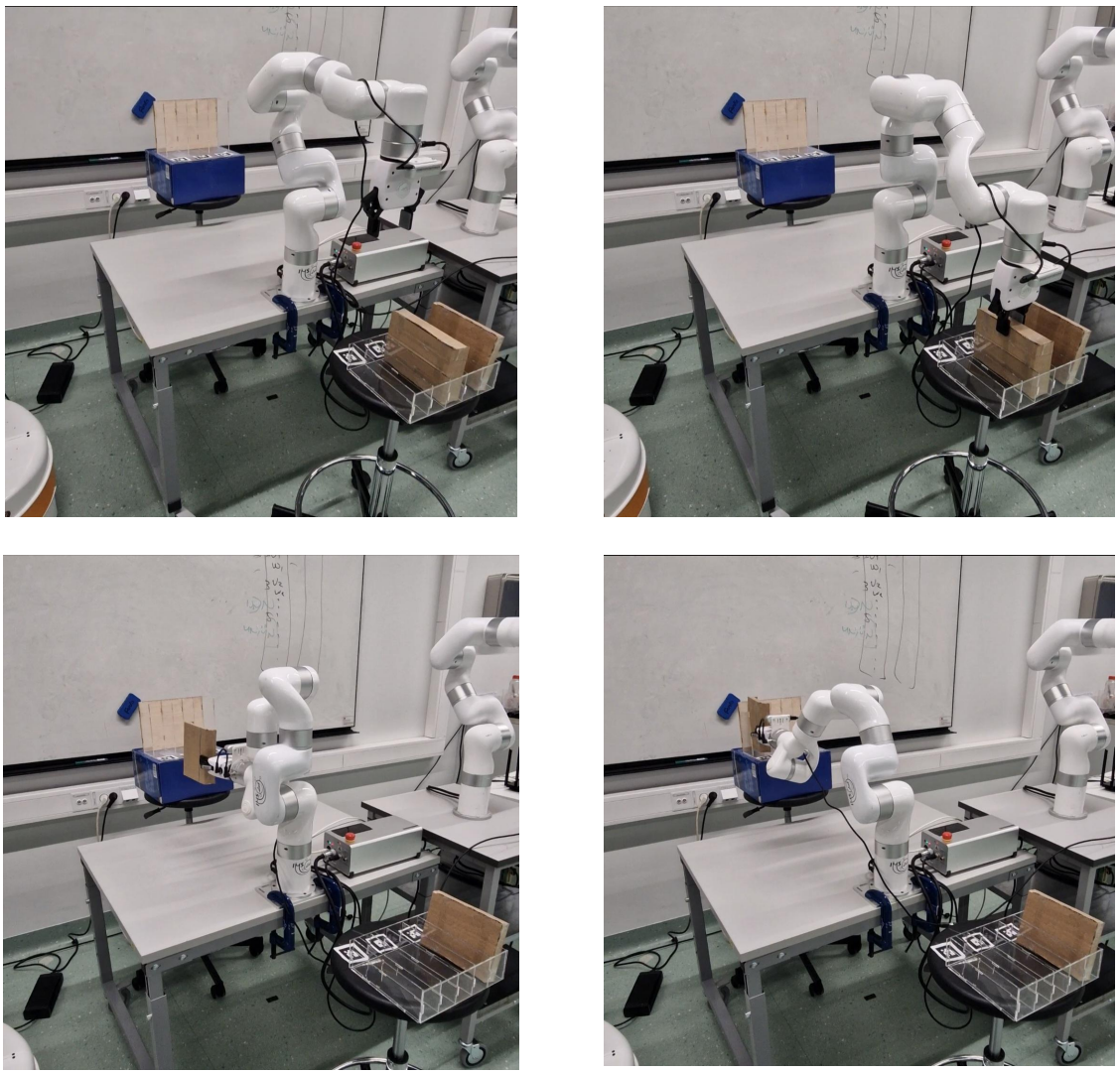

Fig. 16. *Single Arm program. After getting a request from the mobile robot xArm 7 picks up the book and places it on the shelf.*

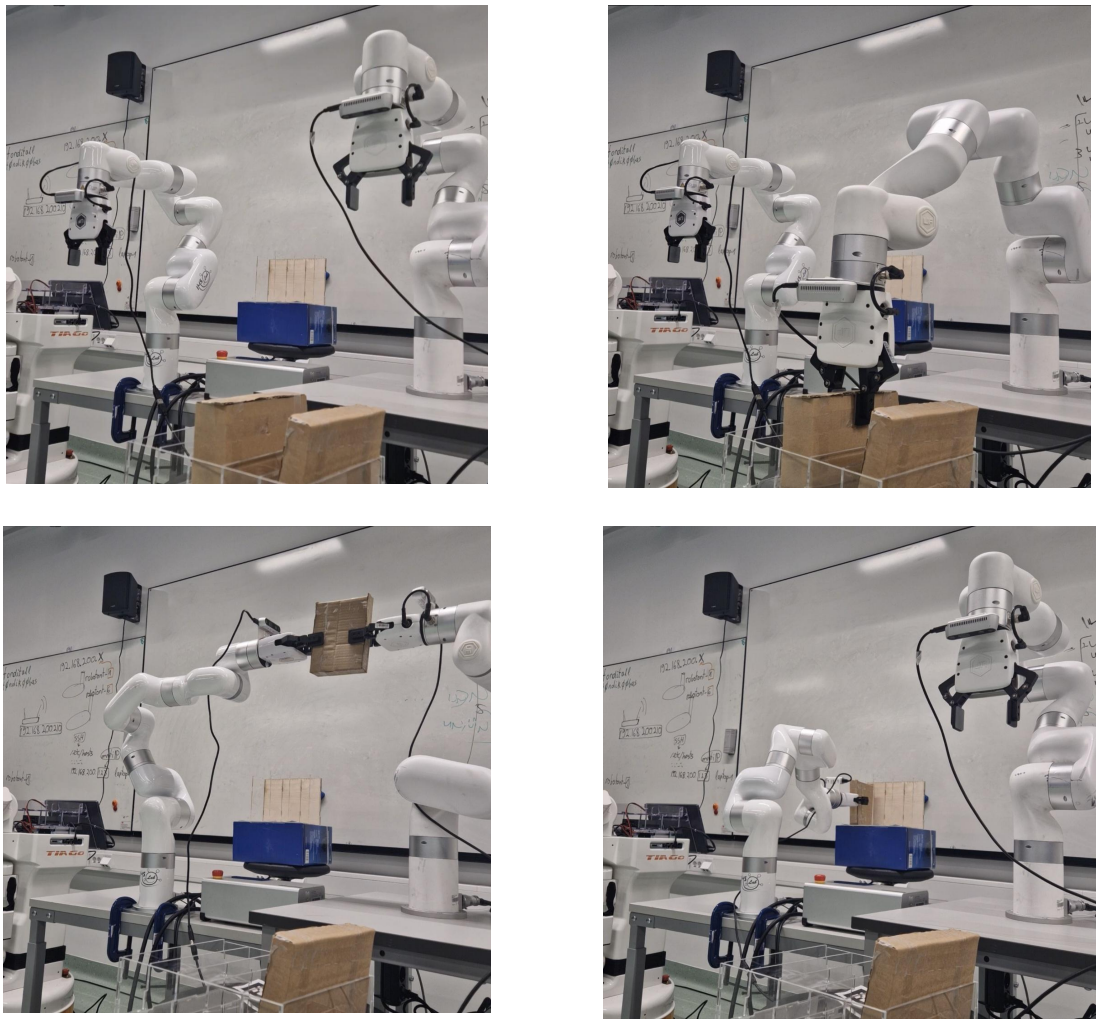

Fig. 17. *Dual Arm program. After getting a request from the mobile robot, right xArm 7 picks up the book and passes it to the left xArm 7, and returns to its initial position, left xArm 7 then places the book on the shelf.*

5.2 Limitations

Even though the kitting station of the learning factory is functioning properly according to the initially stated goal, there are still some limitations regarding the number of manipulator robots that can be added or how easy it would be to replace xArm 7 with different manipulator robots.

The changeability of setup measures how much effort it will require to add or subtract the component without breaking the whole system. This project provides software for setup with one manipulator robot and with two manipulator robots. However, integration of the third robotic arm will necessitate generating another MoveIt configuration package, while the

perfect outcome would be simply including another robot in the launch file under the group namespace. xarm_ros repository provides an example launch file that is supposed to overcome this limitation, but the error is occurring when it is launched. There is a possibility that the cause of this error is the transition to the Noetic distribution of ROS. Nevertheless, an investigation needs to be done to confirm that it is indeed the source of the problem. Testing this launch file with the previous versions of ROS may help to find out the root of the issue and the way to implement the launch of separate MoveIt configuration packages for two robotic arms.

Another characteristic of the changeable setup is hardware agnosticism. ROS makes it possible to create software for manipulator robots that will be compatible with different robotic arms. In theory, the current solution should also work with other robots like Panda Arm or Universal Robot, however, there are still some tests and improvements that need to be done before claiming this component is fully hardware agnostic. The robot replacement should be possible by simply connecting a different robot to the WiFi router with an ethernet cable. However, with the current code, the issue may arise due to the different numbers or organization of joints in different robots.

5.3 Future work

The future work will include solving the limitations described in Section 6.2 and making the system more user-friendly. Testing out the developed software with different manipulator robots will likely reveal some imperfections that should be resolved in the future for better functionality of the desired learning factory. Since this thesis is part of a larger project, which is expected to serve as an environment for future students, it would require to be more intuitive in terms of the code structure. Adding more comments and making documentation for the created ROS package will be the next step in achieving a more understandable structure for the new developers.

REFERENCES

[1] "Top 10 Applications of Robotics in 2020," *GeeksforGeeks*, Nov. 02, 2020. https://www.geeksforgeeks.org/top-10-applications-of-robotics-in-2020/ (accessed Apr. 11, 2023).

[2] "41 Robotics Companies & Startups to Know in 2023 | Built In." https://builtin.com/robotics/robotics-companies-roundup (accessed Apr. 20, 2023).

[3] E. Abele *et al.*, "Learning factories for future oriented research and education in manufacturing," *CIRP Ann.*, vol. 66, no. 2, pp. 803–826, Jan. 2017, doi: 10.1016/j.cirp.2017.05.005.

[4] D. Mavrikios, K. Georgoulias, and G. Chryssolouris, "The Teaching Factory Paradigm: Developments and Outlook," *Procedia Manuf.*, vol. 23, pp. 1–6, Jan. 2018, doi: 10.1016/j.promfg.2018.04.029.

[5] "Automating industrial tasks through mechatronic systems – a review of robotics in industrial perspective," *Teh. Vjesn. - Tech. Gaz.*, vol. 23, no. 3, Jun. 2016, doi: 10.17559/TV-20140724220401.

[6] T. Niemueller, "RoboCup Logistics League." https://robocup.rwth-aachen.de/rcll/ (accessed Apr. 10, 2023).

[7] "LEARN AND EXPERIENCE VPS IN THE BMW LEARNING FACTORY. - PDF Free Download." https://docplayer.net/31133564-Learn-and-experience-vps-in-the-bmw-learning-factory.html (accessed Apr. 10, 2023).

[8] S. Simons, P. Abé, and S. Neser, "Learning in the AutFab – The Fully Automated Industrie 4.0 Learning Factory of the University of Applied Sciences Darmstadt," *Procedia Manuf.*, vol. 9, pp. 81–88, Jan. 2017, doi: 10.1016/j.promfg.2017.04.023.

[9] R. Pittschellis, "Multimedia Support for Learning Factories," *Procedia CIRP*, vol. 32, pp. 36–40, Jan. 2015, doi: 10.1016/j.procir.2015.06.001.

[10] "Value Stream Analysis and Design," *Leonardo Group GmbH*. https://www.leonardo-group.com/en/value-stream-analysis-and-design/ (accessed Apr. 10, 2023).

[11] "Just-in-Time (JIT): Definition, Example, and Pros & Cons," *Investopedia*. https://www.investopedia.com/terms/j/jit.asp (accessed Apr. 10, 2023).

[12] "What Is Line Balancing & How To Achieve It | Tulip." https://tulip.co/glossary/what-is-line-balancing-how-to-achieve-it/ (accessed Apr. 10, 2023).

[13] G. A. Bekey, "On autonomous robots," *Knowl. Eng. Rev.*, vol. 13, no. 2, pp. 143–146, Jul. 1998, doi: 10.1017/S0269888998002033.

[14] "Autonomous Robots in Manufacturing Pros and Cons| Arrow.com | Arrow.com." https://www.arrow.com/en/research-and-events/articles/autonomous-robots-in-manufacturing-pros-and-cons (accessed Apr. 10, 2023).

[15] "Industry 1.0 to 4.0 - Brief History of the Industrial Revolution - Sustainability Success," Jan. 12, 2022. https://sustainability-success.com/industry-1-0-to-4-0-2-3-revolution/ (accessed Apr. 10, 2023).

[16] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0 – A Glimpse," *Procedia Manuf.*, vol. 20, pp. 233–238, Jan. 2018, doi: 10.1016/j.promfg.2018.02.034.

[17] "Industry 5.0 – the essence and reasons why it gets more attention," *i-SCOOP*. https://www.i-scoop.eu/industry-4-0/industry-5-0/ (accessed Apr. 13, 2023).

[18] "Customising the future – The next industrial revolution," Nov. 24, 2020. https://nickelinstitute.org/ (accessed Apr. 13, 2023).

[19] M. Luckcuck, M. Farrell, L. A. Dennis, C. Dixon, and M. Fisher, "Formal Specification

and Verification of Autonomous Robotic Systems: A Survey," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–41, Sep. 2020, doi: 10.1145/3342355.
[20] F. Sherwani, M. M. Asad, and B. S. K. K. Ibrahim, "Collaborative Robots and Industrial Revolution 4.0 (IR 4.0)," in *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, Mar. 2020, pp. 1–5. doi: 10.1109/ICETST49965.2020.9080724.
[21] C. Article, "Three Key Connectivity Requirements for Autonomous Delivery Robots," *Connector and Cable Assembly Supplier*, Nov. 30, 2021. https://connectorsupplier.com/three-key-connectivity-requirements-for-autonomous-delivery-robots/ (accessed Apr. 13, 2023).
[22] E. Aguado, R. Sanz, E. Aguado, and R. Sanz, *Using Ontologies in Autonomous Robots Engineering*. IntechOpen, 2021. doi: 10.5772/intechopen.97357.
[23] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System, 2nd Edition*. Packt Publishing Ltd, 2018.
[24] M. Appo, "Hardware-agnostic compliant control ROS package for collaborative industrial manipulators," Thesis, Tartu Ülikool, 2018. Accessed: Apr. 12, 2023. [Online]. Available: https://dspace.ut.ee/handle/10062/60718
[25] C. Reid, "Networked Heterogeneous Systems in a ROS-Enabled Cloud Environment".
[26] V. M. Vilches, "ROS Robotics Companies." Apr. 10, 2023. Accessed: Apr. 10, 2023. [Online]. Available: https://github.com/vmayoral/ros-robotics-companies
[27] "Collaborative robotic automation | Cobots from Universal Robots." https://www.universal-robots.com/ (accessed Apr. 10, 2023).
[28] "universal_robots - ROS Wiki." https://wiki.ros.org/universal_robots (accessed Apr. 10, 2023).
[29] "Franka Emika - Next Generation Robotics." https://www.franka.de/ (accessed May 16, 2023).
[30] "The difference between UFACTORY xArm5, UFACTORY xArm6 and UFACTORY xArm7 | UFACTORY Help Center." http://help.ufactory.cc/en/articles/4491842-the-difference-between-ufactory-xarm5-ufactory-xarm6-and-ufactory-xarm7 (accessed May 16, 2023).
[31] "xArm," *robots.ros.org*. https://robots.ros.org/xarm/ (accessed May 16, 2023).
[32] "industrial intelligence 4.0_beyond automation," *KUKA AG*. https://www.kuka.com/en-se (accessed Apr. 10, 2023).
[33] "kuka - ROS Wiki." http://wiki.ros.org/kuka (accessed Apr. 10, 2023).
[34] "Industrial Robot Supplier and Manufacturer | ABB Robotics." https://new.abb.com/products/robotics (accessed Apr. 10, 2023).
[35] "abb_driver - ROS Wiki." http://wiki.ros.org/abb_driver (accessed Apr. 10, 2023).
[36] "Industrial Robots - 3D Laser Cutting Robot Manufacturer from Pune," *https://www.neptunesystem.in/industrial-robots.html#3d-laser-cutting-robot*. https://www.neptunesystem.in/industrial-robots.html#3d-laser-cutting-robot (accessed May 16, 2023).
[37] "UFACTORY xArm 7 Robotic Arm: Buy or Lease at Top3DShop," *Digital Manufacturing Store Top 3D Shop*. https://top3dshop.com/product/ufactory-xarm-7-robotic-arm (accessed May 16, 2023).
[38] "Franka Emika - POMO Robotics," Oct. 16, 2021. https://www.pomorobotics.com/robots/frankpanda/ (accessed May 16, 2023).
[39] "How to learn ROS? 5 ROS learning methods | The Construct." https://www.theconstructsim.com/5-methods-learning-ros-one/ (accessed Apr. 10, 2023).
[40] M. Lanz, R. Pieters, and R. Ghabcheloo, "Learning environment for robotics education and industry-academia collaboration," *Procedia Manuf.*, vol. 31, pp. 79–84, Jan. 2019,

doi: 10.1016/j.promfg.2019.03.013.
[41] “RoboLabTampere | Tampere Universities,” *RoboLabTampere*. https://research.tuni.fi/robolabtampere/ (accessed Apr. 10, 2023).
[42] “MoveIt Motion Planning Framework.” https://moveit.ros.org/ (accessed May 21, 2023).

APPENDIX

1. Source code and video demonstration

https://github.com/patsyuk03/thesis-2023-Hurova.git

2. Non-exclusive licence to reproduce thesis and make thesis public

I, **Iryna Hurova**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis **Kitting station of the learning factory**, supervised by **Prof. Karl Kruusamäe and PhD Veiko Vunder**.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Iryna Hurova

24/05/2003