Saimaa University of Applied Sciences
Faculty of Technology, Lappeenranta
Mechanical Engineering and Production Technology

Ilia Dobriakov

Design and Development of a ROS Based LiDAR Platform to Scan Environment for a Mobile Robot's Autonomous Motion

Thesis 2019

Abstract

Ilia Dobriakov
Design and Development of a ROS Based LiDAR Platform to Scan Environment for a Mobile Robot's Autonomous Motion, 75 pages, 8 appendices
Saimaa University of Applied Sciences
Faculty of Technology, Lappeenranta
Mechanical Engineering and Production Technology
Thesis 2019
Instructor: Senior Lecturer Mr. Timo Eloranta, Saimaa University of Applied Sciences.

The objective of the research was to design and develop a LiDAR platform to scan the environment for a TIERA mobile robot's autonomous motion. The development of the system was done in Robot Operational System. The project was a part of a bigger project aimed to implement the autonomous charging system on the robot. The work was commissioned by the head of Laboratory of Intelligent Machines in the Lappeenranta University of Technology, Hamid Roozbahani.

The information for the research was gathered from literature sources of Lappeenranta Academic Library, the Internet and from Master's thesis reports of previous researchers, who were working on TIERA project. This research was carried out in a Laboratory of Intelligent Machines in the Lappeenranta University of Technology.

As a result of this thesis, the robot was completely set up for autonomous navigation utilizing ROS and LiDAR sensor. The 2D map for navigation inside the experimental field was built. In addition, the best concept design for a future charging station was selected. Further study is required to finalize the autonomous charging system implementation. Moreover, further tests of the implemented motion system in various conditions are needed.

Keywords: Robotics, ROS, Autonomous robot, Mobile robot, LiDAR, Navigation, Charging station.

Table of contents

Terminology

BOM	Bill of Materials
BVOA	Best Value Option Analysis
DOF	Degree of Freedom
EPOS	Easy Positioning System
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IIWA	Intelligent Industrial Work Assistant
I/O	Input/Output
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LAN	Local Area Network
LiDAR	Light Detection and Ranging
LUT	Lappeenranta University of Technology
MIR	Mobile Industrial Robots
OPRoS	Open Platform for Robotic Services
ORoCoS	Open Robot Control Software
OS	Operating System
ROS	Robot Operating System
SLAM	Simultaneous Localization And Mapping
UAS	University of Applied Sciences
USB	Universal Serial Bus
VM	Virtual Machine
WLAN	Local Wireless Network

1 Introduction

The idea of robotics was properly described by Nicola Testa:

…I conceived the idea of constructing an automaton which would mechanically represent me, and which would respond, as I do myself, but, of course, in a much more primitive manner, to external influences. Such automation evidently had to have motive power, organs for locomotion, directive organs, and one or more sensitive organs so adapted as to be excited by external stimuli.

The first prototype of the robot appeared as early as 350 B.C. in the ancient Greece. The mechanical "Pigeon", created by philosopher and mathematician Archytas, was able to fly 200 meters on its own jet. In the fifteenth century Leonardo da Vinci designed mechanisms, similar to human body parts, for transmitting motion. The first remote-controlled machine was represented by Nicola Tesla in 1898. The modern word “robot” was first used by novelist Karel Capek in 1921. In translation from the Czech word “robota”, it means tireless work. In the period of 1970s-2000s the robotics science made some significant steps with the invention of controllers, joystick, vision guidance, digital control loops, networking, graphical control interfaces, collision detection systems etc. (Pires 2007, pp. 1-8.)

The history of the development of robotics dates back to ancient times, but it was in the last decades that technology began to develop at a rapid pace. Modern robots combine a lot of advanced systems that give a possibility to find a variety of applications for them and implement them in all areas of our life, from industry to everyday life.

Every year robots are more widely integrated into work in industrial environments through all around the world, as they provide the most flexible existing automation technology. Modern robots provide more and more autonomy by requiring less human intervention in the execution of routine tasks (Pires 2007, p. 1). While at one time Nikola Tesla expected from robots a more primitive, than humans, interaction with the environment, then modern robots sometimes surpass humans in some areas. The robots provide the following benefits in comparison to a human: high repetition with high productivity, superb constant accuracy, immunity

to hazards, intense labour (Davids 2017). In addition, modern sensors and measurement devises allow robots to instantly measure the dynamic parameters of objects and the environment, such as temperature, force, speed, illumination, location, chemical composition and so on, with extreme precision.

It is logical to say, that behind every advanced technology there is a large multidisciplinary work of highly qualified professionals. Intelligent machines cover such areas of science as mechanics, hydraulics, pneumatics, automation, electronics, and information technology. For instance, motion problems cover kinematics, dynamics, control theory; localization and navigation – computer algorithms, artificial intelligence, information, and probability theory (Siegwart & Nourbakhsh 2004, pp. 8-9). Accordingly, robots' developers must be at least familiar with all of these areas.

As it is implied in the title the objective of this work is the autonomous charging system of the tele-operated mobile assembly robot TIERA. The project is carried out in the Laboratory of Intelligent Machines in the Lappeenranta University of Technology. This section describes the background, objectives and contribution of the thesis project.

1.1 Project background

Over the past 25 years, the research group of LUT Laboratory of Intelligent Machines has done a significant number of international projects related to mechatronic machine design and robotics. The projects are done in extensive collaboration with industrial companies, such as MeVEA Oy. Each project implies wide multidisciplinary research with the usage of advanced technologies, such as simulators, industrial robots and virtual reality. (LUT University.)

The laboratory already has a long tradition in creating new innovations for totally new areas of industry in the form of student assignments. The good example of the project is the motion platform based horse riding simulator, which provides genuine motion feedback. Another example of the large-scale project is a Mobile Assembly Robot called TIERA. (LUT University.)

Originally TIERA is a versatile, mobile robot that can sense, navigate, and monitor its surroundings, conduct repairs and assembly tasks in hazardous or dangerous environments where humans cannot operate. The development process of TIERA is done in a format of separate projects of several researchers. Most often, the researchers are working on their thesis projects, improving the robot and implementing new systems. In some cases, the projects are completely independent. However more often, the projects are the parts of bigger projects with more complex objectives. In these cases, the researchers must get familiar with the achievements of previous colleagues and continue the work, based on gained experience and knowledge. For instance, the present project is a part of a bigger project, aimed to give the robot autonomy in motions. Before the start, the researcher must learn the working principle of the network communication, developed by and Efim Poberezkin (2017), and the traction system, developed by Weiting Le (2015), Miguel Cordero Sanchez and Kirill Romanov (2018). The obtained results of the present thesis project will also be used by the next generations of researchers.

1.2 Project objective

The objective of this project is to develop a platform to scan the environment for a TIERA mobile robot's autonomous motion. The process of achieving the objective can be divided into several steps. The first step includes preparation of the robot for the project: disassembling of idle parts, such as head, arms and covers, and installation of the SICK LiDAR. The second step involves the recovery of the communication system between robot and control station, recovery of the current TIERA platform using ROS and testing of the robot’s traction system. The third step includes the study of previous researchers’ experience, getting familiar with ROS and LiDAR working principles. On the fourth step the LiDAR must be connected to ROS and 2D model of the laboratory must be made with the help of it. The fifth step involves the installation and implementation of ROS packages needed for autonomous movements of the robot.

The following outputs are expected as a result of the current thesis project:

- 2D map model of the laboratory made with the help of LiDAR and ROS

- integrated system for autonomous robot's movement from current location to a certain point on pre-defined map based on ROS and utilizing LiDAR sensor
- (additional contribution to TIERA project) conceptual design of the charging station.

It is worth noting that the traction system and network communication system of the robot were already developed by previous researchers and implementation of these systems must not be described in the current project. The only required task is to recover these systems before the start of the project.

It must be noted, that the obstacles avoidance function during autonomous movement is out the scope of the present project. The system must be developed assuming that no unexpected objects will not interrupt the robot's movement on its path.

The results must be aligned with a technical expert and the head of Laboratory of Intelligent Machines in the Lappeenranta University of Technology.

1.3 Contribution of thesis

The most significant part of the present Bachelor's thesis project is developing the system for an autonomous robot's movement. In the scope of the TIERA project, the robot will be prepared for the autonomous motion. Which means, that no operator's control will be needed to move the robot from point A to point B. This feature will broaden the horizons of TIERA application. TIERA robot will get a modified motion system, which can be integrated not only to work within the milling machine service project, but also to perform various autonomous works in the workshop, such as cleaning or transportation.

However, the concepts of the developed autonomous motion system can be used in any similar project, dealing with autonomous mobile robots. The reusing of the code packages without significant modifications is possible due to the Robot Operating System. The only restriction is the same or similar set of equipment.

In addition, as an added value, the contribution to the development of a charging system will be done: the conceptual design of the charging station will be selected.

These outputs of the present thesis can be used in further research in order to finalize the development of the autonomous charging platform. The complete system will increase the autonomy of the robot since it will be able to determine the demand in charging automatically and connect to the charging station itself.

2 Technologies and equipment

2.1 Overall description of TIERA mobile robot

Originally TIERA was designed as a versatile, mobile robot that can sense, navigate, monitor its surroundings, and conduct repairs and assembly tasks in hazardous or dangerous environments where humans cannot operate. Currently, the robot is modified to the second version, which is desired to work with milling lathes autonomously. The modification influences the technical equipment set, functionality and appearance of the robot. The general aim of the modification is to provide autonomous tasks execution utilizing sensing systems with position localization based on Light Detection and Ranging (LiDAR) system.

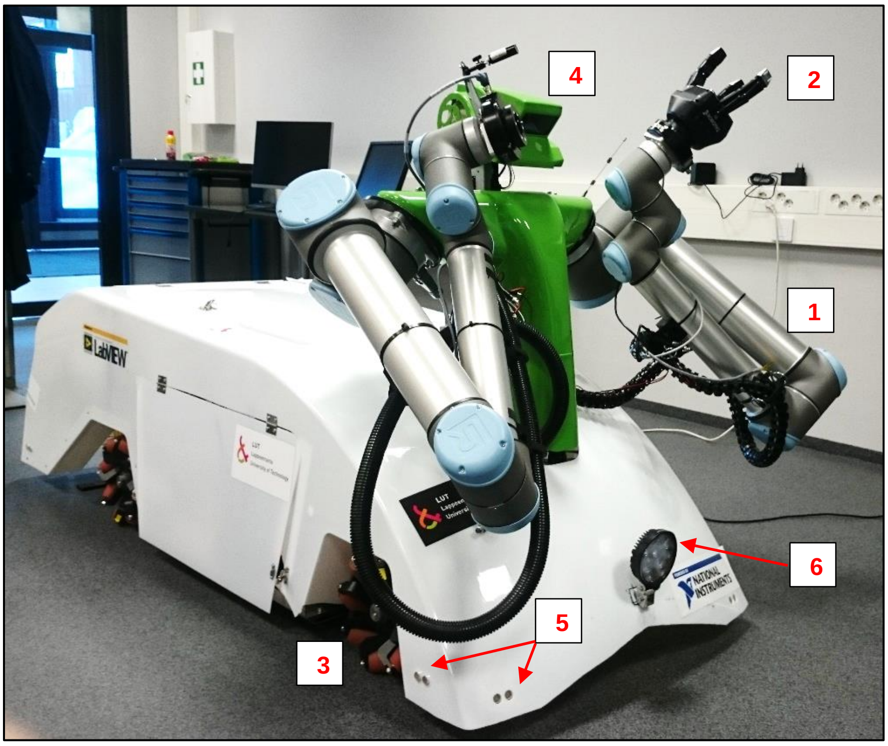


Figure 1. The first version of TIERA mobile robot: 1 – robotic arms, 2 – gripper, 3 – wheels, 4 – head with vision and hearing systems, 5 – sensors, 6 - lighting.

The original design of TIERA robot is presented in Figure 1. Basically, the robot can be divided into the following main parts:

- Control units provide the management of all implemented systems on the robot.
- Battery system supplies the required power for autonomous work of all systems, installed on the robot.
- Industrial robotic arms (number 1 in Figure 1) with gripper (number 2 in Figure 1) provide an ability to pick and move objects of various shapes.
- Traction system with Mecanum wheels (number 3 in Figure 1) makes the robot omnidirectional in movements.
- Head with an implemented vision and hearing system (number 4 in Figure 1) provides the operator with the ability to see and hear the environment around the robot in real time.
- Ultrasonic sensors (number 5 in Figure 1) help the robot to detect the obstacles in the environment and avoid them.

The second version of TIERA, which was the object of the present Bachelor's thesis research, is desired to have a little different composition. The robot will

have only one arm, which will be installed instead of the head. However, this design modification does not influence on the current project.

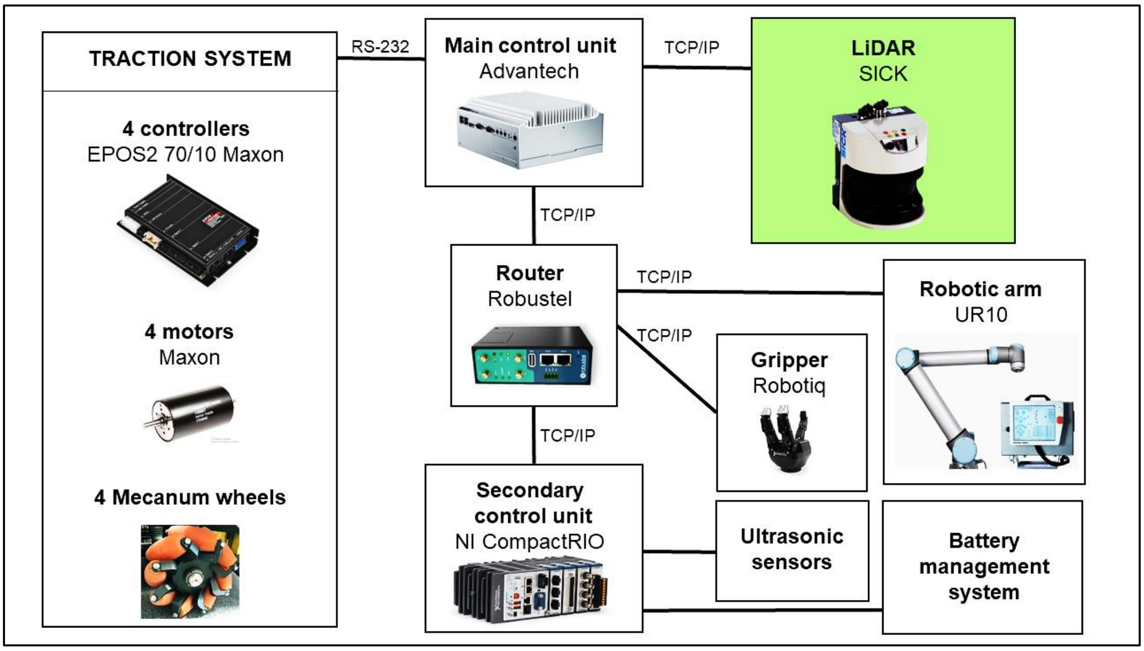


Figure 2. Hardware layout of the second version of TIERA

The simplified layout of the hardware structure of the second version of TIERA can be seen in Figure 2. As can be seen, there are two control units on TIERA: main and secondary. The main control unit deals with systems with high computational operations. While the secondary control unit processes input devices, such as sensors and battery management system. The control over the robotic arm and gripper is done by the operator's station, which sends and receives signals by wireless, with the help of Wi-Fi router installed in the robot. The traction system of the robot is represented by four Mecanum wheels that are powered by independent motors with independent controllers. It is worth noting that the LiDAR sensor was installed on the robot in frames of the present project. The relationship between the components of the robot of the TIERA robot was described in more detail by Belzunce (2015).

Control unit

Drawing an analogy with a human, one can say that a control unit is the brain of a robot. The control unit of the robot includes hardware and software. The control unit has the following roles: information, decision and communication. Information

role covers collecting and processing information from the robot's sensors. Decision role means planning the task's execution and motion of the robot. Communications role consists of organizing the information between the environment and the robot. (Reza 2007., p. 7)

TIERA's main control unit is represented by Andvantech ARK-3440F-U5A2E industrial computer. The main benefit of this computer is high reliability and stability even working under severe conditions (Advantech 2012). The computer supports both Windows and Linux operational systems. Linux OS Ubuntu 14.04 is utilized for TIERA project.

The second control unit is represented by a cRIO (CompactRIO) - an industrial controller manufactured by National Instruments. The device provides extensive I/O (Input/Output) capabilities. The controller supports up to eight I/O module slots, it allows to connect devices via USB, Ethernet, RS232/RS485, Mini Display, and some other types of ports. The following on-board hardware of TIERA robot is connected to cRIO: distance sensors, temperature sensors, lights, battery management system et cetera. The controller can be programmed in a special environment - LabVIEW (Laboratory Virtual Instrument Engineering Workbench) developed by National Instruments.

Robotic arms

The TIERA has got industrial robotic arm UR10, manufactured by Universal Robots. This industrial arm is classified as a collaborative robot because of the human-friendly design. According to it, the arm was made to work in human environments. The main safety measure is the protective stop, which happens when the robot collides with something strong enough. The arm is equipped with a 3-finger gripper, manufactured by Robotiq. Thanks to different grip modes, the device is able to pick the objects of various shapes. (Artigas 2017.)

Battery system

One of the most obvious constraints of autonomous mobile robots is a limited on-board power supply. The capacity of the battery restricts robots' long-term functionality. The rechargeable batteries are utilized to solve the problem. Typically, the batteries may only provide a few hours usage before recharging is necessary. Basic mobile robots require human intervention during the recharging. The robot must be turned offline and attached to a charging device manually. However, in cases when long-term autonomous capabilities are necessary, the more advanced charging systems must be used. In these cases human intervention is unacceptable and the recharging must be done in autonomous mode. (Silverman et al. 2002.)

For the TIERA robot to operate, high powered lithium cell manufactured by GWL power battery is used. Each battery has a nominal voltage of 3,2 V. In order to provide 48V for normal operation of all robot's systems, 16 battery cells are used. The battery cells are connected together into one system, which is controlled by a battery management system BMS123 Smart. The system's software allows one to monitor the most important data like total pack voltage, state of charge, and individual cell voltages (GLW Power 2015). In addition, the system can send the most important data like voltages, temperatures, settings, and status through a CAN bus.

Traction system

Basically, a traction system of the robot has the same function as a musculoskeletal system of human or animal. There is a large variety of locomotion mechanisms, with the help of which the robots can walk, jump, swim, slide, roll or even fly. Mobile robots need locomotion mechanisms that enable them to move boundless in their environment. That is why the design of a mobile robot's traction system must be selected based on a supposed operational environment. In addition, the selected parameters of the traction system must provide three fundamental characteristics: manoeuvrability, stability, and controllability. Manoeuvrability defines the ease of the robot's movement in any desired direction in minimal space. Stability is the characteristic of steadiness of the robot during the operation. It is

usually defined by the number of wheels and their location. Controllability is determined by the ease of converting motion commands to the physical movement of the robot. There is generally an inverse correlation between controllability and manoeuvrability, as the more freedom in movements requires more complicated motion control. (Siegwart & Nourbakhsh 2004, pp. 13-46.)

As TIERA is implied to be an omnidirectional mobile industrial robot, i.e. it should be able to move independently in longitudinal (forward/backwards), lateral (right/left) and rotational directions (Romanov 2018). The robot is equipped with four 254 mm Mecanum wheels which significantly improve its manoeuvrability. The Mecanum wheel is a design for a wheel which makes the vehicle being movable in any desired direction (Ilon 1972). This type of wheel was invented by Bengt Erland Ilon, who was an engineer in the Swedish company Mecanum AB. Due to this fact, the wheels can also be called Swedish. The main advantage of the robot equipped with four Mecanum wheels is that it can rotate and turn around its central axis. This allows the robot to move in very narrow or restricted spaces. The wheel drive components of TIERA robot are shown in Figure 3.

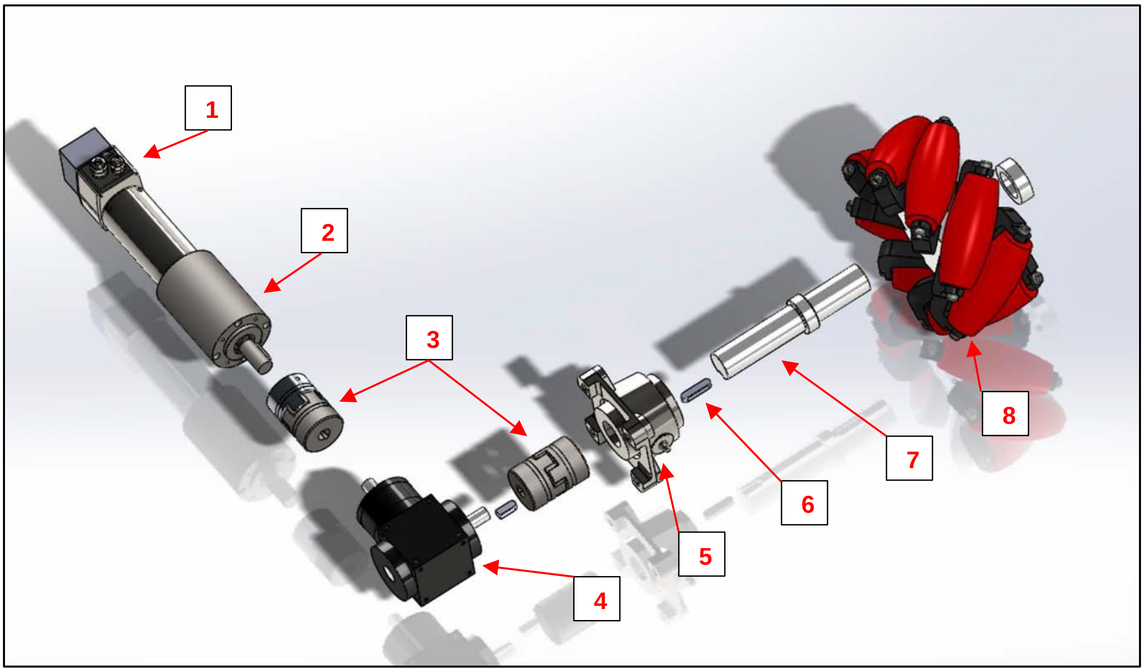


Figure 3. Wheel drive of the TIERA robot: 1 – motor, 2 – planetary gear, 3 – shaft coupling, 4 – L-transmission, 5 – tapered roller bearing, 6 – key, 7 – shaft, 8 – Mecanum wheel (Le 2015).

On TIERA the wheels are powered by Maxon motors which are equipped with magnetic safety brakes. Control of the wheels is performed using digital Easy Positioning System (EPOS) controllers. Each motor is connected with each EPOS controller. EPOS controllers are relatively easy to program with C language or C ++. All the motor drivers are connected and programmed on Ubuntu system installed on Advantech main control unit.

Network communication

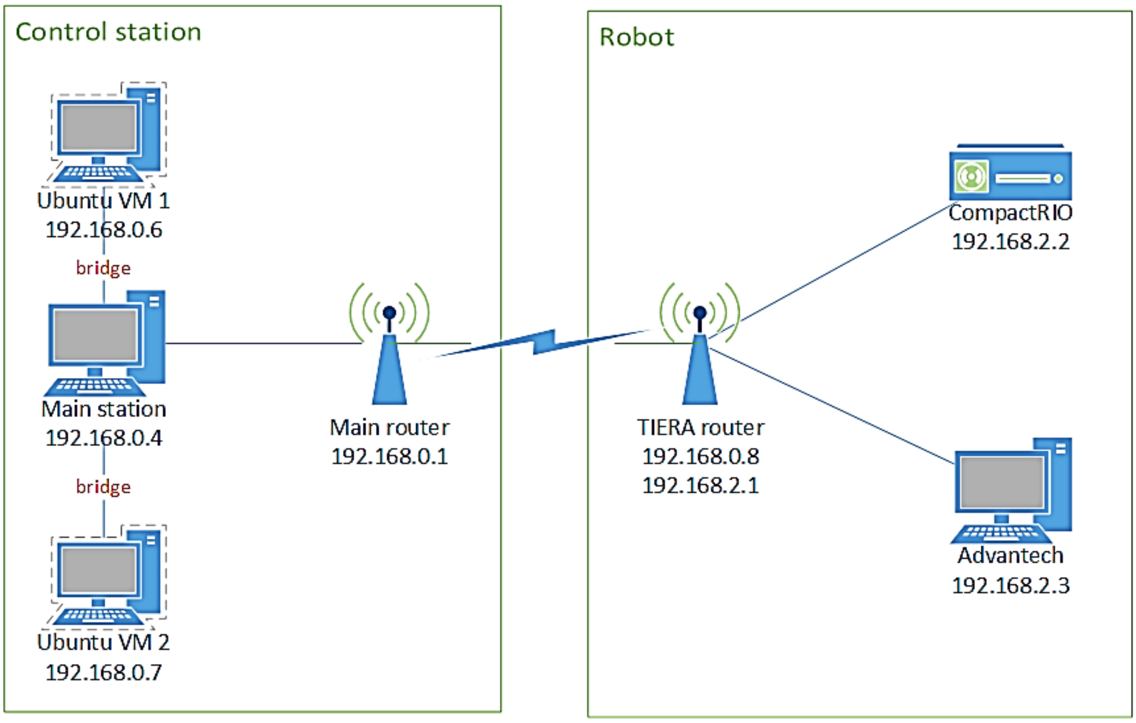


Figure 4. Simplified network topology of the TIERA robot

The wireless network communication system of TIERA mobile robot was developed and described by Efim Poberezkin (2017). According to the classification of wireless networks by the maximum range, the TIERA has a local wireless network (WLAN) with the range of several hundred meters. Every network operates according to a special protocol. The protocol regulates topology of the network, routing, addressing, the order of access of the network nodes to the data transmission channel, the format of the transmitted packets, the set of control commands for the nodes of the network, and the information protection system. The most common WLAN protocol, and the one that is utilized for the communication system of the robot under development is Wi-Fi. Wi-Fi is a trademark of the Wi-

Fi Alliance organization for wireless network protocols based on the IEEE (Institute of Electrical and Electronics Engineers) 802.11 standard, which was developed in 1997. (Labiod et al. 2007.)

The simplified network topology is shown in Figure 4. The general idea is that the network is divided into two main subnetworks: 192.168.0.0 for the operator's control station and 192.168.2.0 for the robot's embedded computer and some of its onboard devices (Poberezkin 2017). The communication between subnets is settled by means of two industrial Robustel routers, with an established Wi-Fi connection. This system enables the robot to be controlled from the operator's control station wireless. The scheme of connection between the components used in the present thesis project can be seen in Figure 5.

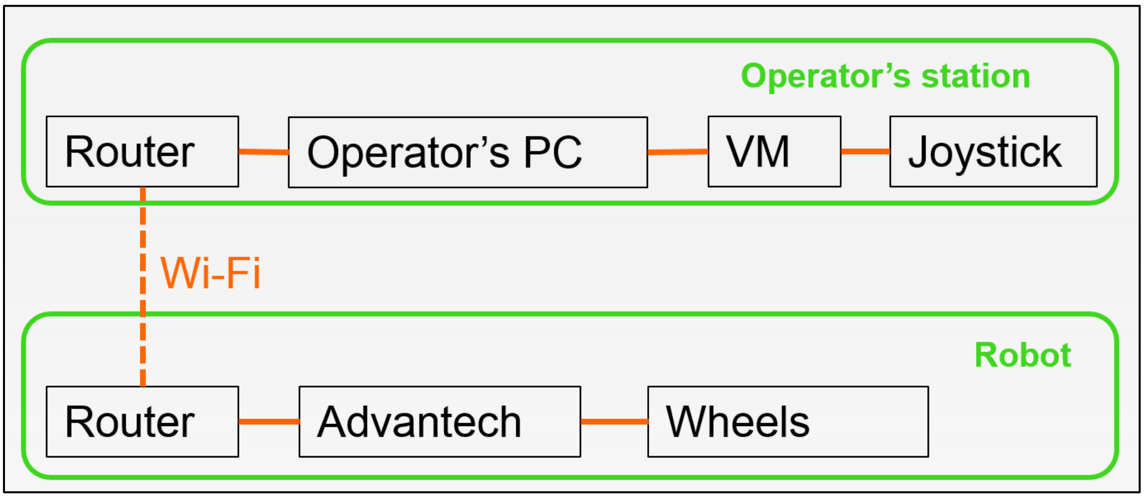


Figure 5. Simplified TIERA's network, related to traction system

As one can see in Figure 5, the control commands for the traction system are sent from a joystick installed on the operator's station. The joystick is connected to Virtual Machine with Ubuntu OS, which is installed on the operator's personal computer (or main station) with Windows OS. The operator's main station communicates with the robot via Industrial Cellular Routers Robustel. They transfer data between each other via Wi-Fi. The data from the router on the robot is processed by Advantech control unit, which in its order with the help of ROS generates the motion commands for wheels.

2.2 LiDAR

From the point of view of the components of the robot, the LiDAR is a sensor. In simple words, sensors allow machines to emulate the human ability to perceive. For instance, the ability to hear is substituted by acoustic sensors; ability to see – by optical and visual sensors; ability to feel the smell or taste – by chemical sensors; ability to recognize the touch – by tactile sensors. (Hesse 2001., p. 9)

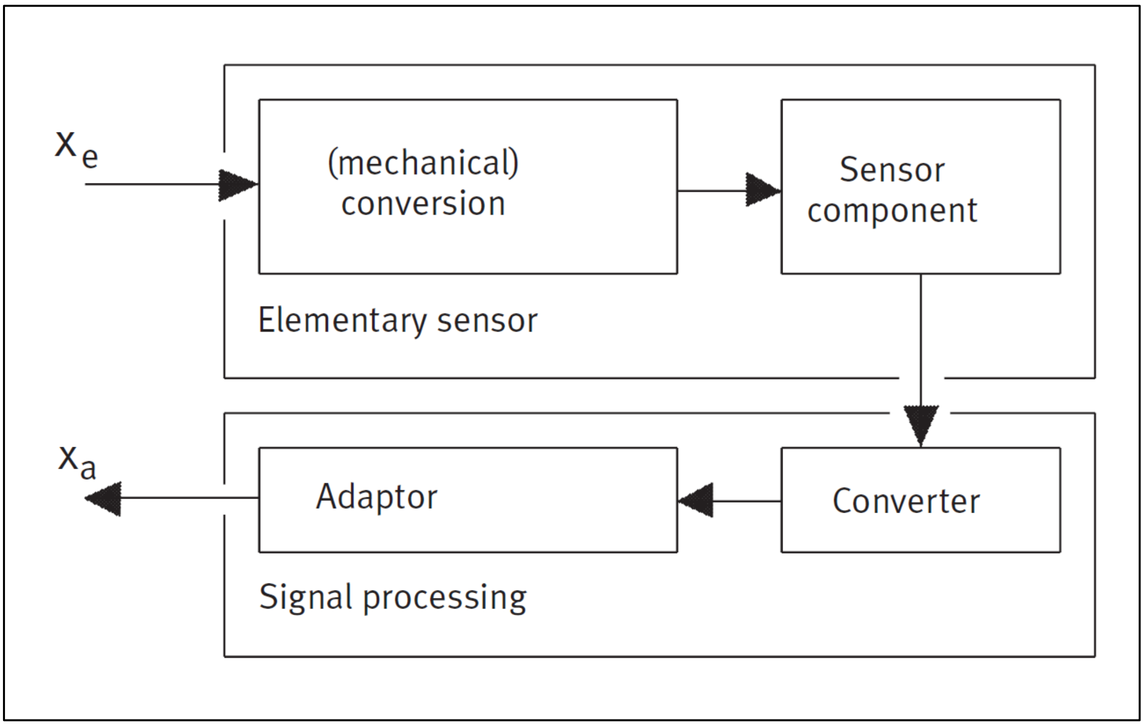


Figure 6. Sensor principle (Hesse 2011)

According to Hesse (2001), the sensors are detecting elements that detect time-variant physical or even electrochemical variables and convert them into a unique electrical signal. The sensor principle can be clearly seen in Figure 6. Basically, the sensor deals with the conversion of the measured non-electrical variable (x_e) into an electrical variable (x_a). Usually, some signal processing is also required before the information is forwarded to an executing system with actuators. (Hesse 2001, pp.11-12.)

Sensors are the elements used for detecting and collecting information about internal and environmental states. Sensors, integrated into the robot, send received information to the control unit, which in its turn determines the configuration of the robot. (Reza 2007.)

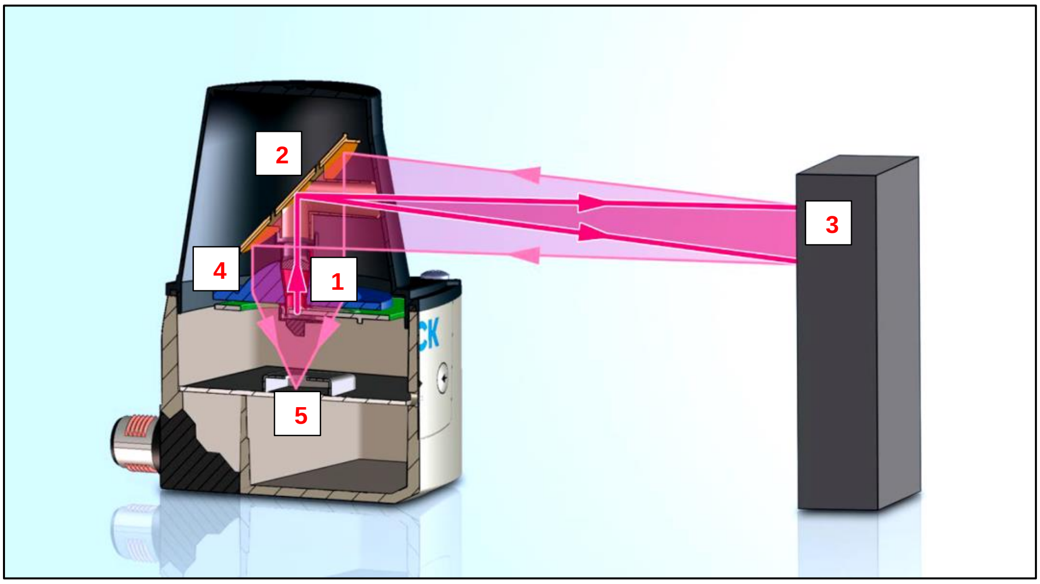


Figure 7. Principle of operation of a 2D LiDAR sensor (SICK 2018)

The basic working principle of LiDAR is shown in Figure 7. The main components of the system are marked by numbers 1-5. The transmitter (1) generates a laser beam, which changes the direction to horizontal with the help of a reflecting rotating mirror (2). After facing an obstacle (3), the bunch of reflected beams returns to LiDAR. The mirror (2) redirects the beams towards focusing medium (4) which in turn sends them to the detector (5).

The main advantage of LiDAR sensors, in comparison to other distance measurement devices, is that it makes measurements around itself with a wide scanning angle and high angular resolution. This means that the LiDAR can measure the distance up to the surrounding objects in the environment almost instantly. The feature is realized with the help of a rotating mirror inside the LiDAR device. The sequential order of the laser pulses is synchronized with the rotation frequency of the motor and the desired angular resolution. This process uses the effect of many individual pulses, which are then grouped together using the known transmission pattern and by way of statistical approaches, providing information on distance and echo signals. (SICK 2018.)

In frames of the current Bachelor's thesis project, the LiDAR LMS511-10100 PRO, manufactured by SICK, was used. The LMS5xx is a non-contact laser measurement sensor that scans the surrounding perimeter radially on a single plane using pulses of light. The LMS5xx makes two dimensional measurements

sending laser impulses around itself (maximum scan angle 190°) and reading received back signals. The output data of LMS5xx is the position of the object including the distance to the object and angle. As can be seen in Figure 8, The LMS5xx cannot scan the environment through objects. (SICK 2017.)

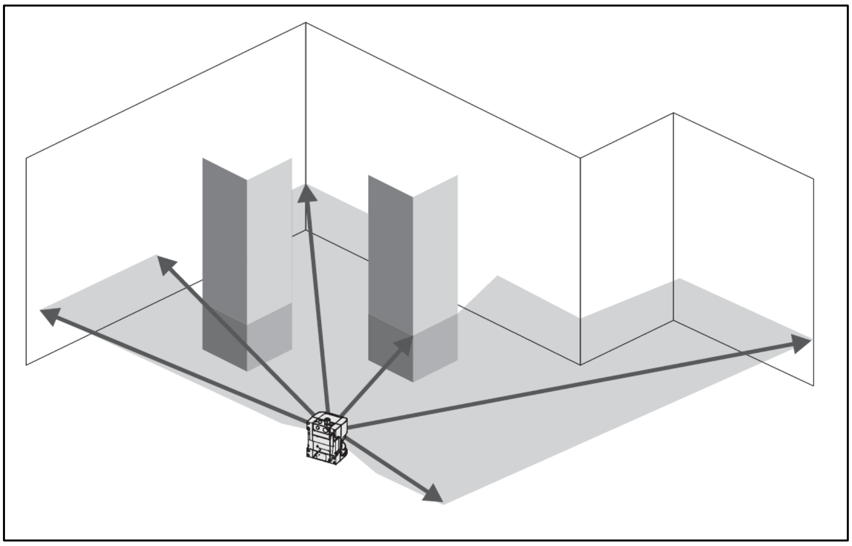


Figure 8. Measuring principle of the LMS5xx (SICK 2017)

The extract from the technical specifications of the used LiDAR (SICK LMS511-10100 PRO) is presented in Table 1.

Functional data	**Value**
Supply voltage	19,2-28,8 V
Scan angle	max. 190°
Scanning frequency	25-100 Hz
Angular resolution	0,1667-1°
Distance measuring range	0,7-80 m
Measurement accuracy	Standard resolution ± 24 mm High resolution ± 12 mm
Laser class	Laser class 1 according to EN/IEC 608251:2014
Beam spot size at the front screen	13,6 mm
Divergence of the collimated beam	Standard resolution 11,9 mrad High resolution 4,7 mrad

Table 1. Extract from Datasheet LMS5xx laser sensor (SICK 2017)

2.3 Robot Operational System

For the operation of any mobile robot, it is necessary to equip it with a variety of devices, sensors and activators. As was established by the Belzunce (2015), one of the optimal solutions for combining these features is implementing Robot Operational System (ROS) which provides both the communication protocol and its implementation for the robot.

The system is popular throughout students and researchers working with robotics projects of different scale, as it provides some significant benefits in contrast to other robotic platforms. The ROS has a lot of ready to use packages that can be implemented for localization, navigation, motion planning etc. of various robots. The system is packed with device drivers and packages of various sensors and actuators in robotics. The modularity of ROS significantly increases the stability of the system during operation, as the execution of different tasks is handled independently. The platform supports the integration of scripts written on C++, C, Java and Python programming languages in one project. In addition, ROS has a lot of useful tools for visualization, debugging, and simulation of robotic systems. Another advantage of the platform is an active online community, where users can ask for the help and share their experience. (Joseph 2015.)

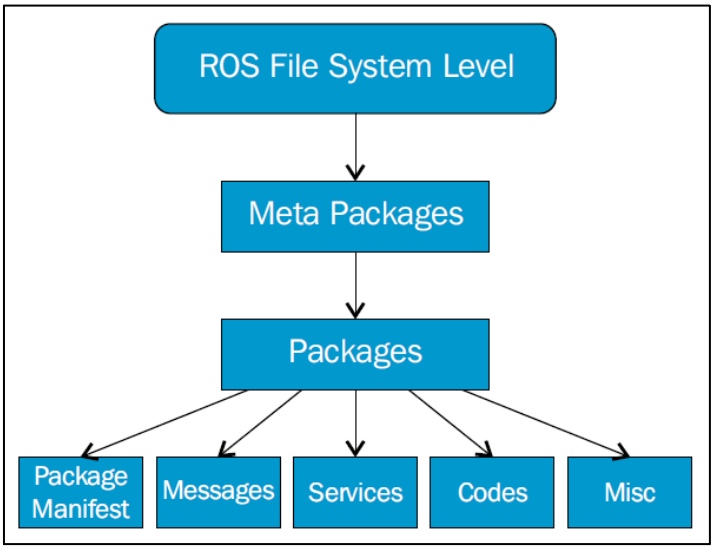


Figure 9. ROS File system (Joseph 2015)

The file system of ROS can be described with the following explanations (Joseph 2015):

- As can be seen in Figure 9, the main structural element of ROS file system is a *package*. Packages lie at the lowest level of the ROS software organization. They can contain libraries, tools, executable files, etc.
- The packages, that need to perform certain operations with individual elements of the robot hardware, can be combined into a *Meta Packages* in order to simplify the code perception.
- A *package manifest* is a file describing a package. The main task of the manifest is to define dependencies between packages.
- The *message* is a type of information that is sent from one ROS process to the other.
- The *service* is a sort of request-reply communication between processes.

The computation in ROS is done using a network of processes called ROS nodes. A *node* is an independent and separately executable code that is responsible for the execution of certain tasks, including hardware initialization or/and control, processing incoming data or operator signals, et cetera. In order to send some information, the nodes can post messages on a *topic*. In case the node needs to receive some information, it can subscribe to a topic. The communication between nodes is managed by a *master*. (Joseph 2015.)

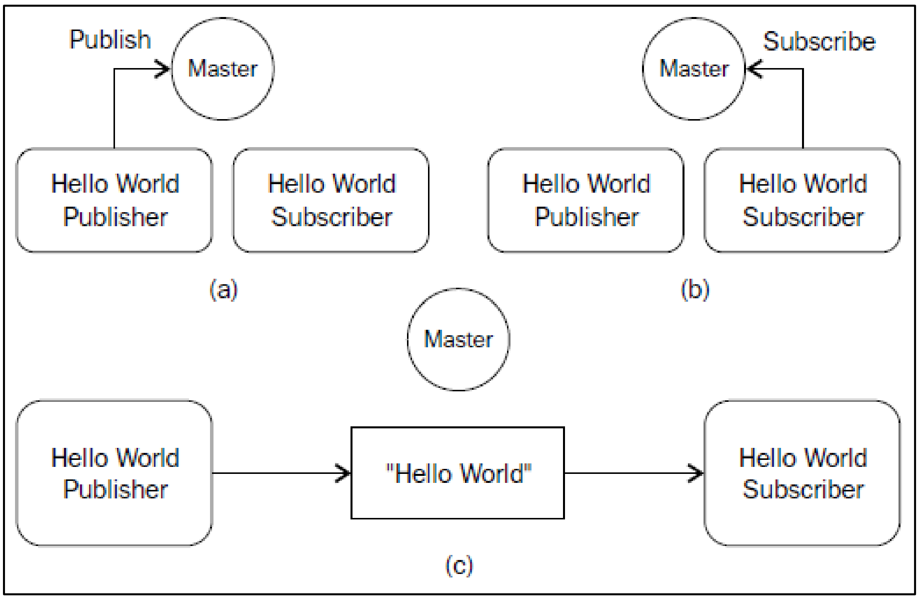


Figure 10. Communication between ROS nodes (Joseph 2015)

The procedure of sending information by nodes over ROS is shown in Figure 10. First of all, the node publishes the topic and notifies the Master about the publication of the data through this topic (step “a” in Figure 10). After that, the Master is waiting for any node which will subscribe to this topic (step “b”). After a subscription, the master establishes a peer-to-peer connection between the nodes through which data is exchanged (step “c”).

ROS on TIERA

Since the ROS can be utilized only on Linux operating systems (OS), the Ubuntu 14.04 which supports ROS Indigo distribution was installed on the Advantech industrial computer and the operator’s control station. The Linux OS was installed on control station with Windows 7 OS using VirtualBox virtual machine.

The main packages of TIERA’s traction system are *epos_hardware* and *lut_track.* The *epos_hardware* package executes a function of the motor’s driver. It was initially developed by Mitchell Wills from RiverLab (Northeastern University) and then customized by Kirill Romanov (2018), the researcher of LUT Laboratory of Intelligent Machines. The package has an *tiera_track_robot.launch* file, that runs one of the most important files in the package - *epos_hardware_node.cpp* which works with EPOS motors’ controllers. (Romanov 2018.)

The developed *lut_track* package is installed on the operator’s station. The package includes the *tiera_track_station.launch* file. The file loads the *joy* node that enables listening for commands from Xbox controller connected to the operator station; runs the *lut_controller.cpp* node which runs the high level controller over traction system; executes *kinematics_server.cpp* which enables the kinematic calculation service. (Romanov 2018.)

3 Theory of autonomous robots

3.1 Autonomous mobile robots

The principle of the autonomous system implies a system that is able to act itself in order to execute the necessary steps towards the achievement of predefined goals. Besides, the system must take into account information about the environment which comes from sensors. (Cardon & Mhamed 2016, pp. 13-14.)

According to Japanese Industrial Robot Association, the robots can be divided into six classes (Reza 2007, pp. 7-8):

- Class 1: Manual handling devices: a device that is actuated by an operator.
- Class 2: Fixed sequence robot: a device that performs the tasks according to a fixed program.
- Class 3: Variable sequence robot: a device that performs the tasks according to a predefined but programmable method.
- Class 4: Playback robot: a robot that can learn and later repeat the motions executed under the guidance of a human operator.
- Class 5: Numerical control robot: the robot, which executes the motion program supplied by the operator.
- Class 6: Intelligent robot: a robot with the ability to understand its environment and the ability to successfully complete a task despite changes in the surrounding conditions under which it is to be performed.

The robot TIERA under the research belongs to the 1st class and it is currently modified to become a robot of the 6th class. While being a manual handling robot, it executes the tasks received from the operator's control station and requires permanent control of the human operator. Becoming the intelligent robot means getting the ability to understand the environment where it operates and the ability to successfully execute the tasks overcoming the obstacles of the environment. However, being an intelligent robot does not mean making any decisions by itself. The robot must understand, what to do in any case, but the behaviour must be programmed by human before switching the robot to autonomous mode.

3.2 Navigation

Navigation is one of the most challenging tasks of the mobile robot. Generally, the navigation implies the autonomous movement of the robot to a certain point. The accuracy in navigation is extremely important in industrial applications. High precision in positioning and movements is extremely important to perform a variety of tasks including docking, transportation or mobile manipulation tasks such as pick and place (Beinhofer et al. 2013).

Successful navigation is achieved by reliable work of four blocks: perception, localization, cognition, and motion control. Perception is the ability of a robot to understand data received from sensors and get useful information out of it. Localization is the ability of a robot to determine its position in the environment. Cognition is the ability of a robot to plan the road to achieve the goal. Motion control – the robot must be able to control its motors in a way that it could precisely follow the planned trajectory. (Siegwart & Nourbakhsh 2004, pp. 181-182.)

The perception and motion control blocks depend on the used equipment and operating system of the robot, so they are specific blocks for every model of the robot. In contrast, the localization and cognition blocks are universal which allows them to be implemented in various types of mobile robots.

Localization

The simplest way to define a robot's location is to install a GPS sensor which will constantly track the location of the robot based on the satellite signal. However, the GPS system provides unacceptable accuracy within several meters (Siegwart & Nourbakhsh 2004, p. 182). Another solution for robot's localization is to track some external references, such as wires embedded in the floor or magnetic tapes on the floor (Röwekämper et al. 2012). However, such a technology limits the flexibility of a robotic system. These technical issues oblige the robot to define and track its location itself, purely with the help of laser sensors with high measurement precision.

The simplified schematic of localization of a mobile robot is presented in Figure 11. As one can notice, the robot identifies its location based on three inputs: pre-

defined map, sensor data, and predicted position. Every comparison of these three components allows the robot to update his assumption on current location. The prediction on a new location, or odometry, is generated with the use of the current position, map, and new data received from the encoder. (Siegwart & Nourbakhsh 2004, pp. 182-183.)

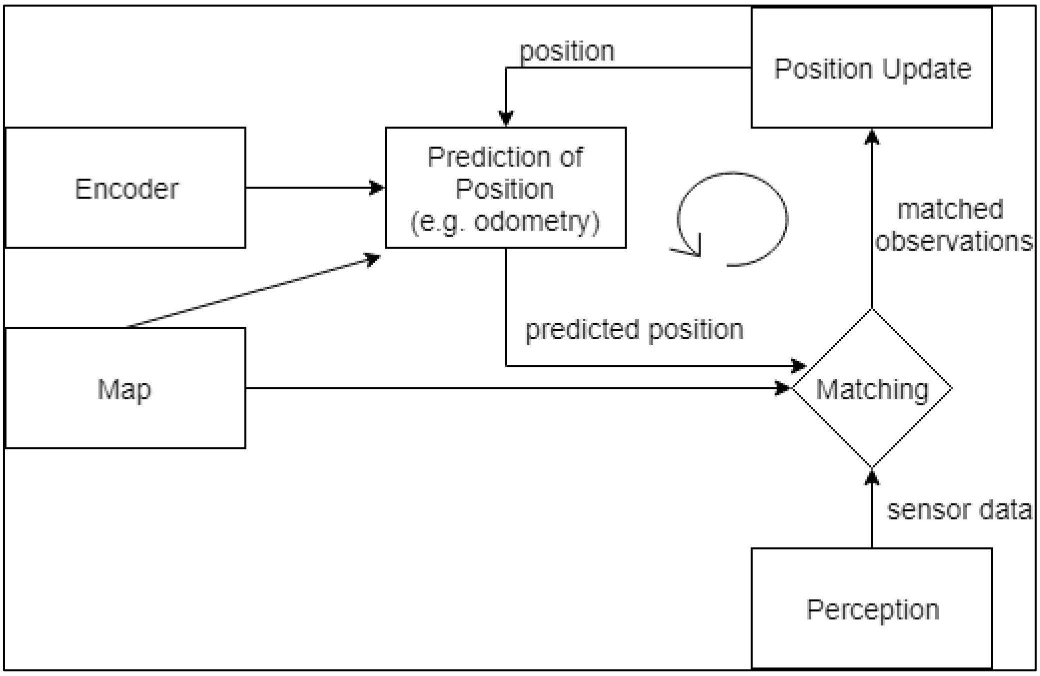


Figure 11. General schematic for mobile robot localization (Siegwart & Nourbakhsh 2004)

The odometry information is a rough estimate of the robot's position, based on the sum of incremental travel distances starting from a known position. Every position of the robot can be represented by the vector shown in Formula 1. The updated position for odometry can be calculated with Formula 2. (Siegwart & Nourbakhsh 2004, pp. 186-188.)

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1}$$

$$p' = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\theta + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \quad (2)$$

where

$\Delta s_r, \Delta s_l$– travel distances for the right and left wheel respectively,
b – distance between the two wheels of robot.

As was mentioned earlier the robot keeps tracking its location in a known environment with the help of odometry. However, due to odometry error, after some movements the robot will become very uncertain about its position. In order to prevent increase of uncertainty, the robot must combine the known data from odometry, map, and on-board sensors. The observed by sensors measurements, when compared to robot's environmental model, tend to provide some clues regarding the robot's possible position. Further localization work is handled by complicated algorithms. (Siegwart & Nourbakhsh 2004, pp.212-213.)

There are two main classes of probabilic map-based localization. They are *Markov localization* and *Kalman filter localization*. The first method, *Markov localization*, uses an explicitly specified probability distribution across all possible robot positions. The benefit of this method is that it allows localization from any unknown position. However, the update of the probability of all possible locations within the whole map requires a discrete representation of the space. The limitations of this method are high memory and computational power requirements. (Gutmann & Konolige 1998.)

The second localization method class, *Kalman filter localization*, uses a Gausan probability density representation of position and scan matching for localization. This method combines multiple measurements into a single estimate of the state. The tracking of location is conducted from the initially known position. That approach increases both precision and efficiency. However, in cases when the uncertainty of the robot becomes too large (for example after collision with an object), the *Kalman filter* may generate multiple possible position options and become completely lost. (Gutmann & Konolige 1998.)

Cognition

Cognition is an ability of the robot to plan the motion path from the current position to some point on the map. The process of path planning always starts with the transformation of the current environmental model into a discrete map suitable for the selected path-planning algorithm. There are three main strategies for path planning: *road map*, *cell decomposition*, *potential field planning*. The illustrations of the strategies are shown in Figures 12, 13, and 14, respectively. The basic principles are described below. (Siegwart & Nourbakhsh 2004, pp.257-261.)

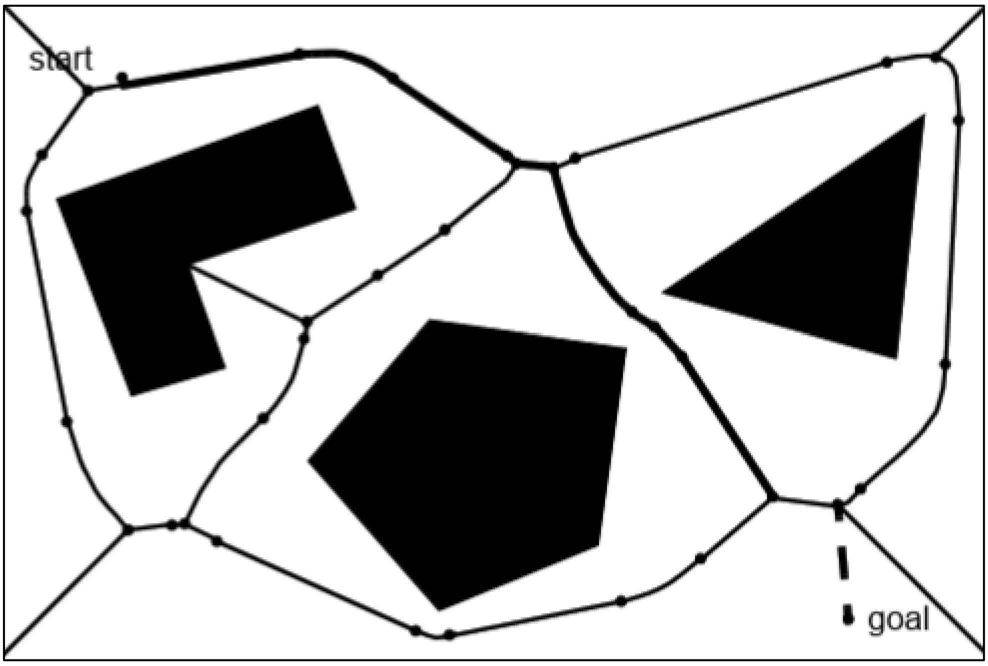


Figure 12. Road map approach, Voronoi diagram (Siegwart & Nourbakhsh 2004)

The *road map* approach implies the creation of a set of routes within a free space between obstacles. The paths are presented by a network of curves or lines, called *road maps*. The path planning process starts after a network of possible roads is constructed. Basically, the motion planning is a process of searching for series of roads from the start position to the goal position. The specific example of road map approach is a *Voronoi diagram*, given in Figure 12. The *Voronoi diagram* is a method that tends to maximize the distance between the planned path and obstacles. The method is achieved by placing each point of network curves equidistant from the nearest obstacles. (Siegwart & Nourbakhsh 2004, pp.261-264.)

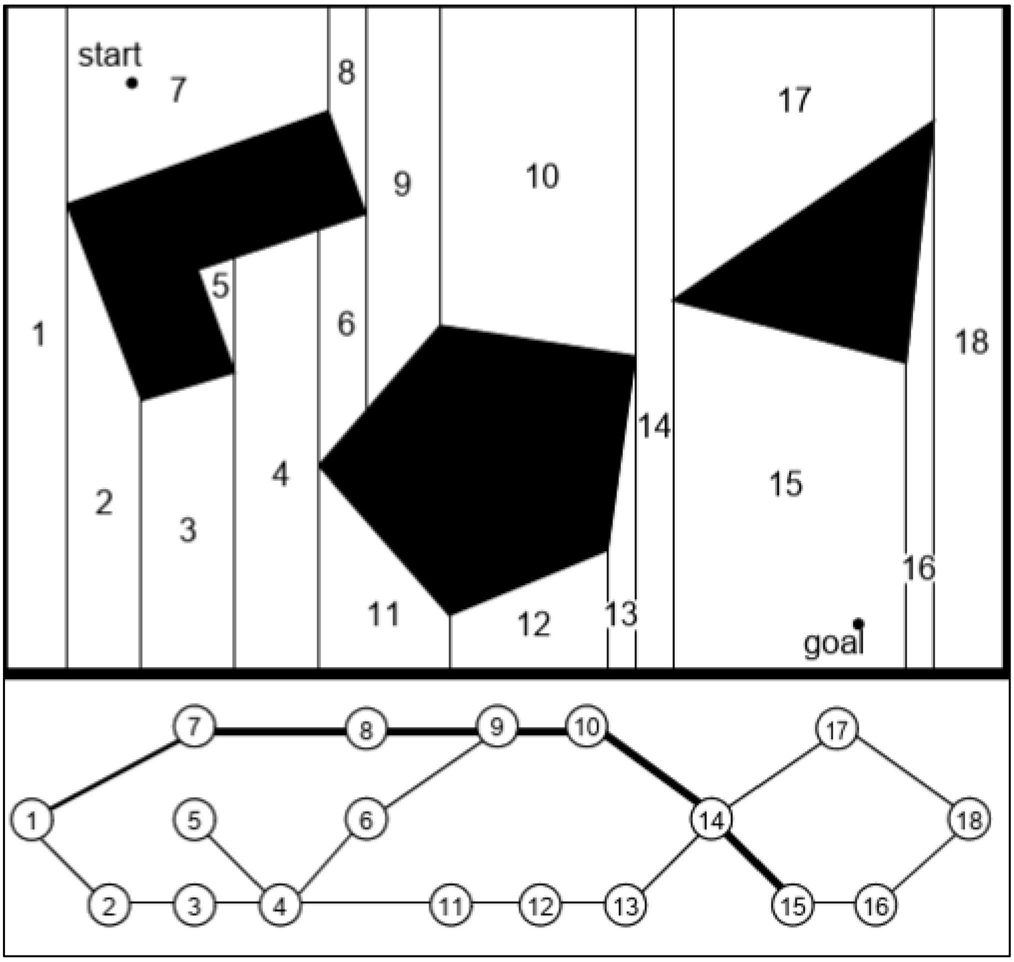


Figure 13. Exact cell decomposition approach (Siegwart & Nourbakhsh 2004)

The *cell decomposition* method selects the path by analysing which cells of the map are free and which are occupied. The visual representation of the approach can be seen in Figure 13. The idea of the method is that the map is divided into regions or cells. The free of obstacles cells are numbered and combined into the connectivity graph. Then the shortest path between the cells with start and goal points is found. After that, the movement algorithm is defined by passing the middle point of each cell on the selected path. (Siegwart & Nourbakhsh 2004, pp.264-267.)

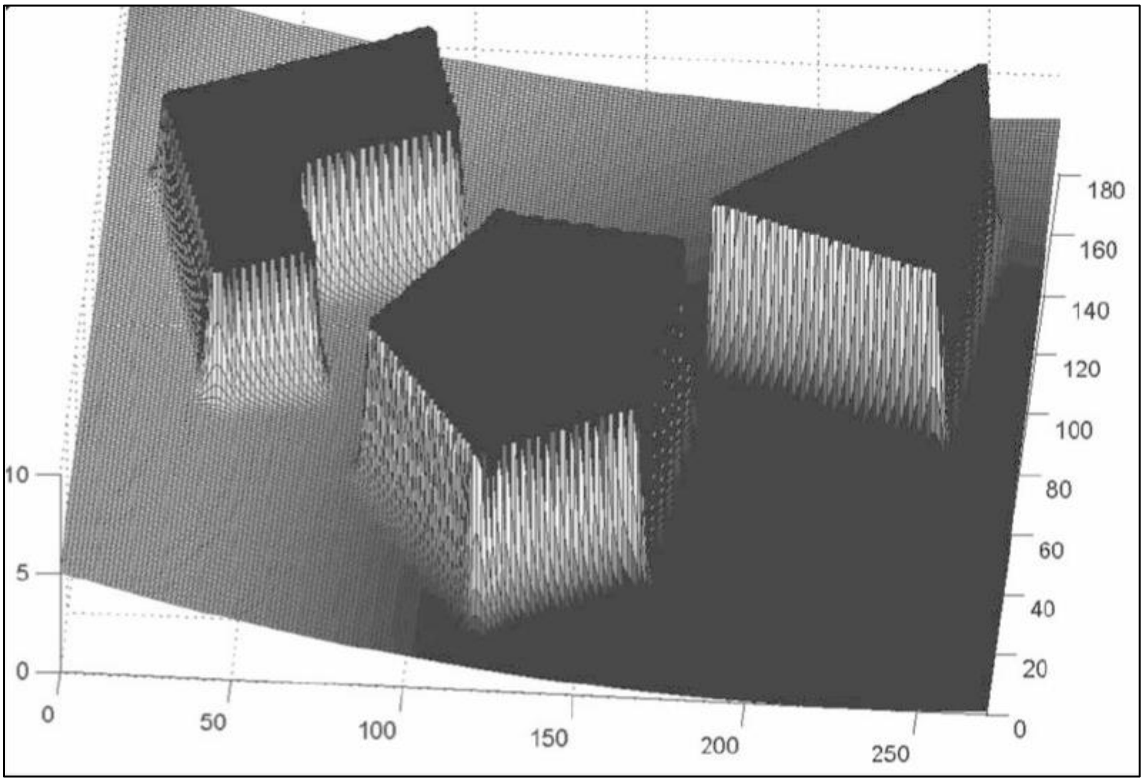


Figure 14. Potential field planning approach (Siegwart & Nourbakhsh 2004)

The *potential field planning* strategy means the generation of a field or gradient across the robot's map that leads a robot to the goal position from multiple prior positions. The visual graph can be seen in Figure 14. In simple words, in this method, the map is extruded with the help of mathematical functions. The highest points of the extruded map are obstacles. The goal point is located lower than the start point. So in order to reach the goal, the robot must just "slide downhill" on the extruded map, like a ball. (Siegwart & Nourbakhsh 2004, pp.267-271.)

3.3 Navigation Stack in ROS

Conceptually the *Navigation Stack* in Robotic Operational System (ROS) is simple. The velocity commands for a mobile base are generated with the usage of odometry information and sensor streams. There are three main hardware requirements for correct integration of *Navigation Stack* (Martinez & Fernández 2013, pp. 140-145):

- The robot mobile base must be a differential drive or holonomic wheeled. It must be controlled by sending desired motion commands in the form of: x velocity, y velocity, theta velocity.

- A planar laser must be mounted on the mobile base.
- The performance of *Navigation Stack* will be best on robots that are nearly square or circular shape. It may have difficulty with large rectangular robots in narrow spaces like doorways.

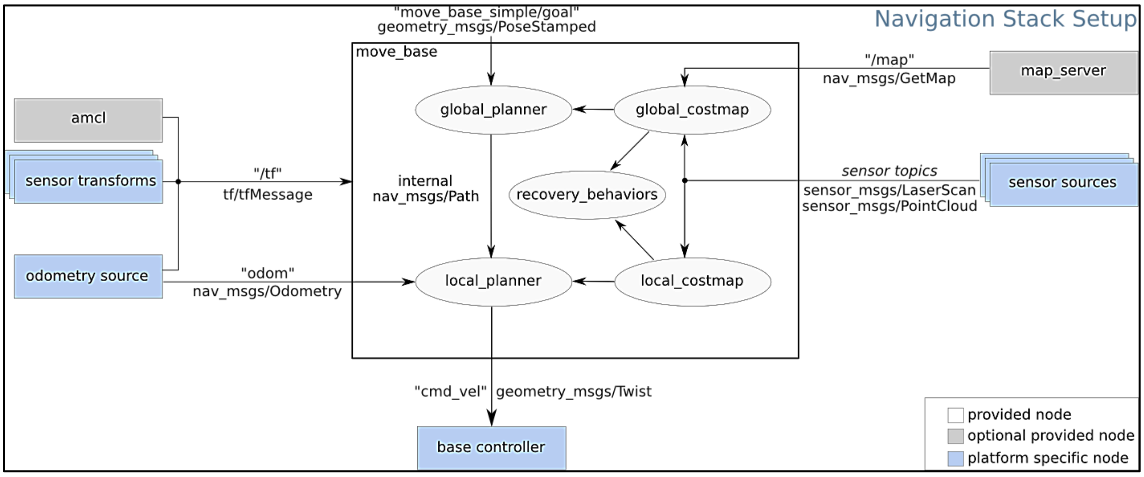


Figure 15. Navigation stack setup in ROS (Martinez & Fernández 2013)

The software configuration of the *Navigation Stack* must be done in a particular manner in order to get a working system. The diagram in Figure 15 shows the configuration principle and connections between *Navigation Stack* packages. The white and grey (optional) components are provided by *Navigation Stack*. The blue components must be created for each robot platform. The required pre-settings of the *Navigation Stack* are listed below (Martinez & Fernández 2013, pp. 141-142):

- The robot must use ROS.
- The robot must publish information about the relationships between coordinate frames using *TF*. *TF* is a ROS package that lets the user transform points, vectors, etc. between any two coordinate frames at any desired point in time.
- The sensors must publish either *sensor_msgs/LaserScan* or *sensor_msgs/PointCloud* messages over ROS.
- The odometry information must be published using *tf* and the *nav_msgs/Odometry* message.
- The Base Controller of the robot must be able to understand motion commands using a *geometry_msgs/Twist* message.

- The pre-defined map of working environment is beneficial for stable work of *Navigation Stack*.

The *Navigation Stack* includes *global_costmap* and *local_costmap* nodes, which are responsible for storing information about obstacles. The *global_costmap* is used for global path planning, and the other costmap is used for local planning and obstacle avoidance. The idea of costmap node is that it takes sensor data about the world, builds a 2D occupancy grid of the data, and converts it to occupancy map. The *global_planner* and *local_planner* are the nodes that create an entire path from start point to the goal and around a few meters around the robot, respectively. (Martinez & Fernández 2013, pp. 142-145.)

4 Competitors review

This chapter includes a brief review of TIERA mobile robot's competitors. The review includes an overall description of robots, their main components, special features, and main specifications.

HelMo mobile robot system

HelMo is an autonomous mobile robot system, manufactured by Swiss company Stäubli. The main advantage of the robot is in its capability to work both completely independently and in collaboration with human employees. The robot was designed to be able to work in almost every industrial sector, like logistics, quality control, assembly or provisioning. The setup of the robot's systems, like gripper, navigation, and rotation unit can be done in a special software with a user-friendly interface. The building of a new map of the working environment can be easily done with the help of provided mobile software. (Stäubli 2017.)

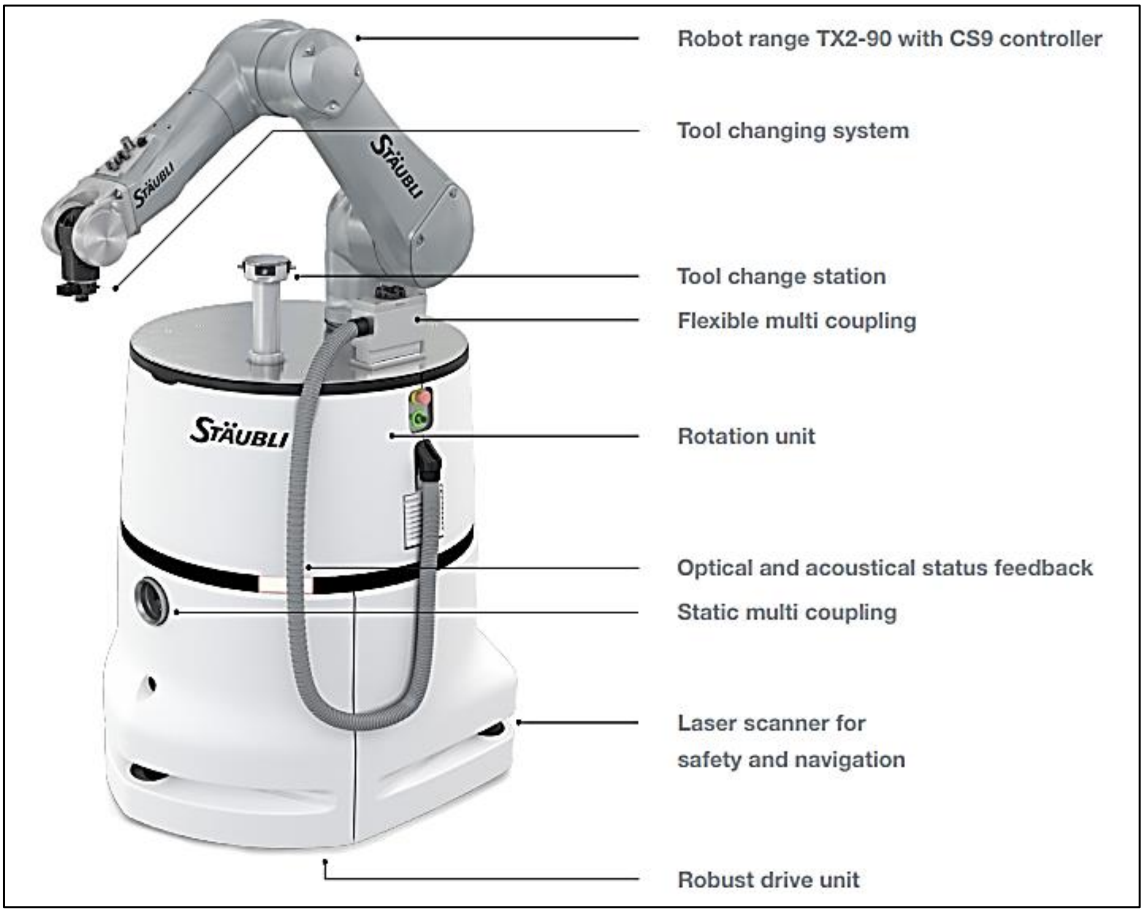


Figure 16. HelMo mobile robot system (Stäubli 2017)

The structure of HelMo robot is shown in Figure 16. As can be seen in the Figure, the main components of the robot are the robust drive unit, rotational unit, and TX2 standard robotic arm. The drive unit provides autonomous navigation of the robot in the workshop based on a laser scanning system. The rotating unit provides a maximum rotation angle of 350° at a maximum speed of 16,8 °/s. The robot arm with six degrees of freedom (DOF) has a maximum carrying capacity of 20 kg. The extract from the technical specifications with the rest important technical data is shown in Table 2. (Stäubli 2017.)

Overall system	
Dimensions (W x H x D)	890 x 2120 x 1000 mm
Weight	710 kg
Battery capacity	4,8 kWh
Power supply	230 V
Communication interfaces	WLAN, LAN, Fieldbus
Compressed air tank	0,75 l / 8 bar
Robot arm TX2-90	
Maximum/nominal load capacity	20/7 kg
Reach (between axis 1 and 6)	1000 mm
Repeatability - ISO 9283	±0,03 mm
Drive unit	
Maximum speed	0,6 m/s
Movement accuracy	± 10 mm
Drive type	Differential drive

Table 2. Extract from technical specifications of HelMo robot (Stäubli 2017)

The navigation system of the drive unit is based on laser scanning data received from three LiDAR sensors which together cover 360° of the environment around the robot. One of the most valuable features of HelMo robot is an implemented safety system that immediately stops the movement in case a contact with an obstacle or human colleague.

KMR iiwa robot

The KMR iiwa is a combination of the sensitive LBR iiwa (intelligent industrial work assistant) lightweight robot and an autonomous mobile platform, both manufactured by KUKA. The robot is a highly flexible, location-independent production assistant with an unrestricted workspace – an ideal basis for the intelligent, networked production worlds of Industrie 4.0. (KUKA 2018.)

Figure 17. KMR iiwa robot (KUKA 2018)

The KMR iiwa robot is presented in Figure 17. As can be seen in Figure, the main components of the robot are the flexible platform and sensitive LBR iiwa light-weight robotic arm. The drive unit with omnidirectional wheels provides maximum flexibility and unrestricted manoeuvrability of the robot. The robot arm has a maximum carrying capacity of 14 kg and seven special joint torque sensors on each axis for high sensitivity to its environment. The extract from the technical specifications is shown in Table 3. (KUKA 2018.)

With the implemented navigation system the KMR iiwa can reliably move around obstacles and look for a new route. The autonomy in motions is achieved by permanent monitoring of the environment by laser scanners. The navigation system uses the Simultaneous Localization And Mapping (SLAM) method. By means of it, the platform is able to pinpoint its location in real time on a map of its environment created from the data of the safety laser scanners and wheel sensors. (KUKA 2018.)

Overall system	
Dimensions (W x H x L)	630 x 2006 x 1080 mm
Weight	412 kg
Robot arm LMR iiwa 14 R820	
Maximum load capacity	14 kg
Reach (between axis 1 and 6)	1300 mm
Repeatability - ISO 9283	±0,1 mm
Drive unit	
Maximum speed	1 m/s
Movement accuracy	± 5 mm

Table 3. Extract from technical specifications of KMR iiwa robot (KUKA 2018)

Summary

The conducted short review of existing autonomous industrial robots shows that TIERA project in general and the present Bachelor thesis project, in particular, are relevant. Mobile robots are widely used in the industry. The main features of advanced robot systems are safe to work in collaboration with human and autonomous navigation utilizing LiDAR sensors. As the present thesis project aimed to build a LiDAR platform to scan the environment of TIERA, it will help the robot to make his first steps in a long transformation process of becoming an autonomous industrial robot.

5 Development of a platform for autonomous motion

5.1 Installation of LiDAR system

5.1.1 Mounting

According to SICK Operating Instructions for LMS5xx Laser Measurement Sensors (2017) the LiDAR's mounting must be:

- robust
- as far as possible without shocks
- as far as possible without vibration
- with the entire field of view not restricted.

The position for the mounting on the robot was selected in a way that LiDAR could scan the environment as close to the ground as possible. That location is suitable for further research in case the obstacle avoidance feature will be implemented. However, it is recommended to separate navigation and obstacle avoidance functions between two separate LiDARs.

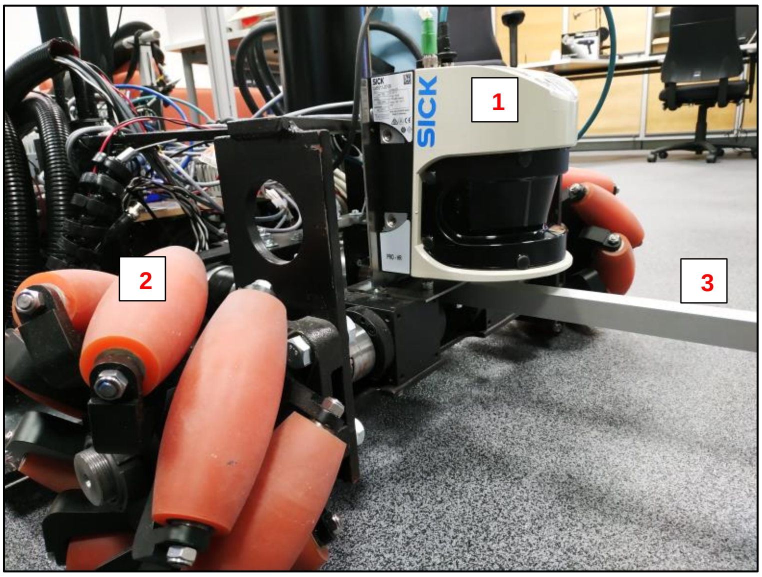


Figure 18. SICK LiDAR installed on robot TIERA: 1 – LiDAR, 2 – front wheels, 3 – charging connection extender.

The mounting part for the LiDAR was designed by the researcher. The mounting was manufactured in LUT workshop by the request of technical expert of LUT Laboratory of Intelligent Machines, Juha Koivisto. The manufacturing drawings

can be seen in Appendix 1. The installed LiDAR is shown in Figure 18. As can be seen in the Figure, the LiDAR (1) was installed between front wheels (2) of the robot, so that none of the robot's components intersect the laser beam.

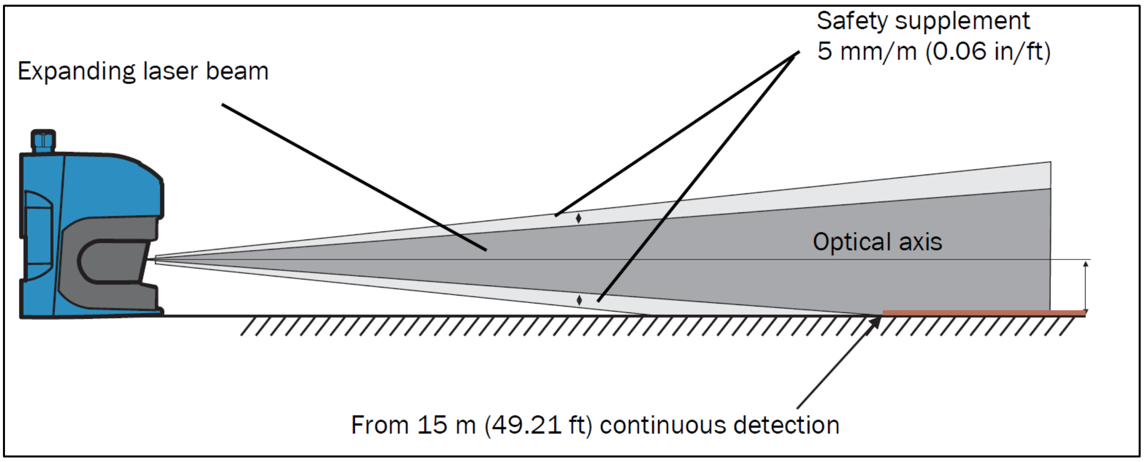


Figure 19. Increase in the size of the beam and safety supplement (SICK 2017)

It is important to notice that mounting of the LiDAR close to the ground brings the scanning distance restriction. That happens due to the laser beam dispersion over the distance. This issue is explained in Figure 19.

In order to achieve a reliable signal for navigation, the laser beam radius must not exceed the distance between the ground and the optical axis. The requirement is due to the fact that continuous detection of the floor may cause confusion for the navigation system during the localization process. According to Operating Instructions of LMS5xx Laser Measurement Sensor (2017), the distance from the front screen till the point with certain beam diameter can be calculated with the help of Formula 3.

$$l = \frac{D - d}{a + s} \tag{3}$$

where

l – distance from front screen (mm),
D – beam diameter (mm),
d – spot size on front screen (mm),
a – divergence (rad),
s – safety supplement, = 0,005 (rad).

For the assessment of the maximum reliable scanning distance, the beam diameter was replaced with the (distance between ground and optical axis) * 2. The

constants, such as spot size on the front screen and divergence, were defined in specifications Table 1. Thus, the maximum scanning distance is:

$$l = \frac{(250 * 2 - 13{,}6)mm}{(11{,}9 * 10^{-3} + 0{,}005)rad} = 28781mm \approx 28m$$

One can notice, that the obtained value of maximum reliable scanning distance is significantly less than the maximum value (80 m) declared by the manufacturer. This means that the selected mounting place restricts scanning potential of the device. The issue can be solved by raising the LiDAR higher from the ground level. That problem can be solved during further development of TIERA project and implementation of a changing system.

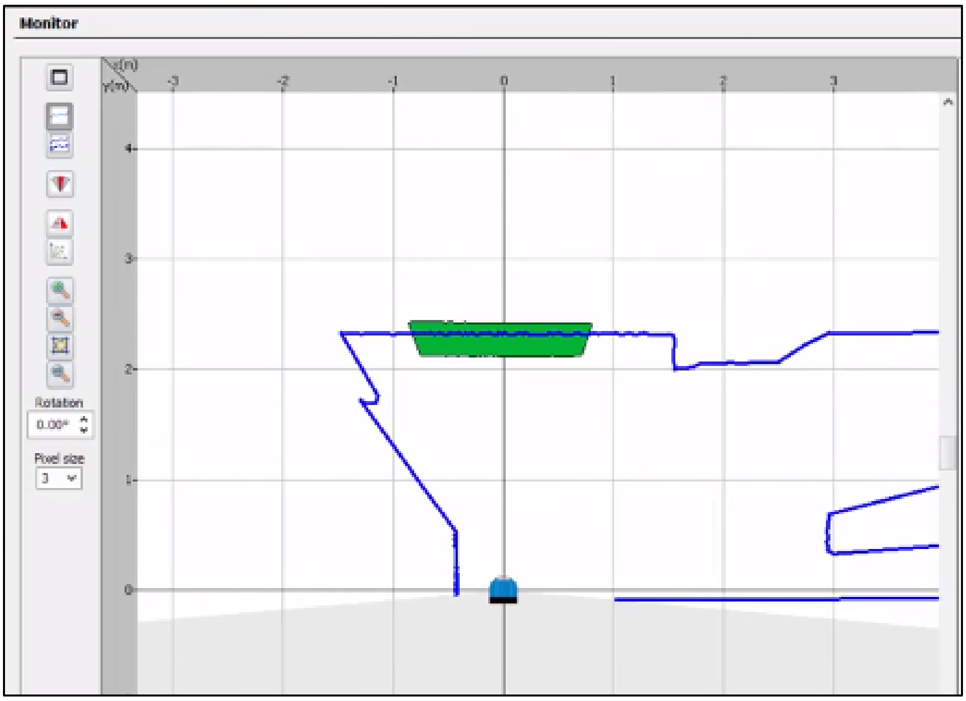


Figure 20. Scans monitoring interface in SOPAS ET

The preliminary setup and testing of SICK LiDAR were conducted with the help of special software - SOPAS Engineering Tool (ET). The software was developed by SICK company – the manufacturer of LiDAR under the research. The software is compatible with Microsoft operating system. The SOPAS ET provides an opportunity to setup device parameters, such as IP address, scanning range, angular resolution, scanning distance, scanning frequency etc. in a user-friendly interface. In addition, the software allows the user to see the obtained measuring data

in a graphical presentation, setup evaluation areas for the scanner, and add the response behaviour for the events of catching an object in evaluation areas. The scans monitoring interface is shown in Figure 20.

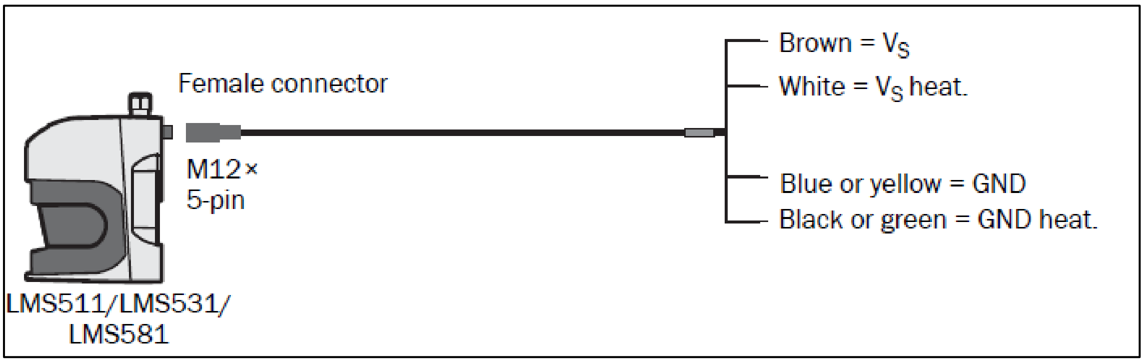


Figure 21. SICK LMS511 connection of the voltage supply (SICK 2017)

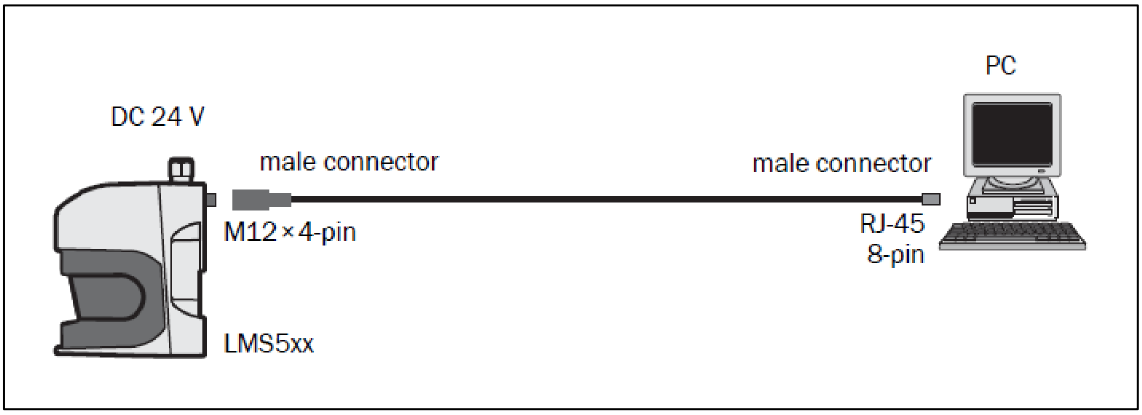
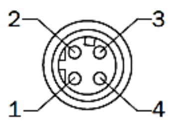


Figure 22. SICK LMS511 Ethernet connection (SICK 2017)



Pin	Signal	Function
1	TX+	Transmitter+
2	Rx+	Receiver+
3	TX–	Transmitter–
4	Rx–	Receiver–

Table 4. Pin assignment of the “Ethernet” M12 female connector (SICK 2017)

The connection of SICK LiDAR to TIERA’s power supply and control unit was done according to SICK LMS5xx Laser Measurement Sensors Operating Instructions (2017). The sensor was connected to a 24V power source. The wires assessment of power cable can be seen in Figure 21. The data transfer cable, the characteristics of which are shown in Figure 22 and Table 4, was connected to the Ethernet port of Advantech control unit.

5.1.2 Integration to ROS

The integration of the SICK LiDAR into ROS environment was done by means of the official *sick_scan* package, developed by Michael, Jochen Sprickerhof and Martin Günther. The package was cloned from GitHub repository ("devel" branch). The package supports a large variety of SICK laser scanners. As an output, the package provides the measurements as *PointCloud2* and *LaserScan* data. The *LaserScan* data was used for project proposes. The transcript of a single *LaserScan* message can be seen in Appendix 2.

After successful installation of *sick_scan* package, the system was tested. The *sick_lms_5xx* launch file was called in the Linux Terminal with the following command:

$ roslaunch sick_scan sick_lms_5xx.launch

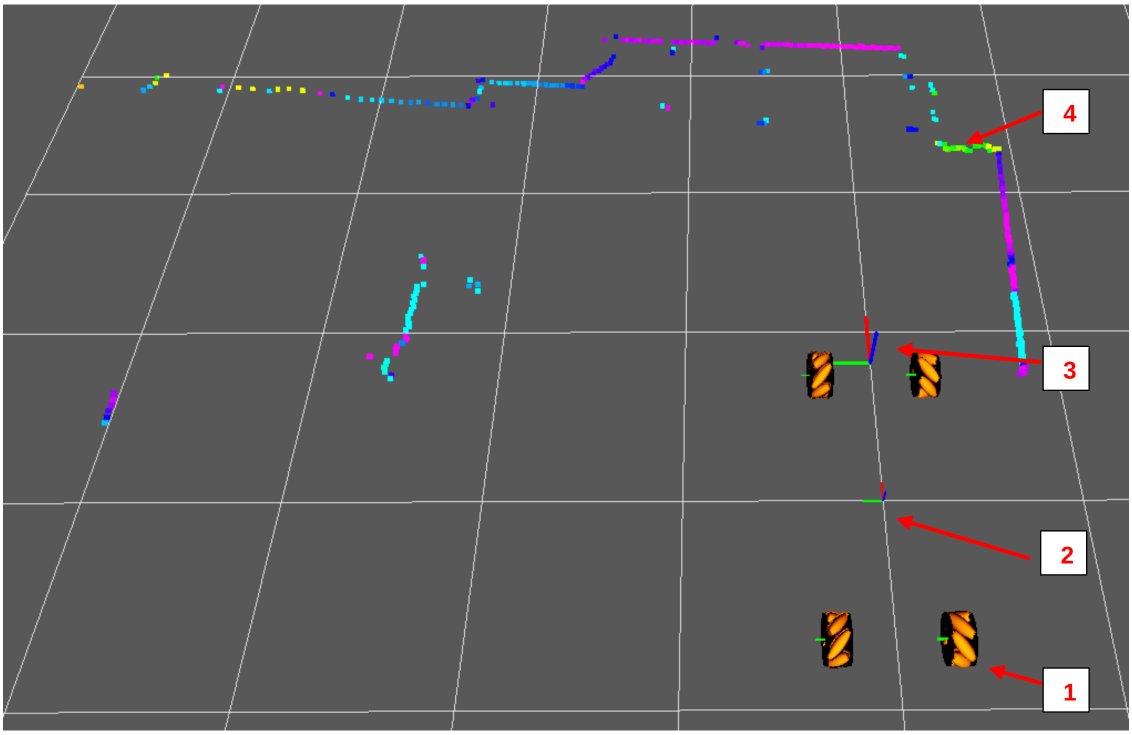


Figure 23. Visual representation of scanned environment using LiDAR: 1 – robot's wheels, 2 – coordinate frame of the robot, 3 – coordinate frame of LiDAR, 4 – measurements representation

The launching file starts the *sick_scan* node with configuration parameters for the device. When launched, the node sends *LaserScan* messages to ROS, through */scan* topic. The visual representation of data obtained with the help of a single scan of LiDAR and published through */scan* topic can be seen in Figure 23. As

one can notice, the scanned objects are represented by a series of dots. Each dot is located at the point where the laser beam faced with the object. It is interesting to notice, that the distance between nearby points increases with the increase of measured distance. That happens due to the dispersion of the laser beams.

5.2 Development of autonomous motion system

The overall aimed algorithm of the charging procedure is shown in Figure 24. As one can see in the figure, the robot must check the battery level every pre-defined time contestant. As soon as the voltage level drops down to the pre-defined minimum level, the robot must finish execution of the current task and go to the charging station. The trip to the charging station starts with the determination of the current location with the help of LiDAR and pre-defined room map. After that the robot must plan the road to the charging station with the means of ROS and go there autonomously. After the accurate docking, the robot must stay stationary until the battery level reaches a pre-defined “full” level. The time constant of pause between voltage level checks must also be pre-defined in the code. Finally, the robot must return to execution of his tasks.

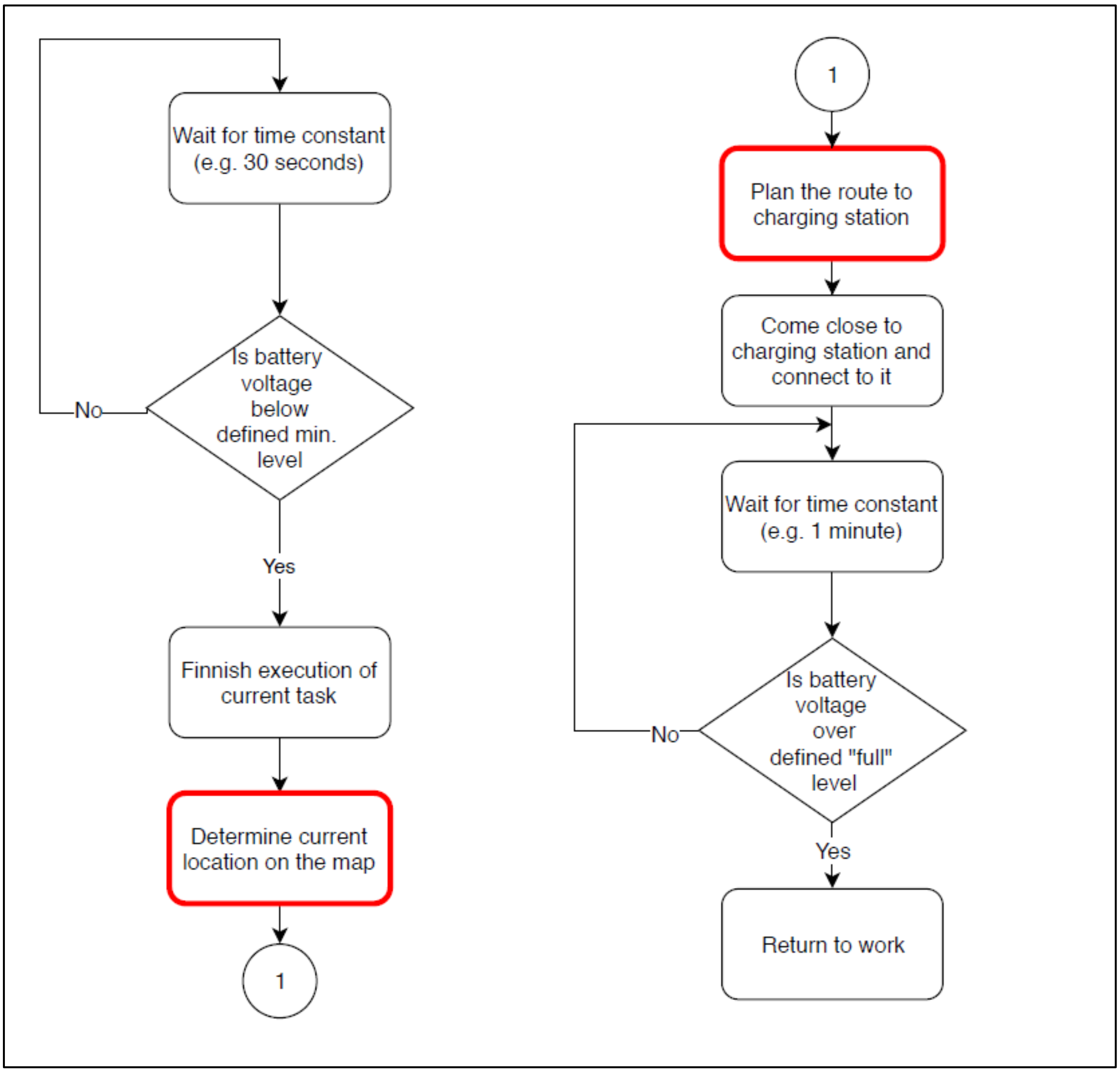


Figure 24. Charging procedure algorithm

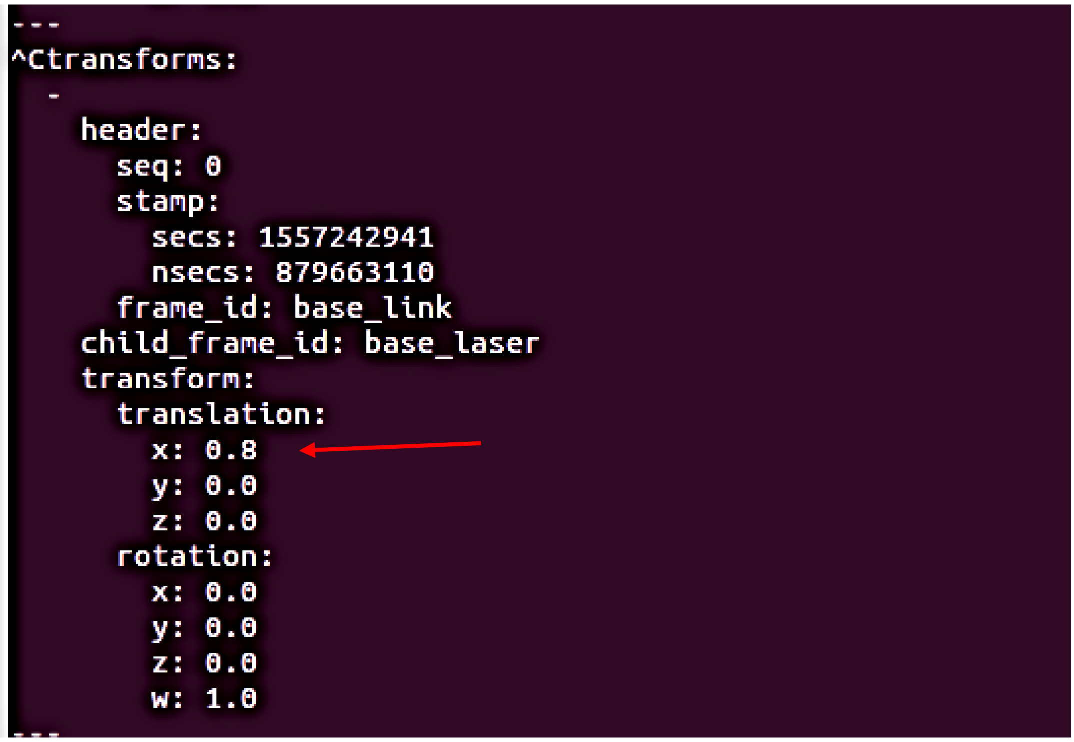
It is important to notice, that the aim of the current Bachelor's thesis was to teach the robot to determine the current location on the map, plan the road and go to a certain point on the map autonomously. The blocks of the algorithm, that must be implemented, were marked with red colour in Figure 24. The implementation of the rest of the charging algorithm was left for further research.

5.2.1 Robot setup

As was described in chapter 3.3, the ROS *Navigation Stack* has a list of requirements for stable work. The setup of the robot started with the creation of the package responsible for the transformation of laser data from the laser coordinate frame to the robot's coordinate frame. The package was named *robot_setup_tf*. The package includes the main executable file written on C++ language and

called *tf_broadcaster.cpp*. The content of the file with comments can be seen in Appendix 3. In order to start sending transformations to ROS, the *tf_broadcaster.cpp* file must be launched in Linux Terminal with the following command:

$ rosrun robot_setup_tf tf_broadcaster



```
---
^Ctransforms:
  -
    header:
      seq: 0
      stamp:
        secs: 1557242941
        nsecs: 879663110
      frame_id: base_link
    child_frame_id: base_laser
    transform:
      translation:
        x: 0.8
        y: 0.0
        z: 0.0
      rotation:
        x: 0.0
        y: 0.0
        z: 0.0
        w: 1.0
---
```

Figure 25. Message of transformation laser frame referenced data to base frame

The messages, which are sent to ROS through */tf* topic while running *tf_broadcaster*, are presented in Figure 25. As one can see in the Figure, the only x coordinate must be transformed for 0,8 meters.

Another required setup for the *Navigation Stack* was the publishing of odometry using *tf* and the *nav_msgs/Odometry* message. The code for odometry calculation and publishing was integrated to *lut_controller.cpp* file which is responsible for high-level control over traction system. The written code with comments can be seen in Appendices 4 and 5.

```
---
header:
  seq: 13761
  stamp:
    secs: 1557241852
    nsecs:  62536657
  frame_id: odom
child_frame_id: base_link
pose:
  pose:
    position:
      x: 0.071502278028
      y: 0.0541172707714
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: 0.0181940156264
      w: 0.999834475198
  covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
twist:
  twist:
    linear:
      x: 0.0
      y: 0.0
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: -0.0
  covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
```


Figure 26. Sample message published through */odom* topic

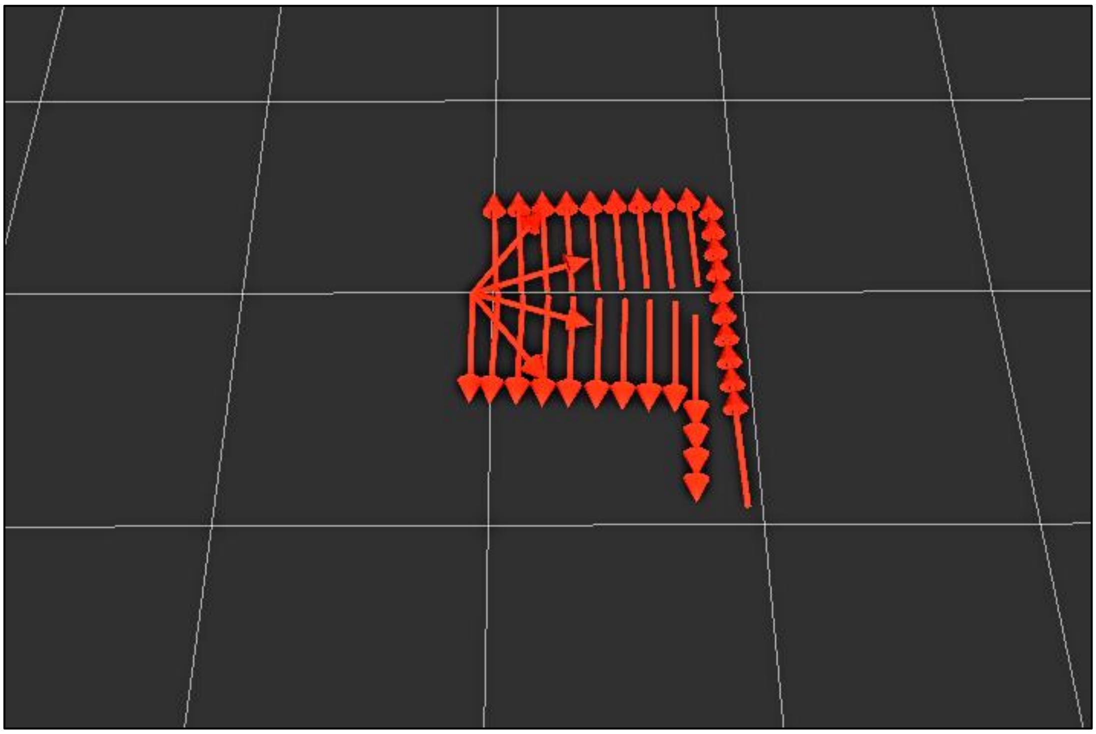


Figure 27. Visual presentation of odometry information

After debugging and compilation of updated *lut_controller.cpp* file, the odometry messages publishing through */odom* topic began. The sample odometry message and visual presentation of odometry are shown in Figures 26 and 27, respectively. As can be seen in Figure 27, each odometry update is represented by an arrow. In the test drive, the robot was driven a few meters forward, then a few meters left; after 180 degrees rotation, it was returned close to the initial position.

The final frames transformation graph can be seen in Appendix 6. The developed frames are marked with red circles.

5.2.2 Building a map

The procedure of map development was done based on guidelines from the book “Learning ROS for Robotics Programming” (Martinez & Fernández 2013, pp.241-243). In order to make a map, the ROS *gmapping* package was used. The procedure started by launching a traction system, *tf_broadcaster.cpp* file, and the *sick_lms_5xx.launch* file (launching commands were described above). After that the creation of a *mylaserdata.bag* file, which records the robot’s transformations and scanned data from LiDAR, was done by the following command:

$ rosbag record -O mylaserdata /scan /tf

During the recording of *mylaserdata.bag* file, the robot was moving around the closed test arena (shown in Figures 28 and 29) with the help of joystick commands.

Then *mylaserdata.bag* file was played and converted to a map with the help of *gmapping* package. The operations were done by execution of two commands in separate Terminals:

$ rosrun gmapping slam_gmapping scan:=scan

$ rosbag play --clock mylaserdata.bag

Finally, the developed map was saved as a picture with the following command:

$ rosrun map_server map_saver -f <map_name>

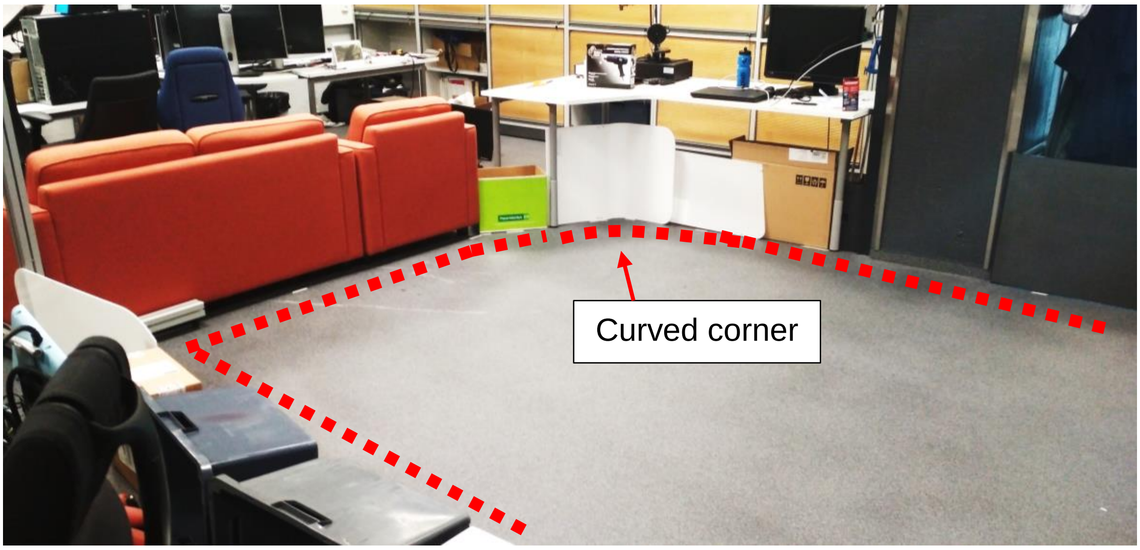


Figure 28. Self-made test arena with for map building process, side 1

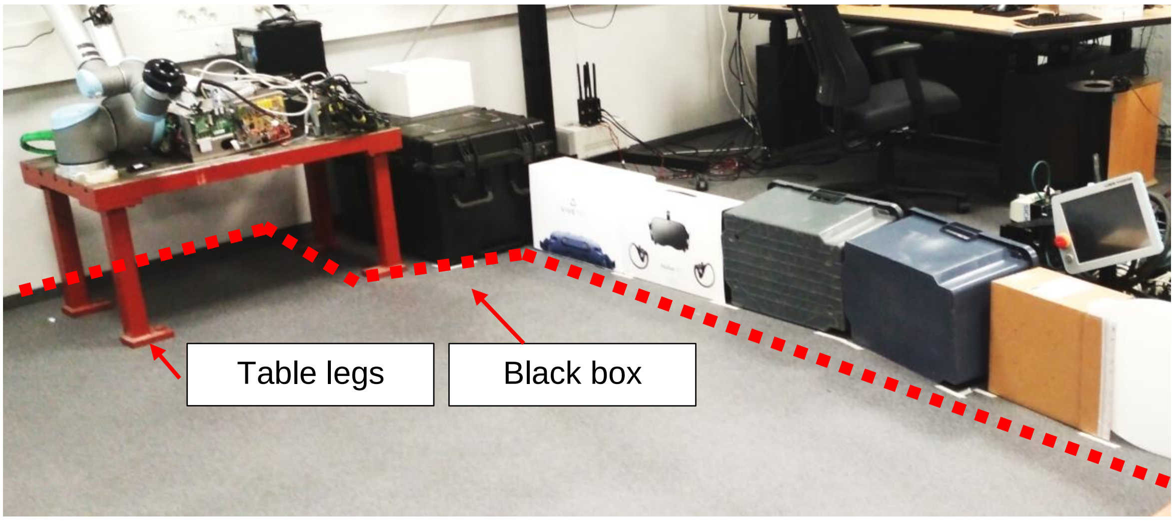


Figure 29. Self-made test arena with for map building process, side 2

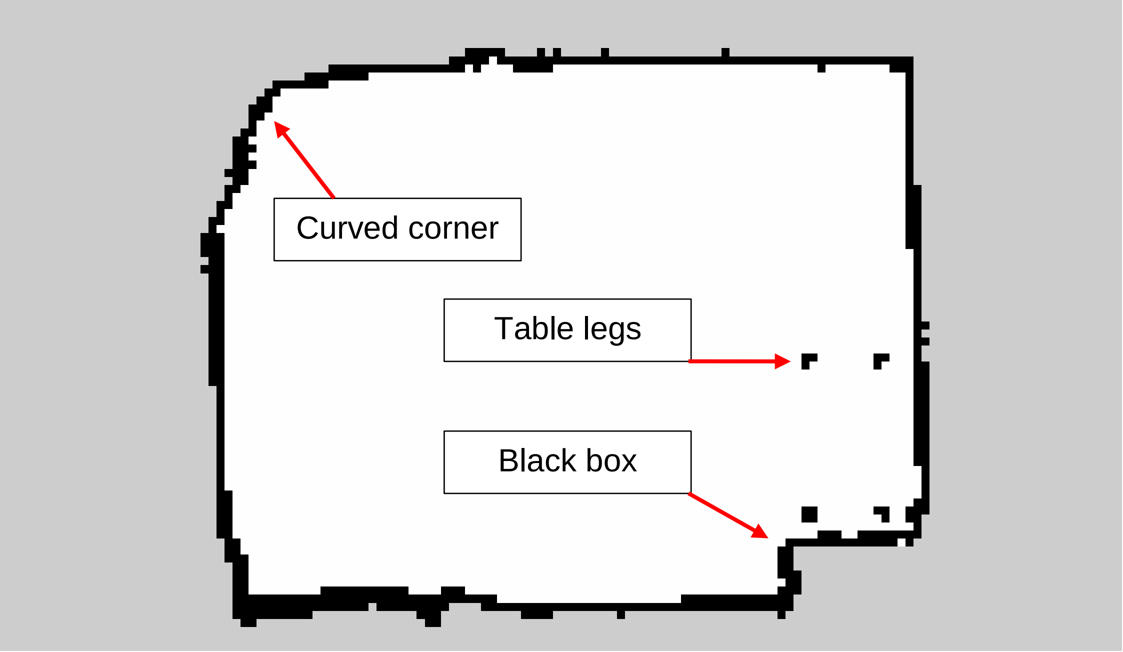


Figure 30. Map obtained with the help of LiDAR system and ROS

The final map obtained with the help of ROS and LiDAR sensor is presented in Figure 30. By comparing Figures 28 and 29 with the map in Figure 30, one can easily notice that all main features of the arena were successfully transferred to the map. Concluding, it can be said, that the map has relatively high precision and it can be used for navigation purposes.

6 Case study: preliminary designing of charging station

The challenge of the present case study was to make a concept design of a charging station, suitable for autonomous docking of TIERA robot. The study was done as an added contribution to the TIERA project.

The chapter describes the design process of the charging station, which was done according to Ullman's mechanical design process. Only specifications definition and concept design phases were conducted in the frames of the Bachelor's thesis project. The specifications definition phase covered the discovering of needs, requirements, and creation of specifications table. The concept design step implied concepts' generation, evaluation, and making the decision about the best concept (Ullman 2010, pp. 81-91).

6.1 Specifications definition

The specifications for a new charging system were defined based on the present equipment specifications, new needs, and safety regulations.

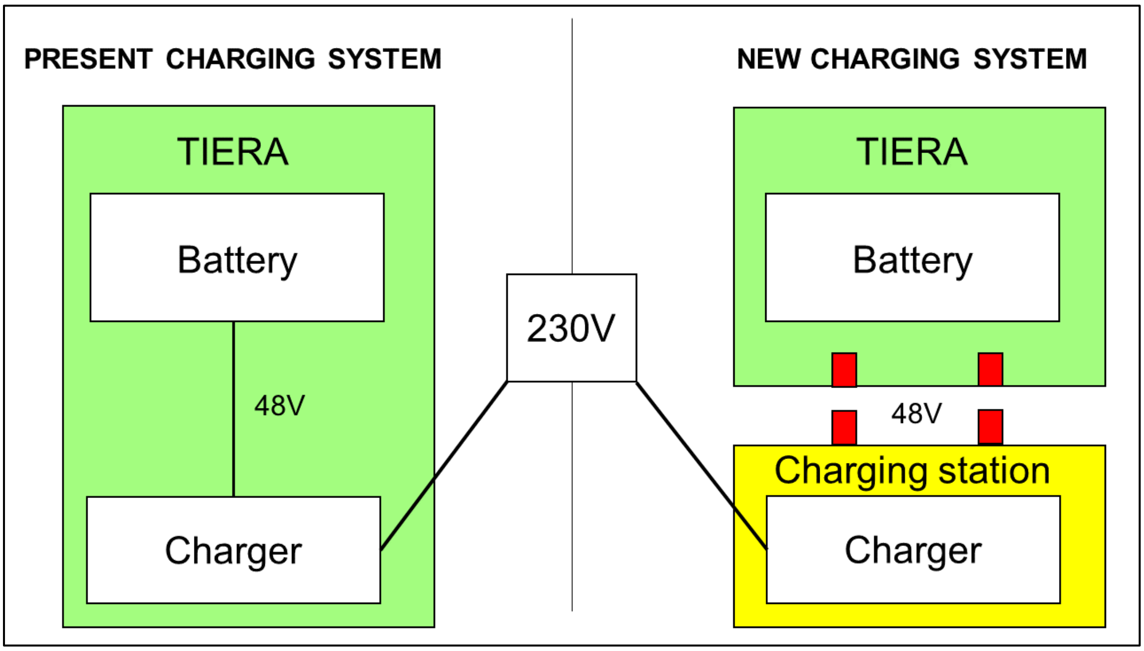


Figure 31. Modification of charging system

The needed modification of the charging system is presented in a simplified scheme in Figure 31. In present, the robot is charged with the help of a battery charger installed inside the robot. The charging plug must be manually inserted to 230V power supply whenever the charging is needed. As soon as the algorithm of autonomous charging will be implemented, the robot must be ready to handle the docking itself. In order to provide such a feature, the new charging station with charging contacts must be built.

As soon as the robot already had a suitable charging device GWL/Power Group. POW48V30A/BLI, it was decided to keep it as a base for a new charging station. Only new housing for the charging station and output charging plug had to be designed.

The summary of the defined specifications for the charging station of robot TIERA is presented in Table 5. The parameters of the charging device, such as input and output power, are listed in Table 5 in section “Battery charger”. Generally, the new charging station had to have a fixed mounting type. It could be either connected to the wall or ground or both. The requirement comes from the need to define in ROS *Navigation Stack* the exact location point of the charging station on the map of working space.

As was described before, the charging station for the autonomous robot must be designed in a way that the robot could easily connect to it. This requirement generates the specification for charging plug type. It should provide as much accuracy allowance during docking as possible.

In order to define safety requirements for new charging plug, the electrical safety standard SFS-EN 61140 + A1 “Protection against electric shock. Common aspects for installation and equipment” was taken into account.

Specification	Value / definition
Battery charger	
Model	GWL/Power Group. POW48V30A/BLI
Input	230V AC 50Hz
Output	48V DC 30A
General	
Mounting type	Fixed to ground/wall/(both)
Charging plug type	Any, providing high accuracy allowance during the docking
Isolation of contacts on steady-state	- *Hazardous-live-parts shall not be accessible and conductive parts shall not be hazardous live* (Finnish Standards Association SFS RY 2007, p.29). - *Placing out of arm's reach barrier may be appropriate to prevent … unintentional simultaneous access to conductive parts between which a hazardous voltage can exist* (Finnish Standards Association SFS RY 2007, p.33). - *Limitation of steady-state touch current and charge shall prevent persons or animals from being subjected to values of steady-state touch current and charge liable to be hazardous or perceptible* (Finnish Standards Association SFS RY 2007, p.35).

Table 5. Specifications for the charging station

6.2 Selection of the design concept

According to the technical specifications defined above, the only two parameters required the creation of technical solution during the design process. The following issues had to be solved for the TIERA charging station: isolation of contacts on steady-state, accuracy of the autonomous connection.

6.2.1 Handling the contacts' isolation issue

The isolation issue implies prevention of accidental contact of the human with charging contacts under the voltage. In case the contacts of the charging station under the voltage have an open excess, the passing by human can touch it unwittingly. The accident may cause an electric shock or even death. The small benchmarking research was conducted in order to solve the issue and create concept design models.

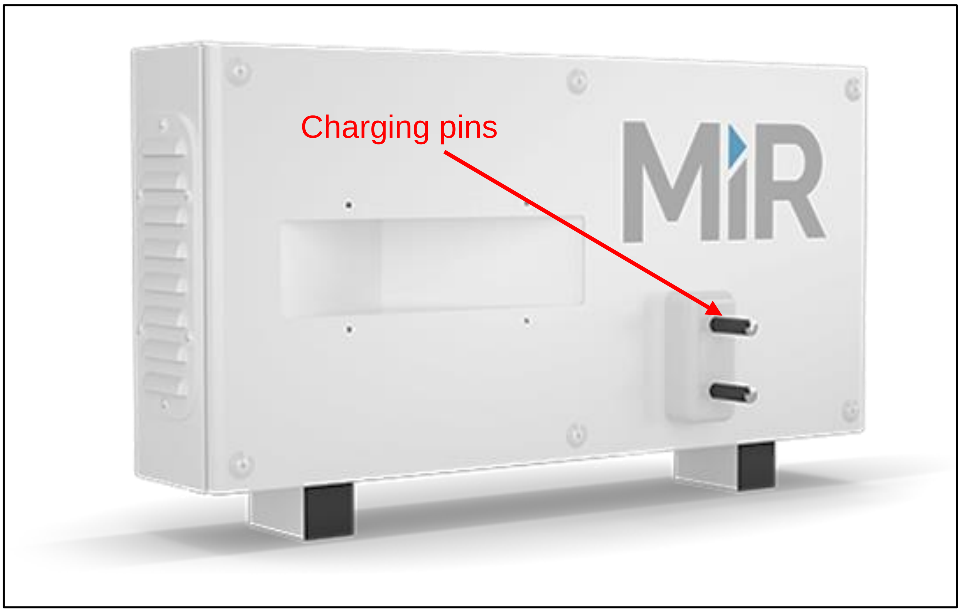


Figure 32. MIR100 charging station (MiR 2017)

The example of charging stations with solved isolation issue are presented in Figure 32. One can see the charging station of MiR100 with charging pins on the figure. The "male" connector is installed on the station and the robot has a "female" contact. The station of MiR100 charges with up to 20A only when both charging pins are pressed and the robot is connected (MiR 2017). In other words, the charging contacts are combined with a push-button switch. During the steady-state, the contacts stay in a normally open position. When the charging pins are pressed, the electrical circuit closes and the charging starts. This is a good solution that prevents accidents in case of light touch. However, there is still a possibility of harm, if the human presses the contacts with the force.

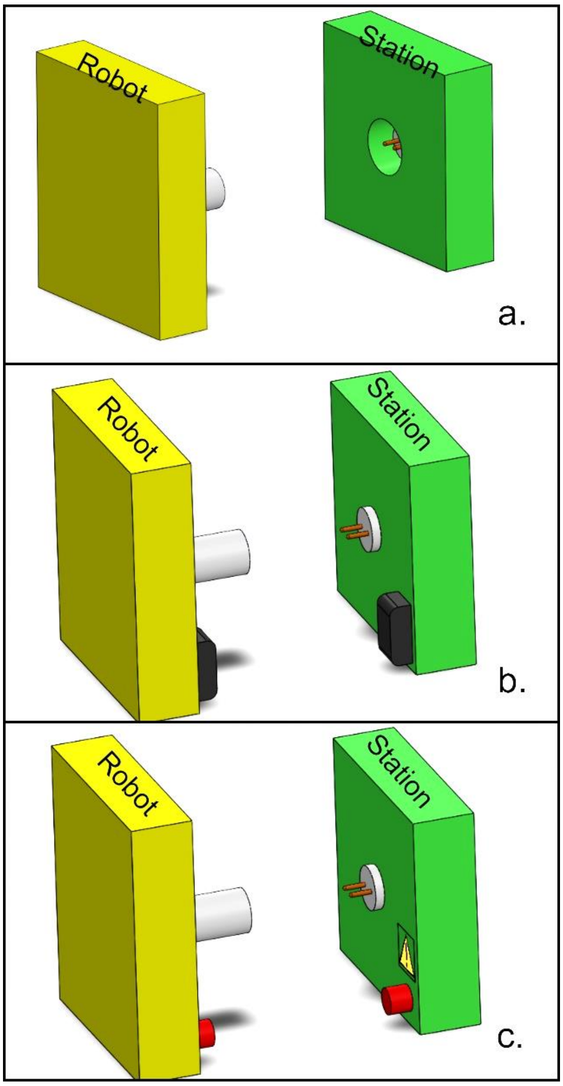


Figure 33. Possible concepts of voltage isolation on the station on steady-state (a.-placing out of arm's reach; b.-magnet key switch; c.-pushbutton switch)

The designed concepts for the solution of contacts isolation issue are presented in Figure 33. The first suggested idea is to put contacts under the voltage out of the human arm's reach. This means that voltage on contacts will not be switched off, but the hazardous parts will be hidden inside the station so that a human will not be able to touch it without a special device. The second and third idea represent systems with various types of switches.

6.2.2 Handling the docking accuracy issue

The accuracy issue covers the possibility of the mobile robot to connect to the charging station precisely and correctly. The carefulness of the connection is defined by the accuracy of movements that are controlled by the traction system.

The field tests of movement accuracy in autonomous mode could help with defining a technical specification for the type of connection. However, the charging station's designing process was conducted in parallel to the development of the autonomous positioning system. The field tests of the accuracy were not possible to perform before the beginning of the concept designing process.

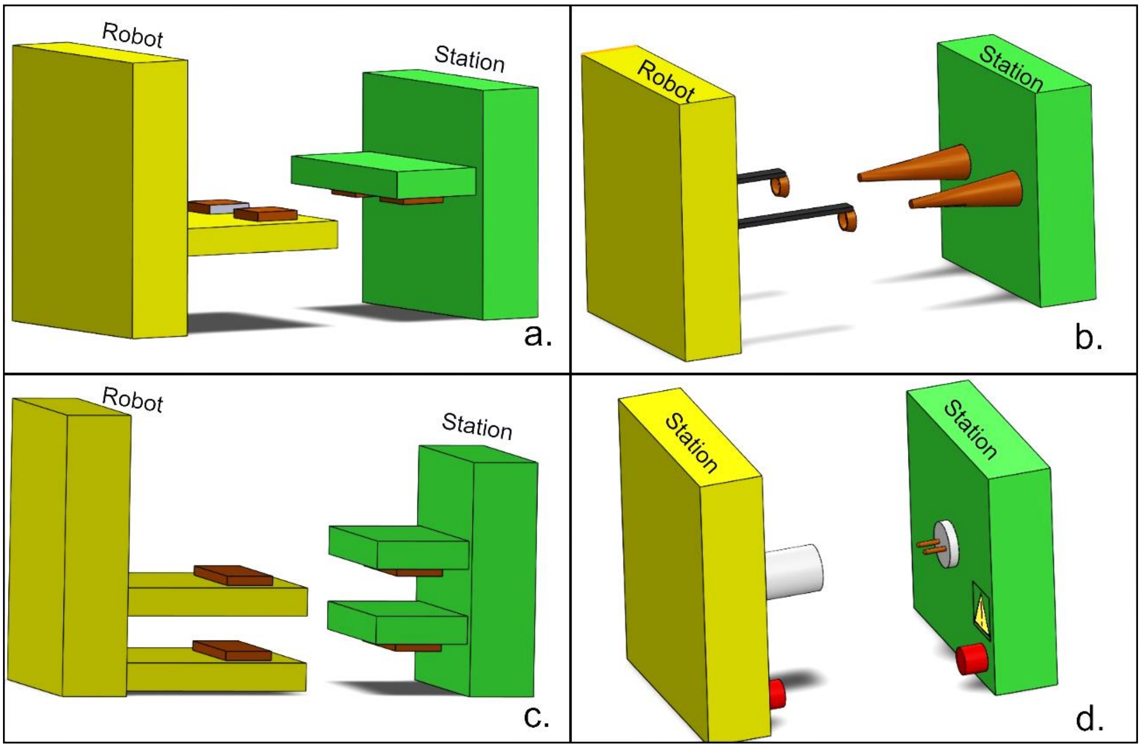
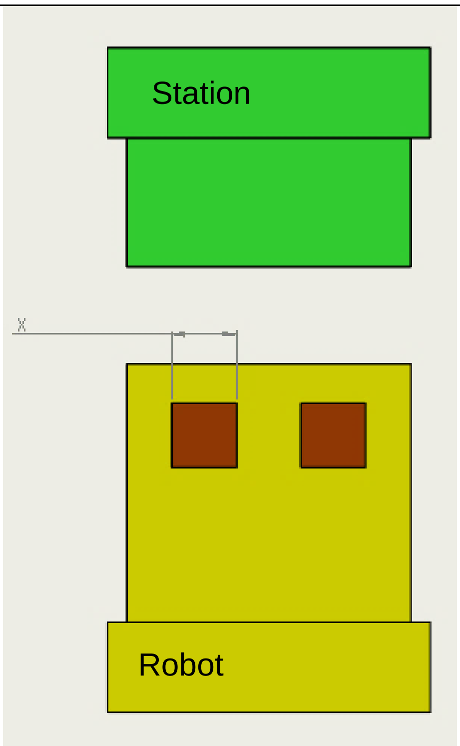
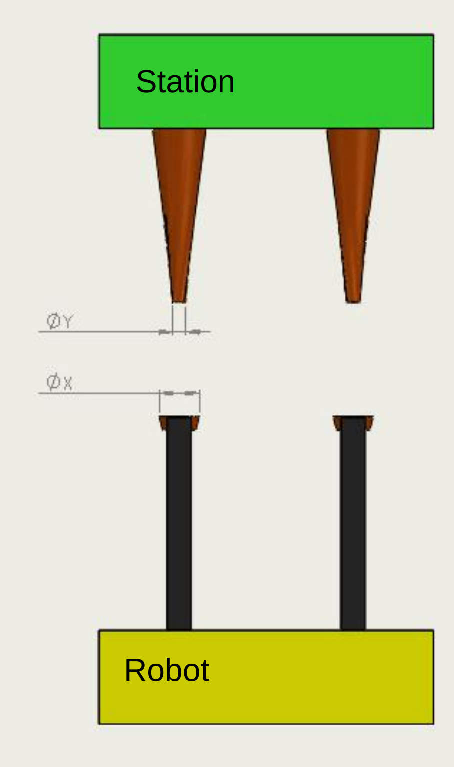


Figure 34. Possible concepts of solving docking accuracy issue (a.-flat contacts on the same level; b.-cone and ring contacts; c.-flat contacts on deferent levels; d.- female-male contacts)

The designed concepts for the solution of accurate docking issue can be seen in Figure 34. The Robot TIERA is represented by yellow block (left), the charging station – by green block (right), and charging contacts are painted brown.

Connection type	**Picture (top view)**	**Estimated accuracy restrictions**
Flat contacts on the same level		$\pm\frac{X}{2}$
Cone and ring contacts		$\pm\frac{\emptyset X - \emptyset Y}{2}$

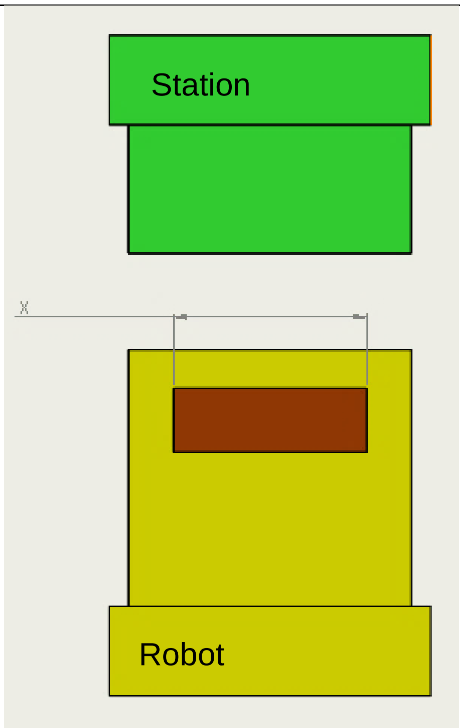
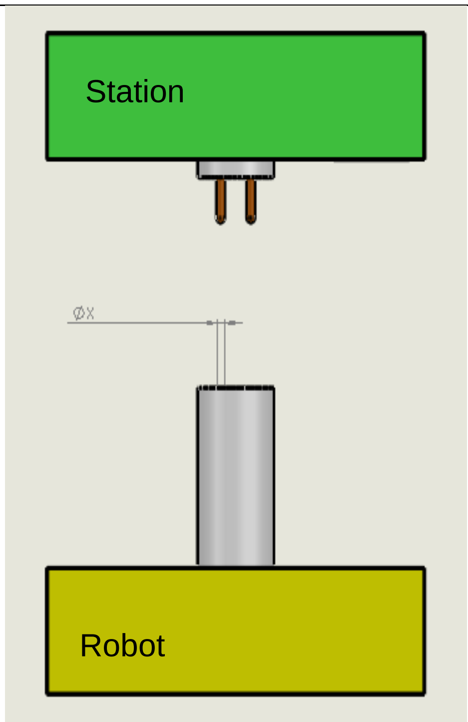
Flat contacts on deferent levels		$\pm\frac{X}{2}$
Female-male contacts		$\pm\frac{\emptyset X}{2}$

Table 6. Accuracy allowance estimation of designed concepts

The estimation of accuracy allowance during docking is shown in Table 6. The idea of given allowances was so that the robot should touch at least half of each charging contact. As can be seen from the Table, the concept with flat contacts on different levels has the biggest relative allowance.

6.2.3 BVOA analysis

As the design of the charging station must solve both accuracy and isolation issues, the concepts handling with two mentioned issues were combined as follows:

- The first concept had female-male charging contacts and a pushbutton switch.
- The second concept had flat contacts located on one plane and a magnetic key switch system.
- The third concept had flat charging contacts on different levels with a narrow space between plates so that human arm could not reach the contacts without extra devices.

The comparison of concepts and selection of the best one was conducted with the help of the Best Value Option Analysis (BVOA). The BVOA table can be seen in Appendix 8. The following criteria for the charging station were selected as the most important: solution of isolation issue, solution of docking accuracy issue. The need in implementation of extra systems was defined as criteria with middle importance. The least important criteria was the ease of development of the station by means of the research group.

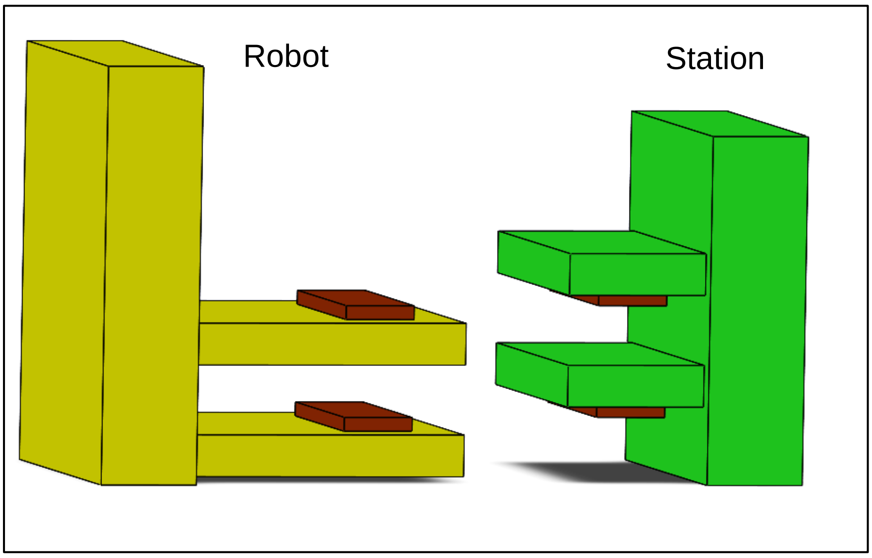


Figure 35. Selected concept of charging station

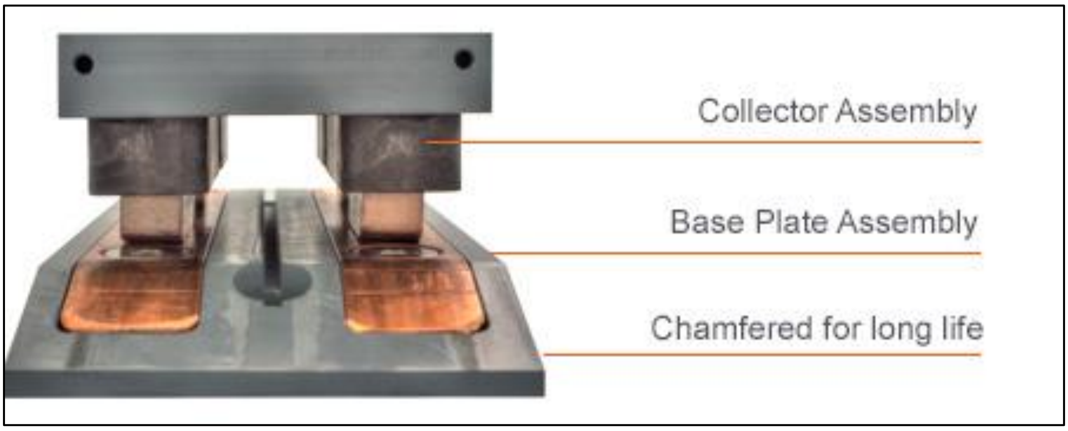


Figure 36. Charging contacts (Conductix-Wampfler 2019)

As a result of the analysis concept number 3, presented in Figure 35 was selected because it received the highest points in all criteria. Further comments on the selection process can be found in Appendix 8. Basically, the selected concept satisfies all safety requirements and it is the easiest to develop. To improve contact pressure, the station can be equipped with contacts with pressure springs, shown in Figure 36.

7 Conclusion

Concluding the results of the present thesis project, it can be said that all goals were achieved. The development and implementation of LiDAR platform based on ROS were successfully done. The 2D map of a self-made arena in the Laboratory of Intelligent Machines was built with relatively high precision. The integration and setup of the ROS Navigation Stack for the autonomous motion was successfully implemented. However, the ground tests of autonomous motion were not done in a proper way due to the time restrictions. The added value studies aimed to select the best conceptual design of the charging station were successfully done as an added contribution to TIERA project.

In addition to the achieved results, the large volume of new knowledge in multiple disciplines was obtained by the researcher during the project. For instance, the network communication principles, the ROS concepts, autonomous navigation approaches, and LiDAR's operational principles were studied from the very basic.

In order to finalize the process of implementation of an autonomous motion of TIERA robot, further study is required. First of all, the experiments with autonomous navigation from various positions must be done. Then the obstacle avoidance system of Navigation Stack must be activated and tested. Moreover, the project aimed to implement autonomous charging process requires conduction of accuracy tests during the autonomous motion, establishment of communication between the battery management system and ROS, and the development of the charging station.

References

Advantech 2012. User manual. ARK-3440 Fanless Embedded Box PC. Edition 2. Beijing: Advantech Co., Ltd.

Artigas, M. I. 2017. Development of Kinematics and Force controller for LUT Mobile Assembly Robot arm. Lappeenranta University of Technology. Degree Program in Mechanical Engineering. Master's Thesis.

Beinhofer, M., Müller, J., Burgard, W. 2013. Effective landmark placement for accurate and reliable mobile robot navigation. Robotics and Autonomous Systems 61(10), pp.1060-1069.

Belzunce, A. 2015. Development of a Control System for a Tele Operated Mobile Robot. Lappeenranta University of Technology. Degree Program in Mechanical Engineering. Master's Thesis.

Cardon, A., Mhamed, I. 2016. New Autonomous Systems: From Reliable Self-Control to Artificial Self. London: ISTE Ltd.

Conductix-Wampfler 2019. Charging Contacts for Charging Current Capacities from 20 amps to 600 amps. https://www.conductix.fi/en/products/charging-contacts. Accessed on 10 May 2019.

Davids, M. 2017. Five Ways Robots Perform Mundane Tasks Better Than Humans. https://blog.robotiq.com/5-ways-robots-perform-mundane-tasks-better-than-humans. Accessed on 10 April 2019.

Finnish Standards Association SFS RY 2007. Protection against electric shock. Common aspects for installation and equipment. SFS-EN 61140 + A1. Helsinki: Suomen standardisoimisliitto.

GLW Power 2015. User manual. BMS123 Smart Extended Module. Prague: GWL Power Ltd.

Gutmann, J.S., Konolige, K. 1998. An Experimental Comparison of Localization Methods. Proceedings 1998 IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS' 98).

Hesse, S. 2001. Sensors in Production Engineering. Esslingen: Festo AG & Co.

Ilon, B. E. 1972, Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base, US3876255A.

Joseph, L. 2015. Mastering ROS for Robotics Programming. Birmingham: Packt Publishing Ltd.

KUKA 2018. KMR iiwa: autonomous and extremely flexible. https://www.kuka.com/en-de/products/mobility/mobile-robots/kmr-iiwa. Accessed on 9 May 2019.

Labiod, H., Afifi, H., Santis, C. 2007. Wi-Fi TM, Bluetooth TM, Zigbee TM and WiMax TM. Netherlands: Springer.

Le, W. 2015. Design of traction transmission and suspension systems for an omni-directional mobile robot. Lappeenranta University of Technology. Degree Program in Mechanical Engineering. Master's Thesis.

LUT University, n.d. Research group of Intelligent Machines. https://www.lut.fi/web/en/school-of-energy-systems/research/intelligent-machines. Accessed on 4 April 2019.

MIR 2017. MiR Charge technical documentation. https://www.mobile-industrial-robots.com/media/2803/mircharge-user-guide-v-14.pdf. Accessed on 17 April 2019.

Martinez, A., Fernández,E. 2013. Learning ROS for Robotics Programming. Birmingham: Packt Publishing Ltd.

Pires, J.N. 2007. Industrial robots programming: building applications for the factories of the future. New York: Springer cop.

Poberezkin, E. 2017. Design and Development of Communication System for Mobile Robot. Lappeenranta University of Technology. Degree Program in Mechanical Engineering. Master's Thesis.

Reza, N., J. 2007. Theory of Applied Robotics. New York: Springer Science+Business Media, LLC.

Romanov, K. 2018. Development of an Omni Based Traction System for a Teleoperated Mobile Robot Utilizing ROS Platform. Lappeenranta University of Technology. Degree Program in Mechanical Engineering. Master's Thesis.

ROS 2018. Sensor_msgs Documentation. http://docs.ros.org/api/sensor_msgs/html/index.html. Accessed on 8 May 2019.

Röwekämper, J., Sprunk, C., Diego, G., Stachniss, C., Pfaff, P., Burgard, W. 2012. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. Proceedings 2012 IEEE/RSJ International Conference on Robotics and Automation. DOI: 10.1109/IROS.2012.6385988.

SICK 2012. Product Information. LMS5xx Laser Measurement Technology. https://www.sick.com/fi/en/detection-and-ranging-solutions/2d-lidar-sensors/lms5xx/lms511-10100-pro/p/p215941. Accessed on 19 April 2019.

SICK 2017. Operating Instructions. LMS5xx Laser Measurement Sensors. https://www.sick.com/fi/en/detection-and-ranging-solutions/2d-lidar-sensors/lms5xx/lms511-10100-pro/p/p215941. Accessed on 19 April 2019.

SICK 2018. Whitepaper. Lidar Sensor Functionality and Variants. https://www.sick.com/fi/en/detection-and-ranging-solutions/2d-lidar-sensors/lms5xx/lms511-10100-pro/p/p215941. Accessed on 19 April 2019.

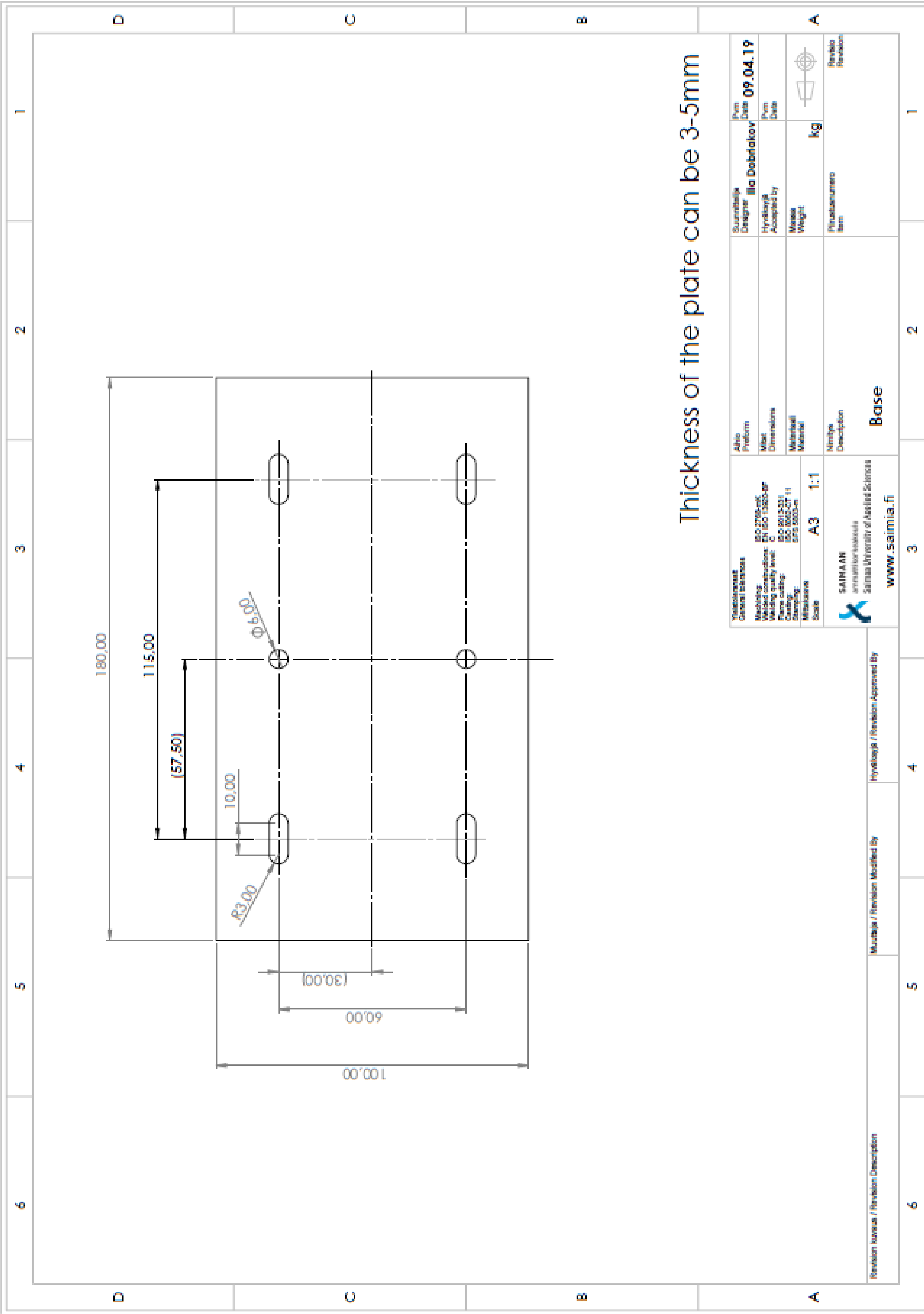
Siegwart, R., Nourbakhsh, I. 2004. Introduction to Autonomous Mobile Robots. Cambridge (MA): The MIT Press cop.

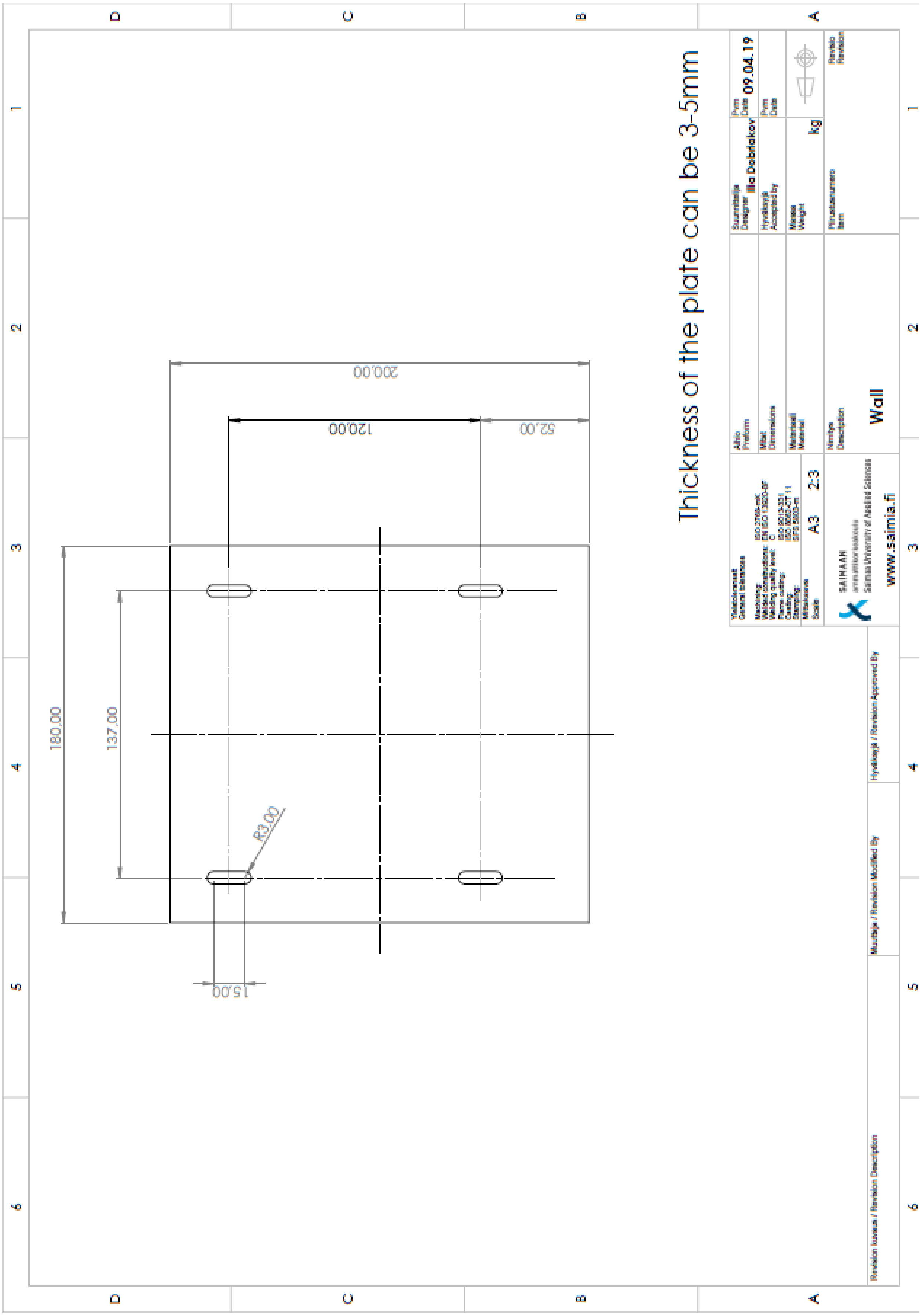
Silverman, M., C., Nies, D., Jung, B., Sukhatme G., S. 2002. Staying alive: a docking station for autonomous robot recharging. Proceedings 2002 IEEE International Conference on Robotics and Automation. DOI: 10.1109/ROBOT.2002.1013494.

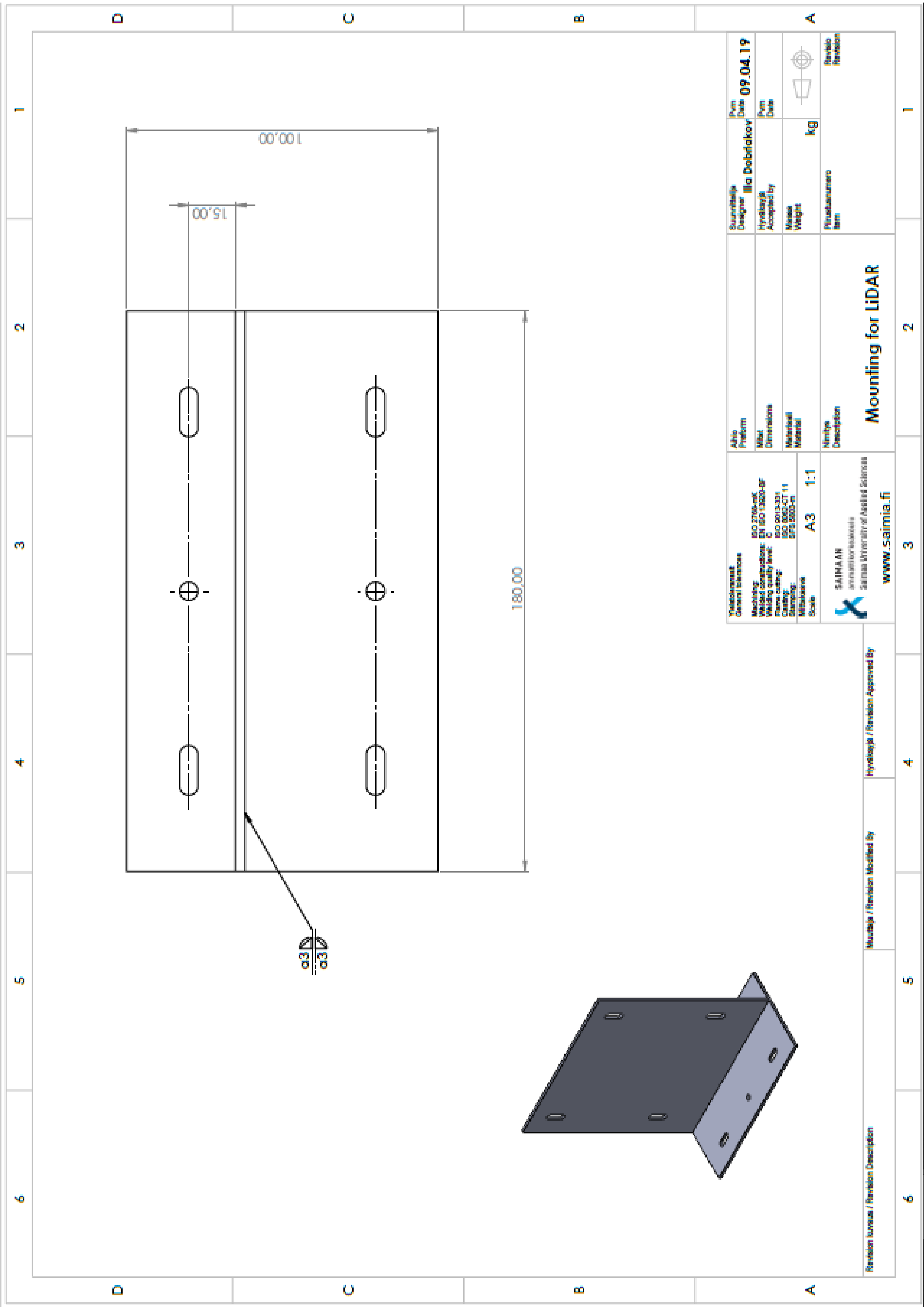
Stäubli 2017. HelMo the mobile robot system. https://www.staubli.com/en-mx/robotics/product-range/mobility/helmo-mobile-robot-system/. Accessed on 9 May 2019.

Ulman, D. 2010. The mechanical design process. 4th ed. New York: McGraw-Hill cop.

Manufacturing drawings for LiDAR mounting




Thickness of the plate can be 3-5mm
200,00
120,00
52,00
180,00
137,00
R3,00
15,00
Ilia Dobriakov
09.04.19
kg
Wall
A3
2-3
www.saimia.fi


100,00
15,00
180,00
a3
a3
Ilia Dobriakov
09.04.19
kg
Mounting for LiDAR
A3
1:1
SAIMAAN
www.saimia.fi

Raw definition of LaserScan message (ROS 2018)

```
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

Header header            # timestamp in the header is the acquisition time of
                         # the first ray in the scan.
                         #
                         # in frame frame_id, angles are measured around
                         # the positive Z axis (counterclockwise, if Z is up)
                         # with zero angle being forward along the x axis

float32 angle_min        # start angle of the scan [rad]
float32 angle_max        # end angle of the scan [rad]
float32 angle_increment  # angular distance between measurements [rad]

float32 time_increment   # time between measurements [seconds] - if your scanner
                         # is moving, this will be used in interpolating position
                         # of 3d points
float32 scan_time        # time between scans [seconds]

float32 range_min        # minimum range value [m]
float32 range_max        # maximum range value [m]

float32[] ranges         # range data [m] (Note: values < range_min or > range_max should be discarded)
float32[] intensities    # intensity data [device-specific units].  If your
                         # device does not provide intensities, please leave
                         # the array empty.
```

Source code tf_broadcaster.cpp

```
#include <ros/ros.h>
 #include <tf/transform_broadcaster.h>

 int main(int argc, char** argv){
    ros::init(argc, argv, "robot_tf_publisher");
    ros::NodeHandle n;
    //Set the frequency to send the data
    ros::Rate r(100);

    tf::TransformBroadcaster broadcaster;

    while(n.ok()){
          //TransformBroadcaster arguments:
          //1-rotation transform between two frames
          //2-vector for translation between frames
          //3-timestamp
          //4-name of the parent node
          //5-name of the child node
          broadcaster.sendTransform(
          tf::StampedTransform(
          tf::Transform(tf::Quaternion(0, 0, 0, 1),
          tf::Vector3(0.8, 0.0, 0.0)),
          ros::Time::now(),
          "base_link",
          "base_laser"));

          r.sleep();
          }

}
```

Source code of modified normal_callback function in lut_controller.cpp

```
void LutMars::normal_callback(lut_track::joy_xy xy_read){

          //<skipped code part: initialization of variables>

  if(ping_state !=0){          //There is connection

          //<skipped code part: sending messages to terminal
          about system state>

    switch(state_test) {   //check system state(1-working, 0-not)
      case 1:         //in case system state is 1 do following

          //<skipped code part: writing velocities read from
          joystick to xy_read>

          //making a request message for Kinematics server
          srv.request.X = xy_read.x;
          srv.request.Y = xy_read.y;
          srv.request.TWIST = xy_read.angle;

          if (client.call(srv))//call Kinematics server
          {
            //print velocity variables with received values from
            //Kinematics server
            ROS_INFO("Command 1: %f", srv.response.n1);
            ROS_INFO("Command 2: %f", srv.response.n2);
            ROS_INFO("Command 3: %f", srv.response.n3);
            ROS_INFO("Command 4: %f", srv.response.n4);
            std::cout<<"n1:"<<srv.response.n1<<std::endl;
            std::cout<<"n2:"<<srv.response.n2<<std::endl;
            std::cout<<"n3:"<<srv.response.n3<<std::endl;
            std::cout<<"n4:"<<srv.response.n4<<std::endl;
            //set velocity variables with received values from
            //Kinematics server
            vel.data=srv.response.n1;
            vel2.data=srv.response.n2;
            vel3.data=srv.response.n3;
            vel4.data=srv.response.n4;

            //convert n received in rpm to w(rad/s)
            w1=(srv.response.n1*2*3.1415)/60;//ADDED BY ILIA
            w2=(srv.response.n2*2*3.1415)/60;//ADDED BY ILIA
            w3=(srv.response.n3*2*3.1415)/60;//ADDED BY ILIA
            w4=(srv.response.n4*2*3.1415)/60;//ADDED BY ILIA
            //calculate vx(m/s), vy(m/s), vth(rad/s)
            vx=R*(w1-w2-w3+w4)/4;                //ADDED BY ILIA
            vy=R*(w1+w2-w3-w4)/4;                //ADDED BY ILIA
            vth=R*(-w1-w2-w3-w4)/(4*(L+l));      //ADDED BY ILIA

          }else{ //if request to Kinematics server failed

            ROS_ERROR("Failed to call service Kinematics");
```

```
            vel.data=0;
            vel2.data=0;
            vel3.data=0;
            vel4.data=0;
          }
          //print current velocity to Terminal
          std::cout<<"Velocity is "<<vel.data<<std::endl;

        if (prev_vel!=vel.data) { //if velocity changed
            std::cout<<"Velocity changed to
            "<<vel.data<<std::endl<<std::endl;
            // publish messages of vel-vel4 with new values
            vel_pub_1.publish(vel);
            vel_pub_2.publish(vel2);
            vel_pub_3.publish(vel3);
            vel_pub_4.publish(vel4);
            prev_vel=vel.data;
        } else {              //if velocity didn't change
            std::cout<<"Velocity didn't
            change"<<std::endl<<std::endl;
        }
      }
      break;
    case 0:          //in case system state is 0 do the following
      vel.data = 0;
      vel2.data = 0;
      vel3.data=0;
      vel4.data=0;
      std::cout<<"Velocity is "<<vel.data<<std::endl;
      // publish messages of vel-vel4 set to zero
      vel_pub_1.publish(vel);
      vel_pub_2.publish(vel2);
      vel_pub_3.publish(vel3);
      vel_pub_4.publish(vel4);
      prev_vel=vel.data;
    break;
    }// END SWITCH

  }else{ //There is no connection

          //<skipped code part: publish messages of vel-vel4 set
          to zero>
  }//end connection ckeck
}//end normal_callback
```

Source code of modified main function in lut_controller.cpp

```
int main(int argc, char** argv)
{
  ros::init(argc, argv, "lut_controller");
  ros::init(argc, argv, "odometry_publisher");    //ADDED BY ILIA
  ros::NodeHandle n;                              //ADDED BY ILIA
  ros::Publisher odom_pub = n.advertise<nav_msgs::Odome
try>("odom", 50);                                 //ADDED BY ILIA
  tf::TransformBroadcaster odom_broadcaster;      //ADDED BY ILIA

  double x = 0.0;                                 //ADDED BY ILIA
  double y = 0.0;                                 //ADDED BY ILIA
  double th = 0.0;                                //ADDED BY ILIA

  ros::Time current_time, last_time;              //ADDED BY ILIA
  current_time = ros::Time::now();                //ADDED BY ILIA
  last_time = ros::Time::now();                   //ADDED BY ILIA
  LutMars lut_controller;

  ros::Rate r(60);
  while(ros::ok()){
          ros::spinOnce();
          //START OF CODE ADDED BY ILIA
          current_time = ros::Time::now();
          //calculate odometry using velocities of robot
          double dt = (current_time - last_time).toSec();
          double delta_x = (vx * cos(th) - vy * sin(th)) * dt;
          double delta_y = (vx * sin(th) + vy * cos(th)) * dt;
          double delta_th = vth * dt;
          //add calculated odometry values to current position
          x += delta_x;
          y += delta_y;
          th += delta_th;
          //since all odometry is 6DOF we'll need a quaternion
          // created from yaw
          geometry_msgs::Quaternion odom_quat = tf::createQua-
ternionMsgFromYaw(th);
          //publish the transform over tf
          geometry_msgs::TransformStamped odom_trans;
          odom_trans.header.stamp = current_time;
          odom_trans.header.frame_id = "odom";
          odom_trans.child_frame_id = "base_link";
          odom_trans.transform.translation.x = x;
          odom_trans.transform.translation.y = y;
          odom_trans.transform.translation.z = 0.0;
          odom_trans.transform.rotation = odom_quat;
          //send the transform
          odom_broadcaster.sendTransform(odom_trans);
          //publish the odometry message over ROS
          nav_msgs::Odometry odom;
          odom.header.stamp = current_time;
          odom.header.frame_id = "odom";
          //set the position
```

```
            odom.pose.pose.position.x = x;
            odom.pose.pose.position.y = y;
            odom.pose.pose.position.z = 0.0;
            odom.pose.pose.orientation = odom_quat;
            odom.child_frame_id = "base_link";


            //set the velocity
            odom.twist.twist.linear.x = vx;
            odom.twist.twist.linear.y = vy;
            odom.twist.twist.angular.z = vth;
            //publish the message
            odom_pub.publish(odom);
            last_time = current_time;
            //END OF CODE ADDED BY ILIA
r.sleep();
}

}
```

Dependency graph of TIEARA frames required for Navigation Stack

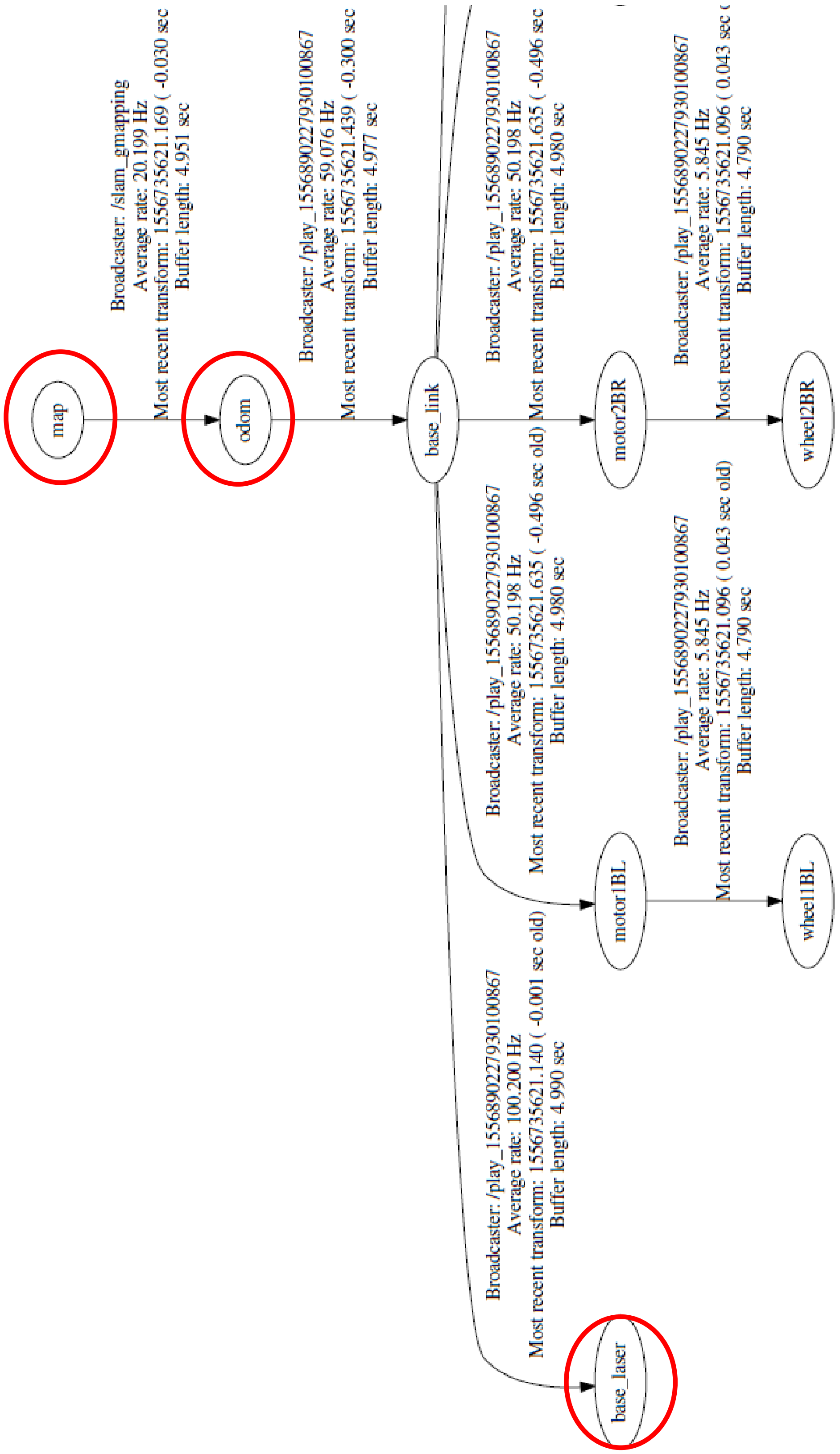
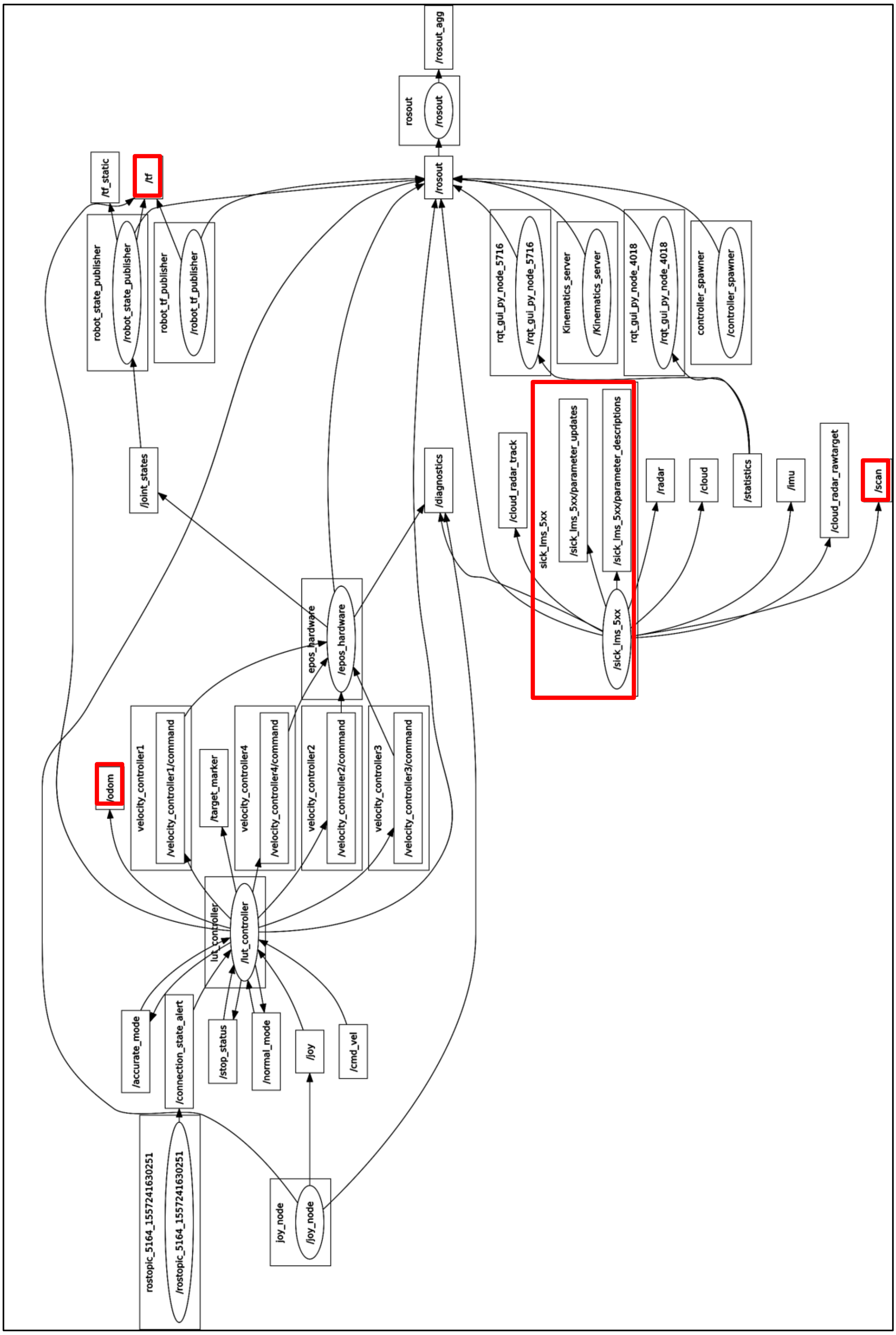


Diagram of package dependencies during mapping process



Best Value Option Analysis of charging station concepts

Criteria	Weight (1-3)	Female-male contacts, pushbutton switch		Flat contacts on same level, magnetic key switch		Flat contacts on difr. level, out of arms reach	
		Comment	Points (1-3)	Comment	Points (1-3)	Comment	Points (1-3)
Solution of isolation issue	3	The one can touch the contacts and push the button simultaneously	2	Voltage is not supplied on contacts until the robot comes to station	3	Human arm can not reach the contacts due to narrow space between contacts	3
Solution of docking accuracy issue	3	No assurance that TIERA can move precise enough	1	The allowance of docking is relatively high	2	The allowance of docking is the highest	3
Need in additional systems implementation	2	Additional sensing system of pattern recognition is needed	1	Integration of magnetic key identification system is needed	2	No need	3
Ease of development	1	The implementation of new systems requires a additional research and tests	1	The implementation of new systems requires a additional research and tests	1	The station can be completely built by research team	3
	TOTAL		12		20		27