

ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING

**THE HYBRID MOBILE MANIPULATOR
COMBINING SOFT AND HARD ROBOTS**

SENIOR DESIGN PROJECT

Berk GÜLER
040150447

Control and Automation Engineering

Thesis Advisor: Prof. Dr. Hakan TEMELTAŞ

JULY 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK ELEKTRONİK FAKÜLTESİ

**YUMUŞAK VE SERT ROBOTLARIN BİRLEŞİMİYLE
GELİŞTİRİLEN HİBRİT MOBİL MANİPÜLATÖR**

BİTİRME TASARIM PROJESİ

Berk GÜLER
040150447

Kontrol ve Otomasyon Mühendisliği

Tez Danışmanı: Prof. Dr. Hakan TEMELTAŞ

TEMMUZ 2020

FOREWORD

This thesis was written as a result of completion of the undergraduate program in Control and Automation Engineering, Istanbul Technical University. I would like to thank my supervisor Prof. Dr. Hakan Temeltaş for his guidance and the opportunity to work in the ITU Robotics Laboratory. I would also like to thank RA Osman Ervan for his time and precious insights shared with me.

In addition, I would like to thank Assoc. Prof. Dr. Tufan Kumbasar for allowing me to use the facilities of the ITU Artificial Intelligence and Intelligent Systems (AI²S) Laboratory.

Finally, and most importantly, I would like to express my gratitude to my family. Without their endless support, I could not have achieved to obtain this education. Especially I would like to thank my sister for always being supportive and being a guide. Last but not least, I thank to my dear friends. I want to thank Pınar for motivation and inspiring talks.

July 2020

Berk Güler

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	v
TABLE OF CONTENTS	vii
ABBREVIATIONS	ix
SYMBOLS	x
LIST OF TABLES	xii
LIST OF FIGURES	xii
SUMMARY.....	xvi
ÖZET	xix
1. INTRODUCTION.....	1
2. SOFTWARE ARCHITECTURE	5
2.1 Robot Operating System (ROS).....	5
2.1.1 Core components	6
2.1.1.1 Communications infrastrucute	6
2.1.1.1 Robot-Specific features	6
2.1.1.1 Tools.....	7
2.1.2 General concepts.....	8
3. HARDWARE AND SYSTEM DESIGN	10
3.1 Technical Specifications of the HUSKY UGV.....	11
3.2 Design and Manufacturing of the Serial Manipulator.....	13
3.2.1 The link between the manipulator and the mobile platform	14
3.2.2 The base link	15
3.2.3 The shoulder link.....	16
3.2.4 The elbow link.....	17
3.2.5 The wrist link	18
3.2.6 The soft gripper	20
3.2.7 The electronic circuit.....	24
3.3 Camera and Sensors Specifications.....	27
3.3.1 Camera	27
3.3.2 LIDAR sensor	28
3.3.3 The Sensor Hub.....	29
3.2.4 Other sensors on the mobile platform.....	30
3.2.5 Pressure and Force Sensor	30
4. KINEMATIC ANALYSIS OF THE SERIAL MANIPULATOR.....	32
4.1 Forward Kinematic.....	33
4.2 Inverse Kinematic.....	37
5. AUTONOMOUS NAVIGATION AND MISSION.....	41
5.1 ROS Navigation Stack.....	41
5.2 Mapping.....	42
5.3 Localization.....	44
5.4 Path Planning	45

5.5 Mission	46
5.6 Marked Box Recognition	46
5.7 Algorithm	50
6. CONCLUSION	53
REFERENCES	55
CURRICULUM VITAE.....	63

ABBREVIATIONS

AMCL	: Adaptive Monte Carlo Localization
DATMO	: Detection and Tracking of Moving Objects
DH	: The Denavit Hartenberg
EKF	: Extended Kalman Filter
FEM	: Finite Element Modelling
GUI	: Graphical User Interface
IMU	: Inertial Measurement Unit
LIDAR	: Light Detection and Ranging
UGV	: Unmanned Ground Vehicle
URDF	: Unified Robot Description Format
PWM	: Pulse Width Modulation
ROS	: Robot Operating System
SLAM	: Simultaneous Localization and Mapping
TF	: The Transform Library
XML	: Extensible Markup Language

SYMBOLS

W_s, W_e, W_w	: Weight of the links
W_L	: Load
M_e, M_w, M_s	: Sum of moments of the joints
D_e, D_w	: Weight of the motors
L_s, L_e, L_w	: Length of the links
L	: Sum of the links
P	: Pressure
R	: Rotation matrix
H	: Homogenous transformation matrix
d	: Distance vector
$\theta_1, \theta_2, \theta_3, \theta_4$: Angles of rotation of the joints
a_i, d_i	: Distance parameters of DH convention
α_i, θ_i	: Angle parameters of DH convention
A	: Homogenous transformation matrix as a product of transformations
T_5^0	: The complete transformation from the base_footprint to the gripper
D, H, L, D', H'	: Parameters for inverse kinematic analysis
h_0	: Elevation of the base link over the ground plane
x_c, y_c	: The position of the target object
L_1, L_2, L_3	: Length of the links
x_a	: The position of the box on the x-axis by ar_track_alvar
y_a	: The position of the box on the y-axis by ar_track_alvar
x_d	: The position of the box on the x-axis by DATMO
y_d	: The position of the box on the y-axis by DATMO
D_{th}	: Diameter of the threshold circle

LIST OF TABLES

	<u>Page</u>
Table 3.1 : Technical Specifications of Husky UGV.	11
Table 3.2 : List of the Actuators.....	12
Table 3.3 : Adjusted Technical Specifications of the C270 Webcam for Simulation Environment	28
Table 3.4 : Adjusted Technical Specifications of the Hokuyo UTM-30LX LIDAR for Simulation Environment	29
Table 3.5 : Operating Characteristics of the MPX4250AP differential pressure sensor.	31
Table 4.1 : DH Parameters.	36

LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : Visualization of TF frames in the RVIZ interface.....	7
Figure 2.2 : Visualization of Camera, LIDAR, and navigation messages in the RVIZ8	8
Figure 2.3 : Asynchronous communication in ROS.....	8
Figure 3.1 : Husky UGV at the ITU Robotics Laboratory.....	10
Figure 3.2 : Force diagram of the serial manipulator.	13
Figure 3.3 : The design of the aluminum platform carrying the manipulator.....	15
Figure 3.4 : The base link with ball-bearings (a) implementation, (b) the fixed part design, (c) the second part design.	16
Figure 3.5 : The shoulder link (a) implementation, (b) design.....	17
Figure 3.6 : The elbow link (a) implementation, (b) design.	18
Figure 3.7 : The wrist link and soft fingers.....	19
Figure 3.8 : The view of the wrist link from different angles.	19
Figure 3.9 : The designs of the pneunet with cellular membrane for fast actuation.	20
Figure 3.10 : The simulation results: (a) 3D model of the PneuNet in the Abaqus, (b) 3D model under gravity load, (c) displacement on the Y-axis in a second under gravity load, (d) 3D model under 20 kPa pressure, (e) displacement on the Y-axis in two seconds under 20 kPa pressure.	21
Figure 3.11 : The simulation results: (a) 3D model under 30 kPa pressure (b) displacement on the Y-axis in two seconds under 30 kPa pressure.....	22
Figure 3.12 : The graph of the equation 3.4.....	23
Figure 3.13 : The 3D model of the hybrid mobile manipulator while grasping the tennis ball.....	23
Figure 3.14 : Electronic components: (a) MG946r servo motor, (b) MG958 servo motor, (c) 6V micro air pump motor, (d) Arduino Uno, (e) L298N dual h-bridge motor driver, (f) 3V 2-way micro solenoid valve.	24
Figure 3.15 : Variable Pulse width control servo position.....	25
Figure 3.16 : 2-Way Valve Symbols	27
Figure 3.17 : Logitech C270 HD Webcam.	27
Figure 3.18 : Hokuyo UTM-30LX LIDAR.....	28
Figure 3.19 : The Sensor Hub.	29
Figure 3.20 : MPX4250AP differential pressure sensor.	30
Figure 3.21 : RP-L-170 thin-film pressure sensor.	31
Figure 4.1 : Coordinate frames attached to the serial manipulator.....	34
Figure 4.2 :Initial configuration ($\theta=[0,0,0,0,0]^T$) of the serial manipulator in the simulation environment	37
Figure 4.3 : Projection of the desired end-effector center onto the x-y plane.....	38
Figure 4.4 : The wrist link is parallel to the ground.	39
Figure 4.5 : The wrist link is perpendicular to the ground.....	40
Figure 5.1 : ROS Navigation Stack.	42

Figure 5.2 : a) World Model in Gazebo Simulation Environment, 2D Occupancy Grid Map by using b) gmapping, c) Google Cartographer.	43
Figure 5.3 : Expected Robot Behaviour.	45
Figure 5.4 : Marked Boxes with AR tags.	46
Figure 5.5 : Visualization of the camera, ARtag and L-shape extraction in RVIZ...	47
Figure 5.6 : Adaptive Breakpoint Detector Algorithm.....	48
Figure 5.7 : L-shape Extraction from Rectangle.....	49
Figure 5.8 : Flow Chart of the Autonomous Visual Fiducial Marker Recognition Algorithm	50
Figure 5.9 : Flow Chart of the Approaching the Visual Fiducial Marker and Manipulation of the Object Algorithms	51

THE HYBRID MOBILE MANIPULATOR COMBINING SOFT AND HARD ROBOTS

SUMMARY

Soft Robotics is the sub-field of robotics science, which includes robots that use materials compatible with living organisms. Soft robots, a new subject, is a rapidly developing branch with the unknowns. Universities and research laboratories all over the world conduct detailed research on this subject. There are robots with a completely soft structure and hybrid robots, some parts of which are combined with hard parts. Soft robots are harmless in interacting with the environment.

In this project, a hybrid design was made and a serial manipulator with a soft gripper at the end was designed and produced. The resulting serial manipulator was combined with the Husky mobile platform manufactured by Clearpath in ITU Robotics Laboratory. If it is desired to draw attention to the importance of this combination, it contains hard bodies like most robots today defined as robots and also a soft body compatible with its environment. The advantage of conventional robots, speed and accuracy, and the advantage of the interaction of soft robots with their environment are combined.

Another point to note for the hybrid mobile manipulator is the cost of construction. A detailed design has been made with the most optimal and accessible parts. The serial manipulator is designed according to the torque of its motors and is compatible for the soft gripper. However, due to its structure, the workspace and the positioning of the camera were not very preferable. This added a different challenge in addition to the software part. At the same time, modeling and control of soft robots is very difficult compared to hard-bodied robots. In this study, this difficulty was also struggled with.

It works autonomously with the sensors and software on the hybrid mobile manipulator working as a task. In addition, it was used in the visual reference system, ARtag, as a guide when interacting with other objects. In this way, whichever object it is desired to interact with, it has been introduced before and has enabled the task to navigate autonomously.

It is aimed to contribute to the soft robotics in the field of mobile manipulator. The main reason why the designed robot is hybrid is that while rigid links are sufficient in terms of speed and accuracy, the errors that occur during the autonomous approach to the target can be compensated with the soft gripper. While grasping with a rigid gripper, the mistakes made may damage the target object, but it is not possible in this task performed using a soft gripper.

The experimental setup is planned to be physical. However, the course of the project was towards the direction of simulation due to the epidemic situation. The biggest disadvantage of working in simulation, simulation environments developed for normal robots are not suitable for soft robots. The software required to run the entire task in the same simulation environment and by providing feedback to each other is currently not available or are under development. However, with the finite element method, simulations on how any soft robot will move under pressure have been made and the serial manipulator has been designed accordingly.

Thanks to autonomous navigation and inverse kinematic analysis, the hybrid manipulator can perform the defined task without any intervention. During autonomous navigation, visual reference system and algorithm of L-shaped extraction from LIDAR data were also used. Defined processes are carried out by making the developed software and mobile platform compatible with the hardware and software. The combination of the soft robot and the hard robot with each other under the head of the mobile manipulator has been successful. However, in the simulation environment, only inverse kinematic analysis was performed in the manipulation stage of the hybrid mobile manipulator in line with the information obtained from the soft robot analysis performed in different environment. Under the roof of higher processing power and more suitable software, this process can be performed.

YUMUŞAK VE SERT ROBOTLARIN BİRLEŞİMİYLE GELİŞTİRİLEN HİBRİT MOBİL MANİPÜLATÖR

ÖZET

Yumuşak Robotik, yapısında canlı organizmalar ile uyumlu malzemeler bulunan robotları kapsayan robotik biliminin altalanıdır. Yeni bir konu olan yumuşak robotlar, içerdiği bilinmezlikler ile hızlı bir şekilde gelişmekte olan bir daldır. Dünyanın dört bir yanında bulunan üniversiteler, araştırma laboratuvarları bu konu hakkında detaylı araştırmalar yapmaktadırlar. Tüm gövdesi yumuşak bir yapıdan oluşan robotlar ve bazı kısımları rijit parçalarla birleştirilmiş yumuşak robotlar bulunmaktadır. Yumuşak robotlar, çevreyle etkileşimde zararsızlardır.

Bu projede hibrit bir tasarım yapılarak, ucunda yumuşak bir gripper bulunan seri manipülatör tasarlanıp, üretilmiştir. Ortaya çıkan bu seri manipülatör ile İTÜ Robotik Laboratuvarı'nda bulunan, Clearpath tarafından üretilmiş Husky mobil platformu ile birleştirilmiştir. Bu birleşimin önemine dikkat çekilmek istenirse, günümüzde robot olarak tanımlanan çoğu robot gibi sert gövdeleri ve aynı zamanda çevresiyle uyumlu yumuşak bir gövdeyi barındırmaktadır. Normal robotların avantajı olan hız ve doğruluk ile yumuşak robotların çevresi ile olan etkileşiminin avantajı birleştirilmiştir.

Hibrit mobil manipülatör için dikkat çekilmesi gereken bir konu ise, yapım maliyetidir. Olabilecek en optimal ve ulaşılabilir parçalar ile detaylı bir tasarım yapılmıştır. seri manipülatör, motorlarının torkları doğrultusunda tasarlanmıştır ve yumuşak gripper için uyumludur. Ancak yapısı nedeniyle çalışma uzayı ve kameranın konumlandırılması, çok tercih edilebilecek şekilde olmamıştır. Bu durum yazılım kısmına ek olarak farklı bir zorluk katmıştır. Aynı zamanda yumuşak robotların modellenmesi ve kontrolü, sert gövdeli robotlara göre oldukça zordur. Bu çalışmada, bu zorluk ile de mücadele edilmiştir.

Görev tabanlı olarak çalışan hibrit mobil manipülatör üzerindeki algılayıcılar ve yazılım ile otonom şekilde çalışmaktadır. Ek olarak, diğer objeler ile etkileşime girerken yol gösterici olması amacıyla, görsel referans sistemi olan ARtag'de kullanılmıştır. Bu sayede hangi obje ile etkileşime girilmesi isteniyorsa, daha önceden tanıtılarak görevin otonom şekilde seyir etmesi sağlamıştır.

Yumuşak robotiğin, mobil manipülatör alanında katkıda bulunulması hedeflenmiştir. Yapılan hatalara karşı, manipüle etmek istediği objeye zarar vermeden çarpabilir. Detaylandırmak gerekirse, mobil platformun serbestlik derecesini düşürmeden, hassas yaklaştığı bir kutunun üzerindeki domatese sert bir gripper ile temas ettiği zaman, eğer doğru yaklaşmadıysa domates zarar görebilir. Ancak yumuşak robotik bu hatayı kompanse edebilir.

Deney düzeneğinin fiziksel olması planlanmıştır. Ancak salgın durumu yüzünden projenin gidişatı simülasyon yönüne doğru olmuştur. Simülasyonda çalışmanın en büyük dezavantajı, normal robotlar için geliştirilen simülasyon ortamları, yumuşak robotlara uygun değildir. Tüm görevi aynı simülasyon ortamında ve birbirine geri besleme sağlayarak çalıştırmak için gerekli yazılımlar şu an bulunmamaktadır veya geliştirme aşamasındadırlar. Ancak sonlu elemanlar yöntemiyle, basınç altında herhangi bir yumuşak robotun nasıl hareket edeceğine dair simülasyonlar yapılmış ve seri manipülatör buna uygun tasarlanmıştır.

Otonom navigasyon ve ters kinematik analiz sayesinde, herhangi bir müdahale olmadan, hibrit manipülatör tanımlanan görevi gerçekleştirebilir. Otonom navigasyon sırasında aynı zamanda görsel referans sistemi ve LIDAR verisinden L şeklindeki objeleri ayrıştırma algoritması kullanılmıştır. Geliştirilen yazılım ve mobil platforma donanımsal ve yazılımsal olarak uyumlu hâle getirilmesi sayesinde tanımlanan işlemler yapılmaktadır.

Yumuşak robot ve sert robotun birbiri ile mobil manipülatör başlığı altında birleşmesi başarıyla sonuçlanmıştır. Ancak, simülasyon ortamında hibrit mobil manipülatörün, manipülasyon aşamasında, farklı ortamda yapılan yumuşak robot analizden alınan bilgiler doğrultusunda sadece ters kinematik analiz yapılmıştır. Daha yüksek işlem gücü ve daha uygun bir yazılım çatısı altında, bu işlem gerçekleştirilebilir.

1. INTRODUCTION

It is obvious that robotics science, developed by both commercial and academic circles, continues to contribute to the lives of people day by day. To contribute as a labor force in the industry, to work in hazardous areas for human life, to facilitate the daily life of people; are the motivation sources for every community that develops this science. Serial manipulators used for object manipulation, especially in factories, have a great deficiency due to their limited workspace. While these robots are sufficient for simple manipulation tasks such as pick-and-place, they are not good enough for interacting with their environment, mobility, and large workspace. For these reasons, the researches on mobile manipulators started with the realization of the points where serial manipulators are insufficient [1].

To complete assigned tasks without any interruption, mobile robots must be able to sense its environment with sensors. As a disadvantage of their mobility, mobile robots must process the information they have acquired from their environment and act according to the task in order not to become dysfunctional in different locations. To bear with uncertainties and system complexities, there are numerous advanced software libraries and sensors. Moreover, robots for research provide hardware support and supply the environment for researchers to develop algorithms. To give an example, the TurtleBot [2], the Husky [3], and the PR2 [4] robots are compatible with the Robot Operating System (ROS) [5] that is a software framework for building robot applications. The Robot Operating System provides hardware support, software packages, visualizer apps.

The autonomous navigation of mobile robots is the utmost challenge for different sectors. While reaching its goal, the robot has to be aware of its surrounding object that could be static or dynamic obstacles. The technologies used by autonomous mobile robots can be examined under three main research topics: to detect and map the surrounding objects (mapping), to know the instant location of the robot at any time (localization) and, finally, to set a path to reach its target and to avoid the obstacle [6]. It is very important for the autonomous mobile manipulator, which is in a simple pick-and-place manipulation task, to reach the environment where it will perform and complete the task autonomously. The

necessary workspace must be provided for the manipulator to successfully complete the grasping object task. Manipulators developed for commercial products or research are successful in terms of high precision, low mechanical flexibility.

In line with the latest developments in material science and advances in nonlinear modeling, soft robotics continues to become increasingly popular and subject to research [7]. This topic is not merely a scientific paradigm, but also invaluable for areas of application (rehabilitation robots, rescue robots, and many more) that require more compatibility and interaction with the environment [8][9]. As an example of interactions between soft robots and humans, they can quickly grasp the object that humans hold [10]. Compatible materials can also better comprehend objects of various shapes and hardness [11].

The kinematic and dynamics of soft robotic systems are not like those of conventional, hard-bodied robots. When it consists of a series of mobilization elements, these robots approach a continuum behavior. Soft robots cannot be limited to planar movements, unlike the control of solid objects whose movements can be defined by six degrees of freedom. It can be seen as offering an infinite number of degrees of freedom, which makes it very difficult to control soft robots. Controlling soft robots requires new approaches to modeling, control, dynamic, and high-level planning[12]. In [13], it was managed to control the "OctArm" developed in the octopus-inspired research despite having a high degree of freedom by dividing it into different sections.

Grasping, a subject heavily researched in the field of robotics is also highly studied in soft robots as it causes safe interactions. For a successful grip, attention should be paid to mechanical design and compliant materials because it affects performance considerably. Soft grippers can be divided into different technologies as materials and grasping technology. Grasping technology has been examined in 3 different categories: controllable stiffness, controllable adhesion, and actuating [14]. These categories are also divided into technologies such as Fluidic Elastomer Actuators (FEAs), Granular Jamming, Geckoadhesion. Soft grippers can be used in different areas like fruit picking [15] and biological sampling [16] since it is capable of grasping fragile and sensitive objects without damaging them. There are publicly accessible online resources for the fabrication

of soft grippers such as Soft Robotic Toolkit [17]. Furthermore, there are well documented academic publications that guide step-by-step instructions of soft gripper [18][19][20]. Different sensors and control algorithms can be used to control it. Strawberry can be correctly grasped by the electro-mechanical system using a fuzzy force controller [15]. Using force and proximity sensor for each soft bending fingers gives proper feedback about the grasping of different types of objects to the controller [21]. Soft gripper with different technologies uses actuator cable with PID Controller that managed the actuating cable pulling length [22]. Another case uses flex sensors to get the bending angle that controlled with PID Controller and electronics valves [23].

Hybrid robots include not only the advantages of hard robots (speed and accuracy) but also the advantages of soft robots (versatility and safe interactions) [24]. From [25], it shows that the custom soft robotic gripper was mounted in Baxter Robot that is from Rethink Robotics and operated together. Thanks to the custom soft gripper, determining grasp success helped to decision making during the operation. In another research, the hybrid robot is able to rapid locomotion, gripping, and retrieving an object and it has two subsystems; the wheeled robot (hard) which is iRobot Create, and a four-legged quadruped (soft) [26]. While the tasks of moving the robot and transporting the necessary parts (battery, microcontroller, pumps, etc.) are responsible for the hard robotic subsystem, the task of grasping of different objects that have complex shapes is the responsibility of the soft robotic subsystem. The OctArm was mounted on a mobile base which is the Foster- Millen Talon platform to transporting the hardware platform for the control system [26].

In this study, it is aimed to design a mobile manipulator that has the hard robotic subsystem and the soft robotic subsystem. The hard subsystem contains a mobile platform which is Husky UGV from Clearpath Robotics and the custom 4 DoF serial manipulator. The soft subsystem contains a soft gripper that made by FEAs (Fluidic Elastomer Actuators). Grasping of an object will be performed through actuators. The hybrid robot works in an indoor environment that contains four marked boxes with different visual fiducial markers. In this environment, the hybrid robot can recognize and approach to the boxes using LIDAR and camera. As a goal, the hybrid robot will be able to interact with the

object above the boxes that the robot approaches with the inverse kinematic algorithms and soft gripper that has three pneumatic fingers. As a result, the main reason why the designed robot is hybrid is that while rigid links are sufficient in terms of speed and accuracy, the errors that occur during the autonomous approach to the target can be compensated with the soft gripper. While grasping with a rigid gripper, the mistakes made may damage the target object, but it is not possible in this task performed using a soft gripper. Besides, the serial manipulator design has built-in LIDAR and camera holder. During autonomous navigation, the mobile platform can identify its surroundings using sensors that fixed with the custom holder. The absence of any feedback regarding the position of the target object with respect to the box is the utmost challenge for this task. Therefore, mapping of the boxes in the environment is very important. In recognizing the boxes process, not only visual fiducial markers but also the L-shape extraction algorithm from the LIDAR data has been used.

2. SOFTWARE ARCHITECTURE

In this section, Robot Operating System (ROS), the architecture of the hybrid mobile manipulator software will be introduced.

2.1. Robot Operating System (ROS)

Robot Operating System (ROS) is an open-source framework for building robotic applications. Unlike its name, it is not an operating system that we know in a traditional sense. Instead, it functions as a highly organized layer of communications above various operating systems of a heterogeneous compute cluster [28]. The ROS offers a platform that contains driver-level software, a collection of libraries, tools, and visualization and monitoring environments for researchers. To give an example of the potential offered by ROS, it eliminates the time of dealing with the necessary software of the other components of the robot during the preparation phase required for the laboratory to develop an algorithm in the field of navigation. Also, ROS is open-source and free. It means that any researcher can contribute to the ROS and take advantage of the contributions posted by other researchers. With highly organized communications layers, different parts of the robot can communicate with each other above the ROS framework. ROS supports several programming languages such as C++, Python, Octave, and, LISP. To support these languages, ROS uses a language-neutral interface definition language (IDL). To give an example of why it should be, the software used to process the information of any sensor returns meaningful data written in C++ and can be used in a different program written in Python.

From [29], the beginning of ROS was formed in the mid-2000s with Stanford University's flexible and dynamic system development prototypes. In 2007, the core components of ROS began to be created by Willow Garage that a research laboratory and technological incubator which produces hardware and software for service robotics. From the start, many institutes with the PR2 robots produced by Willow Garage and countless researchers have participated in the development of ROS. Since 2013, the project of maintaining and developing ROS has been transferred to the Open Source Robotics Foundation (OSRF) which is a non-profit organization [30]. In the year 2010, ROS 1.0 was released. In the subsequent months, ROS Box Turtle and ROS C Turtle distributions were released.

Currently, the last three distributions are supported and among them, ROS Kinetic Kame, ROS Melodic Morenia, ROS Noetic Ninjemys. ROS has a community that questions and answers are available on the answer.ros.org website. Also, there is a ROS developers conference named ROSCon which started in 2012 [31].

2.1.1. Core components

ROS is a middleware between the hardware of the robots and control. To understand how ROS works, it is necessary to have an idea about the basic components of it. These are communications infrastructure, robot-specific features, and tools [32].

2.1.1.1. Communications infrastructure

The most important need when building robot applications is the communication system. The Anonymous Publish / Subscribe Mechanism provides great convenience in the communication between the different nodes that will be mentioned in the following sections. Another benefit is that the code written in the nodes makes it more readable and improves encapsulation which is one of the fundamentals of OOP (object-oriented programming) that refers to the bundling of data with the methods that operate on that data. The structure of the messaging interface is defined as message IDL (Interface Description Language).

The messages can be easily captured and replayed, since the publish and subscribe system that will be mentioned in the following sections is anonymous and asynchronous. Thanks to this feature, while working on a different algorithm, messages that the system has previously published and subscribed to can be used.

2.1.1.2. Robot-Specific features

Robot-specific libraries and tools are provided by ROS to make the robot under development faster. Standard message definitions for robots include geometric transformations, vectors used in most robots, and messages used by sensors such as camera and IMU. As these messages are standard, there is no difficulty in using different packages in the robot project.

There is a transform (tf) library that is used to define the relations of the links of the robot with each other. To give an example of using this library, as used in this project, since the distances and orientations of the camera and LIDAR sensors relative to each other are known, this information is used when analyzing the data from each other. The relationship information between each part of the robot provided by the TF library is updated at a very high frequency.

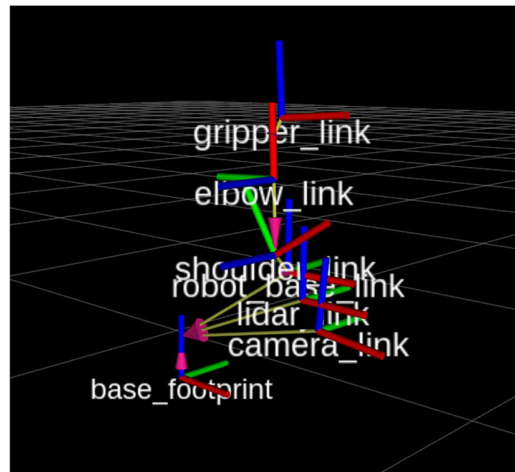


Figure 2.1: Visualization of TF frames in the RVIZ interface.

ROS provides tools for describing a robot in a machine-readable way. These tools publish the robot model to the rest of the ROS system. Unified Robot Description Format (URDF) is the format for describing the robot in ROS. It contains the length of the links, sizes of wheels, joint properties as XML (Extensible Markup Language) document.

2.1.1.3. Tools

ROS has the development toolset to introspecting, debugging, plotting, and visualizing. Without a GUI (Graphical User Interface), all functionality and introspection tools can be used by the command-line. Besides, there is a powerful three-dimensional visualization tool named the RVIZ which visualizes many of the common message types, the robot model, and coordinate frames from the tf library. There is a tool called rqt that allows users to prepare custom interfaces by configuring the library of built-in rqt plugins. The user is also allowed to write the custom rqt plugin. The rqt_graph plugin visualizes a live ROS system, the rqt_plot plugin plots anything that can be represented as a number that

varies over time, the `rqt_topic` and the `rqt_publisher` plugins let the user monitor and introspect any number of topics being published in the ROS system.

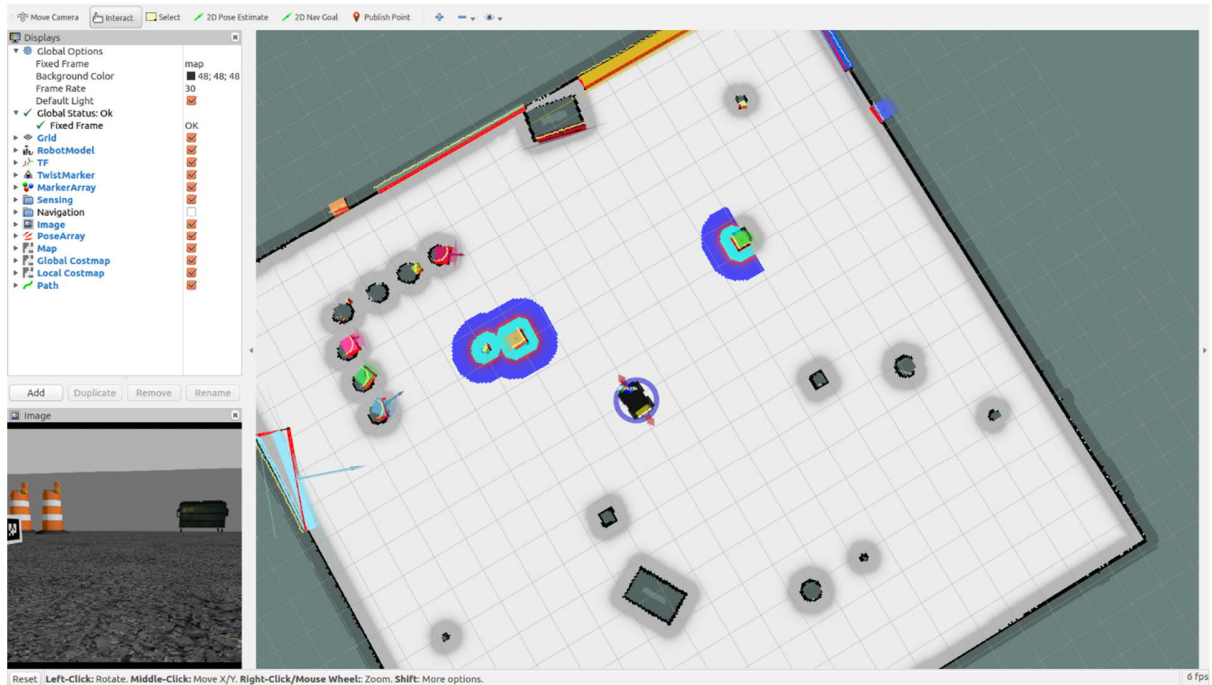


Figure 2.2: Visualization of Camera, LIDAR, and navigation messages in the RVIZ

2.1.2. General concepts

ROS Computational Graph is a network of the process that can communicate with each other. General concepts consist of nodes, master, parameter server, messages, topics, services, and bags.

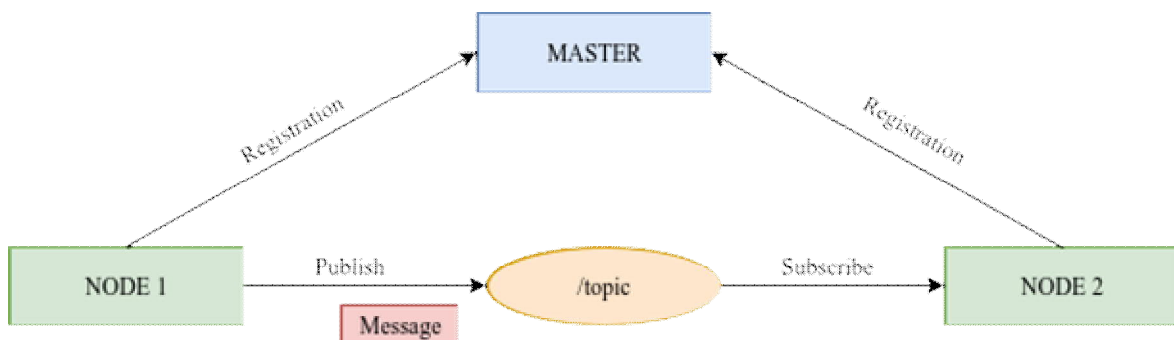


Figure 2.3: Asynchronous communication in ROS

Figure 2.3 shows the basic form of communication in ROS between two nodes over a single topic. It clearly shows the general concepts of ROS. ROS starts with the ROS Master which contains all the names and addresses. It ensures that all nodes in the system find other nodes and communicate with each other. The Parameter Server serves as a place to store public data in central storage which is a part of the ROS Master. It is not used for runtime communication since it is not very fast.

Two nodes, i.e. NODE 1 and NODE 2 are processes for the mentioned work in figure 2.3. This is where the computation and processing of data. A single robot may have lots of nodes such as one handles localization, another one handles image processing, one node provides the GUI to control the robot, and so on. Here, NODE 1 publishes messages to the topic, and NODE 2 subscribes to the topic for receiving messages. With this asynchronous communication, the topic name has to be unique. There is no limit for the number of subscribers and publishers for the specific topic.

When request-reply communication is required, Services can be used. A node can provide a service with a unique name to any other node can send a request and wait for the reply. The synchronous communication is done through services. Bags files are used for storing recorded data and reproducing as live. Bags perform this by providing a mechanism for storing ROS message data.

In this study, there are several nodes to perform autonomous navigation and recognizing tags. To make the visual fiducial markers meaningful, one node publishes a message which contains the position and the ID of the markers. Another node publishes a message which contains box shapes from provided real-time LIDAR data using clustering and L-shape extracting. The node named Fusion combines these topics and performs mapping of boxes and approaching to the selected object.

3. HARDWARE AND SYSTEM DESIGN

In this section, the hybrid mobile manipulator is examined in two different branch as the mobile platform and the serial manipulator.

3.1. Technical Specifications of the Husky UGV

Husky was the ROS-enabled field robotics platform that supports ROS with software packages. The moving platform of the hybrid robot is also shown in figure 3.1. It has high-resolution encoders that are important for state estimation[3]. Also, it can carry up to 75 kg. In this project, air pumps, robotic arm, and other technical items will be carried by the Husky. In addition, Husky will be used as a power supply for the serial manipulator, microcontroller, and other sensors. At the same time, with the manipulator package, Husky is sold with the integrated UR5 robot produced by Universal Robotics.



Figure 3.1: Husky UGV at the ITU Robotics Laboratory

Table 3.1: Technical Specifications of Husky UGV [3]

Size and Weight		Speed and Performance	
External Dimensions	990 x 670 x 390 mm	Max. Speed	1.0 m/s
Internal Dimensions	296 x 411 x 155 mm	Run Time (Typical Use)	3 hours
Weight	50 kg	User Power	5V, 12V, and 24V fused at 5A each
Max. Payload	75 kg	Drivers and APIs	ROS, C++ Library, Mathworks

3.2. Design and Manufacturing of the Serial Manipulator

The following features were taken into consideration while designing the serial manipulator:

- Low-cost
- Maximum Workspace
- Easy to implement

As the robot's degree of freedom increases, the robot's mobility increases. A serial manipulator with 6 degrees of freedom can do all the desired movements in the three-dimensional space of the object it grasps. However, in the task assigned to the serial manipulator in this project, it is not necessary to perform different roll, pitch, or yaw rotation without moving the x, y, and z axes of the object it holds. For this reason, considering its low cost, 3 degrees of freedom are sufficient. However, in the robot's pick-and-place task, it is necessary to bring the gripper at the desired angle, so another degree of freedom has been added. Thus, if each joint of the robot is named with anthropomorphic names, there are 4 degrees of freedom, namely base, shoulder, elbow, wrist. Also, all joints are rotational joints.

Cost, mass, and durability are taken into consideration in the selection of actuators. RC servos were selected for their closed-loop position control, lightweight, and providing high torque. Two high-torque digital servos in the shoulder joint have been selected to work synchronously. These servos are the MG958 model produced by the Tower Pro [33]. The three servos used in other joints are the servos in the MG946r model produced by the Tower Pro [34]. In which joints these servos are used, torque, weight, and speed information are shown in table 3.2. Another point that is particularly important in the selection of these servos is that it has a metal gear structure. In this way, they are more resistant to mechanical stresses. Mounting on links was done using 25T (25 teeth) spline aluminum servo horn.

Table 3.2: List of the Actuators

Location	Type	Torque	Speed	Weight
Base	MG946r	13 kg.cm	0.17 sec / 60 degree	55 g
Shoulder	2 x MG958	40 kg.cm	0.15 sec / 60 degree	2 x 65 g
Elbow	MG946r	13 kg.cm	0.17 sec / 60 degree	55 g
Wrist	MG946r	13 kg.cm	0.17 sec / 60 degree	55 g

Figure 3.2 shows the force diagram used for determining the link lengths. Since two high torque RC servo motors are used in the shoulder joint, there is no need to pay attention to this joint in force calculations. Under the constraints such as production, transportation, and durability, the length of the shoulder link has been chosen to be 25 centimeters, to greatly contribute to the increase of the farthest distance the manipulator can reach. Equations obtained from the force diagram are used in the length calculations required for the shoulder and elbow links. Microsoft Office Excel software was used in link calculations using mathematical optimization methods. Simplex LP was chosen as the solution method [35]. The sum of the lengths of the links was chosen as the cost function because it represents the length that the manipulator can reach the farthest point. The total weight of the soft gripper and the held object was determined as 300 grams. The torque calculation of the motors has been chosen as constraint conditions and compared to a torque less than the limit torque to be safe. The weights of the wrist link and elbow link

are considered according to their design and are chosen independently of the length, weighing 50 grams and 80 grams, respectively.

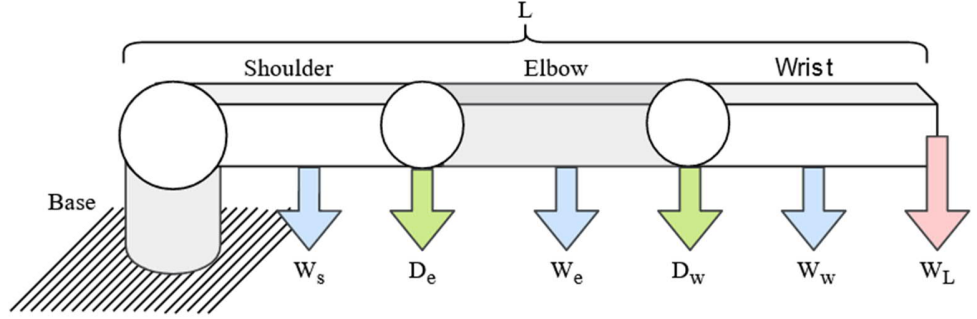


Figure 3.2: Force diagram of the serial manipulator

The variables used for the torque calculations:

$W_s = 0.2$ kg (weight of shoulder link)

$W_e = 0.08$ kg (weight of shoulder link)

$W_w = 0.05$ kg (weight of shoulder link)

$W_L = 0.3$ kg (load)

L_s (length of the shoulder link), L_w (length of the wrist link), L_e (length of the elbow link)

$L = L_s + L_w + L_e$ (the sum of the links)

$D_e = D_w = 0.055$ kg (weight of motor)

M_e (sum of moments around axes of motion of the elbow joint), M_w (sum of moments around axes of motion of the wrist joint), M_s (sum of moments around axes of motion of the shoulder joint)

$$\sum M_w = -(W_L L_w) - \left(W_w \frac{L_w}{2}\right) + M_w = 0 \quad (3.1)$$

$$\sum M_e = -(W_L (L_w + L_e)) - \left(W_w \left(\frac{L_w}{2} + L_e\right)\right) - (D_w L_e) - \left(W_e \frac{L_e}{2}\right) + M_e = 0 \quad (3.2)$$

$$\begin{aligned} \sum M_s = & -(W_L (L_w + L_e + L_s)) - \left(W_w \left(\frac{L_w}{2} + L_e + L_s\right)\right) - (D_w (L_e + L_s)) - \\ & \left(W_e \left(\frac{L_e}{2} + L_s\right)\right) - (D_e L_s) - \left(W_s \frac{L_s}{2}\right) + M_s = 0 \end{aligned} \quad (3.3)$$

Constraints:

$$L_e \geq 5 \text{ cm} \quad L_w \geq 5 \text{ cm}$$

$$M_s \leq 36 \text{ kgcm} \quad M_e \leq 11 \text{ kgcm} \quad M_w \leq 11 \text{ kgcm}$$

With the above constraint conditions, the optimal elbow and wrist link lengths that L will be maximum are calculated as 15 centimeters. As a result, link lengths of 25 cm, 15 cm, and 15 cm were chosen, respectively, to compensate for error margins resulting from positioning the motors and to stay below the safety torque limits. The length that the manipulator can reach the farthest is 55 cm without gripper. However, considering factors such as modular structure and shear force, the distance between the two joints will be different from the selected distances. It will be covered in more detail in the design section. On the other hand, the serial manipulator integrated into the mobile platform. With two degrees of freedom added to the robot from the mobile manipulator, the only thing that matters is that it can access the object in front of the vehicle.

3.2.1. The link between the manipulator and the mobile platform

The main purpose of the design made in the form of a balcony on the front of the vehicle is that the manipulator's workspace has wider access to the outside of the robot. It is designed to be secured with plastic clamps and screws between Husky's front bumper and the sigma profiles on it. Aluminum is preferred for durability and lightness. 6061 aluminum alloy was preferred in the 10 mm thick top plate. The stress test was carried out to determine that it was durable at different loads for the aluminum plate.



Figure 3.3: The design of the aluminum platform carrying the manipulator

3.2.2. The base link

It is understood that considering the axial force coming to the motor in the base joint of the manipulator, it meets the weight of the entire manipulator. This problem can be solved by moving the base link over any structure on the aluminum platform to prevent the entire weight from being loaded onto the motor. However, due to the friction between the rotating part of this link and its fixed part, the moment affecting the base joint will increase and it is thought that it will not be able to overcome static friction at small turning angles. It was decided to use a bearing to reduce the friction between them and the moment that the servo motor meets. The axial deep groove ball bearing was preferred for this link and its type is 51213 [36].

As shown in figure 3.4 (c), there are parts for fixing the electronic card and two servos on the rotating part of the base link. It was produced by using PLA filament with a 3D printer in two parts in the base link.

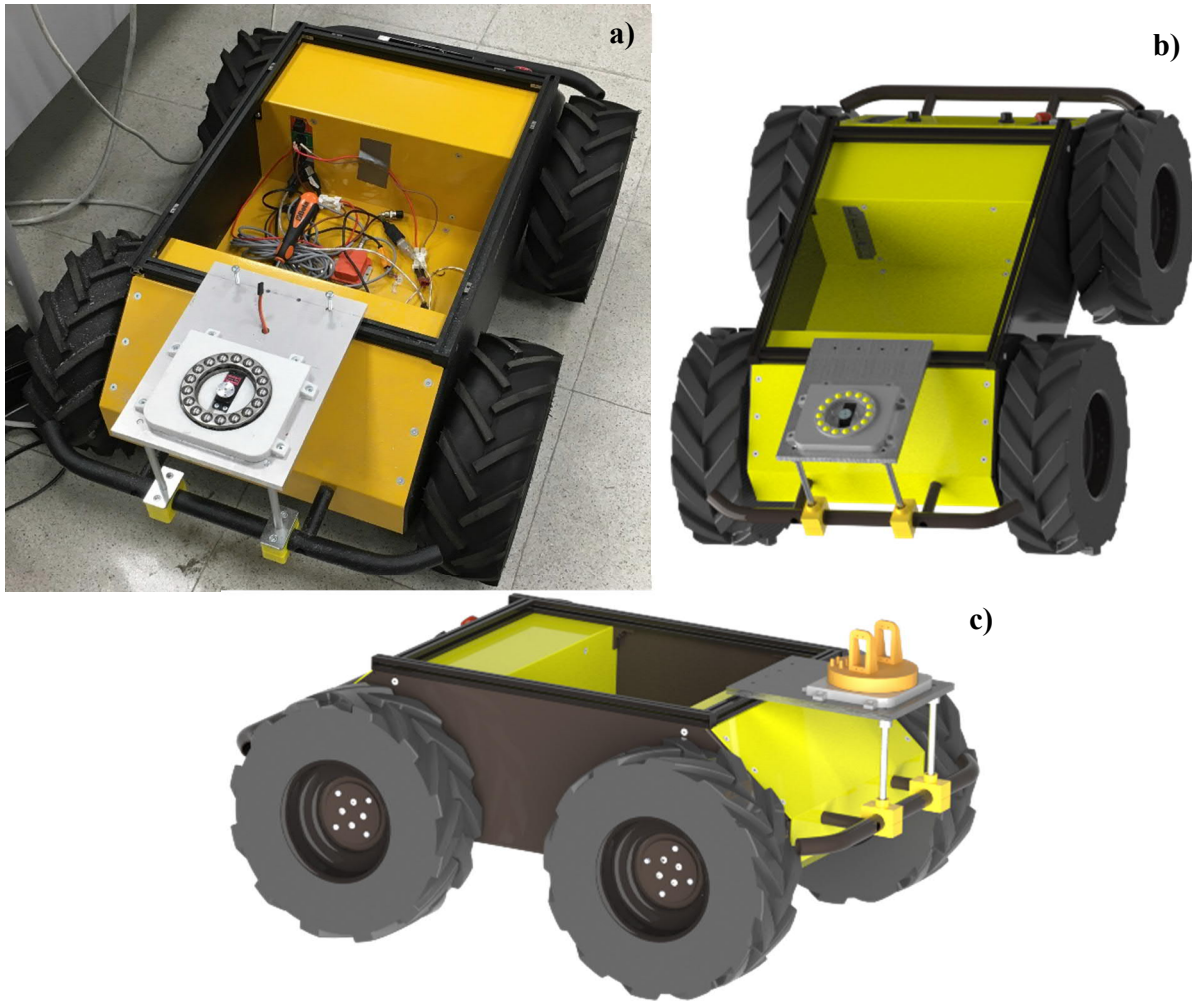


Figure 3.4: The base link with ball-bearings (a) implementation, (b) the fixed part design, (c) the second part design

3.2.3. The shoulder link

Two servos motors attached to the shoulder link must move parallel and attach the appropriate angle value. Because even in overload, the whole load should affect these two servos, so it should affect them equally. At the same time, 6 spacer pieces used between two plates prevent different angles from forming while moving in parallel. The distance between the shoulder joint and elbow joint was selected 200 mm.

As shown in figure 3.5 (b), there are three different sized holes on the plates. The smallest holes are drilled for M3 screw holes used to make connections, the largest holes are made for the slot of the additional cylinder part in the elbow link and the other hole is for the

lightness of the shoulder link without compromising the durability. The plates were manufactured by cutting 5 mm plexiglass with a laser CNC. Distances were produced by PLA filament with a 3D printer.

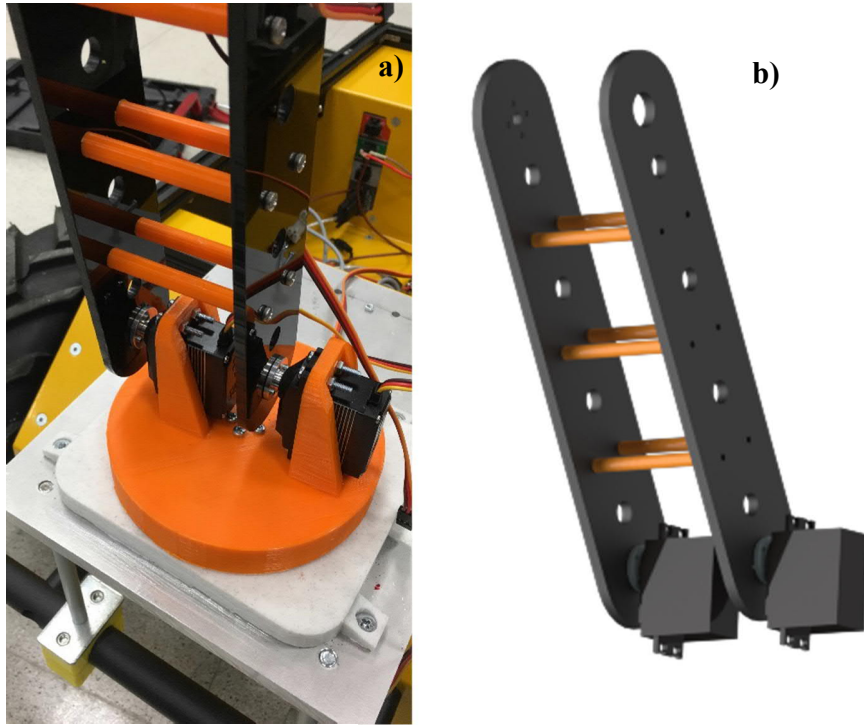


Figure 3.5: The shoulder link (a) implementation, (b) design

3.2.4. The elbow link

There are two servo motors in the elbow link. One is to rotate the elbow link that is attached, and the other is to rotate the wrist link. 4 spacer pieces are used to link to two plates. Additional cylinder parts on the second plate have been added to share the entire load acting on the shaft of the respective motor and the previous link. The distance between the elbow joint and wrist joint was selected 109 mm.

As shown in figure 3.6, plates have holes with the same radius. Since the gap between the motors, the pneumatic hose which connected the wrist link can easily move without any squeeze. Plates and spacer pieces were produced by using PLA filament with the 3D printer.

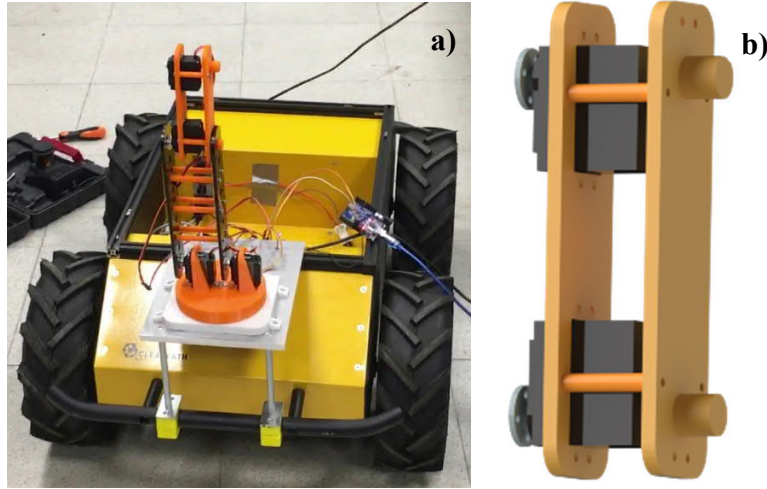


Figure 3.6: The elbow link (a) implementation, (b) design

3.2.5. The wrist link

The most important reason for being called the wrist link is the last rigid joint between independent soft finger movements and the rotational movements that occur depending on the angle of a servo motor. However, it is possible to think of it as part of the soft gripper. It will be examined under different titles under this section.

The large hole on the side of the wrist link closest to the elbow link is divided into three different 3.5 mm diameter tunnels. Each of these air tunnels retains the pneumatic tube that will go to the soft fingers. Thus, it is aimed that pneumatic pipes can enter smooth fingers perpendicular to their surfaces without bending. The orange-colored structure that appears in the figure 3.7 is aimed to be produced with a 3D printer, but this part was not produced since the next parts of the study had to continue only through simulation.

As shown in figure 3.8 (a), the position of the slots where the soft fingers are fixed to the wrist link, and the angle between the surface were calculated. To perform this calculation, it is necessary to know the curvature of the air pressure applied fingers. In order to know this curve, it is necessary to analyze with the finite element method that will be mentioned in the next sections.

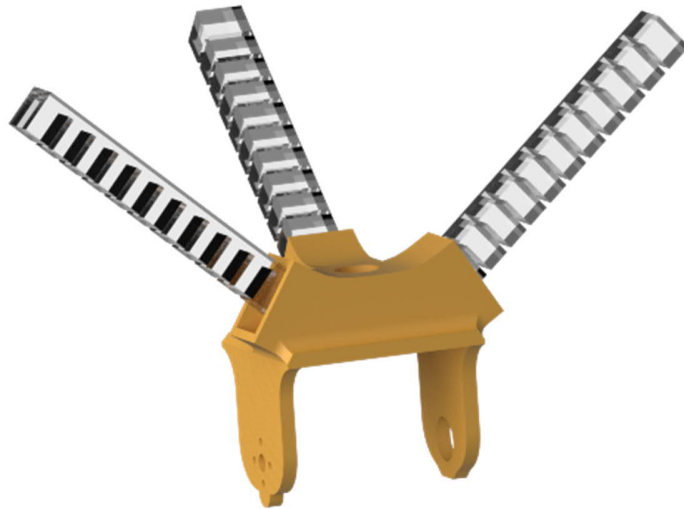


Figure 3.7: The wrist link and soft fingers

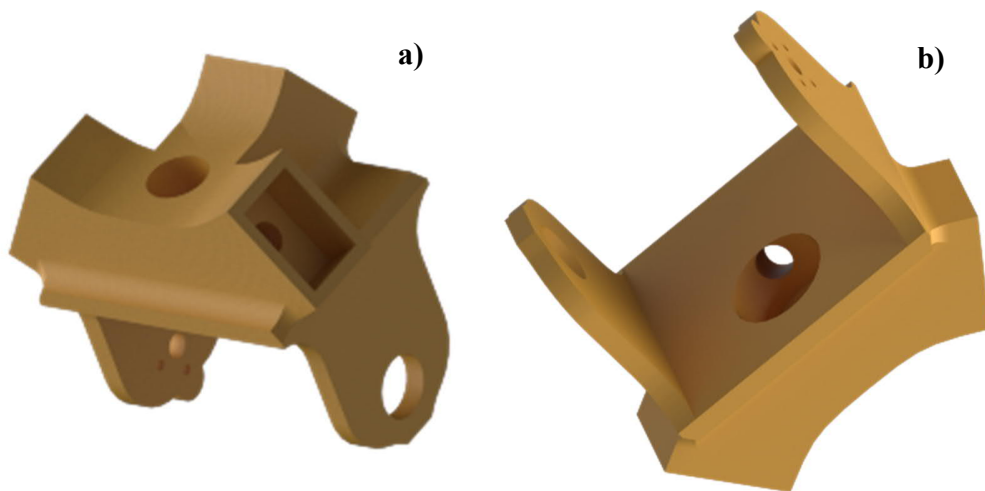


Figure 3.8: The view of the wrist link from different angles

3.2.6. The soft gripper

The structure formed by inflating the air channels inside the elastomer like a balloon by a single pressure source is called pneunet for actuation [37]. A series of parallel chambers embedded are used as repeating components in elastomers. In summary, pneunet actuators have two parts: the upper extensible part and the bottom inextensible part made of an elastomer that is extensible, but the bottom part contains embedded inextensible sheet in the elastomer to prevent it from straining. These layers are then linked together to form a closed channel. The walls between the individual chambers are mostly attached to the substrate, with the exception of a relatively small opening for air to enter the next chamber. This allows more forces to be transformed through the two layers, resulting in a more pronounced bending motion. The stiffness of the pneunet finger can be changed by thickening a wall, the stiffness of the actuator can be easily increased [40]. Height and number of chambers were determined the same value with [38].

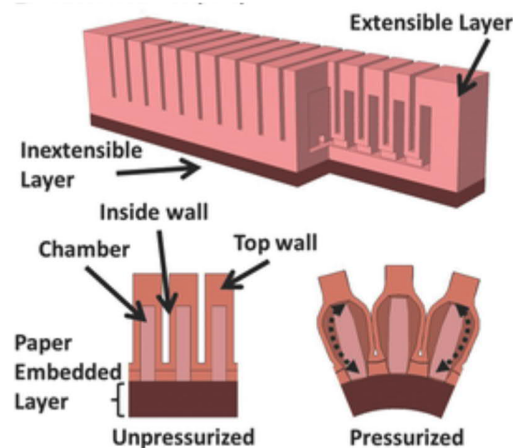


Figure 3.9: The designs of the pneunet with cellular membrane for fast actuation [39]

In this project, the EcoFlex 00-30 product of Smooth-On [41] was used as elastomer and a piece of paper was used as the inextensible layer. EcoFlex 00-30 is the preferred elastomer for durability, ease of use, and relative accessibility. The properties of these materials are taken into consideration in the analysis procedures described in the next paragraph.

Finite Element Method (FEM) provides model the behavior of the soft actuator and see the effect of pressure such as displacement on the cartesian coordinate systems [42]. The Abaqus is software for finite element analysis by Dassault Systems [43]. The process started with designing the 3D model of the actuator in the FEM software. Materials assigned to the model with inserting values of physical properties. Then, Physical loads were defined and applied to the relevant parts. After meshing the geometry and running jobs, FEM simulation results can be ready for review.

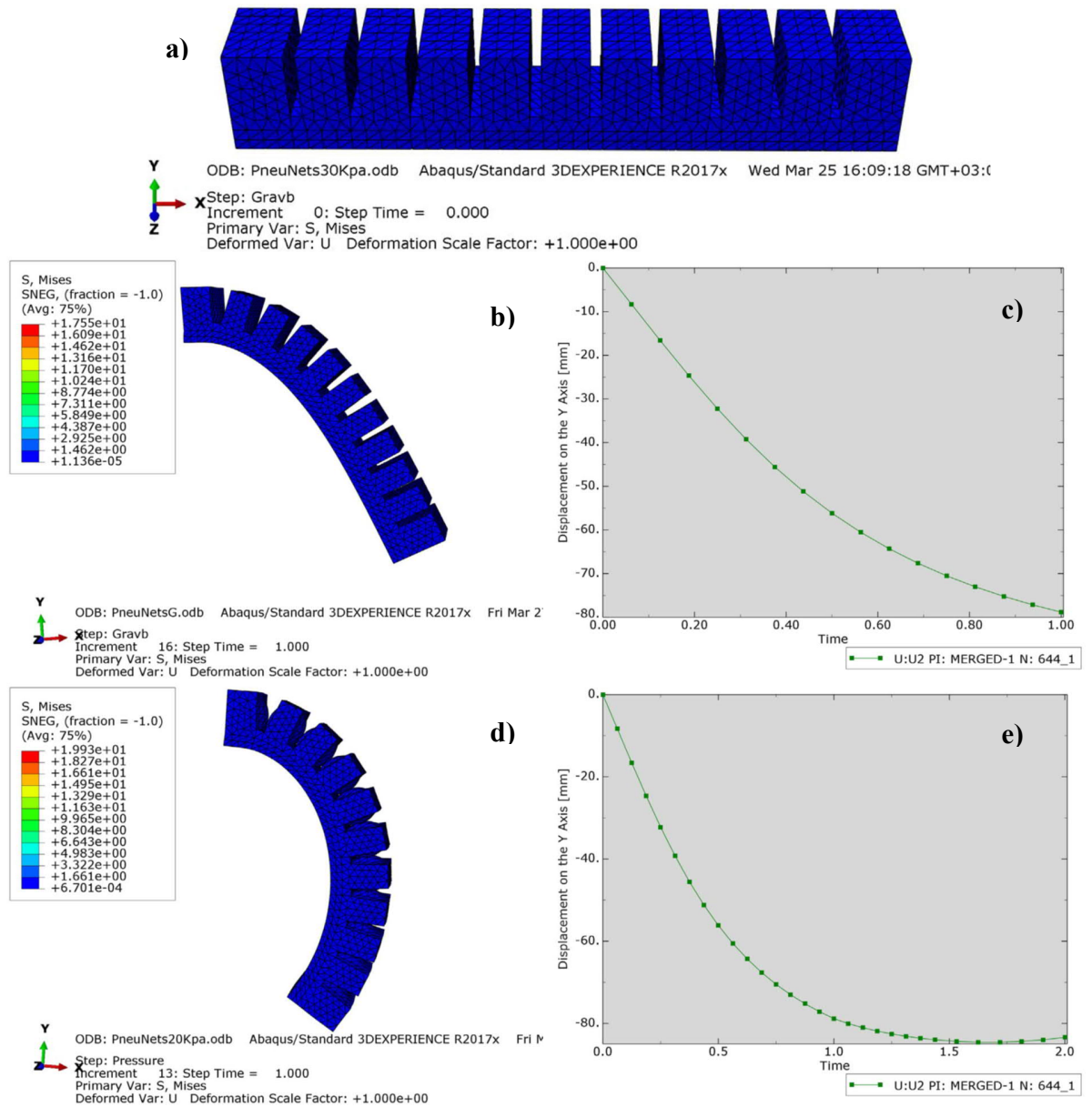


Figure 3.10: The simulation results: (a) 3D model of the PneuNet in the Abaqus, (b) 3D model under gravity load, (c) displacement on the Y-axis in a second under

gravity load, (d) 3D model under 20 kPa pressure, (e) displacement on the Y-axis in two seconds under 20 kPa pressure

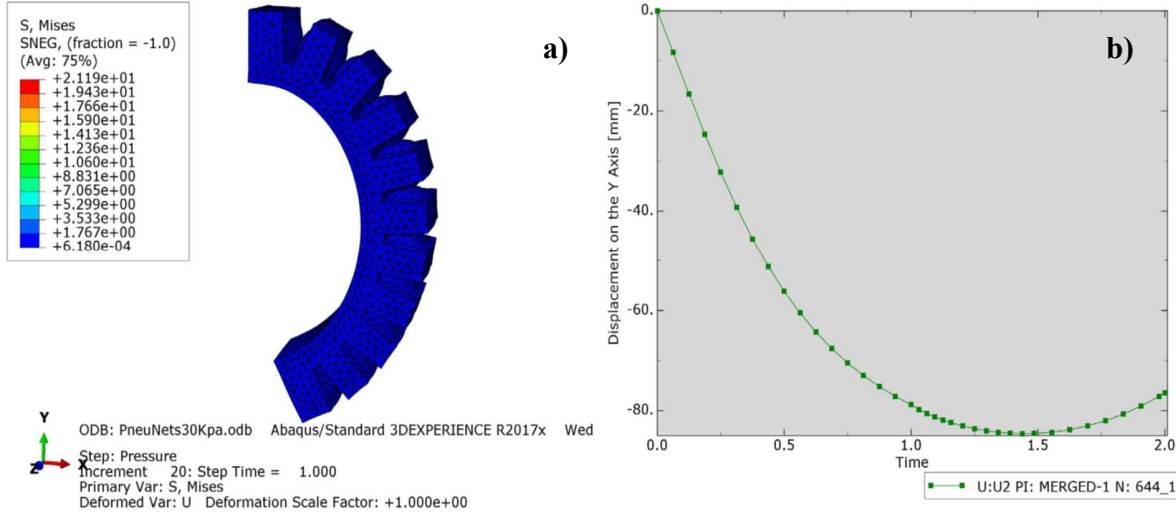


Figure 3.11: The simulation results: (a) 3D model under 30 kPa pressure (b) displacement on the Y-axis in two seconds under 30 kPa pressure

In simulations, the reason that the pneunet finger is inflated with a maximum pressure of 30 kPa is that the motors used as the air source mentioned in the electronic circuit section cannot provide pressures above 30 kPa. As shown in figure 3.10 and figure 3.11, bending increases with increasing pressure. After a certain pressure, the shape of the pneunet finger is twisted and started to become round. The air used in the simulations is independent of the air outlet of the motor intended to use the experiment set.

Equation 3.4 was obtained by fitting the 3rd order polynomial equation with the displacement data on the y-axis against time under the pressure of 30 kPa for the curve in the graph in figure 3.11 (b). A comparison of the obtained equation with simulation data is shown in figure 3.12.

$$y = -13.408x^3 + 81.141x^2 - 147.47x \quad (3.4)$$

The 3D models obtained in the Abaqus simulation were transferred to Siemens Nx 12.0 CAD environment and the number of fingers and their position on the grip of the tennis ball were determined. Three pneunet fingers are used in the gripper to prevent grip failure. For enveloping grasp, one of the 3 fingers is placed in the middle of the other two fingers. The placement of these pneunet fingers will also be efficient for the pinch grasp. For

robust grasping, the depth of the pneumatic finger slots in the gripper has been calculated by taking into account the chambers of the 3D models in the simulation results that have less bending angle against the variable pressure. The angle between these slots and the surface of the gripper is determined so that the pneumatic fingers under pressure can grip the tennis ball diameter of 6.5 cm without touching each other.

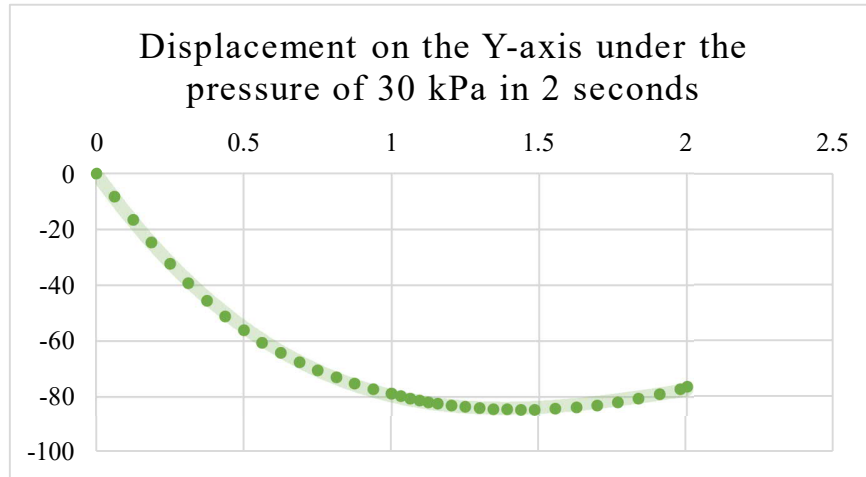


Figure 3.12: The graph of the equation 3.4



Figure 3.13: The 3D model of the hybrid mobile manipulator while grasping the tennis ball

3.2.7. The electronic circuit

Rigid links are actuated by the digital servo motors, and the soft fingers are actuated by the DC air pumps. To communicate and control these motors with the computer, a microcontroller should be used as an interface. Arduino Uno is used in this project as seen in figure 3.14 (d).

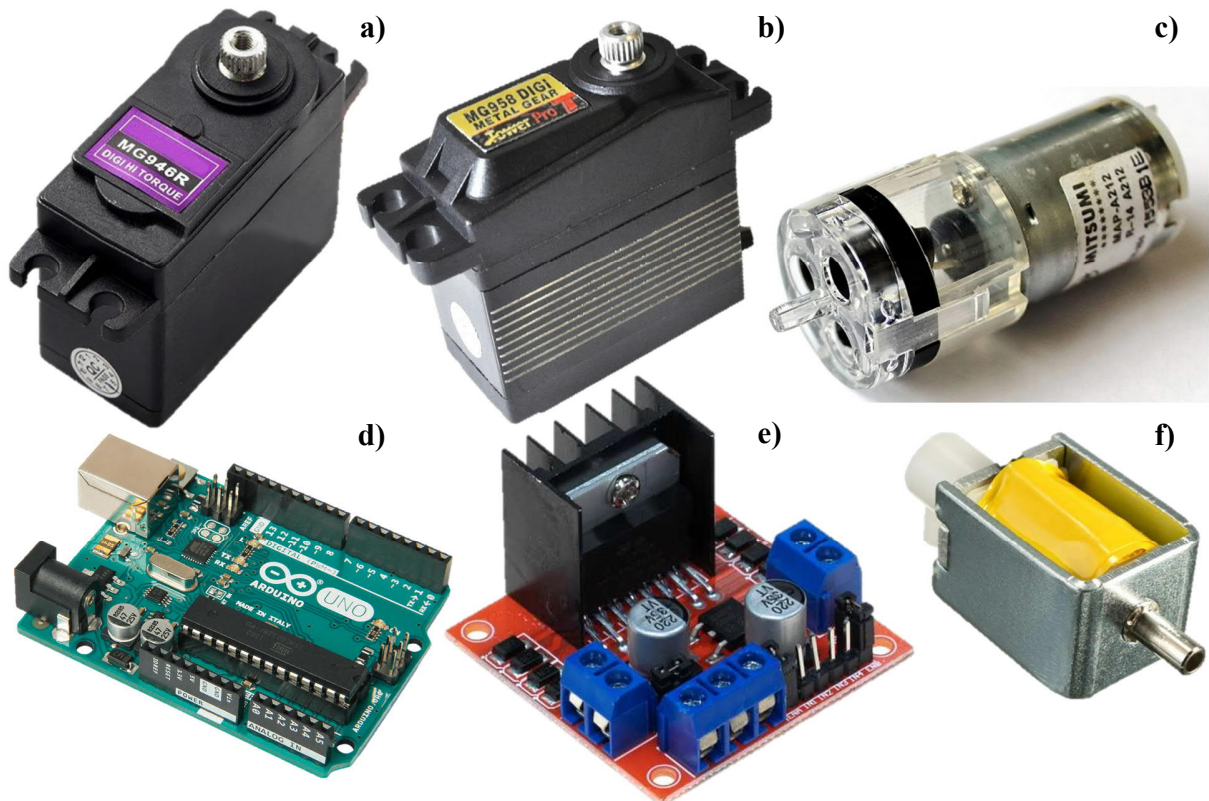


Figure 3.14: Electronic components: (a) MG946r servo motor, (b) MG958 servo motor, (c) 6V micro air pump motor, (d) Arduino Uno, (e) L298N dual h-bridge motor driver, (f) 3V 2-way micro solenoid valve.

Servo motors receive position feedback from the potentiometer located inside the motor chase. Therefore, no other component was used to measure the angle between rigid links. When servo motors are examined in more detail, it contains a small dc motor, potentiometer, and control circuit. The motor is powered up until the motor shaft reaches the reference angle. The desired reference angle is sent to the motor via the pulse width modulation (PWM) signal. The speed of rotation of the motor is adjusted by the difference between the angle of the motor and the reference angle. There is a proportional controller

algorithm in the control circuit. To make the PWM signal meaningful, there is a minimum pulse, a maximum pulse, and a repetition rate. MG946r and MG958 servo motors can only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined by the same distances to both sides. The PWM signal determines the angle of the shaft, and based on the duration of the pulse sent by the microcontroller; the rotor will turn to the desired angle. The servo motor controller circuit expects to see a pulse every 20 milliseconds (ms) and the pulse width transmits the desired angle to the circuit.

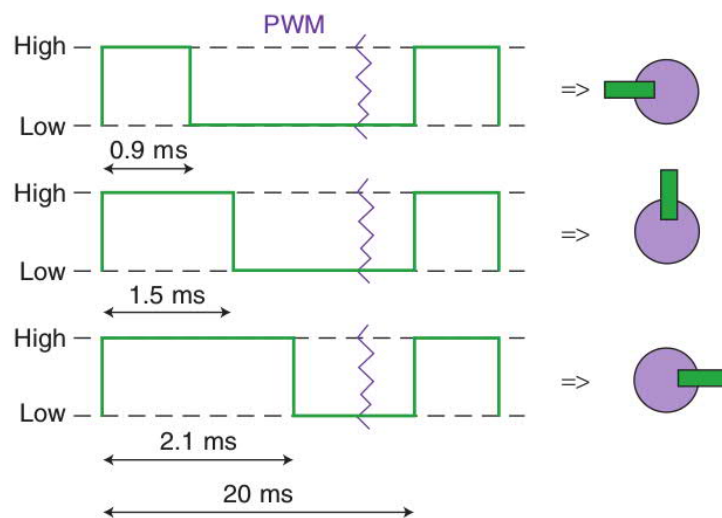


Figure 3.15: Variable Pulse width control servo position [44]

In the electronic circuit powered by the Husky's power outputs, 5A glass fuse was added to the input of the power, except for its own protection. A capacitor has been added to reduce interference and noise the voltage input of servo motors. In other words, motors may suddenly need high currents during sudden movements. This current requirement cannot be provided instantly by the power supply and the motor cannot find the power it needs. As a result of this, sudden fluctuations in motors occur. To prevent this, a capacitor is added so that it can instantly switch on the sudden power needs. The capacitor is connected parallel to the power input of the motors. Thus, movement is provided by protecting from millisecond power cuts and making less vibration. Also, the capacitor absorbs the back emf of the motors and keeps healthy the rest of the circuit which contains sensors.

Blood pressure measuring blood pressure devices include a pressure sensor, a 6V micro air pump, and a 3V 2-way solenoid valve. Due to the cost of medium diaphragm air pumps, 6V or 12V air pumps in digital blood pressure meters can be used to press air into soft fingers. However, these pumps usually have not a label that contains the model information. Therefore, the specifications of the pump are unknown in general. In this project, three air pumps are used to provide the air supply without storing air in air tubes. The outlet of the three motors and the solenoid valve was combined with an airtight piece and used in the system. These three motors are controlled with the same voltage which adjusted by L298N dual h-bridge motor driver. The L298N controls the speed of the DC motor by controlling the input voltage to the motor and the method of doing that is by using the PWM signal as servo motors do. The average value of the voltage by turning on and off the power at a fast rate depends on the duty cycle. For controlling the rotation directions, the direction of the current flow through the motor should inverse. An H-bridge circuit contains four switching elements such as transistors or MOSFETs. By activating two particular switches of four MOSFETs in L298N dual h-bridge motor driver at the same time, the direction of the current flow can changes. However, in this project, the L298N is used to control the speed of air outlet speed of the three air pumps by performing on-off operations in line with the PWM signal thanks to the MOSFETs inside.

The 3V 2-way micro solenoid valve is electromechanically operated. It is used to air release inside the pneuNet fingers when releasing the object after grasping with the soft gripper. The valve has 2 ports. When the valve is open, the two ports are connected and airflow occurs between the ports. The valve is termed normally open (N.O) since the valve opens when the solenoid is not energized. The valve is controlled by a digital pin output that supplies 5v or 0V on Arduino Uno. As shown in figure 3.16, the 2-way valve has two different states.

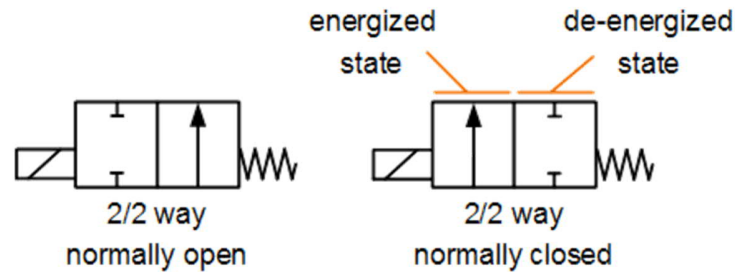


Figure 3.16: 2-Way Valve Symbols

3.3. Camera and Sensors Specifications

In this project, LIDAR and camera were used to identify autonomous navigation and visual fiducial reference tags. It is aimed to use pressure and force sensors for the gripping process of the soft gripper.

3.3.1. Camera

The camera was used to process images. The C270 model camera by Logitech is used. To use this camera in the simulation environment, some features that were changed with the data are seen in Table 3.3 were obtained and used [45]. The utmost reason for not using a stereo camera in this project is the size and weight. The required depth information is provided by LIDAR as 2D.



Figure 3.17: Logitech C270 HD Webcam [45]

Table 3.3: Adjusted Technical Specifications of the C270 Webcam for Simulation Environment

Specifications	Value
Image Resolution	800 x 800 px
Update Rate	30.0
Image Format	R8G8B Bu8
Sensor Noise	Gaussian type with 0.007 standard deviations and 0.0 mean
Near-Far Plane Distances	0.009 m – 1000 m
Dimensions (LxWxH)	71x31x24 mm
Weight	75 grams

3.3.2. LIDAR sensor

2D light detection and ranging (LIDAR) sensors provide horizontal information at high spatial resolutions. The UTM-30LX 2D LIDAR by the Hokuyo is used for localization, obstacle avoidance, and scanning the surrounding objects. As with the camera, the technical specifications of the LIDAR sensor and other information adapted for the simulation environment are shown in table 3.4 [46]. The scanning angle of LIDAR is limited due to its position on the mobile platform.



Figure 3.18: Hokuyo UTM-30LX LIDAR [46]

Table 3.4: Adjusted Technical Specifications of the Hokuyo UTM-30LX LIDAR for Simulation Environment

Specifications	Value
Power Source	12V Max. 1A, Normal 0.7A
Update Rate	40.0
Accuracy and Range	0.1 to 10m:±30mm, 10 to 30m:±50mm
Sensor Noise	Gaussian type with 0.01 standard deviations and 0.0 mean
Angular Resolution	0.25°(360°/1,440 steps)
Scan Angle Range	-1.38 rad – 1.38 rad with 720 samples
Dimensions (WxDxH)	60x60x87 mm
Weight	370 grams

3.3.3. The Sensor Hub

A component is made as shown in figure 3.3.3.1 to fix the camera and the LIDAR between the Husky and the aluminum plate which is located under the manipulator. Its main feature is to ensure that the LIDAR and the camera align exactly on the y-axis. It can be seen in Figure 3.3.3.1.

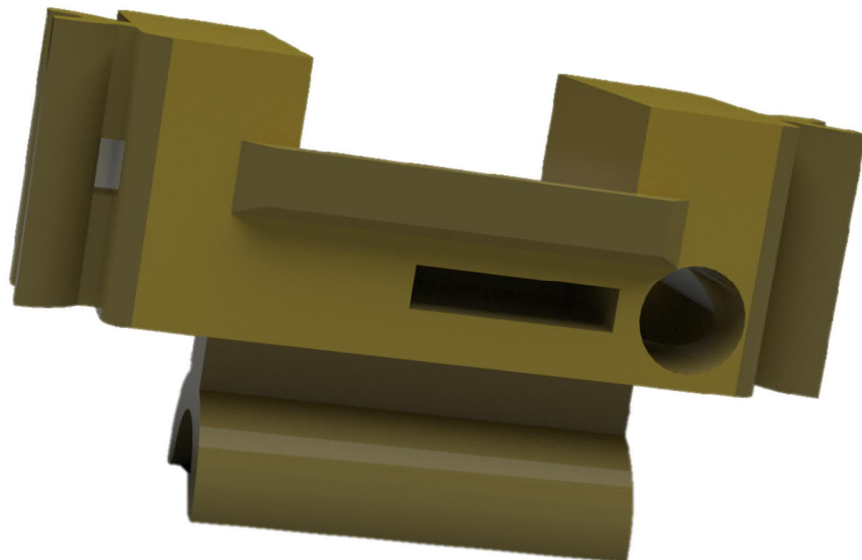


Figure 3.19: The Sensor Hub

3.3.4. Other sensors on the mobile platform

The four-wheeled skid-steer platform Husky is equipped with several sensors that allow it to know the orientation of unmanned ground vehicle (UGV) and how far UGV has gone. These sensors are wheel encoders and Microstrain 3DM-GX4-15 inertial measurement unit (IMU).

Wheel encoders located on Husky's wheels for the purposes of UGV state estimation and localization. IMU attached to the Husky's base frame for the purposes of UGV state estimation and localization.

3.3.5. Pressure and Force Sensor

The pressure sensor measures the sum of the pressure inside the three pneumatic fingers, allowing the speed of the air pumps to be adjusted according to the pressure. There are two situations where pressure is important. These situations are to provide pressure feedback to the controller while trying to reach the reference pressure during the gripping of the object and keeping the reference pressure constant during the transport of the gripped object. Considering the maximum pressure capacity of the air pumps and the design of the gripper, the reference pressure value is a maximum of 35 kPa. For this reason, sensor selection was made accordingly. Due to its accessibility and pressure range, the MPX4250AP differential pressure sensor by Freescale was preferred. MPX4250AP differential pressure sensor is seen in figure 3.20 and technical details are in table 3.3.5.1. The pressure equation obtained from the supply voltage and output voltage is seen in equation 3.3.5.1.



Figure 3.20: MPX4250AP differential pressure sensor [47]

Table 3.5: Operating Characteristics of the MPX4250AP differential pressure sensor [47]

Characteristic	Value
Pressure Range	0 – 250 kPa
Supply Voltage	4.85 – 5.35 V
Minimum Pressure Offset @ $V_s = 5.1$ Volt	0.204 V
Full Scale Span @ $V_s = 5.1$ Volts	4.705 V
Accuracy	± 1.4 % Full-Scale Span
Sensitivity	18.8 mV/kPa
Response Time	1.0 ms

$$P = \frac{\frac{(V_{out} - 20 \times 0.00488758)}{V_s} + 0.04}{0.004} - 93.5 \quad [kPa] \quad (3.5)$$

The force sensor and pressure sensor can be used jointly to understand whether the gripped object is slipping or gripping. In this project, feedback can be obtained that an object is gripped by the thin-film pressure force sensor placed inside the soft three fingers on the soft gripper. RP-L-170 by DFRobot, which is structurally suitable and accessible, is a 170 mm long, flexible, and thin-film pressure sensor that can be used to perform high precision pressure sensing. The sensor is durable and designed to detect static and dynamic pressure at a high response rate. Makes force density and frequency understandable [48].

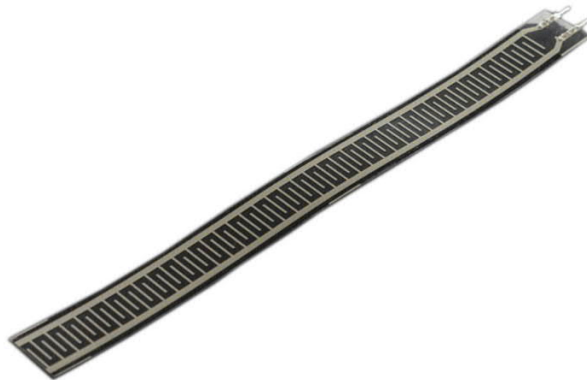


Figure 3.21: RP-L-170 thin-film pressure sensor [48]

4. KINEMATIC ANALYSIS OF THE SERIAL MANIPULATOR

Solving the kinematic problem is of great importance for the trajectory tracking of serial manipulators. Two different approaches, kinematic and dynamically based, are used to solve this problem. The kinematic problem of serial manipulators involves the examination of forward and inverse kinematics. Forward kinematics determines the position and the orientation of the end effector with given joint angles and link lengths of the serial manipulator. Inverse kinematics is the opposite of the forward kinematics. Inverse kinematics determines the joint angles of the serial manipulator for the desired end-effector position and orientation.

A large part of robot kinematics is concerned with the creation of various coordinate systems to represent the positions and orientations of rigid objects and the transformations between these coordinate systems. Three-dimensional space geometry and solid movements play an important role in all aspects of robot manipulation. In robotics, analytical solutions are generally used because tasks assigned to the robot are often defined using the Cartesian coordinate system. To assign coordinates, it is necessary to specify a coordinate frame. Any vector, point, or another coordinate frame can be defined according to any other coordinate frame. It is important to define all coordinate vectors according to the same coordinate frame to perform algebraic manipulations using coordinates [49].

A coordinate frame is defined to each rigid body to represent the relative position and orientation relative to a reference rigid body. Position and orientation information is needed to be used in the geometric relationship between any two coordinate frames. For the position relationship, vectors defined between the origin points of the two coordinate frames are used. Rotation matrixes are used when defining the orientation of a coordinate frame relative to another coordinate frame. Each column and row of these special matrixes are unit vectors. The inverse matrix and transpose matrix of these matrixes are equal. Rotation matrixes are elements of the special orthogonal group ($SO(n)$) and their determinants are equal to 1. It belongs to the $SO(3)$ group of the 3x3 rotation matrix that occurs when rotating in three dimensions. A rotation matrix can be used not only to represent the orientation of the coordinate frame with respect to other coordinate frame

but also to transform the coordinates of a point from one frame to another. In summary, the rotation matrix represents a coordinate transformation relating the coordinates of a point in two different frames, and provides an orientation of a transformed coordinate frame with respect to a fixed coordinate frame.

Rigid motion and homogeneous transformation can be defined by combining the position and orientation concept mentioned in the above paragraphs. The rigid motion is defined by the distance vector (d) and the rotation matrix (R) pair. The group that includes all rigid motions is known as the Special Euclidean Group and is denoted as $SE(3)$.

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}; R \in SO(3), d \in \mathbb{R}^3 \quad (4.1)$$

Transformation matrices of the form given in Equation 4.1 are called homogeneous transformation. A homogeneous transformation is a matrix representation of a rigid motion and it provides the Special Euclidean Group interchangeably to represent both the set of rigid motions and the set of all 4×4 matrices H . The inverse transformation H^{-1} is given by

$$H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix} \quad (4.2)$$

4.1. Forward Kinematic

Each of the rigid links of the serial manipulator depends on the movement of a single revolute joint. Although the serial manipulator is linked to the mobile platform, forward and inverse kinematic analyzes were assumed to be on a fixed platform. Also, the coordinate frame attached to the mobile platform is considered the base frame which defined *base_footprint*, so as not to transform again during autonomous navigation and general perception tasks. The base frame and other coordinate frames are seen in figure 4.1.

The serial manipulator with n joints will have $n+1$ links, as each joint is connected to two links. Joints are numbered from 1 to n , and links are numbered from 0 to n starting from the base link. According to this convention, joint i connects link i and link $i-1$ together. Link 0 (the initial link) is fixed and does not move when the joints are actuated. When the joint i is revolute, joint variable denoted by θ_i that represents the angle of rotation. The serial manipulator has 4 revolute joints. q_i is the joint variable of joint i . Assume that A_i is the homogenous transformation matrix that contains the position and orientation information of $o_i x_i y_i z_i$ with respect to $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$.

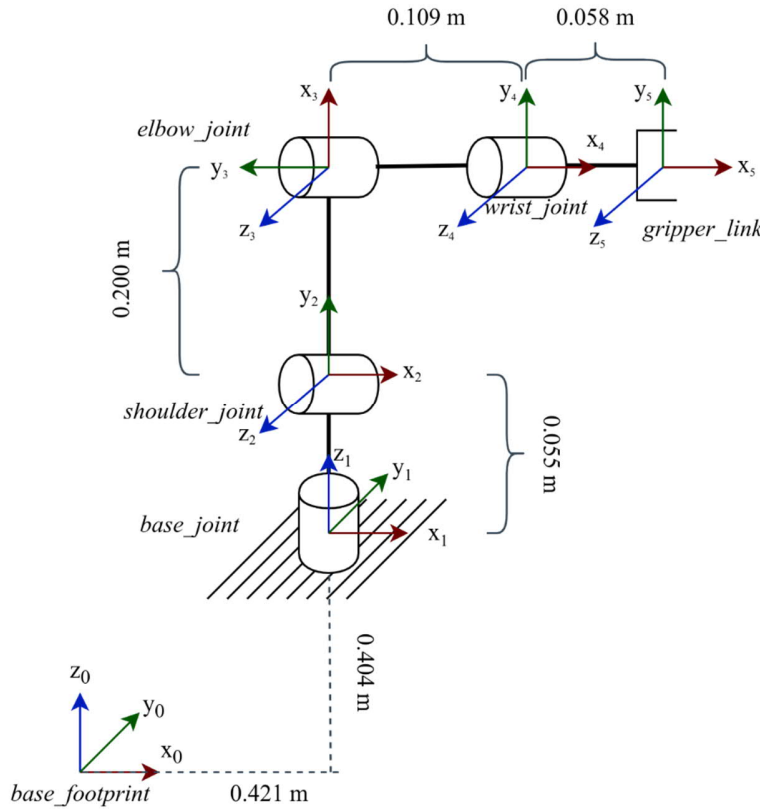


Figure 4.1: Coordinate frames attached to the serial manipulator

The homogeneous transformation matrix that expressed the position and orientation of frame j with respect to frame i is called a transformation matrix that denoted by T_j^i .

$$T_j^i = A_{i+1} A_{i+2} \dots A_{j-1} A_j \quad \text{if } i < j \quad (4.3)$$

$$H = \begin{bmatrix} R_n^0 & O_n^0 \\ 0 & 1 \end{bmatrix} \quad (4.4)$$

$$A_i = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (4.5)$$

$$H = T_n^0 = A_1 A_2 \dots A_{n-1} A_n \quad (4.6)$$

To achieve the equation 4.6 easier, the Denavit-Hartenberg (DH) convention can be used for the forward kinematics. Denavit-Hartenberg convention is based on attaching coordinate frames at each joint. There are four DH parameters to complete the DH table. These parameters:

a_i = the distance along the axes x_i from o_i to the intersection of the axes z_{i-1} and x_i

α_i = the angle between the axes z_{i-1} and the axes z_i around the axes x_i

d_i = the distance along the axes z_{i-1} from o_{i-1} to the intersection of the axes z_{i-1} and x_i

θ_i = the angle between the axes x_{i-1} and x_i around the axes x_i

In the DH convention, A_i is represented as a product of transformations:

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (4.7)$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

DH parameters for the serial manipulator is given in table 4.1. When calculating these parameters, link lengths and coordinate frames in figure 4.1 are taken into consideration. In the robot model transferred to the simulation environment, the direction of rotation, and the initial configuration of the joints affected the angle value of the joints.

Table 4.1: DH Parameters

link	a_i	α_i	d_i	θ_i
1	0.421 m	0	0.404 m	0
2	0 m	$\frac{\pi}{2}$	0.055 m	θ_1^*
3	0.200 m	0	0	$\frac{\pi}{2} - \theta_2^*$
4	0.109 m	0	0	$-\theta_3^* - \frac{\pi}{2}$
5	0.058 m	0	0	$-\theta_4^*$

According to the DH parameters seen in table 4.1, the homogeneous transformation matrices were calculated using equation 4.8. The complete transformation T_5^0 from the base_footprint frame to the gripper is calculated as follows:

$$H = T_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 = A_1 A_2 A_3 A_4 A_5 \quad (4.9)$$

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0.4210 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.4040 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

$$A_2 = \begin{bmatrix} c_{\theta_1} & -0.9996 * s_{\theta_1} & 0.02741 * s_{\theta_1} & 0 \\ s_{\theta_1} & 0.996 * c_{\theta_1} & -0.02741 * c_{\theta_1} & 0 \\ 0 & 0.02741 & 0.9996 & 0.055 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

$$A_3 = \begin{bmatrix} c_{(\theta_2-1.571)} & s_{(\theta_2-1.571)} & 0 & 0.2 * c_{(\theta_2-1.571)} \\ -s_{(\theta_2-1.571)} & c_{(\theta_2-1.571)} & 0 & -0.2 * s_{(\theta_2-1.571)} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

$$A_4 = \begin{bmatrix} c_{(\theta_3+1.571)} & s_{(\theta_3+1.571)} & 0 & 0.109 * c_{(\theta_3+1.571)} \\ -s_{(\theta_3+1.571)} & c_{(\theta_3+1.571)} & 0 & -0.109 * s_{(\theta_3+1.571)} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.13)$$

$$A_5 = \begin{bmatrix} c_{\theta_4} & s_{\theta_4} & 0 & 0.058 * c_{\theta_4} \\ -s_{\theta_4} & c_{\theta_4} & 0 & -0.058 * c_{\theta_4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

If equation 4.10, equation 4.11, equation 4.12, equation 4.13, and equation 4.14 are multiplied respectively, a forward kinematic solution is obtained as a result of equation 4.9.

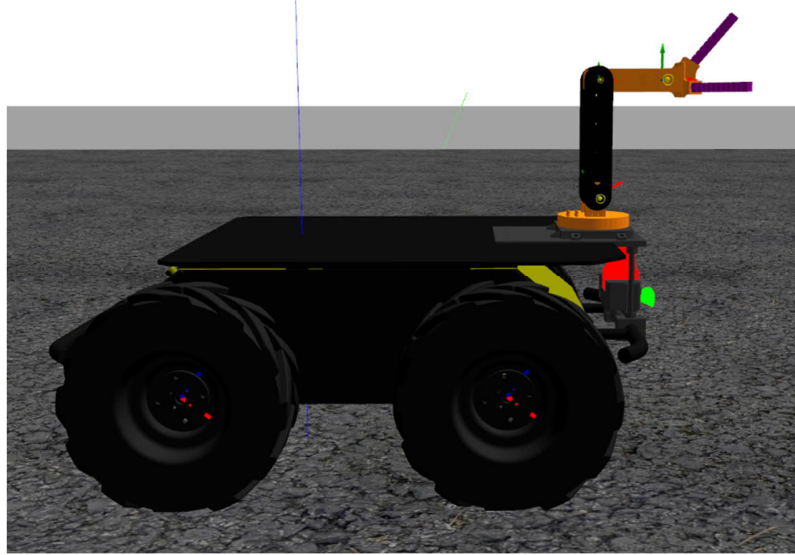


Figure 4.2: Initial configuration ($\theta = [0, 0, 0, 0, 0]^T$) of the serial manipulator in the simulation environment

4.2. Inverse Kinematic

In inverse kinematics analysis, knowing that there are multiple configurations for the position and orientation of the end effector allows using this flexibility when applying the inverse kinematics algorithm to the system. In the gripping process, the soft gripper can have two different orientations due to the position of the fingers and the distance between the target object and the moving platform. This orientation depends on the position of the wrist link. It can be positioned perpendicular to the ground or parallel. When choosing these two different configurations, if the horizontal distance between the target object and the mobile platform is more than normal, parallel; if the vertical distance is more than normal, it is taken into consideration that it is perpendicular to the ground. However, considering the grip point of the wrist link and gripper, a total of 58 mm can contribute in both scenarios.

Inverse kinematics calculations can be done in different ways. In this project, the geometric approach is more suitable for the serial manipulator due to the scenarios mentioned in the upper paragraph. While doing inverse kinematics calculation, different angle values will be calculated in two different scenarios. However, for both scenarios, the angle of the base joint was calculated as seen in figure 4.3.

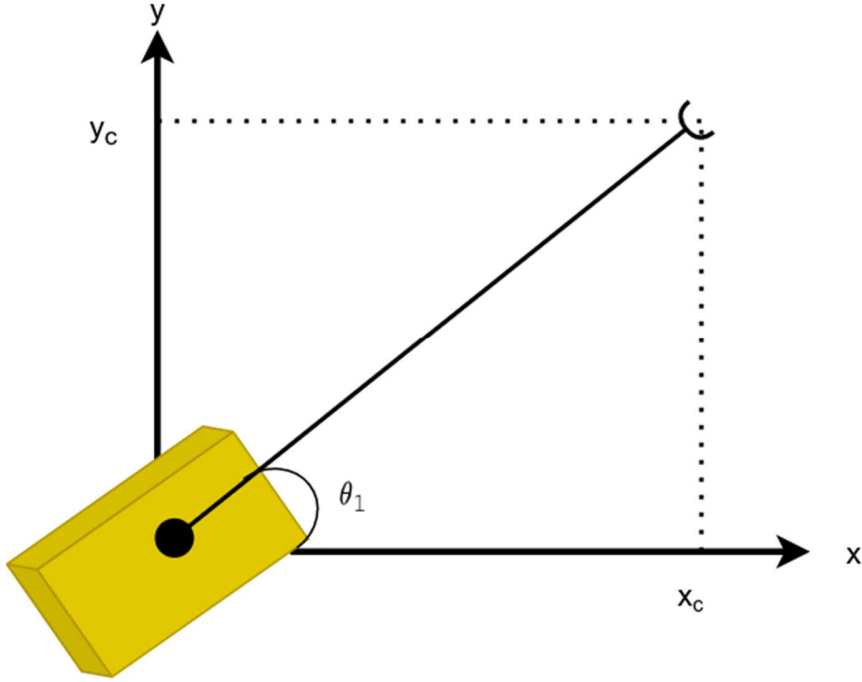


Figure 4.3: Projection of the desired end-effector center onto the x-y plane

$$\theta_1 = \text{atan2}(y_c, x_c) \quad (4.15)$$

$$\theta_1 = \text{atan2}(y_c, x_c) + \pi \quad (4.16)$$

There are two different solutions to the base joint for the desired x and y positions. These solutions are equation 4.15 and equation 4.16. As the study in [50], parameters D' and H' take two values. The D and H parameters are used to show the position of the desired point with respect to the position of the mobile platform. h_0 shows the elevation of the aluminum plate over the ground plane. The variable D in figure 4.4 and figure 4.5 represents the distance to the desired point and is solved as follows:

$$D = \sqrt{x_c^2 + y_c^2} \quad (4.17)$$

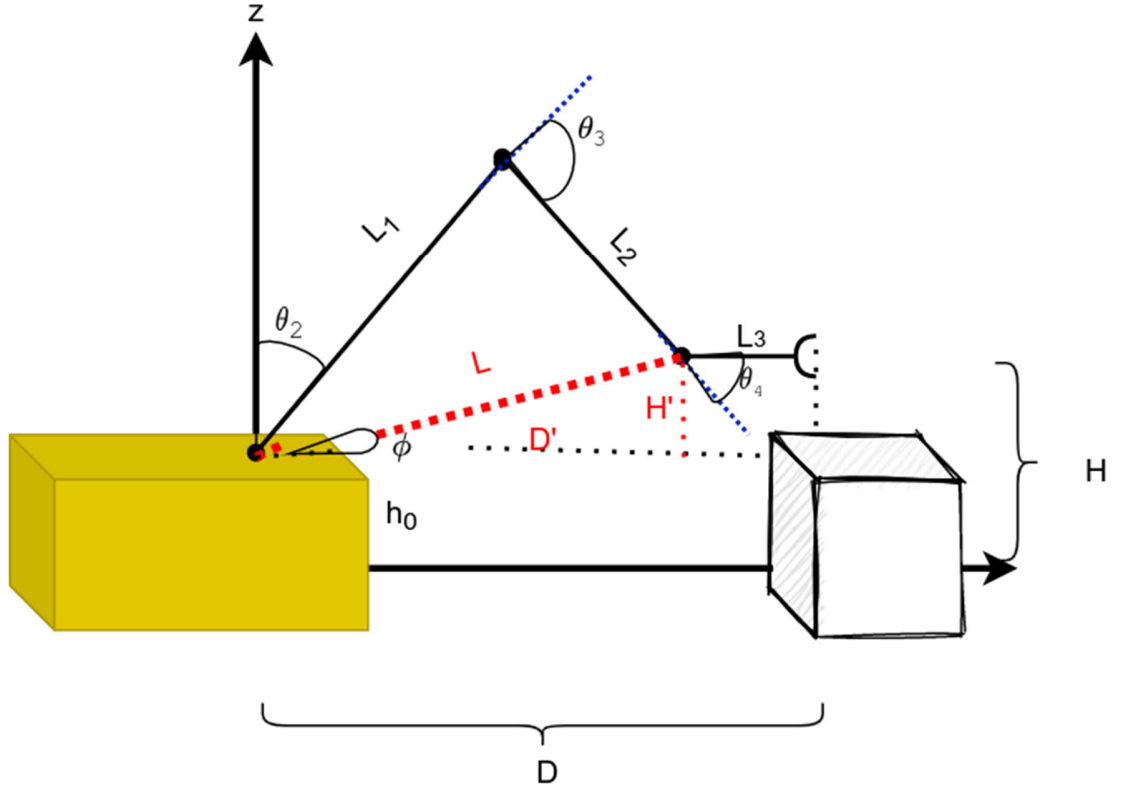


Figure 4.4: The wrist link is parallel to the ground

As seen in figure 4.4, link L_3 does not affect the parameters of H and H' , but the distance of *base_footprint* frame to the desired point could be reduced.

$$D' = D - L_3 \quad (4.18)$$

$$H' = H - h_0 \quad (4.19)$$

$$L = \sqrt{D'^2 + H'^2} \quad (4.20)$$

$$\cos(\theta_3) = \frac{-L^2 + L_1^2 + L_2^2}{2L_1L_2} = Z \quad (4.21)$$

$$\sin(\theta_3) = \pm \sqrt{1 - Z^2} \quad (4.22)$$

There are two solutions for θ_3 . However, θ_3 is used as in equation 4.23 according to the defined height of the boxes and mobile base.

$$\theta_3 = \text{atan2}(-\sqrt{1 - Z^2}, Z) \quad (4.23)$$

$$\theta_2 = \frac{\pi}{2} - \theta_2' \quad (4.24)$$

$$\theta_2' = \frac{\text{acos}(L^2 + L_1^2 - L_2^2)}{2L_1L} + \phi \quad (4.25)$$

$$\theta_2 = \frac{\pi}{2} - \arccos\left(\frac{L^2 + L_1^2 - L_2^2}{2L_1L}\right) - \phi \quad (4.26)$$

$$\theta_4 = -(\theta_2 + \theta_3) \quad (4.27)$$

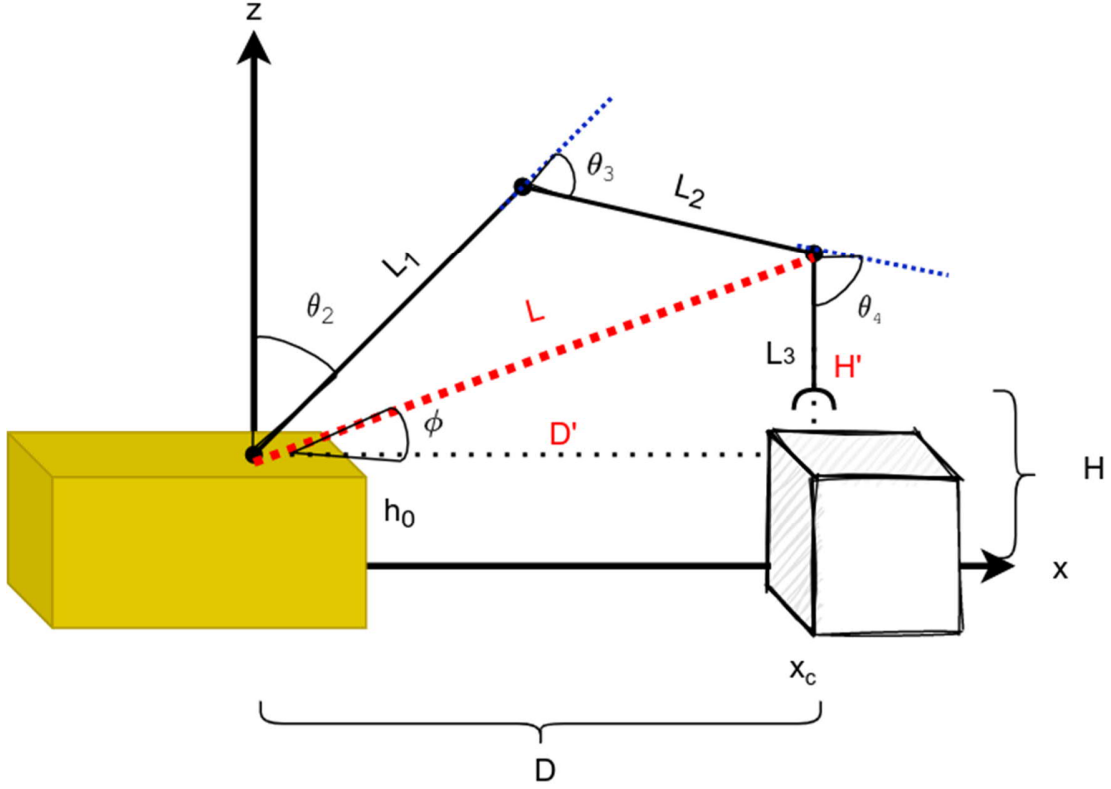


Figure 4.5: The wrist link is perpendicular to the ground

As shown in the figure 4.5, when the inverse kinematic analysis of the serial manipulator with consideration of the wrist link is perpendicular to the ground, equation 4.20, equation 4.21, equation 4.22, equation 4.23, equation 4.24, equation 4.25, equation 4.26 can be used in the same way.

$$D' = D \quad (4.28)$$

$$H' = H + L_3 - h_0 \quad (4.29)$$

$$\theta_4 = \frac{\pi}{2} - (\theta_2 + \theta_3) \quad (4.30)$$

The preferability of the two scenarios depends on environmental factors. Thanks to this flexibility, the wrist link of the serial manipulator is used more conveniently. The angle value of each joint is resolved separately for the desired point. However, during the application, the above equations can be revised according to the joint limits and direction of rotation.

5. AUTONOMOUS NAVIGATION AND MISSION

The hybrid mobile manipulator, which is composed by integrating the custom serial manipulator into the mobile platform, can move autonomously with the help of the sensors on it. Thanks to the soft gripper, which is the reason for the hybrid of this robot, it is aimed to successfully complete the defined task without a very precise approaching. How the hybrid mobile manipulator performs autonomous navigation is described in the next sections.

5.1. ROS Navigation Stack

The basic understanding of the ROS structure has been established, a comprehensive presentation of the autonomous navigation feature can be presented. Autonomous navigation in ROS is implemented in the navigation stack, where it requires different information to make an accurate calculation to the desired destination. In this section, each component is briefly introduced to understand what kind of information the component provides and how the information interacts with each other. The features that can be seen in Figure 5.1 are in three different colors, white, gray, and blue. The white color represents the stack in the ROS for direct autonomous navigation. The blue color represents the platform where code should be written to accommodate the ROS and navigation stack. These platforms are odometer and sensor sources, base controller for the mobile platform, sensor transforms. However, the base controller and odometer source are included in the mobile platform software package called *husky_control*. The gray colors are AMCL (Adaptive Monte Carlo Localization) and *map_server*. Also, the *husky_navigation* ROS package contains *map_server* and AMCL configurations. The software packages of the mobile platform, UGV, have been customized due to the serial manipulator and sensors attached to it. The locations of these added sensors and their features specified in section 3.3 are defined in the simulation environment.

The differential drive controller included in the *husky_control* package contains the skid-steer control. However, with the addition of the serial manipulator, following the reference in the simulation environment has become difficult in some cases. This reduced the accuracy of localization, especially after turns. Some changes have been made to address

this issue. Some movements are defined by using LIDAR data to increase localization accuracy.

The transform configuration provides information about the relationship between coordinate frames using TF (Transform Library) that mentioned in Section 2.1.1. To work successfully, the ROS navigation stack needs to know the position of each wheel, sensor, and joints. It also needs to understand the relationship between the robot's positions compared to the world frame. Laser scan results are published by LIDAR driver as sensor_msgs / LaserScan messages to the navigation stack. Odometer information provides velocity information for the navigation stack and this data is used during localization.

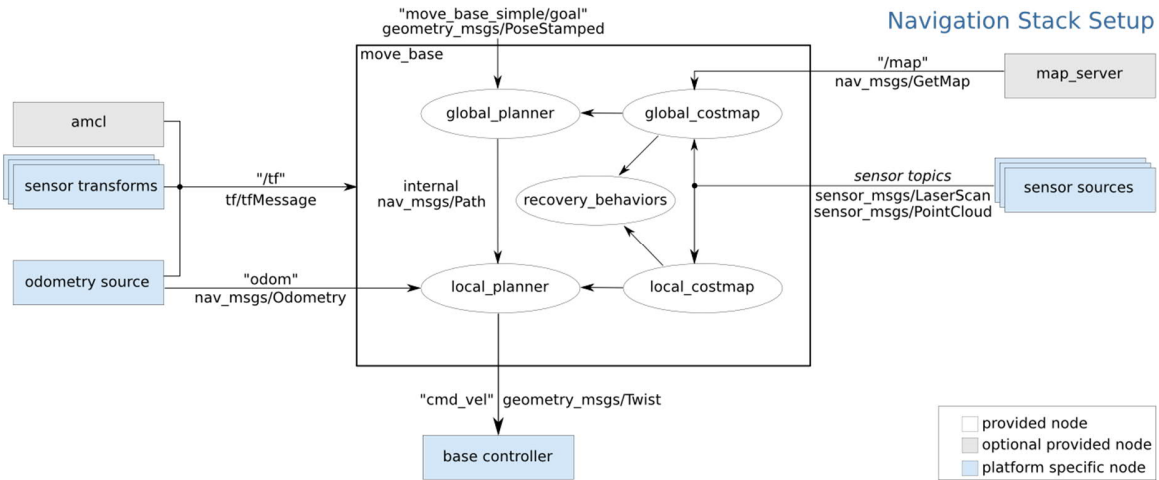


Figure 5.1: ROS Navigation Stack [51]

Details of the map, amcl, planner terms mentioned in Figure 5.1 will be examined in the next sections.

5.2. Mapping

Mapping is the task to model the environment. Mobile robots use maps to define their location and to make motion plans. Although there are different types of maps, the 2D occupancy grid map is used in this project. This occupancy grid map divides the environment in a grid and each cell represents the probability of this area is occupied by an obstacle in this grid. Therefore, the map is a matrix in which elements are probabilities of occupation. In the mapping process, each cell in the grid is continuously updated by a binary Bayes filter [53]. The procedure starts with the creation of a zero $n \times m$ matrix.

Zero value corresponds to a 0.5 probability of occupation. After that, the mapping algorithm updates the probabilities according to the LIDAR measurements and the robot's state by odometry. The method is called SLAM (Simultaneous Localization and Mapping).

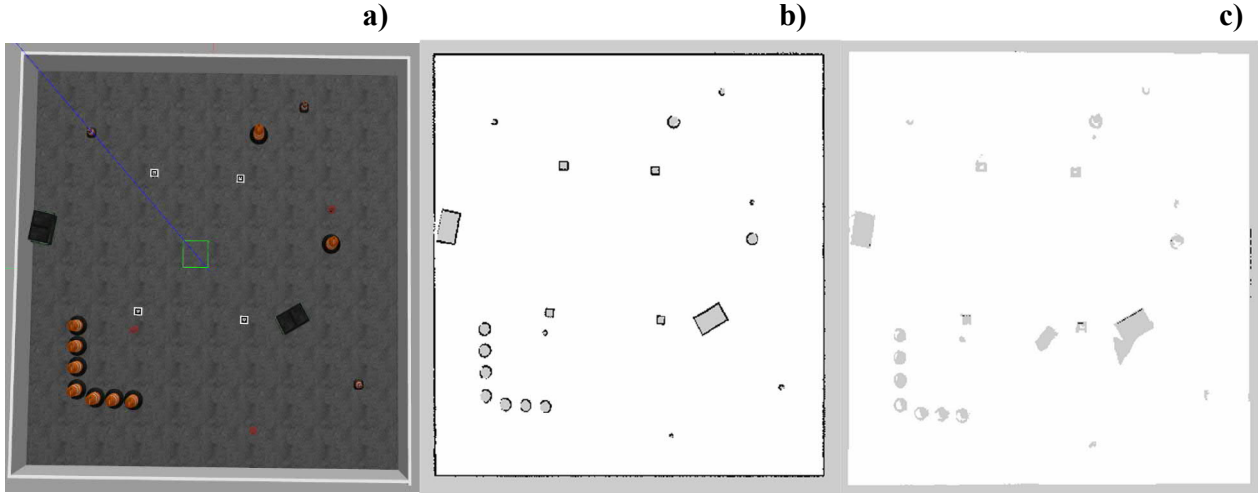


Figure 5.2: a) World Model in Gazebo Simulation Environment, 2D Occupancy Grid Map by using b) gmapping, c) Google Cartographer

The gmapping package in *husky_navigation* is a default tool to create a 2D occupancy grid map from the laser and pose data collected by a mobile robot for ROS. Gmapping uses the Rao-Blackwellized particle filter to learn grid maps using measurements of LIDAR and odometry [54]. The Google Cartographer SLAM algorithm is also tried in this project for comparing the gmapping algorithm. As distinct from gmapping, Google Cartographer does not use a particle filter algorithm and solves the problem of error accumulation by pose estimation [55]. This algorithm uses submaps which represented in the form of probability grid points. Scan matching uses a non-linear optimization for aligning the submaps. In [56], the gmapping algorithm has performed better than google cartographer only in the fast ride with sharp rotations. In the experiment with the hybrid mobile manipulator, the resulting 2-dimensional occupancy grid maps and the world model in the simulation are shown in figure 5.2. Although the loop was closed on the map created with Google Cartographer, the transition between some possibilities on the map was not correct. The 2d occupancy grid map created by the gmapping algorithm with the same movements is more successful and this map has been chosen to be used.

5.3. Localization

The approach of Monte Carlo Localization is based on tracking the robot's position using the particle filter on the known map. Adaptive Monte Carlo Localization is located in the ROS Navigation Stack and it is described in [53]. Basically, samples around the uncertain pose given by odometry data and apply a weight to each sample. Weight information contains similarities between laser scans and static obstacles on the map. The distribution of samples in each step depends on their weight, in other words, the higher the weight, the higher the likelihood of drawing that sample. By doing this, sample distribution after some steps tends to concentrate near the robot's actual pose. The accuracy of odometry data is important, since there may be situations where AMCL is uncertain in the robot position, for example when the map feature is detectable or when many similar features exist. Therefore, AMCL should be coupled with another localization system that combines the information provided by the sensors to produce as precise exposure estimation as possible. For this reason, husky's software package fuses IMU and odometry data with Extended Kalman Filter (EKF) with a 6D model (3D position and 3D orientation).

Based on the given information, Kalman Filter generates an optimal estimation of system state and it is designed for linear discrete-time dynamical systems. However, the Extended Kalman Filter is designed for discrete-time nonlinear systems. EKF based on linearizing dynamics and output functions for a current estimate and propagating an approximation of the conditional expectation and covariance.

According to the guide prepared in [57], AMCL parameters were optimized. For instance, the *odom_alpha4* parameter, which specifies the expected noise in odometry's translation estimate from the rotational component of the robot's motion, increased from 0.2 to 0.8 to minimize the localization error experienced after the robot's turn. Amcl parameters related to odometry data are *odom_alpha1*, *odom_alpha2*, *odom_alpha3* and *odom_alpha4*. These parameters are expected noise in rotation estimation from the rotational element and translational element, and in translation estimation from the translational element and rotational element, respectively. These and other parameters are heuristically optimized in line with the guide.

5.4. Path planning

The path planning can be stated as follows: given a map, an initial state, and a goal state, the planning algorithm must calculate a sequence of states which conforms a free-collision trajectory from initial state to goal state. As seen in figure 5.1, the *move_base* package contains a global and local planner to accomplish its global navigation task. It consists of five nodes as follows:

- The *global_costmap* node that retrieves map and sensor data as input and forms a cost map of the given part. In general, the cost map is created according to the map and the size of the robot by assigning values to each map cell that is similar to the possibility of the collision. These values are between 0 and 255 that 255 represents the cell inside the obstacle.
- The *local_costmap* node, which receives only sensor data as an input and creates a cost map only depending on the laser scan message. The local cost map is used for local planning and obstacle avoidance
- The *recovery_behaviors* node that receives global and local cost maps as inputs and, in dangerous situations, tries to rescue the robot from an accident-like situation.

move_base Default Recovery Behaviors

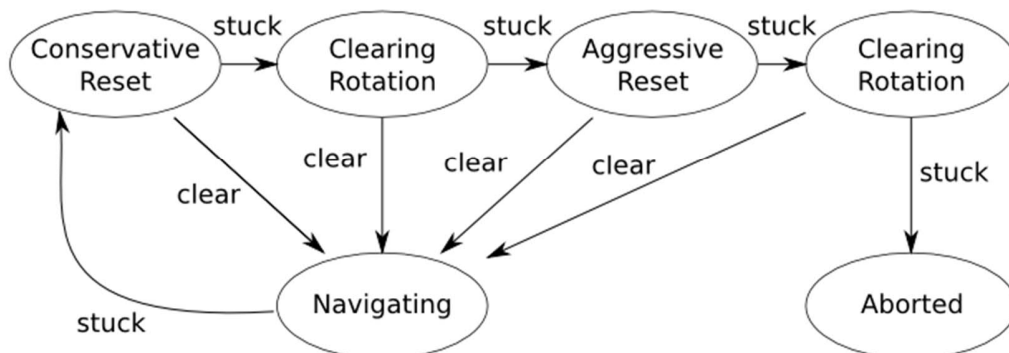


Figure 5.3: Expected Robot Behaviour [58]

- The *local_planner* node uses the global planner path and calculates the maximum velocity of the motors based on the next points of the path that is the output of the entire *move_base* package.

5.5. Mission

For the purpose of the project, the hybrid mobile manipulator needs to go to a certain point during the autonomous navigation and interact with the object there. The husky mobile platform is the outdoor field research UGV. It is desired to adapt the tasks defined to the hybrid mobile manipulator to a scenario. According to this scenario, the hybrid mobile manipulator records the locations of the defined zones in the given map where it is located to perform the defined tasks. It represents the defined zones by the marked boxes with the visual reference markers shown in figure 5.4. The tasks are the logistics operations of the serial manipulator. For the flexibility of the task, the boxes can be anywhere on the known map and there are no number limits.

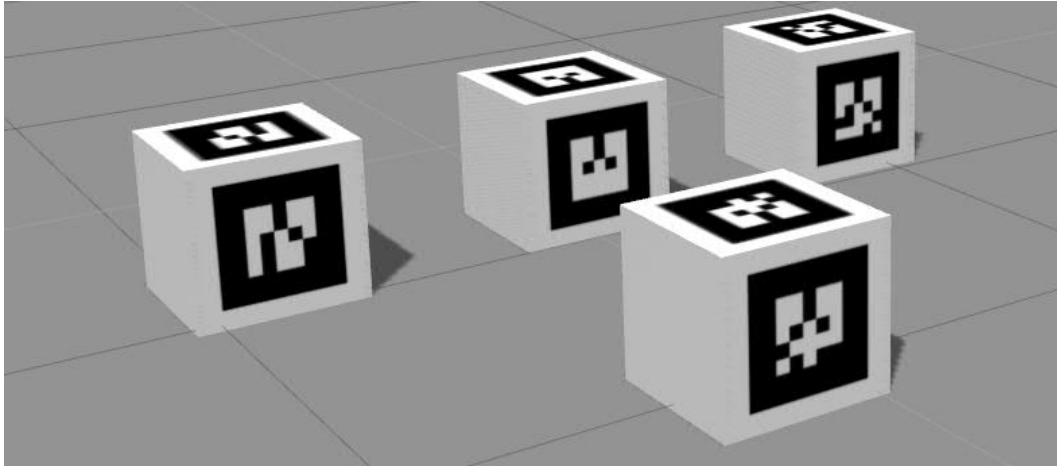


Figure 5.4: Marked Boxes with AR tags

In the world model, which is prepared for simulation and contains different static obstacles, 4 different marked boxes are located in 4 different coordinate regions. When the box recognition processes are finished, the source box and the target box with the goal object will be identified by the operator.

5.6. Marked Box Recognition

There are multiple visual fiducial markers for pose estimation. However, the ARtag system was preferred in this project. There is a ROS package named *ar_track_alvar*, which is adapted from the open-source AR-tag tracking library [59]. The *ar_track_alvar* package has different features. It can generate AR tags of varying size and ID encoding, identifying and tracking the pose of individual AR tags with respect to other coordinate

frames. For tracking the markers, configuration file contains a 0.3 m marker size and 0.02 max_track_error parameters in this study. For the camera to detect more successfully, a white area is left around the tag and a rectangular prism model with a height of 0.4 m, a depth of 0.4 m and a width of 0.42 m has been created. In the launch file created in the *ar_track_alvar* package, it is written which camera topic and which transformation frame is used for calculation. Incorrect results can be obtained in the recognition process performed at different angles. Therefore, in order to make a more successful recognition process, the box objects are extracted from the LIDAR data and matched with the output of the AR tags. As can be seen in figure 5.5, the colored output of the camera, AR tag transformation frames, and the L-shape extractions from LIDAR data are visualized in the RVIZ environment.

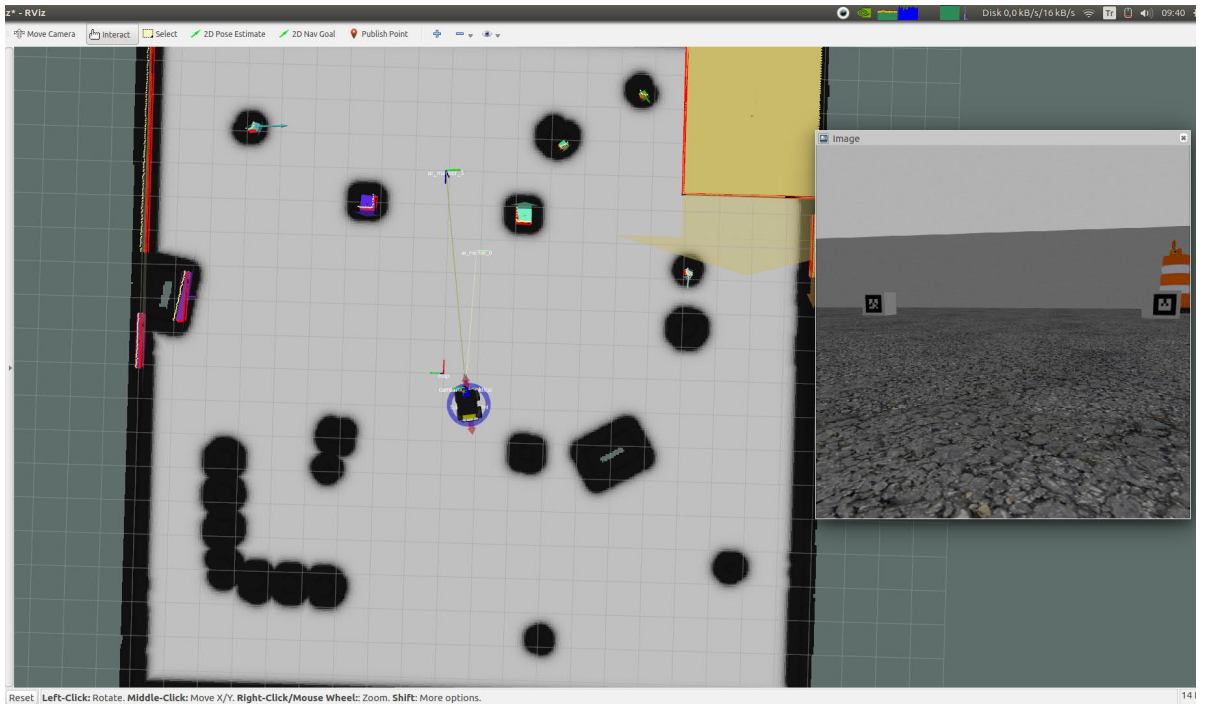


Figure 5.5: Visualization of the camera, ARtag and L-shape extraction in RVIZ

To L-shape extraction, the detection and tracking of moving objects (DATMO) algorithms are used. DATMO has three steps to extract L-shape i.e. LIDAR data clustering, rectangle fitting, and L-shape extraction [60].

In the clustering step, laser scan messages are divided into clusters. The clusters contain different objects in the simulation environment. This can be done simply by separating clusters based on the initial Euclidean distance of LIDAR measurements. Therefore, if the distance of two consecutive LIDAR measurements are greater than a predefined threshold distance, the two points are divided into two separate clusters. However, since the LIDAR measurements become more intense as the distance from the sensor increases, the threshold distance used in object separation should be adapted to increase with respect to the range distance. As can be seen in figure 5.6, Adaptive Breakpoint is obtained using the equation 5.1.

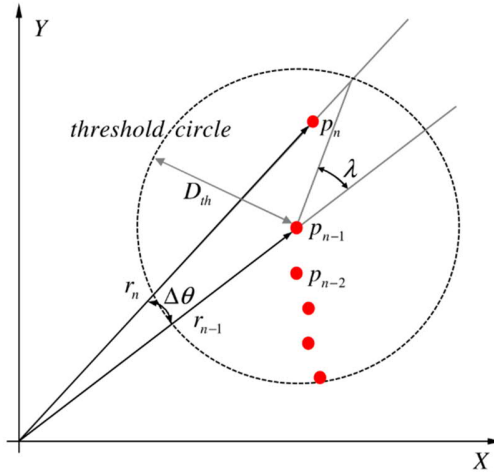


Figure 5.6: Adaptive Breakpoint Detector Algorithm [60]

$$D_{th} = \min(r_n, r_{n-1}) \frac{\sin(\Delta\theta)}{\sin(\lambda - \Delta\theta)} \quad (5.1)$$

In rectangle fitting and L-shape extraction steps, rectangles are fitted in extracted clusters. The Search-Based Rectangle Fitting algorithm used for the shape estimation of rectangular objects. After rectangle fitting, L-shapes are extracted from all the assigned rectangles. ROS message contains x_{corner} , y_{corner} , θ , L_1 , L_2 measurements.

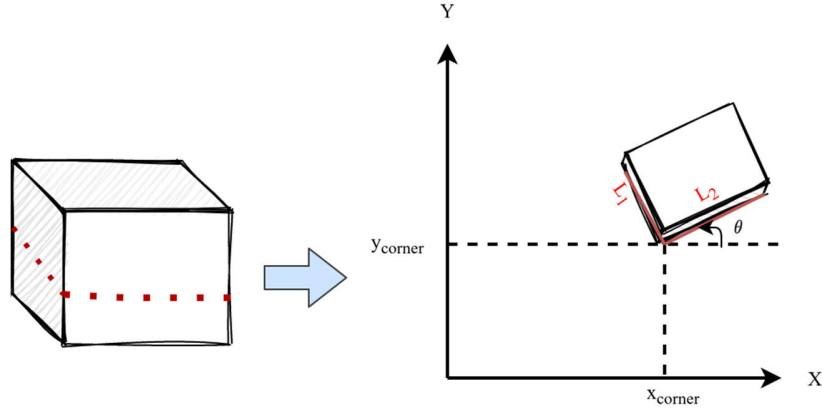


Figure 5.7: L-shape Extraction from Rectangle

To fuse the outputs from *DATMO* and *ar_track_alvar* packages, the 2D Euclidean distance is calculated as follows:

$$distance = \sqrt{(x_a - x_d)^2 + (y_a - y_d)^2} \quad (5.2)$$

x_a = The position of the box on the x-axis of the vehicle, provided by *ar_track_alvar*

x_d = The position of the box on the x-axis of the vehicle, provided by *DATMO*

y_a = The position of the box on the y-axis of the vehicle, provided by *ar_track_alvar*

y_d = The position of the box on the y-axis of the vehicle, provided by *DATMO*

All the data obtained in each measurement interval are listed according to the distance parameter, and the position information with the smallest distance value is obtained. Homogeneous transformation matrixes are used to mapping the box locations.

$$H = T_{base_footprint}^{map} T_{artag_x}^{base_footprint} \quad (5.3)$$

The transformation matrix of the mobile platform by the static *map* frame and the transformation matrix of the marked box's transformation frames relative to the *base_footprint* which attached to the mobile base is multiplied. As a result of this matrix multiplication, the locations of the boxes relative to the map are determined and saved.

5.7. Algorithms

There are two different algorithms associated with each other to complete the defined tasks. The first algorithm scans around on the given map to record the locations of recognized boxes and allows the other algorithm to use this location information.

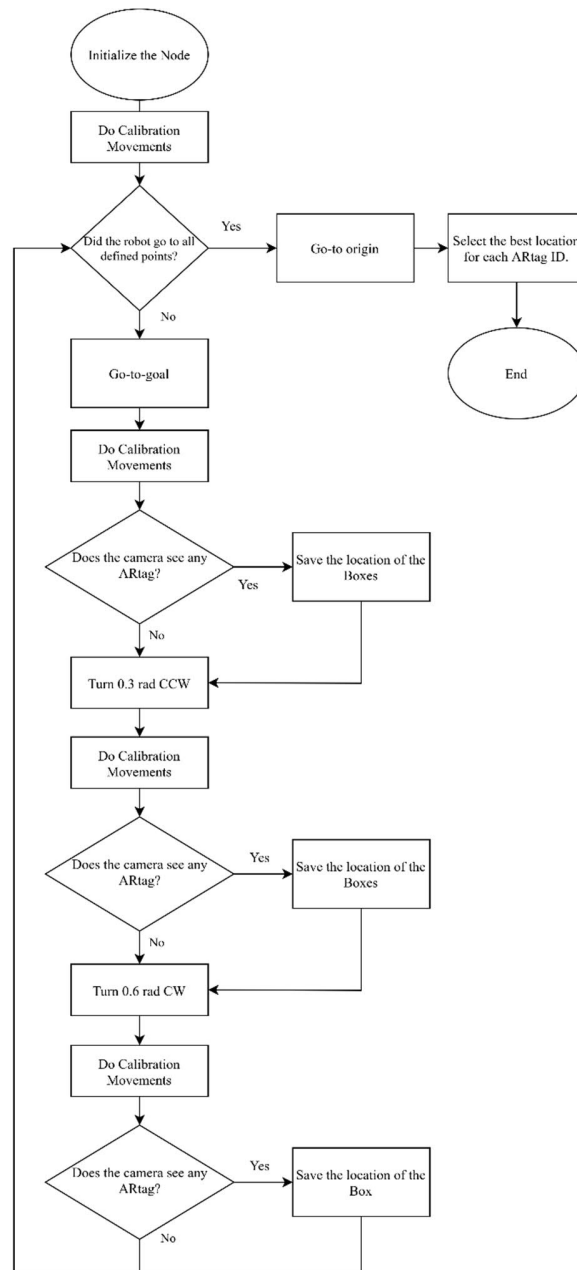


Figure 5.8: Flow Chart of the Autonomous Visual Fiducial Marker Recognition Algorithm

The autonomous visual fiducial marker recognition algorithm based on four different views of different angles of the given map. The *move_base* package is used to go to these predefined points. Moving in forward and in reverse called calibration movement. With this movement, the accuracy of the output of the *amcl* package is increased. Measurements are taken to 3 different yaw angles from each visited location and calculations mentioned in section 5.6 are made. After scanning at different angles at all points, the locations of boxes are recorded in each ARtag ID according to the smallest distance parameter.

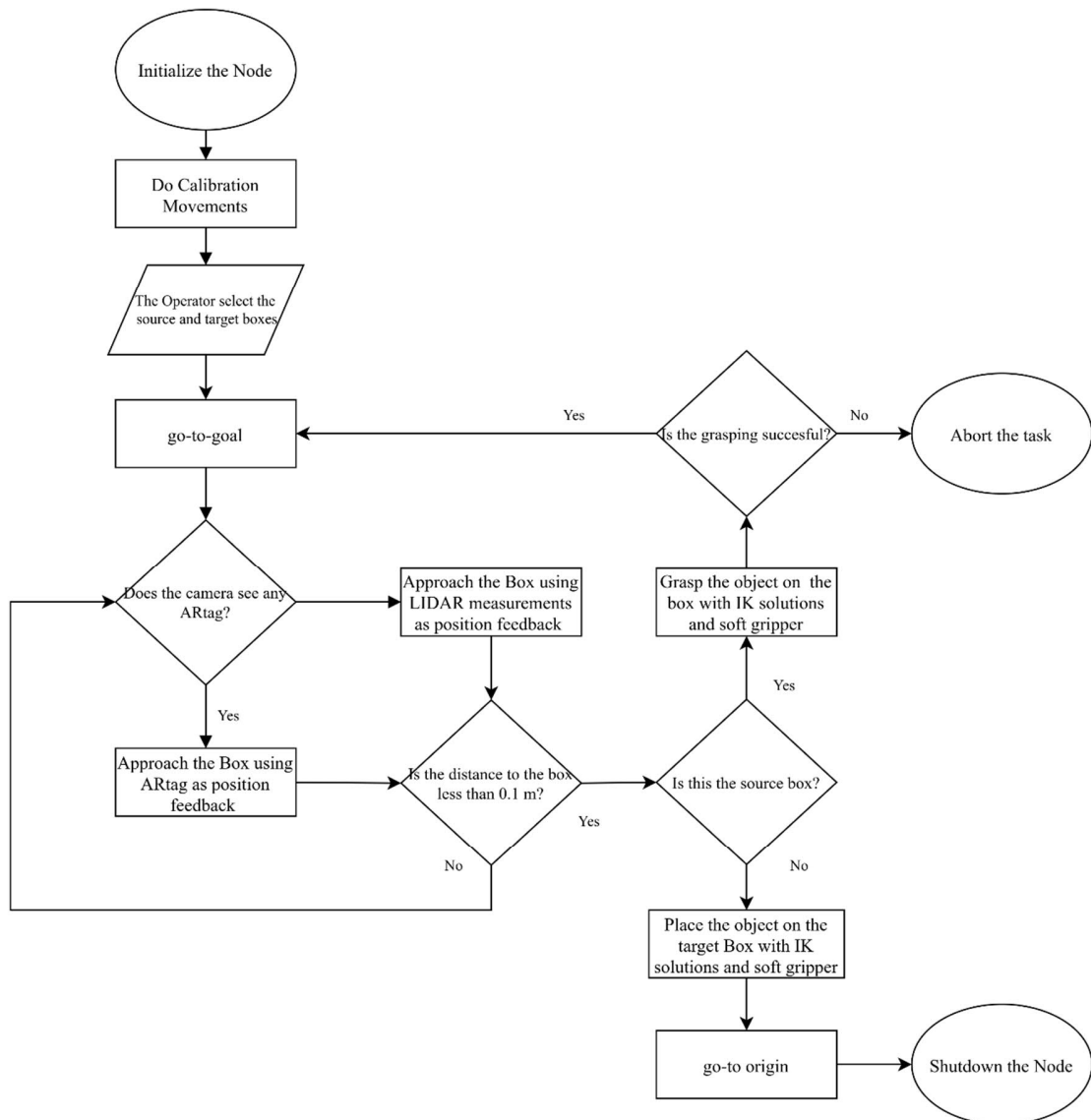


Figure 5.9: Flow Chart of the Approaching the Visual Fiducial Marker and Manipulation of the Object Algorithms

For approaching the visual fiducial marker and manipulation of the object algorithms, the precise approach is the utmost challenge. Due to the field of view of the camera, visual reference identification does not work at distances of less than 0.6 m. To handle this issue, when any part of the reference tag is out of the camera's sight, the precise approach is achieved with a maximum distance of 0.1m by using LIDAR data. After the precise approach, thanks to the inverse kinematics solutions mentioned in section 4.2, the serial manipulator can contact the target object. The success of grasping objects can be measured by pressure and force sensors. This control process is in the algorithm even if it is not included in the experimental setup.

6. CONCLUSION

In this project, a custom serial manipulator with a soft and rigid links was designed and integrated into the mobile platform. Although the whole experimental setup has been prepared according to the physical system, it has been successfully transferred to the simulation environment. Defined tasks were performed in line with the capabilities of the simulation environment in addition to the physical capabilities of the robot. In this developed mechanical system, autonomous navigation and manipulation were studied using the Robot Operating System. One of the important parts of this study is the successful operation of different software packages and different hardware with each other.

The hybrid mobile manipulator includes the compatibility of the soft system with its surroundings and the combination of the rigid system's speed and accuracy capabilities. Taking into account the soft robots has added a different perspective to the analysis that made for many years. It has contributed to the topic of soft robots with a hybrid mobile manipulator which is created with a low-cost and effective design.

Since the Gazebo and Rviz simulation environments do not have software support for soft robots, a full hybrid task has not been performed. However, the FEM analysis showed how the physical experiment system will act. Moreover, very accurate positions were obtained by using visual fiducial markers and the L-shape extraction algorithm together. With this approach, more reliable action can be taken on unknown maps.

The hardware of the computer on which the simulation runs is very crucial. Due to the lack of optimization and lack of computation power during the development of algorithms, stable and repeatable test environment may not be created. In more powerful hardware, a simulation platform with stable and repeatable features for hybrid robots can be developed by establishing a relationship between the simulation tools that suitable for soft robots and simulation environments that designed for traditional robots.

REFERENCES

- [1] **Khatib, O.** (1998). Mobile Manipulators: Expanding the Frontiers of Robot Applications. In A. Zelinsky (Ed.), *Field and Service Robotics* (pp. 6–11). Springer London.
- [2] **Turtlebot** (n.d.). Retrieved July 2, 2020, from <https://www.turtlebot.com/>
- [3] **Husky UGV – Outdoor Field Research Robot by Clearpath** (n.d.). Retrieved July 2, 2020, from <https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>
- [4] **The PR2 Overview | Willow Garage** (n.d.). Retrieved July 2, 2020, from www.willowgarage.com/pages/pr2/overview
- [5] **ROS.org | About ROS** (n.d.). Retrieved July 2, 2020, from <https://www.ros.org/about-ros/>
- [6] **Silva, V. De, Roche, J., & Kondo, A.** (2018). Robust Fusion of LiDAR and Wide-Angle Camera. <https://doi.org/10.3390/s18082730>
- [7] **Laschi, C., & Cianchetti, M.** (2014). Soft robotics: New perspectives for robot bodyware and control. *Frontiers in Bioengineering and Biotechnology*, 2(JAN), 1–5. <https://doi.org/10.3389/fbioe.2014.00003>
- [8] **Chu, C. Y., & Patterson, R. M.** (2018). Soft robotic devices for hand rehabilitation and assistance: A narrative review. *Journal of NeuroEngineering and Rehabilitation*, 15(1), 1–14. <https://doi.org/10.1186/s12984-018-0350-6>
- [9] **Mazzolai, B., Margheri, L., Cianchetti, M., Dario, P., & Laschi, C.** (2012). Soft-robotic arm inspired by the octopus: II. from artificial requirements to innovative technological solutions. *Bioinspiration and Biomimetics*, 7(2). <https://doi.org/10.1088/1748-3182/7/2/025005>

- [10] **Marchese, A. D., Tedrake, R., & Rus, D.** (2016). Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *International Journal of Robotics Research*, 35(8), 1000–1019.
<https://doi.org/10.1177/0278364915587926>
- [11] **Deimel, R., & Brock, O.** (2016). A novel type of compliant and underactuated robotic hand for dexterous grasping. *International Journal of Robotics Research*, 35(1–3), 161–185. <https://doi.org/10.1177/0278364915592961>
- [12] **Rus, D., & Tolley, M. T.** (2015). Design, fabrication and control of soft robots. *Nature*, 521(7553), 467–475. <https://doi.org/10.1038/nature14543>
- [13] **McMahan, W., Chitrakaran, V., Csencsits, M., Dawson, D., Walker, I. D., Jones, B. A., Pritts, M., Dienno, D., Grissom, M., & Rahn, C. D.** (2006). Field trials and testing of the OctArm continuum manipulator. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006(May), 2336–2341.
<https://doi.org/10.1109/ROBOT.2006.1642051>
- [14] **Shintake, J., Cacucciolo, V., Floreano, D., & Shea, H.** (2018). Soft Robotic Grippers. *Advanced Materials*, 30(29). <https://doi.org/10.1002/adma.201707035>
- [15] **Dimeas, F., Sako, D. V., Moulianitis, V. C., & Aspragathos, N. A.** (2015). Design and fuzzy control of a robotic gripper for efficient strawberry harvesting. *Robotica*, 33(5), 1085–1098. <https://doi.org/10.1017/S0263574714001155>
- [16] **Galloway, K. C., Becker, K. P., Phillips, B., Kirby, J., Licht, S., Tchernov, D., Wood, R. J., & Gruber, D. F.** (2016). Soft Robotic Grippers for Biological Sampling on Deep Reefs. *Soft Robotics*, 3(1), 23–33.
<https://doi.org/10.1089/soro.2015.0019>
- [17] **Soft Robotics Toolkit** (n.d.) Retrieved 15 November, 2019 from, <https://softroboticstoolkit.com/>

- [18] **Bharath Vamsi, P., & Ragavendra Rao, V.** (2018). Design and fabrication of soft gripper using 3D printer. IOP Conference Series: Materials Science and Engineering, 402(1). <https://doi.org/10.1088/1757-899X/402/1/012026>
- [19] **Mac Murray, B. C., An, X., Robinson, S. S., Van Meerbeek, I. M., O'Brien, K. W., Zhao, H., & Shepherd, R. F.** (2015). Poroelastic Foams for Simple Fabrication of Complex Soft Robots. *Advanced Materials*, 27(41), 6334–6340. <https://doi.org/10.1002/adma.201503464>
- [20] **Finio, B., Shepherd, R., & Lipson, H.** (2013). Air-powered soft robots for K-12 classrooms. ISEC 2013 - 3rd IEEE Integrated STEM Education Conference. <https://doi.org/10.1109/ISECon.2013.6525198>
- [21] **Li, Y., Chen, Y., & Li, Y.** (2019). Pre-charged pneumatic soft gripper with closed-loop control. *IEEE Robotics and Automation Letters*, 4(2), 1402–1408. <https://doi.org/10.1109/LRA.2019.2895877>
- [22] **Manti, M., Hassan, T., Passetti, G., D'Elia, N., Cianchetti, M., & Laschi, C.** (2015). An under-actuated and adaptable soft robotic gripper. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9222, 64–74. https://doi.org/10.1007/978-3-319-22979-9_6
- [23] **Elgeneidy, K., Lohse, N., & Jackson, M.** (2018). Bending angle prediction and control of soft pneumatic actuators with embedded flex sensors – A data-driven approach. *Mechatronics*, 50(September 2017), 234–247. <https://doi.org/10.1016/j.mechatronics.2017.10.005>
- [24] **McKenzie, R. M., Barraclough, T. W., & Stokes, A. A.** (2017). Integrating Soft Robotics with ROS-A hybrid pick and place arm. *ArXiv Preprint ArXiv:1702.00694*.

- [25] **Homberg, B. S., Katzschnann, R. K., Dogar, M. R., & Rus, D.** (2015). Haptic identification of objects using a modular soft robotic gripper. IEEE International Conference on Intelligent Robots and Systems, 2015-Decem, 1698–1705.
<https://doi.org/10.1109/IROS.2015.7353596>

- [26] **Stokes, A. A., Shepherd, R. F., Morin, S. A., Ilievski, F., & Whitesides, G. M.** (2014). A Hybrid Combining Hard and Soft Robots. *Soft Robotics*, 1(1), 70–74.
<https://doi.org/10.1089/soro.2013.0002>

- [27] **McMahan, W., Chitrakaran, V., Csencsits, M., Dawson, D., Walker, I. D., Jones, B. A., Pritts, M., Dienno, D., Grissom, M., & Rahn, C. D.** (2006). Field trials and testing of the OctArm continuum manipulator. *Proceedings - IEEE International Conference on Robotics and Automation*, 2006(May), 2336–2341.
<https://doi.org/10.1109/ROBOT.2006.1642051>

- [28] **Morgan Quigley et al.** (2009) “ROS: an open-source Robot Operating System”. In: ICRA Workshop on Open Source Software. Vol. 3. Jan. 2009

- [29] **Robot Operating System** (n.d.). Retrieved 15 November, 2019 from
<https://www.ros.org/history/>

- [30] **OSRF,” ROS@OSRF”** (n.d.). Retrieved 2 July, 2020 from
<http://osrfoundation.org/blog/ros-at-osrf.html>

- [31] **ROSCON** (n.d.). Retrieved 2 July, 2020 from <https://roscon.ros.org/2020/>

- [32] **CORE COMPONENTS** (n.d.). Retrieved 2 July, 2020 from
<https://www.ros.org/core-components/>

- [33] **MG958** (n.d.). Retrieved 2 July, 2020 from
<http://www.towerpro.com.tw/product/mg958/>

- [34] **MG946R** (n.d.). Retrieved 2 July, 2020 from
<http://www.towerpro.com.tw/product/mg946r/>

- [35] **G. B. Dantzig** (1948) “Programming in a linear structure,” *Comptroller*, USAF, Washington, D.C.,
- [36] **51213 BALL BEARINGS** (n.d.). Retrieved 2 July, 2020 from <https://www.skf.com/group/products/rolling-bearings/ball-bearings/thrust-ball-bearings/productid-51213>
- [37] **F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides**, (2011) Soft robotics for chemists, *Angew. Chemie - Int. Ed.*, vol. 50, no. 8, pp. 1890–1895.
- [38] **PNEUNET Fabrication** (n.d.). Retrieved 2 July, 2020 from <https://softroboticstoolkit.com/synergistic-design/fabrication/pneunet>
- [39] **J. D. Holt**, (2017) “Design and Testing of a Biomimetic Pneumatic Actuated Seahorse Tail Inspired Robot,”
- [40] **R. Khajuriwala** (2018) “Soft Gripper Using Pneumatic Network Actuators,” pp. 2–6, 2018.
- [41] **SMOOTH-ON | ECOFLEX 00-30** (n.d.). Retrieved 2 July, 2020 from <https://www.smooth-on.com/products/ecoflex-00-30/>
- [42] **PNEUNET MODELLING** (n.d.). Retrieved 2 July, 2020 from <https://softroboticstoolkit.com/book/pneunets-modeling>
- [43] **Abaqus** (n.d.). Retrieved 2 July, 2020 from <https://www.3ds.com/products-services/simulia/products/abaqus/>
- [44] **Pinckney, N.** (2006). Pulse-width modulation for microcontroller servo control. *IEEE Potentials*, 25(1), 27–29. <https://doi.org/10.1109/MP.2006.1635026>
- [45] **C270** (n.d.). Retrieved 14 July, 2020 from <https://www.logitech.com/tr-tr/product/hd-webcam-c270>

- [46] **UTM-30LX Specification** (n.d.). Retrieved 16 July, 2020 from <https://www.hokuyo-aut.jp/search/single.php?serial=169>
- [47] **MPX4250dp Fark Basinc Sensoru** (n.d.). Retrieved 10 July, 2020 from <https://pdf.direnc.net/upload/mpx4250dp-fark-basinc-sensoru.pdf>
- [48] **Wiki DFrobot** (n.d.). Retrieved 2 July, 2020 from <https://wiki.dfrobot.com/Thin%20Film%20Pressure%20Sensor>
- [49] **Spong, M. W., Hutchinson, S., & Vidyasagar, M.** (2006). Robot modeling and control. In IEEE Control Systems (Vol. 26, Issue 6). <https://doi.org/10.1109/MCS.2006.252815>
- [50] **Clothier, K. E., & Shang, Y.** (2010). A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation. *Journal of Robotics*, 2010, 1–10. <https://doi.org/10.1155/2010/984823>
- [51] **Robot Setup** (n.d.). Retrieved 2 July, 2020 from <http://wiki.ros.org/navigation/Tutorials/RobotSetup>
- [52] **Husky Repo** (n.d.). Retrieved 2 July, 2020 from <https://github.com/husky/husky>
- [53] **S. Thrun, W. Burgard, D. Fox.** (2005) Probabilistic Robotics, MIT Press
- [54] **Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard.** (2007) Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, *IEEE Transactions on Robotics*, Volume 23, pages 34-46
- [55] **W. Hess, D. Kohler, H. Rapp, and D. Andor** (2016) Real-Time Loop Closure in 2D LIDAR SLAM, in *Robotics and Automation (ICRA)*, 2016 IEEE International Conference on. IEEE, pp. 1271–1278.
- [56] **Yagfarov, R., Ivanou, M., & Afanasyev, I.** (2018). Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth. 2018 15th International

Conference on Control, Automation, Robotics and Vision, ICARCV 2018, January 2019, 1979–1983. <https://doi.org/10.1109/ICARCV.2018.8581131>

- [57] **Zheng, K.** (2017). ROS Navigation Tuning Guide. 2019, 1–23.
<http://arxiv.org/abs/1706.09068>
- [58] **Move Base** (n.d.). Retrieved 15 July, 2020 from http://wiki.ros.org/move_base
- [59] **Fiala, M.** (2005). ARTag, a fiducial marker system using digital techniques.
Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, II, 590–596.
<https://doi.org/10.1109/CVPR.2005.74>
- [60] **Konstantinidis, K., Alirezai, M., & Grammatico, S.** (n.d.). Development of a Detection and Tracking of Moving Vehicles system for 2D LIDAR sensors.

CURRICULUM VITAE



Name Surname : Berk Güler
Place and Date of Birth : Istanbul, TURKEY
E-mail : gulerb15@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2020, Istanbul Technical University, Faculty of Electrical and Electronic Engineering, Control and Automation Engineering

PROFESSIONAL EXPERIENCE:

- Oct, 2019 – Jun, 2020 Part-Time Working Student at Siemens A.Ş.
- Sep, 2019 – Jan, 2020 Undergraduate Teaching Assistant at Istanbul Technical University
- Aug, 2019 – Sep, 2019 Summer Intern at Baykar Technologies
- Jul, 2018 – Aug, 2018 Summer Intern at Arçelik A.Ş.