

**REPUBLIC OF TURKEY  
AYDIN ADNAN MENDERES UNIVERSITY  
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
MECHANICAL ENGINEERING  
2020-Ph.D.-003**



**DEVELOPMENT OF A ROBOTIC SYSTEM  
WITH HYBRID LOCOMOTION FOR  
BOTH INDOOR AND OUTDOOR  
FIRE DETECTION OPERATIONS**

**Hilmi Saygin SUCUOGLU**

**Supervisor:  
Prof. Dr. Ismail BOGREKCI**

**AYDIN**



**REPUBLIC OF TURKEY**  
**AYDIN ADNAN MENDERES UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**  
**AYDIN**

The thesis with the title of “Development of a Robotic System with Hybrid Locomotion for Both Indoor and Outdoor Fire Detection Operations ” prepared by the Hilmi Saygin SUCUOGLU, PhD Student at the Mechanical Engineering Program at the Department of Mechanical Engineering was accepted by the jury members whose names and titles presented below as a result of thesis defense on 13.01.2020.

	<b>Title, Name Surname</b>	<b>Institution</b>	<b>Signature</b>
President :	Prof. Dr. Ismail BOGREKCI	Aydin Adnan Menderes University	
Member :	Prof. Dr. Zeki KIRAL	Dokuz Eylul University	
Member :	Prof. Dr. Hasan OZTURK	Dokuz Eylul University	
Member :	Assoc. Prof. Dr. Pinar DEMIRCIOGLU	Aydin Adnan Menderes University	
Member :	Asst. Prof. Dr. Adem OZCELIK	Aydin Adnan Menderes University	

This PhD Thesis accepted by the jury members is endorsed by the decision of the Institute Board Members with ..... Serial Number and ..... date.

Prof. Dr. Gönül AYDIN  
Institute Director



**REPUBLIC OF TURKEY**  
**AYDIN ADNAN MENDERES UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**  
**AYDIN**

I hereby declare that all information and results reported in this thesis have been obtained by my part as a result of truthful experiments and observations carried out by the scientific methods, and that I referenced appropriately and completely all data, thought, result information which do not belong my part within this study by virtue of scientific ethical codes.

13/01/2020

Hilmi Saygin SUCUOGLU



## ÖZET

### AÇIK VE KAPALI ALANLARDA YANGIN ALGIMA YETENEĞİNE SAHİP HİBRİT LOKOMOSYONLU BİR ROBOT SİSTEMİNİN GELİŞTİRİLMESİ

Hilmi Saygın SUCUOĞLU

Doktora Tezi, Makine Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. İsmail BÖĞREKÇİ

2020, 247 sayfa

Bu tez çalışmasında; merdiven çıkma, engele tırmanabilme ve yangın algılama yeteneklerine sahip bir robotik sistem geliştirilmiştir. Robotik sistem kapalı ve açık alanlarda çalışma yeteneğine sahip olacak şekilde tasarlanmış ve üretilmiştir. Robotun hareket sistemi hibrit olacak şekilde; üç tekerlekli bacak sistemi ile oluşturulmuş, böylece hem merdiven tırmanmasına hem de düz yolda ilerlemesine imkan sağlanmıştır. Üç tekerlekli bacak sistemi için gerekli matematiksel modeller geliştirilmiştir. Tez çalışması için önerilen “Yön Tabanlı Açık Hesaplama” yaklaşımının gerekli deneysel testleri robotik sisteme uygulanmıştır. Ayrıca, robotun karşılaştığı engelleri boyuna ve şekline göre sınıflandıran bir algoritma önerilmiş ve geliştirilmiştir. Bunların yanı sıra, yangın kaynağını bulan ve kaynağın yangın olma olasılığını belirleyen algoritmalar tasarlanmıştır. Robotun mekanik sistemlerine ve algoritmalarına çeşitli deneysel testler uygulanmıştır (Hareket ve transmisyon sistemleri performans testleri, lokal güzergah planlama ve engelden kaçınma, hareket türü belirleme, yangın algılama ve tespit). Bu testlerin sonucunda; mekanik iletim ve hareket sistemlerinin merdiven çıkma ve düz yolda ilerleme uygulamaları için yeterli olduğu sonucuna varılmıştır. Ayrıca geliştirilen şekil tabanlı açı hesaplama yaklaşımının, güzergah planlama ve engelden kaçınma işlemleri için uygun olduğu gözlemlenmiştir. Faster R-CNN modeli kullanılarak geliştirilen yangın algılama algoritmasının yangın kaynağını %93 doğrulukla tespit ettiği gözlemlenmiştir.

**Anahtar Kelimeler:** Engelden Kaçınma, Faster R-CNN Derin Öğrenme Modeli, Hibrit Lokomasyon, Lokal Güzergah Planlama, Yangın Algılama Robotu, Yangın Arama ve Bulma





## ABSTRACT

### DEVELOPMENT OF A ROBOTIC SYSTEM WITH HYBRID LOCOMOTION FOR BOTH INDOOR AND OUTDOOR FIRE DETECTION OPERATIONS

Hilmi Saygin SUCUOGLU

PhD Thesis, Department of Mechanical Engineering

Supervisor: Prof. Dr. Ismail BOGREKCI

2020, 247 pages

In this thesis; a robotic system with ladder & obstacle climbing and fire detection capabilities was developed. The robotic system was designed and produced as adaptable for both indoor and outdoor applications. An adaptive three wheel legged locomotion system to provide the obstacle climbing and straight motions was developed. The mathematical models were derived for the hybrid locomotion system. “Direction Based Angle Calculation” approach was proposed and the required experimental tests were applied to check the performance of the robotic system with the developed algorithm. A motion decision algorithm to classify the obstacles as ladder, negligible or negotiable according to the height and shape of the obstacle was structured. “Fire Search and Find” and “Fire Detection” algorithms were constructed to find the fire candidates and to determine the probabilities. Several experimental tests were applied to the mechanicals systems and algorithms of the ladder climbing fire detection robot (Performance tests of motion and transmission systems, local path planning and obstacle avoidance, motion mode decision, fire detection). The obtained test results showed that the transmission and motion systems are capable for both ladder climbing and linear motions. The direction based angle calculation approach is suitable and satisfactory for local path planning and obstacle avoidance applications. The developed fire detection algorithm using the Faster R-CNN deep learning model also determines the probability of a fire source with the accuracy of 93%.

**Key Words:** Obstacle Avoidance, Faster R-CNN Deep Learning Model, Hybrid Locomotion, Local Path Planning, Fire Detection Robot, Fire Search and Find



## ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor, Prof. Dr. Ismail BOGREKCI who was abundantly helpful and offered invaluable support with his sincerity and belief in me. In addition, I am especially grateful to Assoc. Prof. Dr. Pinar DEMIRCIOGLU for her useful advices and comments.

I would also like to express my sincere thanks to thesis committee and jury members Prof. Dr. Zeki KIRAL, Prof. Dr. Hasan OZTURK and Asst. Prof. Dr. Adem OZCELIK for their valuable recommendations.

I wish to acknowledge the efforts of my friends Burak KAYA, Neslihan DEMIR, Ogulcan TURHANLAR, Emrah GUVEN and Emre ALTUN.

In addition, I would like to thank my lovely families of SUCUOGLU and KULAH, for their invaluable efforts when I felt hopeless and weak in solving problems.

Finally, I would like to send my special thanks to my lovely wife Nur SUCUOGLU for her understanding, help and encouragement and faith in me.

Hilmi Saygin SUCUOGLU



## TABLE OF CONTENTS

ÖZET.....	vii
ABSTRACT.....	ix
ACKNOWLEDGEMENTS.....	xi
TABLE OF CONTENTS.....	xiii
LIST OF ABBREVIATIONS.....	xvii
LIST OF FIGURES.....	xix
LIST OF TABLES.....	xxix
1. INTRODUCTION.....	1
1.1. Problem Description.....	1
1.2. Motivation.....	1
1.3. Objectives.....	2
1.4. Contributions.....	2
1.5. Hypothesis.....	3
1.6. Overview of the Thesis.....	3
1.7. Robotic System Overview.....	4
2. LITERATURE REVIEW.....	6
2.1. Firefighting and Detection Robots.....	6
2.2. Locomotion Systems of Robots.....	16
2.3. Obstacle Avoidance and Path Planning Systems.....	25
2.4. Sensor Based Fire Detection Methods.....	33

2.5. Computer Vision and Deep Learning Based Fire Detection Methods .....	34
3. MATERIAL AND METHOD.....	40
3.1. General System Architecture and Functions .....	40
3.2. The Mechanical Design and the Components .....	42
3.2.1. General Dimensions and the Weight.....	46
3.2.2. Material and Manufacturing of the Components.....	47
3.3. Motion Calculations and Analyses .....	60
3.4. Mechanical Strength Check of the Components .....	70
3.5. Electronic Systems and Hardware.....	77
3.6. Algorithms and Software.....	87
3.6.1. Motion Mode Decision.....	87
3.6.2. Direction Based Angle Calculation .....	89
3.6.3. Fire Search and Find.....	91
3.6.4. Fire Detection .....	92
4. RESULTS AND DISCUSSION.....	97
4.1. Performances of Transmission and Motion Systems .....	97
4.2. Lidar Interface .....	100
4.3. Fire Detector.....	104
4.4. Tests of the Algorithms .....	104
4.4.1. Direction Based Angle Calculation .....	106
4.4.2. Fire Search and Find.....	115

4.4.3. Fire Detection.....	121
4.5. Tests of Ladder Climbing Robot.....	126
5. CONCLUSIONS.....	128
6. RECOMMENDATIONS AND FUTURE WORKS.....	129
REFERENCES.....	130
APPENDICES.....	146
Appendix 1 (Matlab Scripts).....	146
Motion Calculations.....	146
Mechanical Strength Check of the Components.....	152
Appendix 2 (C# Codes).....	163
Serial Control Software.....	163
Appendix 3 (Arduino Codes).....	195
Control of the Transmission and Motion Systems.....	195
Appendix 4 (Faster R-CNN).....	209
Training.....	209
Implementation.....	215
Appendix 5 (Python Codes).....	219
Robot Control Software.....	219
RESUME.....	243





## LIST OF ABBREVIATIONS

A*	A Star
Ah	Ampere Hour
CNN	Convolutional Neural Network
CSV	Comma Separated Value
DOF	Degrees of Freedom
DWA	Dynamic Window Approach
FDM	Fused Deposition Modelling
FGM	Follow the Gap
GPIO	General Purpose Input Output
GPS	Global Positioning System
GUI	Graphical User Interface
Hi	Hit Point
HIS	Hue Intensity Saturation
IR	Infrared
Li	Leave Point
nm	Nanometer
PFM	Potential Field Method
PID	Proportional Integrate Derivative
PLA	Polylactic Acid
PWM	Pulse Width Modulation
R-CNN	Regional Convolutional Neural Network
RGB	Reg Green Blue
Rpm	Rotation per Minute
RPN	Regional Proposal Network
SVM	Support Vector Machine
UV	Ultraviolet
VFH	Vector Field Histogram
VNC	Virtual Network Computing
VO	Velocity Obstacle
Wh	Watt Hour
ZMP	Zero Moment Point



## LIST OF FIGURES

Figure 1.1 Developed Ladder Climbing Fire Detection Robot (Personal Archive)	5
Figure 2.1 LUF60 Firefighting Robot (Tan et al., 2013) .....	6
Figure 2.2 Firemote 4800 (Kim et al., 2015).....	7
Figure 2.3 Archibot (Hong et al., 2012).....	7
Figure 2.4 FFR-1 Firefighting Robot (Levendis et al., 2011) .....	8
Figure 2.5 Thermite T3 (Tan et al., 2013; Jia et al., 2018) .....	8
Figure 2.6 FireRob (Tan et al., 2013; Aliff et al., 2019).....	9
Figure 2.7 MVF-5 Firefighting Robot (Wagoner et al., 2015; Lakshmi et al., 2018) .....	9
Figure 2.8 Colossus (Selek et al., 2019, Edlinger et al; 2019) .....	10
Figure 2.9 Rainbow 5 (Miyazawa, 2012).....	10
Figure 2.10 JMX-LT50 Robot (Nikitin et al., 2019).....	11
Figure 2.11 Anna Konda (Liljebäck et al., 2006).....	11
Figure 2.12 Intelligent Security System (Luo and Su, 2007).....	12
Figure 2.13 AFFMP (Khoon et al., 2012) .....	12
Figure 2.14 Hoyarobot (Kim et al., 2009).....	13
Figure 2.15 QRob (Aliff et al., 2019).....	13
Figure 2.16 Firefighting Robot of Rakip (Rakip and Sarkar, 2015) .....	14
Figure 2.17 Firefighting Robot of Hassanein (Hassanein et al., 2015) .....	14
Figure 2.18 Fire Detection & Fighting Robot of Perumal (Perumal et al., 2019)..	15

Figure 2.19 Octavia Robot (Martinson et al., 2012).....	15
Figure 2.20 Fire Detection Robot of Sucuoglu (Sucuoglu et al., 2018) .....	15
Figure 2.21 Operation Environment Samples (a) Aerial, (b) Underwater, (c) Urban (Chen et al., 2009) .....	16
Figure 2.22 First Dynamically Stable System by Raibert (Garcia et al., 2006) ....	17
Figure 2.23. Gait types (a) Statically Stable, (b) Dynamically Stable.....	17
Figure 2.24 Wabot 2 (Lim and Takanishi, 2006; Behnke, 2008) .....	18
Figure 2.25 Passive Dynamic Bipedal Locomotion Element .....	19
Figure 2.26 Comparison of the Locomotion Systems (Bruzzone and Quaglia, 2012) .....	20
Figure 2.27 Mantis Robot (Luca and Pietro, 2014).....	22
Figure 2.28 Wheeleg (Tadakuma et al., 2010) .....	22
Figure 2.29 Robotic System with Retractable Modules (Tadakuma et al., 2010).	23
Figure 2.30 Quadruped Robotic System Lebel (Gonzalez et al., 2011).....	23
Figure 2.31 Space Cat (Siegwart et al., 1998).....	24
Figure 2.32 MSRox Robot (Dalvand and Moghadam, 2006) .....	24
Figure 2.33 Bug1 Algorithm (Oroko and Nyakoe, 2012; Minguez et.al., 2008; Nguyen and Le, 2016; Lumelsky, 1987) .....	26
Figure 2.34 Bug2 Algorithm (Lumelsky and Skewis, 1990; Nguyen and Le, 2016) .....	27
Figure 2.35 Tangent Bug Algorithm Choset, 2005; Oroko and Nyakoe, 2012) ...	27
Figure 2.36 Potential Field Method (Xi an Chaoi, 2013; Nguyen and Le, 2016; Khatib, 1985).....	28

Figure 2.37 Vector Field Histogram (Koren and Borenstein, 1991; Laumond, 1993; Laumond and Risler, 1996; Babinec et al., 2014).....	29
Figure 2.38 Velocity Obstacle Approach (Fiorini and Shiller, 1998; Cheng and LaValle, 2001; Zhong and Zhou, 2011; Kluge and Prassler, 2007).....	30
Figure 2.39 Dynamic Window Approach (Alsaab, 2015; Simmons, 1996) .....	31
Figure 2.40 Follow the Gap Method (Demir and Sezer, 2017; Kim, 2014).....	31
Figure 2.41 Application of Dijkstra Algorithm (Jasika et al., 2010; Alsaab, 2015) .....	32
Figure 2.42 Wavefront Algorithm (Zidane and Ibrahim, 2017; Alsaab, 2015) .....	33
Figure 2.43 Typical Application of Smoke Detector (Porteous, 2011).....	34
Figure 2.44 Computer Vision Based Fire Detection Software (Cetin, 2007) .....	35
Figure 2.45 Moving Object Detection with Background Subtraction (Cetin, 2013) .....	36
Figure 2.46 Dynamic Texture and Pattern Analysis for Fire Detection (Toreyin et al., 2007).....	37
Figure 2.47 CNN Structure for Fire Detection in Video Stream (Frizzi et al., 2016) .....	38
Figure 2.48 The Algorithm Schematic of Proposed Approach (Zhong et al., 2018) .....	38
Figure 2.49 Test Results of Fire Detector (Sucuoglu et al., 2019).....	39
Figure 3.1 The Developed Robotic System (Personal Archive) .....	41
Figure 3.2 Designed Modular Chassis .....	43
Figure 3.3 Transmission of Robotic System .....	43
Figure 3.4 General Design View .....	44

Figure 3.5 Exploded View of Designed Robotic System.....	46
Figure 3.6 General Dimension of Design (in mm).....	47
Figure 3.7 Off-Road Type Wheels (Pololu Robotics & Electronics, 2018 a).....	48
Figure 3.8 Housing Rubbers.....	48
Figure 3.9 Connectors for Shaft to Wheel Connection .....	49
Figure 3.10 Bearings for inner and outer gear shafts housing (Makparsan, 2017 a) .....	49
Figure 3.11 The Outer Cover.....	50
Figure 3.12 The Inner and Outer Gear Shafts .....	50
Figure 3.13 The Manufactured Helical Type Main Gear .....	51
Figure 3.14 Technical Drawing of the Gears .....	53
Figure 3.15 The Main Gear Shafts .....	53
Figure 3.16 The Connector Couplings .....	54
Figure 3.17 Linear Motion Ball Bearings (Sahin Rulman, 2017).....	54
Figure 3.18 Axial Ball Bearing Sets (Makparsan, 2017 b) .....	55
Figure 3.19 Side Covers .....	55
Figure 3.20 Side Frames.....	56
Figure 3.21 DC Motor Covers.....	56
Figure 3.22 Connection Brackets .....	57
Figure 3.23 Designed Linear Actuation Mechanism.....	57
Figure 3.24 Designed Motion System .....	58

Figure 3.25 Designed and Manufactured Transmission System .....	60
Figure 3.26 Rolling Down on an Inclined Plane Assumption.....	61
Figure 3.27 The Condition of Equilibrium.....	61
Figure 3.28 The Torque and Acceleration Calculations for Rolling Motion .....	65
Figure 3.29 Angular Velocity and Power Calculations for Rolling Motion .....	66
Figure 3.30 The Forces in Linear Motion .....	67
Figure 3.31 Angular Velocity, Torque and Power Calculations for Linear Motion .....	68
Figure 3.32 Simulation of Climbing Motion.....	68
Figure 3.33 Simulation of Linear Motion .....	69
Figure 3.34 Selected DC Motor for Actuation (Robotistan, 2018 a) .....	69
Figure 3.35 Lipo Batteries for Energy Supply (Robotistan, 2018 b) .....	70
Figure 3.36 The Affected Forces on the Gears .....	70
Figure 3.37 Technical Drawing of the Main Shaft with Shear and Moment Diagram .....	71
Figure 3.38 The Minimum Shaft Diameter Calculation of Main Shaft .....	72
Figure 3.39 Structural Analysis of Main Shaft.....	73
Figure 3.40 Analysis of the Gear .....	75
Figure 3.41 Coupling Pin Diameter Calculation .....	76
Figure 3.42 Motion and Strength Calculator.....	77
Figure 3.43 Circuit Design of the Motion Control System .....	78
Figure 3.44 Hokuyo 30-LX Lidar (Hokuyo, 2019).....	80

Figure 3.45 Scan Angle of Hokuyo 30-LX Lidar (Kazuya, 2015).....	80
Figure 3.46 Raspberry pi 3b (Robotistan, 2018 c) .....	82
Figure 3.47 IR Flame Sensor (Robotistan, 2018 d).....	82
Figure 3.48 Lipo Battery for Lidar (Robotistan, 2018 e) .....	82
Figure 3.49 Ultrasonic Distance Sensor (Robotistan, 2018 f).....	83
Figure 3.50 Xiaomi Power bank (Xiaomi, 2019).....	83
Figure 3.51 Arduino Mega 2560 (Robotistan, 2018 g) .....	84
Figure 3.52 Pololu Dual Motor Driver (Pololu, 2019).....	84
Figure 3.53 DC Motor of Motion System (Robotistan, 2018 a).....	85
Figure 3.54 DC Motor of Linear Actuation System (Robotistan, 2018 h).....	85
Figure 3.55 Fire Detection Webcam (A4 Tech, 2019).....	86
Figure 3.56 The Dimensions of Ladder Step (Megep, 2006).....	87
Figure 3.57 The Flowchart of the Motion Type Decision .....	88
Figure 3.58 The Defined Regions in Direction Based Angle Calculation .....	89
Figure 3.59 The Situation and the Reaction of the System in Direction Based Angle Calculation Approach.....	90
Figure 3.60 The Flowchart of the Fire Search and Find Algorithm .....	92
Figure 3.61 The Structure of CNN (Sucuoglu et al., 2019).....	93
Figure 3.62 The Selected Images from the Training Process (Personal Archive). 94	
Figure 3.63 Total Loss Values in Training Process.....	94
Figure 3.64 Implementation of Fire Detector Software .....	95



Figure 3.65 Serial Control Software .....	95
Figure 3.66 Ladder Climbing Robot Control Software.....	96
Figure 4.1 Wooden Test Carrier.....	97
Figure 4.2 General Test Setup (Personal Archive).....	98
Figure 4.3 Linear Motion Test (Personal Archive) .....	99
Figure 4.4 Climbing Motion Test (Personal Archive) .....	100
Figure 4.5 The Condition of the Stop Area (Personal Archive).....	101
Figure 4.6 The Distance Measurement in UrgBenri Software .....	102
Figure 4.7 Stop Area Reaction .....	102
Figure 4.8 Front Right Obstacle Condition (Personal Archive).....	103
Figure 4.9 Front Right Obstacle Reaction.....	103
Figure 4.10 Implementation of Fire Detector Software (Personal Archive) .....	104
Figure 4.11 Test Robot (Personal Archive) .....	105
Figure 4.12 General Dimensions of Test Robot(Personal Archive).....	105
Figure 4.13 Plan Drawing of the First Test Bed.....	106
Figure 4.14 Test Bed with Front Left & Right Obstacles and Left Corner (Personal Archive).....	107
Figure 4.15 Experimental Results from Test Bed One (a) Front Left Obstacle, (b) Left Corner (Personal Archive).....	107
Figure 4.16 Plan Drawing of the Second Test Bed .....	108
Figure 4.17 Test Bed with Right & Left Corners and Right Obstacle .....	108
Figure 4.18 Experimental Results from Test Bed Two.....	109

Figure 4.19 Plan Drawing of the Third Test Bed .....	110
Figure 4.20 Test Bed with Right & Left Corners and Front Right & Left Obstacles (Personal Archive).....	110
Figure 4.21 Experimental Results from Test Bed Three .....	111
Figure 4.22 Plan Drawing of the S Type Test Bed.....	112
Figure 4.23 S Type Test Bed (Personal Archive).....	112
Figure 4.24 Experimental Results from the S Type Test Bed .....	113
Figure 4.25 Test Bed with Complicated Obstacles (Personal Archive) .....	113
Figure 4.26 Test Bed with Complicated Obstacles .....	114
Figure 4.27 First Case of Dynamic Obstacle Avoidance (Personal Archive) .....	114
Figure 4.28 Second Case of Dynamic Obstacle Avoidance (Personal Archive). .....	114
Figure 4.29 Placement of the IR Sensors (Personal Archive) .....	116
Figure 4.30 Front Fire Source Detection Process (a) Placement of the Fire Source, (b) Detection of the Fire Source in Front of the Robotic System, (c) The motion of the Robotic System After Fire Source is Removed (Personal Archive).....	117
Figure 4.31 Right Side Fire Source Detection Process (a) Placement of the Fire Source at the Right Side, (b) Right Turning Motion to Detect the Fire Source, (c) Moving Forward Action, (d) Second Right Turning to Find the Fire Source, (e) Detection of the Fire Source in Front (Personal Archive).....	119
Figure 4.32 Left Side Fire Source Detection Process (a) Placement of Fire Source at Left Side Maze, (b) Left Turning Motion to Detect the Fire Source, (c) Detection of the Fire Source in the Maze (Personal Archive) .....	120
Figure 4.33 Different Fire Sources Used in Fire Detection Test.....	122
Figure 4.34 Detection of the Fire Sources with Fire Detection Algorithm .....	123

Figure 4.35 Obtained Fire Probabilities in Detection Process (a) Candle Fire (b) Wood Fire, (c) Complex Fire .....125

Figure 4.36 Obstacle Avoidance and Local Path Planning Tests (a) Front Left Obstacle, (b) Front Obstacle, (c) Left Corner (Personal Archive) .....126

Figure 4.37 Ladder Climbing Test (Personal Archive).....127





**LIST OF TABLES**

Table 1.1 General System Specifications.....	4
Table 3.1 Component List of the Robotic System .....	44
Table 3.2 General Dimensions .....	46
Table 3.3 The Dimensions of the Gears .....	52
Table 3.4 The Components of the Motion System.....	58
Table 3.5 The Components of the Transmission System.....	59
Table 3.6 Components of the Motion Control System.....	77
Table 3.7 Technical Specifications of Hokuyo Lidar.....	79
Table 3.8 Technical Specifications of Raspberry pi 3b (Shinde and Mane, 2015)	81



# 1. INTRODUCTION

## 1.1. Problem Description

Fire is a type of disaster that causes to harm in areas such as home, industrial zones and forest. Although firefighting includes the dangerous tasks such as fire extinguishment and victim rescue, it is still conducted by human operators so the firefighters jeopardize their life. It is impossible to control a fire disaster after it spreads and so hard to recover a damaged area. Therefore, the best way for firefighting is to search and detect it, before it reaches the point of no return. Many different devices and systems have been designed and developed for the purpose of early fire detection. Optical flame and smoke detectors, camera systems equipped with fire detection algorithms and capability are the different types of early fire detection devices. These systems have some disadvantages such as high cost, necessity of fixing and inflexibility. This situation leads to usage of mobile systems for the purpose of early fire detection.

## 1.2. Motivation

When the relationships between this thesis study and real life applications are observed, the requirements can be listed as below:

1. Using a mobile system can reduce the cost of the fire detection and protection systems. To establish fixed equipment require many expensive devices such as cameras, sensors and control center which increase the investment cost.
2. The fixed systems can be used for only one area. However, mobile system can be adaptable for many different places; it can be indoor or outdoor according to adjustments and settings.
3. Currently, the work safety in industrial zones is a hot topic discussed everyday by everyone. The developed mobile robotic system can serve for work safety applications with the issue of fire protection.
4. There are firefighting and fire detection mobile systems in literature. However, many of them need the human operator for operation and others need physical path lines for the motion. The autonomous capability is required to create more reliable results from the fire detection and protection operations.

The listed below are motivated us to design and manufacture an autonomous mobile system for fire detection.

### **1.3. Objectives**

- 1.To design and fabricate a mobile robot system that is adaptable for both indoor and outdoor environments.
- 2.To develop an adaptive three wheel-legged hybrid locomotion mechanism with a transition system that provide motion for both ladder climbing, terrestrial and indoor environments.
- 3.To derive mathematical model for developed tracking system, kinematic and dynamic analyses.
- 4.To develop static and dynamic obstacle avoidance system with direction based angle calculation approach.
- 5.To design and develop a decision mechanism for obstacle classification. According to the results of this algorithm, system is capable to climb ladders or move fast in indoor straight and outdoor terrestrial environments. For straight or off-road usage system is tracked with two wheels and other wheel is disabled with mechanic system and decision algorithm. For ladder climbing mission; three-wheel rotation is activated and rotation movement is applied to the mechanism.
- 6.To develop fire search and find & fire detection algorithms. According to the sensor fusion data, system decides that it may be a fire at the front, left or right sides. Then, the developed algorithm for fire detection using Faster R-CNN (Faster Regional Convolutional Neural Network) model decides the exact fire probability.

### **1.4. Contributions**

- 1.Mathematical-kinematic models, design and fabrication of three wheel-legged hybrid locomotion system that can be used for both indoor and outdoor applications.
- 2.Mathematical model and algorithm of “Direction based angle calculation approach” for obstacle avoidance and local path planning.
- 3.A decision control mechanism named as “Obstacle classification” that activates or disables the transition system for climb ladder climbing or fast motion at indoor and outdoor environments.



4. “Fire Search and Find” algorithm for fire detection; a sensor fusion algorithm is developed to increase the process efficiency, detection of fire tendency.
5. Development of “Fire detection” algorithm using Faster R-CNN model to decide fire probability.

## **1.5. Hypothesis**

The hypothesis of this thesis is to develop a robotic system with hybrid locomotion that has fire detection unit and operated with direction based angle calculation approach for obstacle avoidance and path planning.

## **1.6. Overview of the Thesis**

The outline of the thesis is as follows;

In the next chapter “Literature Review”,

1. The firefighting and detection robots are presented,
2. Locomotion systems of robots are described,
3. Obstacle avoidance and path planning systems are explained,
4. Computer vision and Deep Learning based fire detection methods are presented.

In the third chapter “Material and Method”,

1. General system architecture and the functions are explained,
2. The mechanical design and the components of the robotic system are presented,
3. Motion calculations and analyses are presented,
4. Strength check of the components are explained,
5. Design and the development of the electronic systems and hardware are described,
6. Development of the algorithms and software are explained.

In the fourth chapter “Results and Discussion”,

1. The experimental tests are described,
2. The results from the tests are presented and discussed.

In the fifth chapter “Conclusion”. The conclusions of this thesis study are explained. In the last chapter “Recommendation and Future Works”. Recommendations and the possible future works are discussed.

## 1.7. Robotic System Overview

The general view and system specifications of the designed and developed mobile robotic system are given in Table 1.1 and Figure 1.1, respectively.

Table 1.1 General System Specifications  
General Dimensions and Weight

<b>Total Weight</b>	10.5 kg
<b>Maximum length</b>	792 mm
<b>Maximum height</b>	346 mm
<b>Maximum width</b>	466 mm
<b>Maximum radius</b>	210 mm
<b>Actuation System Features</b>	
<b>Number of motor</b>	4
<b>Motor operating voltage</b>	12 V DC
<b>Maximum operation current per one DC motor</b>	3 Amperes
<b>Power per one DC motor</b>	40 Watt
<b>Torque per one DC motor</b>	4.2 Nm
<b>Control Structure</b>	
<b>Obstacle Avoidance Unit</b>	Hokuyo UTM 30 LX Lidar
<b>Main Controller</b>	Raspberry Pi 3b
<b>Motion Control Unit</b>	Arduino Mega 2560
<b>Fire Detection Unit</b>	Infrared sensors and webcam

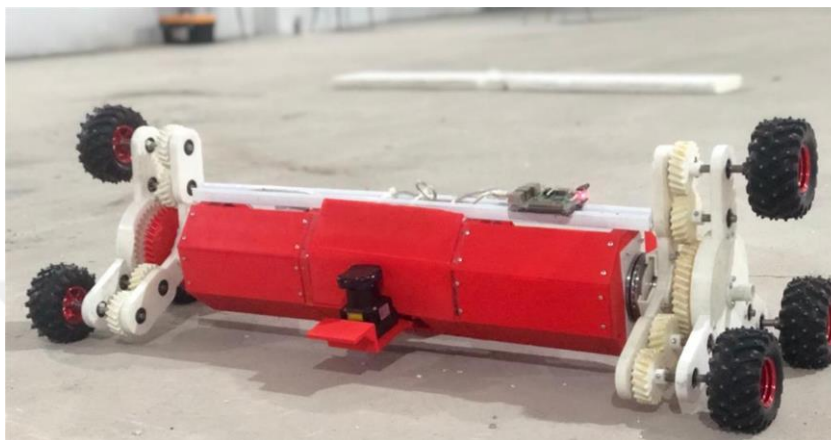


Figure 1.1 Developed Ladder Climbing Fire Detection Robot  
(Personal Archive)

## 2. LITERATURE REVIEW

Despite the precaution, protection applications and safety knowledge all over the world the human still confronts with the fire disaster and hazards. This causes to environmental pollution, spreading of carcinogenic material and loss of living quarters, forests and life (Mishra et al., 2013; Haukur et al., 2010). Firefighting is a dangerous task for the firefighters. They are injured and death caused by exposure to fire conditions. Smoke inhalation, burns of a building, being trapped in an area can be the reasons. (Ronchi et al., 2013; Wright et al., 2001). To handle these situations; effective technologies and methods are required for fire detection & extinguishment and victim rescue operations (Bertram et al., 2013). In order to help to firefighters, to reduce the loss of life and to increase the operation performance, fire detection and fighting robots have been developed in scientific and industrial areas.

### 2.1. Firefighting and Detection Robots

The Robot LUF 60 (Figure 2.1) was developed in Germany. This firefighting machine was powered with a diesel engine and it was equipped with air blower and water beam fog to destroy the dangerous obstacles and extinguish the fire sources. This machine was especially designed to operate in the areas of aircraft hangers, garages and chemical plants. The monitor nozzle system of the LUF60 machine had the capability to reach flow rate of 3,000 liters per minute with the range of 80 meters. This machine also could climb the stair and operate in the maximum environmental temperature of 230 °C (Tan et al., 2013).



Figure 2.1 LUF60 Firefighting Robot (Tan et al., 2013)

Firemote 4800 (Figure 2.2) was developed by Ryland Research Development Company in China. Firemote was an electrically powered unmanned ground vehicle. It had capability to be controlled with a panel via a control camera system that provide the environmental information to the operators. Firemote was designed to operate especially in factories, power plants, aircrafts and tunnels. It was equipped with high pressure water system could reach up to flow rate of 4,800 liters per minute, navigation & thermal cameras and local cooling system (Kim et al., 2015).



Figure 2.2 Firemote 4800 (Kim et al., 2015)

Archibot (Figure 2.3) was designed and manufactured as a firefighting robot in South Korea. It was especially designed to use in the areas that hard and dangerous to reach by human operator. It was equipped with a special suspension system for ladder climbing. This machine had a waterproof body and cooling system (Hong et al., 2012).



Figure 2.3 Archibot (Hong et al., 2012)

FFR-1 firefighting robot (Figure 2.4) was manufactured in United States as a supporter of the firefighters. It could be controlled with a remote controller panel.

This robot was suitable to operate in narrow streets, industrial buildings and military installations. It was equipped with internal cooling system to protect the body from high environmental temperature (Levendis et al., 2011).



Figure 2.4 FFR-1 Firefighting Robot (Levendis et al., 2011)

Thermite version T3 (Figure 2.5) was designed and fabricated in United States for hazardous substance, forest and chemical plant fires. It had 2,200 liters per minute pumping capacity and high temperature durability. It could also be controlled remotely up to distance of 400 meters (Tan et al., 2013; Jia et al., 2018).



Figure 2.5 Thermite T3 (Tan et al., 2013; Jia et al., 2018)

FireRob (Figure 2.6) was a firefighting robot could be used for search and seek in fire environment and controlled via remote controller. It was equipped with heat shield to protect the body from environmental conditions. It had thermal camera and heat sensors to observe the fire environment and provide information to the human operator (Tan et al., 2013; Aliff et al., 2019).



Figure 2.6 FireRob (Tan et al., 2013; Aliff et al., 2019)

MVF-5 (Figure 2.7) was a firefighting robot manufactured by DOK-ING Company in Croatia. It was fabricated to control the fires especially in unreachable places. It could be remotely controlled with human operator and equipped with GPS (Global Positioning System) for the location information. It had water pump capability up to range of 55 meters. MVF-5 had fire durability of 400 °C for 30 and 700 °C for 15 minutes. It could carry 2,200 liters waters and 500 liters foam in the storage tanks (Wagoner et al., 2015; Lakshmi et al., 2018).



Figure 2.7 MVF-5 Firefighting Robot (Wagoner et al., 2015; Lakshmi et al., 2018)

Colossus (Figure 2.8) was developed in France by the company of Shark Robotics. It had capability to extinguish the fire source, to climb the the stairs and it could be remotely controlled up to 300 meters range. The weight of the Colossus was about 400 kg and it had a payload of 1,000 kg. It was actuated with electric motors with the power of 8 kilowatts (Selek et al., 2019, Edlinger et al; 2019).



Figure 2.8 Colossus (Selek et al., 2019, Edlinger et al; 2019)

The Rainbow 5 (Figure 2.9) was developed in Japan for the fighting of complex, oil and aircraft crash fires. It was equipped with a system to clear the obstacle such as fallen objects and dangerous materials. It could extinguish the fire source using water and foam with the maximum flow rate of 5,000 and 3,000 liters per minute, respectively (Miyazawa, 2012).



Figure 2.9 Rainbow 5 (Miyazawa, 2012)

JMX-LT50 (Figure 2.10) was firefighting robot manufactured in China. It had remote control capability. It had also a mist spraying system to protect the body from high temperature. The traction system of the ground machine were established with wheel tire to overcome different kind of the obstacles (Nikitin et al., 2019).





Figure 2.10 JMX-LT50 Robot (Nikitin et al., 2019)

The mentioned above robots had remotely control ability and they were suitable for the usage as an assistant for the human firefighters. Remotely controlled systems could provide some advantages to human operators such as environment observations and to help to make decision about situation from the safe area. The most important one was the preventing the human firefighter life from the injuries and death. However, these systems had some limitations. The human operator had to rely on the limited information from the firefighting vehicle. Another, some communication problem could be occurred between the human operator and the vehicle that caused to unstable connection and insufficient information about the fire environment (Kim et al., 2015; Hong et al., 2012, Chang et al., 2006).

The semi & fully autonomous systems have been also developed for fire detection and fighting.

Anna Konda (Figure 2.11) was snake type firefighting robot was developed in Norway. It was a snake robot with the length of 3 meters and 70 kg. Snake type robot had washers to extinguish the fire source with the maximum flow rate of 30 liters per minute. Anna Konda was equipped with 20 hydraulic cylinders to actuate the joints. It had also contact force sensors to sense the ground and for adaptive motion according to the operation area (Liljebäck et al., 2006).



Figure 2.11 Anna Konda (Liljebäck et al., 2006)

Luo and Su (2007) developed an intelligent security system with the autonomous navigation (Figure 2.12). It was a cylindrical shape robot with the diameter and height of 500 and 1,400 mm, respectively. The robotic system was equipped with multiple sensors (UV, ionization smoke and temperature) to generate the reliable fire detection data.



Figure 2.12 Intelligent Security System (Luo and Su, 2007)

Khoon et al. (2012) designed and manufactured AFFMP (Autonomous Firefighting Mobile Platform) to detect and extinguish the small size of the fire sources (Figure 2.13). The platform had ability to patrol and monitor the prescribed area that indicated with physical path lines before. The robot was equipped with flame detection, obstacle avoidance sensors and DC fan.

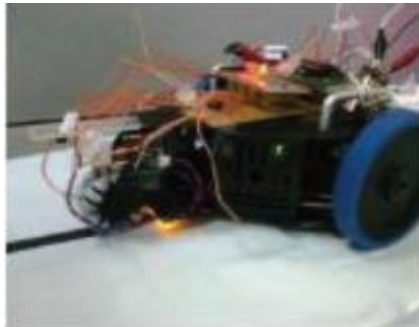


Figure 2.13 AFFMP (Khoon et al., 2012)

Kim et al. (2009) produced a guidance robotic system with the name of Hoyarobot for the purpose of victim detection at indoor environment (Figure 2.14). The robot had ability to explore the environment with obstacle avoidance algorithm autonomously and to provide the information to the operator. By this way, the

operator could control the other system's robot members and could get the information about the environmental situation.



Figure 2.14 Hoyerobot (Kim et al., 2009)

Aliff et al. (2019) developed the autonomous fire detection and extinguishment robot (QRob). The system could find the fire location using the data from flame sensor and avoid the obstacles with ultrasonic sensors. It had also provide the environmental conditions to the operator with smartphone camera (Figure 2.15).



Figure 2.15 QRob (Aliff et al., 2019)

Rakip and Sarkar (2016) designed and produced an autonomous firefighting robot. The fire detection unit was established on multi sensor detection and control of the robot was conducted via PID (Proportional integrate derivative) controller. The body of the system was constructed with fire resistant and waterproof material (Figure 2.16)

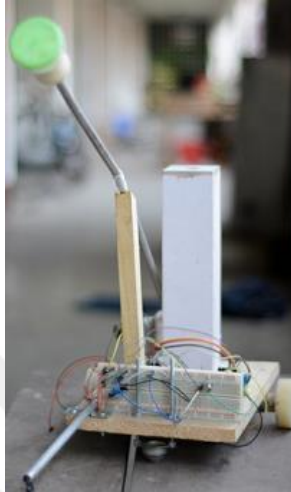


Figure 2.16 Firefighting Robot of Rakip (Rakip and Sarkar, 2015)

Hassanein et al. (2015) developed a firefighting robot with autonomous motion capability. The robotic system was equipped flame sensor for fire detection, infrared & ultrasonic sensors for navigation and a fan for fire extinguishment. A map representation system was also added to robot (Figure 2.17).

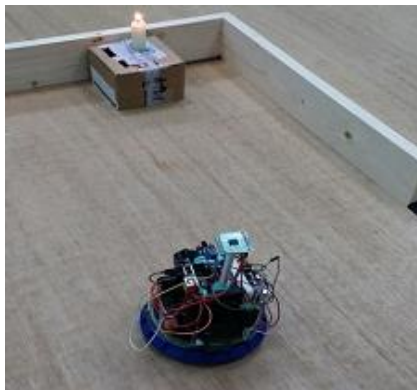


Figure 2.17 Firefighting Robot of Hassanein (Hassanein et al., 2015)

Perumal et al. (2019) fabricated a fire detection & fighting robot. The system was controlled via Raspberry pi and equipped with thermal, infrared cameras for fire detection and water tank for fire extinguishment. The robot had also capability to provide live video recording to save and share the fire extinguishment process (Figure 2.18).



Figure 2.18 Fire Detection & Fighting Robot of Perumal (Perumal et al., 2019)

Martinson et al. (2012) developed Octavia Robot as a team member of firefighters. It was a humanoid robot had the ability of speech recognition, navigation using Lidar (Laser range finder). It could find the exact location of the fire source and extinguish the fire with compressed air foam (Figure 2.19).

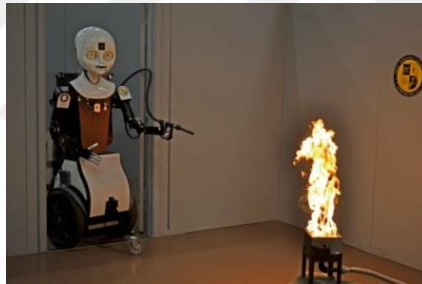


Figure 2.19 Octavia Robot (Martinson et al., 2012)

Sucuoglu et al., (2018) designed and produced a mobile fire detection robot. The robot could track the virtual path line for patrolling while scanning the environment to detect the fire source with sensor fusion. It had also obstacle avoidance capability through placed ultrasonic sensors and developed algorithm (Figure 2.20).

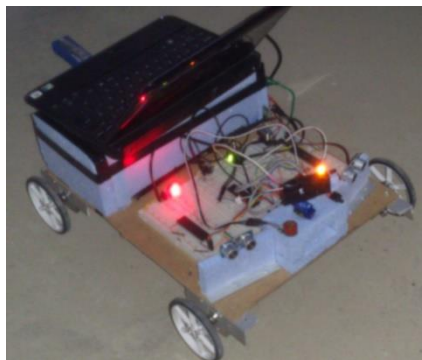


Figure 2.20 Fire Detection Robot of Sucuoglu (Sucuoglu et al., 2018)

## 2.2. Locomotion Systems of Robots

The locomotion systems of the robots are firstly depend on the operation environment. With a simple classification, the operation environment can be aerial, aquatic or terrestrial as shown in Figure 2.21 (Chen et al., 2009).

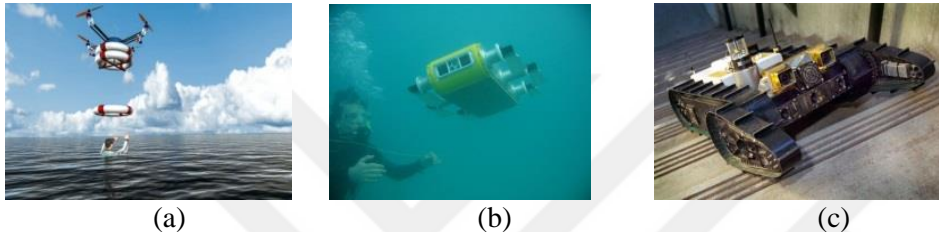


Figure 2.21 Operation Environment Samples (a) Aerial, (b) Underwater, (c) Urban (Chen et al., 2009)

The propellers and screws are generally used for aquatic and aerial operation environments. The necessary motion requirements of terrestrial environment is more complicated than the others (Valgren, 2007; Hahnel et al., 2003). Therefore, different suitable types of the elements have to be selected for the motion of terrestrial environment. These elements can be wheels, tracks, legs and the combination of them according to the necessities (Garcia et al., 2007; Patnaik, 2007). Apart from the wheels, tracks and propellers, there is a different locomotion tool named as adaptive legs for the motion of biologically inspired robots. The motion produced by the legs according to the surface situation soft or hard is walking. This walking motion generally based of the DOF (Degree of Freedom) of the manipulators (Ceccarelli and Carbone, 2005). The walking type of the locomotion systems have some pros and cons as listed below:

1. The robotic systems with legs can overcome the irregular terrain better than the wheeled locomotion. This situation keeps the robotic system motion more stable and protect it from the abrupt motions and effects.
2. For the obstacle avoidance, ladder climbing and stair mobility legs can provide some advantages to robotic system.
3. The robots with leg can easily walk on the surfaces such as loose and sandy terrains.
4. The effects of the environmental damages are less for the walking than the robots with wheels.

Although the listed advantages below, body stability and the walking gait design, are challenges for the robotic systems.

The researches of the legged locomotion systems are generally concentrate on leg motion, coordination and the stability. Basically, if a legged robot can keep its balance, the system is stable. It is the idea of static stability. In the static ability idea, the inertia of the robot limbs are neglected. However, in the real life applications, some inertial effects and the other dynamic issues such as friction are restricted the motion of the robotic system. In order to handle this restriction, dynamic stability has to be taken into account. The first dynamically stable system shown in Figure 2.22 was developed by Raibert in 1986 (Garcia et al., 2006).

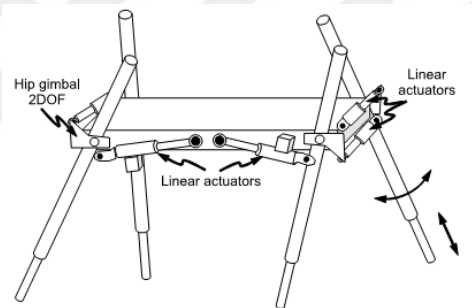


Figure 2.22 First Dynamically Stable System by Raibert (Garcia et al., 2006)

The gait is another important research topic for the legged locomotion systems. The type of gait and the sequence of the legs directly affect the stability and balance of the robotic system. Statically and dynamically stable gaits are two developed methods in the literature as shown in Figure 2.23 (Garcia et al., 2007, Zielinska, 2004).

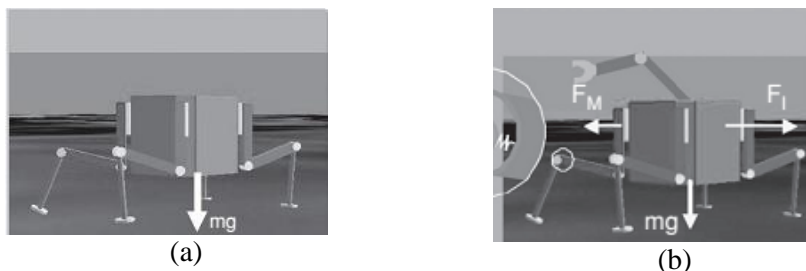


Figure 2.23. Gait types (a) Statically Stable, (b) Dynamically Stable (Garcia et al., 2007, Zielinska, 2004)



In the statically stable gait method, the legged system simply controlled with the heavy limbs. This stability criterion can be classified as periodic and aperiodic. While periodic gait is created with the cyclically repeated movement of predefined sequence, aperiodic gait provides more flexible motion to the system especially for the complicated terrain.

The actual autonomous humanoid robotic systems are required a different locomotion system named as bipedal locomotion. There are some challenges to develop a real humanoid bipedal walking system as listed below: (Nonami et al., 2014).

1. Weight & volume to torque and power ratios are important problems to create an energy efficient and reliable walking robotic system.
2. The requirement of the reliable sensors to measure the dynamic torque and power necessities and contact forces.
3. The high cost of the mechanically strong and lightweight material
4. Computation speed problems of the main processors.
5. Power supplying problems.

Wabot series anthropomorphic robots had bipedal locomotion capability was developed by Kato et al. in Waseda University in Japan (Figure 2.24). These robots were equipped with 80 microprocessors to create the motion of 50 DOF (Lim and Takanishi, 2006; Behnke, 2008).

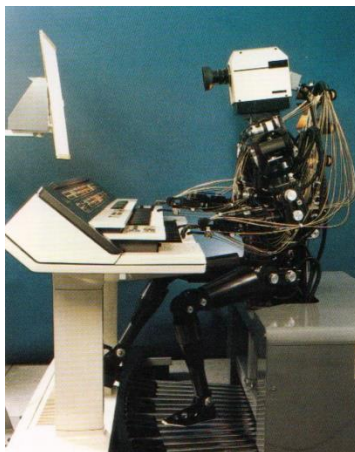


Figure 2.24 Wabot 2 (Lim and Takanishi, 2006; Behnke, 2008)



There are two main categories for the bipedal locomotion in the literature. ZMP (Zero moment point) is the first approach of the bipedal locomotion. According to the ZMP, at the defined zero point sum of the moments and active forces have to be equal to zero. If the defined zero point is within the support polygon of all contact points among the ground and feet, the legged locomotion system is accepted as dynamically stable. The second one is to use the legged system dynamics. A model includes the system dynamics was studied by McGeer. In this model, the possibility of the walk down in slope using passive walking elements without any actuator was demonstrated. In the model shown in Figure 2.25, the wheel had periodic motion in defined slope that led to enough stable region. According to this approach, if the initializing velocity caused to rolling and the defined slope are sufficient, the system never fell to forward. This phenomenon was used to produce the stable passive walking element in bipedal locomotion. Then, a pin type joint was assembled to the leg. If the weight of the walking element was assumed as negligible, the locomotion element motion never disturbed the swing motion. Therefore, the legs rolled on the floor with constant speed. Using these analyses and experiments many different bipedal locomotion elements have been developed (Vukobratovic and Borovac, 2004; Narukawa et al., 2010; Hobbelen and Wisse, 2007). The passive dynamic bipedal locomotion system of McGeer is shown in Figure 2.25.

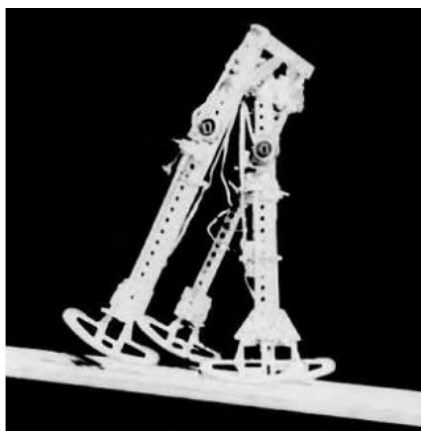


Figure 2.25 Passive Dynamic Bipedal Locomotion Element  
(Vukobratovic and Borovac, 2004; Narukawa et al., 2010; Hobbelen and Wisse, 2007)

Honda and Sony Companies produced the robotic systems with bipedal locomotion. The developed robot series with the project name of ASIMO had the capability of

the walking, climbing the ladders and obstacle avoidance (Behnke, 2008; Duran and Thill, 2012; Gebbert, 2014).

For the ground mobile robots locomotion categorization, several mechanical architectures have been proposed from academic and industrial researches. There are three main categories as W (wheeled), T (tracked) and L (Legged) and four hybrids derived from the combination of the mains as LW (legs-wheels), LT (legs-tracks), WT (wheels-tracks) and LWT (legs-wheels tracks). The listed locomotion systems have their own pros and cons according to several performance features listed below (Bruzzone and Quaglia, 2012; Vidoni et al., 2015; Siegwart et al., 2002; Yang et al., 2009):

1. Velocity,
2. Obstacle avoidance and climbing,
3. Step & stair and ladder climbing,
4. Motion on the slope,
5. Walking capability on the soft and uneven terrain,
6. Energy efficiency.

The comparison chart of the different locomotion types according to the performance criteria listed above is shown in Figure 2.26 (Bruzzone and Quaglia, 2012).

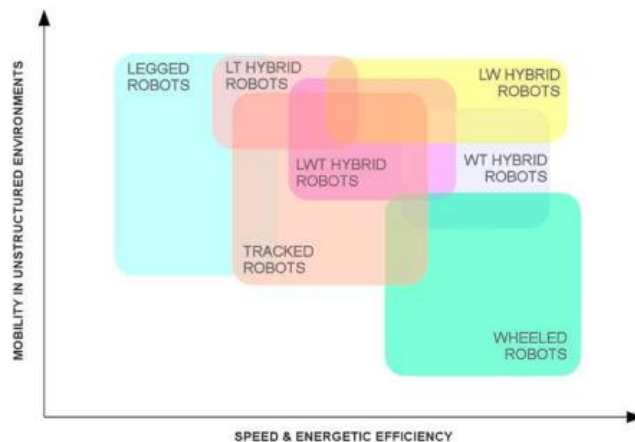


Figure 2.26 Comparison of the Locomotion Systems (Bruzzone and Quaglia, 2012)

While the mobility in unstructured environment is considered in the axis of “Y”, the speed and energy efficiency performance is at the axis of “X”. Wheeled robots are in the right lower zone that means better in the criteria of speed and energy efficiency worse in the mobility in unstructured environments than other locomotion systems. Legged locomotion is at the left upper zone and tracked system places in the middle of chart. In fact, locomotion by wheels provides high speed and energy efficiency as the contact with the ground occurs with lower level of impacts and vibrations. The robotic systems derived from the automotive industries are designed and produced with the combination of different suspension systems, and steering methods such as Ackermann to maximize the performance. On the other hand, the robotic systems with wheels are less suitable to climb the obstacles and to cope with the irregularities of the operation terrain. This limits the motion capability of the wheeled systems in unstructured environments (Seeni et al., 2010; Silva and Machado, 2012). The robotic systems with leg are more suitable for unstructured environment operation. They have better capabilities to overcome the obstacles and terrain irregularities. However, these systems are relatively slow, mechanically more complicated and high-energy consumer as their motion is not continuous unlike wheeled systems (Koditschek et al., 2004).

The other system is the tracked locomotion system placed at the middle of the comparison chart. Tracked systems can climb the obstacles and handle with irregularities better than the wheeled due to their large contact surfaces. However, they are less effective at the criteria of speed and energy efficiency because of the more vibration effects and lower level of the damping systems (Westervelt et al., 2018).

As understood from the Figure 2.26, the hybrid categories are created to take benefit from the main categories but this is not completely occur because the extra payload of the added locomotion system. LW locomotion systems are suitable when both high speed and climbing ability are required for the robotic system operation. This combination can be applied for small and lightweight robotic systems, as the inertial effects and motion discontinuous caused from the legs are less dangerous from the structural point of view (Alamdari et al., 2013; Bruzzone and Quaglia, 2012).

The legs and wheels can be combined in four ways as listed below (Raibert et al., 1989; Manchester et al., 2011):

1. Connection of a leg to the body of wheeled robotic system.
2. Combined wheels and legs working together for the locomotion.
3. Usage of the retractable modules.
4. Placed wheels at the end of the legs.

In the first approach, the design of the robotic system is conceptually simple. According to the environmental conditions, the wheels and legs are used alternatively. The main drawback of this hybrid system is the mass of the system caused from the two locomotion elements. The Mantis Robotic system (Figure 2.27) is an example of the usage of the approach one (Luca and Pietro, 2014).

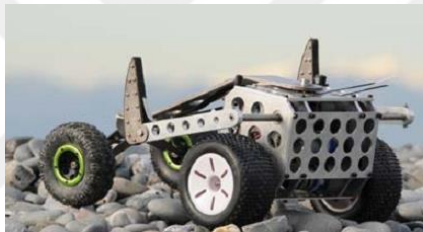


Figure 2.27 Mantis Robot (Luca and Pietro, 2014)

In the systems developed using approach two, the wheels are not enough for any motion mode without the contribution of the wheels. These type of the systems are generally actuated with lower number of motors when compared with the approach one. A sample robotic system produced with approach two Wheelleg is shown in Figure 2.28 (Tadakuma et al., 2010).

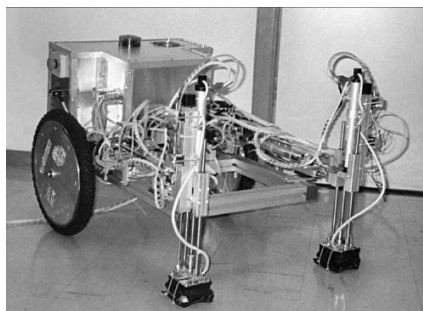


Figure 2.28 Wheelleg (Tadakuma et al., 2010)

The third approach is an interesting one. In this approach robotic system has retractable modules can be used as both leg and wheel according to the requirement. It is a complicated and not reliable method because of the low shock resistivity in

the dirty environments. A sample of a robotic system designed and produced using approach three is shown in Figure 2.29 (Tadakuma et al., 2010).

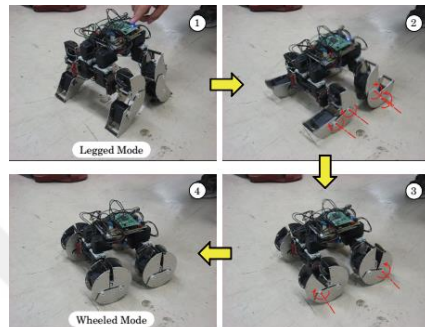


Figure 2.29 Robotic System with Retractable Modules (Tadakuma et al., 2010)

The quadruped robotic system with wheels placed at the end of the rear legs Lebel developed by Gonzalez et al. (2011) is shown in Figure 2.30.



Figure 2.30 Quadruped Robotic System Lebel (Gonzalez et al., 2011)

There are also different types and approaches of hybrid locomotion in the literature. Space Cat Rover developed by the team of the Siegwart is the one of them. The name of the concept of the locomotion system is Stepping Triple Wheels (Figure 2.31). The developed hybrid system was constructed both wheels and legs. In the idea of the Space Cat Rover, the frames supported two driven set of the wheels. With the connections of the wheels and frames, left and right sides of locomotion were constructed. The sides of the locomotion could rotate independently around the main (payload) frame. This allowed to robotic system to climb the obstacles and steps.

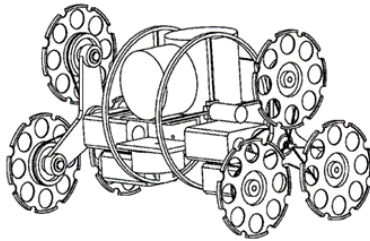


Figure 2.31 Space Cat (Siegwart et al., 1998)

Eight actuators were used to establish the drive module. Six of them were connected directly to the wheels and other two were assembled to the main frame. During the climbing motion, the center of gravity of the robotic system was moved to a location outside of the contact surface of four driven wheels. Therefore, Space Cat was fell with its upper wheel on climbed obstacle. On the other hand, in the straight motion there was no change of the center of gravity so the robotic system could move forward or pass the small size of the obstacles without rolling (Siegwart et al., 1998).

Another researched hybrid locomotion system in the literature is the MSRox developed by the team of the Dalvand (Figure 2.32). The driving unit of this robotic system is named as Star Wheel. This unit was also was designed and developed to climb the steps, stairs and obstacles. It was three legged wheel unit and had three radially located wheels at the end of each spoke. The Star Wheel had two rotary axes. The first one was used for the rotation of the wheels when MSRox moved on the flat and slope surfaces. Second was dedicated to rolling motion of the wheels that provide the obstacle and stair climbing. The four unit of the locomotion system were connected to the central body. According to the motion strategy, the robot could move on the ground with wheel rotation and climb when the rolling system of the locomotion was driven (Dalvand and Moghadam, 2006).



Figure 2.32 MSRox Robot (Dalvand and Moghadam, 2006)

### 2.3. Obstacle Avoidance and Path Planning Systems

The obstacle avoidance and path planning are the keys issues for the navigation of the robotic system. An autonomous robot can have ability to execute its task if it has well performed obstacle avoidance and path planning systems. Simply, obstacle avoidance is behavior of robotic systems that protects the robot from collision to other objects (Chean, 2002; Fox et al., 1997). These objects can be static or dynamic. The issue of the path planning is to create to free path to the goal of the robot with the considerations of fixed objects such as walls. For the path planning, geometric characterization of the obstacles and the kinematic constraints of the robotic system are important criteria. The capabilities of an autonomous robotic system with obstacle avoidance and path planning are listed below (Kunchev et al., 2006):

- 1.The model of the operation environment can be created as a map,
- 2.The free path without collision to the goal or to an obstacle in operation environment can be calculated,
- 3.The obstacles can be passed to reach to final goal with specified velocity and acceleration.

The path planning topic can be divide into two categories as local and global. The global path planning is also known as static motion planning that the motion trajectory of the robotic system calculated before the starting of the motion. It can be a suitable approach when the operation environment is pre-defined and structured. The other method global path planning is also known as dynamic motion planning. In this approach, the motion trajectory of the robotic system is generated online and dynamically based on the information of perception and sensing units. These perception and the sensing units can be different types of the sensors, cameras etc. The dynamic motion planning is accepted as more reliable than the static as it has quick response to the changeable environmental conditions such as shape of the way and position of the obstacles (Oroko and Nyakoe, 2012; Coste and Simmons, 2000, Minguez et.al, 2008).

Although, in the real life autonomous mobile robot applications it is a useful way to use the local path planning there are some limitations of this approach as listed below (Nguyen and Le 2016):

1. The global converge to the final target cannot be computed exactly without the information from the global.
2. The re-computation of the waypoints may be required many times during the motion of the robotic system.

To handle the problems mentioned above and to construct the real autonomous robotic systems different types of the path planning and obstacle avoidance methods and algorithms have been developed in the literature (Floreano et al., 1999; Jingang, 2016; Takeshi et al., 2012).

The Bug algorithms are accepted as the simple obstacle avoidance system in literature. In the first version of the Bug algorithms (Bug1), the robotic system move through the boundary lines of the obstacle from the  $H_i$  (Hit Point; this is the first point that robot hits the obstacle) to find the  $L_i$  (Leave Point). The  $L_i$  point has the shortest distance to the final target. The robotic system repeats this procedure until it reaches the final goal (Figure 2.33). Although it is a safe method and guarantee to reach the final point, it is not a good application method for the obstacles with long boundaries. It is a time and energy consuming method (Oroko and Nyakoe, 2012; Minguez et.al., 2008; Nguyen and Le, 2016; Lumelsky, 1987).

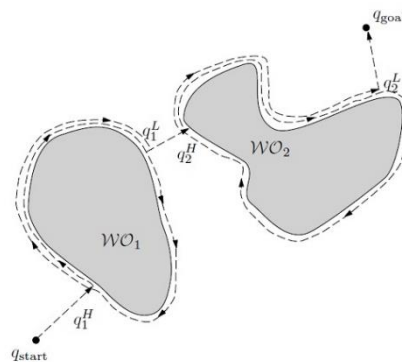


Figure 2.33 Bug1 Algorithm (Oroko and Nyakoe, 2012; Minguez et.al., 2008; Nguyen and Le, 2016; Lumelsky, 1987)

To handle the problems mentioned above, Bug2 algorithm has been developed (Figure 2.34). According to the Bug2, the robot leaves from the obstacle boundary when it finds the  $L_i$  point. Therefore, it does not need to move through the arc between  $H_i$  and  $L_i$  two times. However, if the leave point is not located at the optimal point to reach to the final target, Bug2 algorithm may cause infinite loops (Lumelsky and Skewis, 1990; Nguyen and Le, 2016).



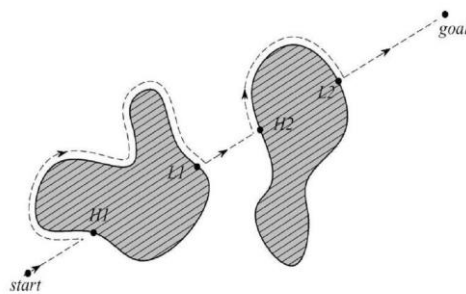


Figure 2.34 Bug2 Algorithm  
(Lumelsky and Skewis, 1990; Nguyen and Le, 2016)

In the expanded and improved version of the Bug algorithms named as Tangent Bug algorithm (Figure 2.35), the range sensors are mounted to the robotic system. With the knowledge of current position of the robot, final target pose and the range measurement sensor outputs, the robotic system can plan to reach to the target. The Tangent Bug algorithm gives shorter path to reach to the final goal (Choset, 2005; Oroko and Nyakoe, 2012).

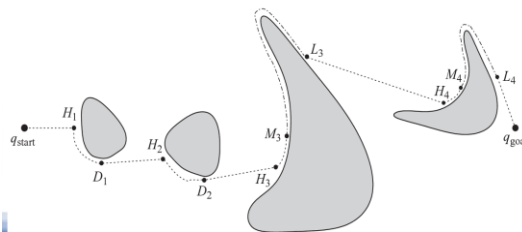


Figure 2.35 Tangent Bug Algorithm Choset, 2005; Oroko and Nyakoe, 2012)

The Bug Algorithms have some pros and cons as listed below (Kamon et al., 1996; Oroko and Nyakoe, 2012; Nguyen and Le, 2016):

1. They are not too hard and complex to use in a different trajectory planning.
2. The Bug2 is less time and energy consumer than the Bug1.
3. The application in the open spaces are more suitable for Bug1 algorithm.
4. With the usage of the Bug2, some situations may be occurred cause to trap.
5. These algorithms do not consider the robotic kinematics. This may cause some motion problems caused from the structure of the robotic system and some issues as inertia.

6. The noise of the range sensor used in Tangent Bug algorithm may cause to large mistake and performance loss as the method use the most recent data from the sensor.

In the other method named as PFM (Potential field method) uses the potential fields of the electrical forces (Figure 2.36). In this approach, the robotic system is assumed as a particle and its movement is adjusted through the electrical force field influence. In the process, while the final target creates the force to attract the robotic system, the obstacle scatter the repulsive forces. In this method, the potential field is the energy field so the gradient of this field at each point is a force. As the robotic system is effected from the field, the combination of the attractive and repulsive fields drives it to the final target without collision (Koren and Borenstein, 1991; Ishay et al., 1996; Minguez et.al., 2008).

The magnitude of the force is inversely proportional to the distance between particles (obstacles). In an application sample, the potential field treats the robotic system as positive charge same with the obstacles, the charge of the target is opposite as negative. When the robotic system navigates within the operation environment, it is effected from the both positive and negative charges. The resultant of these forces help and allow the robot to reach to the final goal and minimize the collision risks (Xi an Chaoi, 2013; Nguyen and Le, 2016; Khatib, 1985).

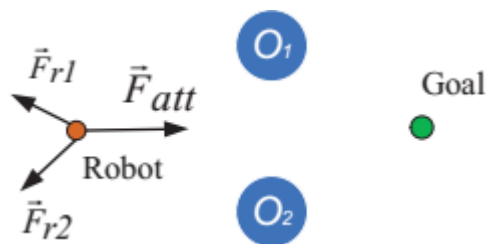


Figure 2.36 Potential Field Method  
(Xi an Chaoi, 2013; Nguyen and Le, 2016; Khatib, 1985)

Although the potential field method is suitable for the applications of offline and online navigation, it has some limitations (Kamil et al., 2015; Subramanian and Thondiyath, 2012; Oroko and Nyakoe, 2012):

1. The oscillation in the motion and the trap situation can be occurred, especially for operation environments with small size and narrow spaces.

2. In the application, the attractive and repulsive forces may effects and cancel each other that cause to the deterioration of the navigation.

The VFH (Vector field histogram) is another method for obstacle avoidance (Figure 2.37). VFH is a real time obstacle detection and avoidance method. It helps and allows to robot to pass the unknown obstacles and to reach to final goal. Two dimensional Cartesian histogram grid model is used in the navigation system of the VFH. The created world model as grid is updated continuously with the measured data from the range measurement sensors. Two stage data reduction process is used in order to compute and create the commands required for the robotic system motion. Firstly, the momentary location of the robot is compressed and one dimensional polar histogram is established. In this histogram, each part includes the value that represent the polar obstacle density. Then, in the second stage, the most suitable path that has the lowest level of obstacle density was found and the direction of the robotic system is adjusted (Koren and Borenstein, 1991; Laumond, 1993; Laumond and Risler, 1996; Babinec et al., 2014).

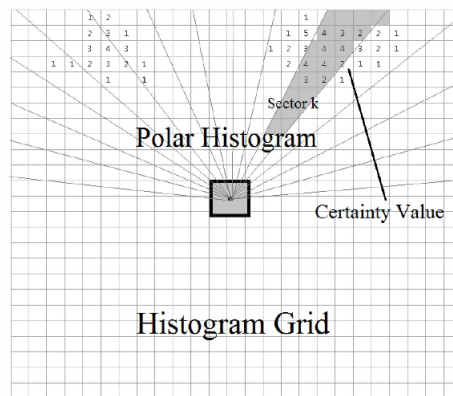


Figure 2.37 Vector Field Histogram (Koren and Borenstein, 1991; Laumond, 1993; Laumond and Risler, 1996; Babinec et al., 2014)

Although VFM can be used for real time obstacle avoidance it has some limitations as (Oroko and Nyakoe, 2012; Molinos et al., 2014):

1. The local trap occurrence cannot be handled.
2. The dynamic constraints are not considered and cannot be coped.
3. The motion in the narrow spaces may be problematic.

The VO (Velocity obstacle) method is especially suitable for moving obstacle avoidance (Figure 2.38). In this approach, the collision cone is used to find the collision between two objects with circular shapes moving with linear motions. In the VO, the collision between a robotic system and an obstacle can be calculated and predicted. For the prediction, the resulting vector of the robot velocity and speed of the obstacle have to be in the collision cone. According to this approach, any speed of the robot that leads the any relative velocity located in the collision cone, causes to the collision. As a result, the collision risk can be minimized with the control of the relative speed of the robot (Fiorini and Shiller, 1998; Cheng and LaValle, 2001; Zhong and Zhou, 2011; Kluge and Prassler, 2007).

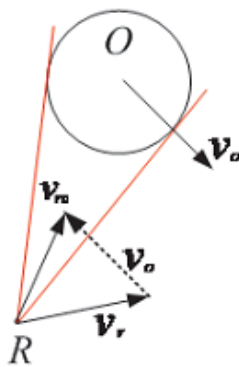


Figure 2.38 Velocity Obstacle Approach (Fiorini and Shiller, 1998; Cheng and LaValle, 2001; Zhong and Zhou, 2011; Kluge and Prassler, 2007)

VO approach has some limitations for real world applications (Oroko and Nyakoe, 2012; Kluge, 2004):

1. It is hard to create the collision cones of the objects, which do not have the shape of circular.
2. The robotic system cannot reach to its target if the all velocities to goal are locate in the collision cone.

The DWA (Dynamic Window Approach) is another method for obstacle avoidance and path planning (Figure 2.39). In this method, the kinematic constraints of the robotic system (acceleration and maximum velocity etc.) are taken into consideration. A robotic system with DWA approach can avoid the obstacles through following the circular trajectories. The radius of these trajectories can be computed using velocities of linear and rotation. The optimum path can be selected

with maximum value derived from the cost function. This cost function has three factors. These are; the position of the robot path according to the final goal, move forward speed and distance to obstacle (Alsaab, 2015; Simmons, 1996).

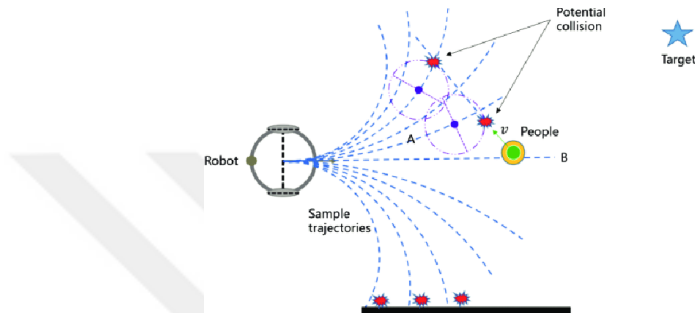


Figure 2.39 Dynamic Window Approach (Alsaab, 2015; Simmons, 1996)

The FGM (Follow the Gap Method) is another approach to calculate the safe path to the final goal (Figure 2.40) (Demir and Sezer, 2017; Kim, 2014).

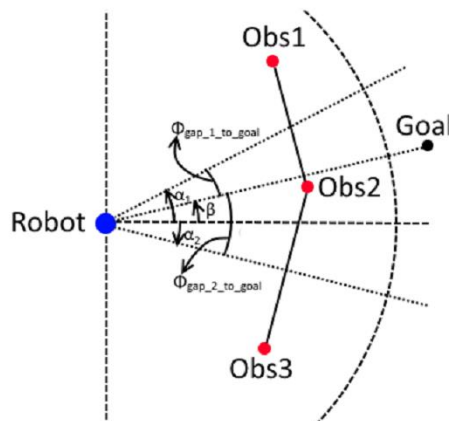


Figure 2.40 Follow the Gap Method (Demir and Sezer, 2017; Kim, 2014)

In this process, the robotic system has also ability to avoid the static and dynamic obstacles by creating a gap map using the obstacles and calculating the angle to the center of the maximum gap. However, FGM has some limitations as listed below:

1. In the complex geometries, the computing of the safe trajectory may be a problem because the center point of the gap may not be the safe route.
2. FGM needs the information of X & Y axes and the radius of the obstacle to find the path exactly. If the obstacle is continuous such as long wall, it is impossible to calculate the required information.

3. It is not suitable use at indoor environment as it contains large number of obstacles.

The mentioned above especially useful for path planning with obstacle avoidance. There are different algorithms and methods in the literature generally used for graphic representation of robotic operation environment to find and to compute the optimal path to reach the final goal.

In the Dijkstra Algorithm (Figure 2.41), each point named as node is defined as unvisited. The current position of the robotic system is assigned as the value of “0”. Then, different weight values are assigned to the other nodes according to the distance to starting point. For the motion plan, the point with the lowest weight is selected as the new current and the other nodes are weighted one more time according to the new current node. This cycle is applied until the robotic system is reached to the final goal (Jasika et al., 2010; Alsaab, 2015).

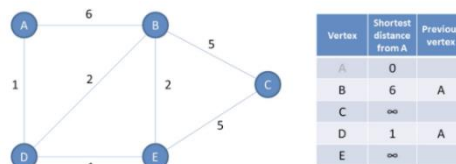


Figure 2.41 Application of Dijkstra Algorithm (Jasika et al., 2010; Alsaab, 2015)

The A\* search algorithm is the expanded version of Dijkstra. A cost function which is the sum of the distance from the current node to final target and the distance between the starting point of the motion and the final goal is added to compute the shortest path to goal (Alsaab, 2015; Zheng and Church, 2009; Vignesh et al., 2012).

The other path planning method Wavefront Algorithm (Figure 2.42) uses the similar principle with heat radiation of conductive material. In this method, the initial position of the robot is assumed as heating source. According to this, if there is any path to final target, the heat radiates toward robot. The obstacles considered as zero conductivity isolates the heat from the goal. The areas without obstacle are defined as large conductivities while the obstacles are assigned as lower level of conductivity. To compute the path to reach to goal, the cells with obstacles are assigned to value of “1”. The value of final target is given as “2”. These values are increased to cover the free nodes. Then, the path is started from the current position

of the robot. For the motion, the robotic system follows the sequence of the nodes with decreasing values (Zidane and Ibrahim, 2017; Alsaab, 2015).

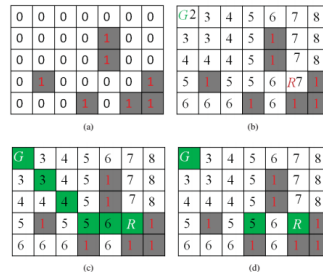


Figure 2.42 Wavefront Algorithm (Zidane and Ibrahim, 2017; Alsaab, 2015)

## 2.4. Sensor Based Fire Detection Methods

Optical flame, smoke and heat detectors are the commonly used for the sensor based fire detection applications. The first one, optical flame detectors can sense the fire up to range of 65 m. They are extremely fast due to the high speed of the light. Various types of optical flame detectors with various spectral band are utilized for fire detection. Optical flame detectors usually employ several optical sensors that work in specific spectral ranges and these sensors record the incoming radiation at the selected wavelength. Optical flame detectors are categorized as:

1. UV (Single sensor type)
2. IR (Single sensor type)
3. UV/IR (Dual sensor type)
4. IR2 (Dual IR sensor type)
5. IR3 (Triple IR sensor type)
6. CCTV (Image processing type)

The second type of the sensor based fire detection method is the usage of the smoke detectors. There are five types of the smoke detectors. Photoelectric and Ionization are the most common types. The others especially used for special applications; (optical) beam, aspirating and video smoke detectors.

The last type of the fire detection sensor is the heat detector. There are two methods to detect the fire from the change of the heat:

1. When the ambient temperature increases sufficiently to predetermined temperature level, system operates and alarm is activated (Fixed temperature heat detector).
2. When the ambient temperature increases over time equal to or greater than the rate of change, detector operates and activates the alarm system (Rate of rise heat detector).

The heat detectors can be categorized as electromechanical, optomechanical, electropneumatic and electronic also known as thermistor (Porteus, 2011; Sucuoglu, 2015).

## 2.5. Computer Vision and Deep Learning Based Fire Detection Methods

The fire detection task is so important and crucial for safety of the human, life areas and industrial zones. Many different fire detection systems have been designed and developed to conduct the fire detection task. The common detection methods are sensor based. In these conventional methods, point smoke and fire detectors are generally used. They generally sense and detect the fire source with the presence of particles from the smoke or fire (Kong et al., 2016; Zhong et al., 2018). The detection type can be ionization and photometry (Figure 2.43) (Porteous, 2011).

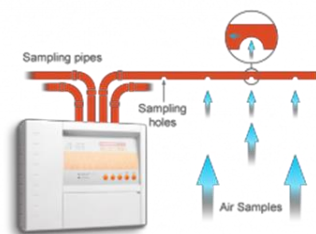


Figure 2.43 Typical Application of Smoke Detector (Porteous, 2011)

These types of detectors need the close proximity to the sensor unit for detection. They are generally suitable for indoor usage. The outdoor detection using these types of sensor is impractical. In addition, these types of detection systems cannot provide the detailed information about the fire such as fire location, direction of the propagation, and growth rate etc.

Currently, with the development of the digital camera technologies there is an important tendency to use the computer vision based detection methods instead of sensor based for early fire detection systems. These systems are suitable to detect



the fire source in large and open environments. They have also capability to analyze the fire and to provide the critical information such as initial starting point of fire etc. The typical application of the computer vision based fire detection is shown in Figure 2.44. The computer vision based fire detection systems are generally consisted of three major steps. The first stage is the pixel classification of the flame, the second one is the segmentation of the moving object and the last one is candidate analysis (Celik and Demirel, 2009; Celik and Ma, 2008).



Figure 2.44 Computer Vision Based Fire Detection Software (Cetin, 2007)

The computer vision based fire detection techniques generally concentrate on the detection and analysis of the flame and smoke in video sequences. These systems varies according to the spectral range of the used camera and the purpose of the system (flame or smoke detection). The techniques used in the video fire detection systems are listed below (Cetin et al., 2013):

1. Color detection,
2. Moving object detection,
3. Motion and flicker analyses,
4. Color variation analysis,
5. Dynamic texture and pattern analysis.

The color detection is the most common approach of the computer vision based fire detection. In this method, the RGB (Red, green, blue) space and sometimes HIS (Hue, saturation and intensity) are used (Chen et al., 2004; Qi and Ebert, 2009; Chen and Wang, 2010). In the flame pixels, the RGB values are sorted as  $R > G > B$ . However, the values of RGB are so close to each other in smoke pixels. It is not possible to detect the fire source using only color information because of the variation of the color, density and background. The color detection can be used as a

step of more comprehensive system (Cetin et al., 2013; Calderara et al., 2008; Phillips et al., 2002).

Moving object detection is another method widely used for fire detection as the flames and smoke have relative motions. The moving object detection algorithms uses the subtraction, temporal differencing and optical flow analyses to determine that moving object is flame and smoke or not (Ko et al., 2010; Yuan, 2008). An example of moving object detection with subtraction is shown in Figure 2.45.



Figure 2.45 Moving Object Detection with Background Subtraction (Cetin, 2013)

The flickering of the flame (flicker detection) can be used to detect the fire situation. In this method, the researchers generally focus on the temporal behavior of the smoke and flame. The flame and smoke colored pixels are located at the edges of fire scenes. However, it is not possible to find a single flickering frequency for flame or smoke in fire scenes so it is not reliable to use only flicker detection to detect the fire source. Flicker detection can be used a part of the multi feature fire detection technique (Xiong et al, 2007; Marbach et al., 2006).

The flames of a fire source have varying colors. Color variation based methods use the variation characteristics of the flame. In this method, different range filters, variance and wavelet analyses are used to distinguish a fire colored scene. The concept focuses on the standard deviation of the green color band (Toreyin et al., 2006; Qi and Ebert, 2009; Borges and Izquierdo, 2010).

Dynamic texture and pattern in a video stream of smoke or flame is defined as the time varying texture with the motion, which forms image sequence. The applications of texture analyses are used for fire detection. The geometry, model

and motion based techniques are applied for dynamic texture detection (Doretto et al., 2003; Hamme et al., 2010; Yuan, 2011). An application of dynamic texture and pattern analysis for fire detection is shown in Figure 2.46.



Figure 2.46 Dynamic Texture and Pattern Analysis for Fire Detection (Toreyin et al., 2007)

Christensen et al. (2019) presented a fire detection method, which allows the detection of the fire in dusty environments. This method is not dependent on the amount of the light or wavelength. It detects the fire with refractive index of the fire source that randomly fluctuates because of the heat convection. They constructed the fire detection system with a standalone laser system consist of Linux based Red Pitaya, 650 nm laser diode and a positive intrinsic negative photo detector.

The several methods and studies listed above are the computer vision video based systems to detect the fire source. They focus on the color, shape, behavior of flame and smoke characteristics or the combination of these features to detect the fire sources. The general aim of these methods are to establish a rule-based algorithm that used as inputs for the classification such as SVM (Support vector machine). For these systems, an expert is required to create the model and define the rules (Frizzi et al., 2016; Verstockt et al., 2010; Poobalan and Liew, 2015).

Currently, different types of the deep learning algorithms such as CNN (Convolutional neural network) have been used to detect the flame and smoke in video streams. Frizzi et al. (2016) proposed a system using CNN for flame and smoke detection. They combined the convolution and max pooling. In this structure, an RGB color image went through two convolutional operations with the kernel size of 3x3. A max pooling with the stride value of two was added the convolutional

layers of two and five. The output of the fully connected layers were connected to the layer of three to produce the class labels. The ReLu activation function with the coefficient value of  $1/3$  was used for the iterations (Figure 2.47). The training set was prepared with 27,919 RGB images with the size of  $64 \times 64$  pixels. 60, 20 and 20% percentages of the images were used for training, validation and test, respectively. The initial learning rate was selected as 0.01. Then, this learning rate was decreased by factor value of 0.95 for each 5 epochs.

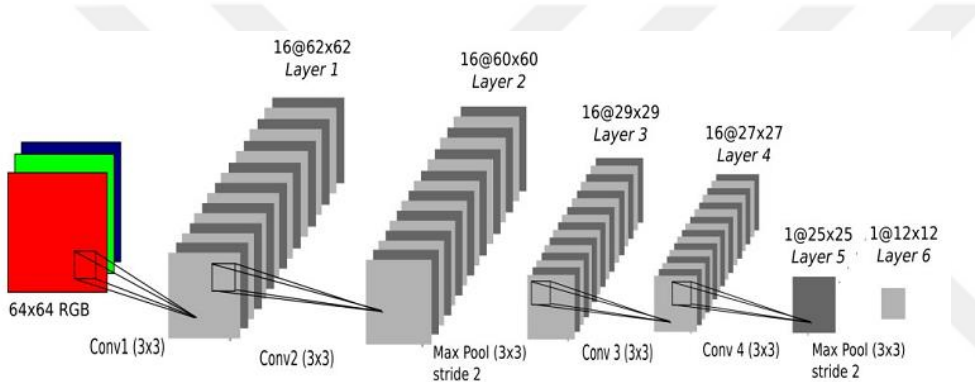


Figure 2.47 CNN Structure for Fire Detection in Video Stream (Frizzi et al., 2016)

Zhong et al. (2018) developed a fire detection system based on CNN in video sequence. In the structure, A RGB model based fire image went through three convolutional layers with the kernel size of  $11 \times 11$ ,  $5 \times 5$  and  $3 \times 3$ , respectively. Then, a window for max pooling with the size of  $3 \times 3$  was adapted to structure to reduce the cost of the computing function. In the fully connected layer stage the dropout method was adopted to the system to reduce the over smoothing. 40,000 and 4,000 images were used for train and test, respectively. The algorithm schematic of the proposed model is shown in Figure 2.48.

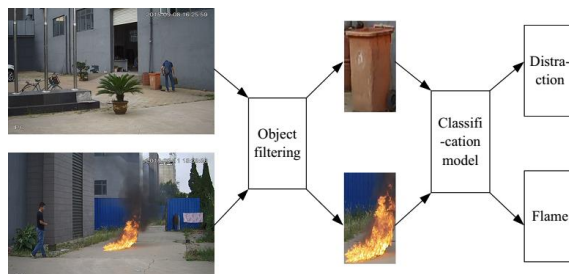


Figure 2.48 The Algorithm Schematic of Proposed Approach (Zhong et al., 2018)

FLIR Company produced the Neuro Technology (2019). In this technology, the trained fire detection model via AI (Artificial Intelligence) was embedded into the microprocessor of the thermal camera. This reduced the cost and complexity of the early fire detection system.

Sucuoglu et al. (2019) developed a real time fire detector using Faster R-CNN (Faster region based convolutional neural network). In the learning process 1,000 images with fire and non-fire scenes were used. 80 and 20% percent of these images were dedicated for training and test, respectively. The Softmax was chosen as converter function. The number of steps for learning determined as 40,000 and epoch number was used as three. As the results of the training process, it was observed that the detector could the fire source with the accuracy of 99% (Figure 2.49).

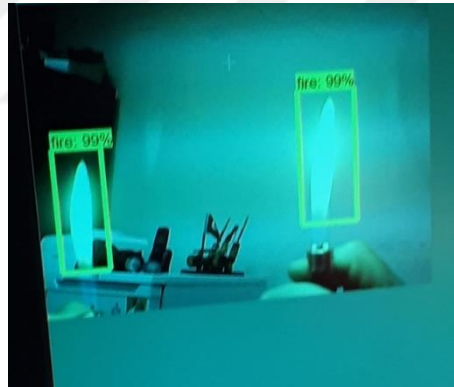


Figure 2.49 Test Results of Fire Detector (Sucuoglu et al., 2019)

### 3. MATERIAL AND METHOD

#### 3.1. General System Architecture and Functions

In this study, a mobile robotic system with hybrid locomotion was designed and produced to operate in both indoor and outdoor environments. Hybrid locomotion was developed through a three-wheel mechanism that could operate as wheel at the ground motion and behave as a leg for climbing. For the adaptive three wheel, a transition system was developed with a decision mechanism had ability to classify the obstacles. The robotic system is capable to determine about the motion type. For ground motion, robot is tracked with two wheels. In the climbing task, three wheel locomotion is activated with transition mechanism and the rotation movement was conducted to climb ladder or obstacle.

A real time path planning and obstacle avoidance algorithm named as “Direction Based Angle Calculation” was also developed in this thesis study. According to this algorithm, the paths and the obstacles in the operation environment were classified with their shape and angle features. In this approach, the fire source and the paths to reach the fire source were defined as target and sub-target, respectively. The sub-targets (safe path) were updated according to situation, shape and the angle of the obstacles. As the sub-targets were defined before to the decision system, the robot did not need to calculate the motion trajectory for every operation. Therefore, the computational burden was prevented.

To localize the fire source and to find the probability “Fire Search and Find” was developed. In this structure three IR flame sensor and a web camera connected Raspberry pi were used. In the first stage, the location of the fire source was detected with flame sensors and the head of the robot direction was adjusted according to the fire source position. The robot was driven to the fire source until it reaches safe detection distance and it was stopped. The safe detection distance was determined as 800 mm. Then, the developed “Fire Detection” algorithm with Faster R-CNN deep learning model (Operated in the web camera connected raspberry pi) determined the probability of the fire. The probability of the fire determined through the developed algorithm were exported as CSV (Comma separated value) file and it could be monitored by operator via wireless connection. The developed robotic system with the functions listed above is shown in Figure 3.1.

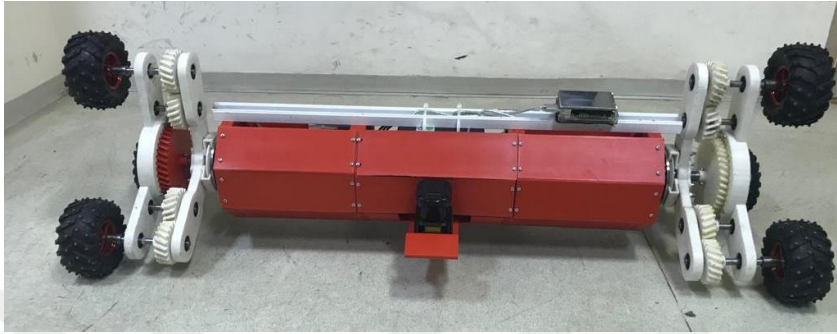


Figure 3.1 The Developed Robotic System (Personal Archive)

The objectives of this thesis study are listed as below:

1. Development of a mobile robot with hybrid locomotion system that operates as wheel and leg according to operation environment as indoor-outdoor and ladder & obstacle climbing.
2. Design and implementation of a motion type decision mechanism that identify the structures as ladder, straight path or negligible obstacle.
3. Development of a real time path planning and obstacle avoidance algorithm named as direction based angle calculation approach that localize the main target (fire), avoid the obstacles, and determine the sub-targets with classification to prevent from computational burden.
4. Design and development of a fire detector via fire detection algorithm using the Faster R-CNN deep learning model.

According to the aim of thesis study, the design goals and criteria are listed as:

1. The mechanic part of the robot such as chassis and carriers should be robust and scalable to carry payloads to allow the robotic system to complete its tasks.
2. The complexity level of the usage of the robotic system should be low. The GUI (Graphical user interface) should be designed according to these criteria.
3. The velocity of the robot should be adjusted according to the functions. If the velocity is higher than the required, there can be some problems in communication and sensor data read. If it is too low, the robot cannot complete its task at the optimum time.
4. The weight of the robot should be optimized. It should be not too heavy or too light. The weight affects the power consumption and balance. The material of the

- chassis and the other mechanical components should be selected carefully according to their target and intended purpose.
5. Energy consumption of the robotic system is an important parameter. It is affected by dimensions, weight and motor selection. Thus, the mechanical and electronic designs should be done carefully; components should be selected according to these criteria.
  6. The obstacle avoidance and path planning systems should be adaptable for different operation areas usage. In the outdoor environment, the robot should have ability to pass or climb the obstacle if it is required. In the indoor usage, the maneuverability of the system should be sufficient to pass the complex obstacles.
  7. The robotic system should have the capability to pass the dynamic obstacles.
  8. The location of the fire source should be determined correctly and the head of the robotic system should be adjusted according to the fire source.
  9. The safe distance to the fire source should be determined carefully to protect the robot from fire.
  10. The target & sub-targets and the motion plan according to the targets should be planned to prevent from computational burden.

### **3.2. The Mechanical Design and the Components**

In the mechanical process of the robotic system, several issues were taken into considerations as:

1. Modularity for easy connection, assembly and disassembly,
2. Compact structure for sufficient payload,
3. Lightweight for energy efficiency,
4. Tolerances of the components for assembly process,
5. Strength of the components and assembly.





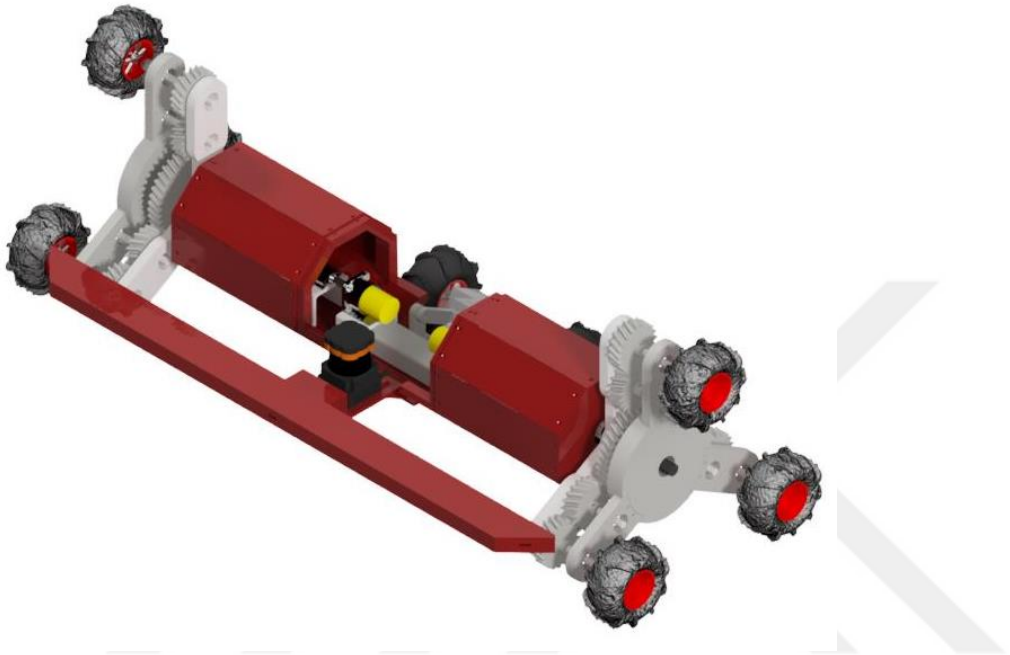


Figure 3.4 General Design View

The components of the robotic system and the exploded view of the general assembly are given in Table 3.1 and Figure 3.5, respectively.

Table 3.1 Component List of the Robotic System

No	Name of the Components	Pieces	Explanation
1	Off-road wheels	8	They were used for traction of the system.
2	Housing rubbers	2	They were dedicated for housing of the main shaft at the rolling motion for climbing.
3	Shaft to wheel connectors	8	The connection of the wheels and shafts were provided with these components.
4	Ball bearings	24	They were used for housing of inner-outer gear shafts.
5	Outer housing covers	2	Transmission system elements were supported with outside-inside housing covers.
6	Inner-Outer gear shafts	12	They were used to transmit the motion from main gear to wheels.

Table 3.1 Component List of the Robotic System (Continued)

7	Main gears	2	The required torque and angular velocity were distributed by the main gears for linear and rolling motions.
8	Inner-outer gears	12	They were used to transmit the motion from main gear to wheels.
9	Inner housing covers	2	Transmission system elements were supported with outside-inside housing covers.
10	Main gear shafts	2	They were the main element to distribute the torque and angular velocity.
11	Couplings	2	They were used to connect the DC motor shaft to main shaft.
12	Axial ball bearing sets	2	They were used as both thrust and housing for transmission system.
13	Linear motion ball bearings	2	They were selected to provide both axial and radial motions to transmission system.
14	Side covers	2	They were used to connect the transmission system to main chassis of the robot.
15	Side frames	2	They were also dedicated for connection the transmission system to main chassis
16	Brackets	18	They were used to build the chassis of the system.
17	DC motor covers	2	DC motors protected from the environmental conditions with these covers.
18	DC motors	2	System was actuated with DC motors.
19	Linear slides	2	They were used as support for the linear motion actuated by the linear actuators.
20	Linear actuators	2	They were used to provide linear actuation.

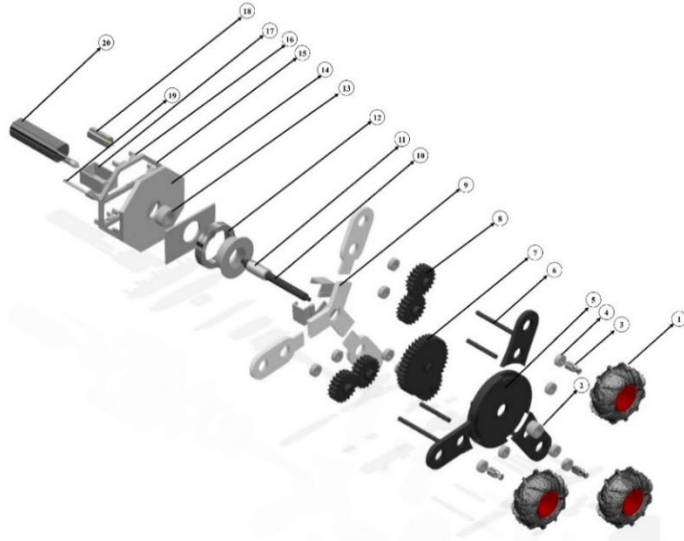


Figure 3.5 Exploded View of Designed Robotic System

### 3.2.1. General Dimensions and the Weight

The dimension and the weight of the system was an important issue as they affected the general performance, maneuverability and the energy consumption. These criteria was taken into consideration in the component design and selection processes. The selected important dimension values of the robotic system are given in Table 3.2 and Figure 3.6, respectively.

Table 3.2 General Dimensions

No	Name of the Dimension	Value (mm)
1	Maximum length	792
2	Chassis length	464
3	Cover length	382
4	Gear system length	70
5	Maximum height	346
6	Maximum width	466
7	Maximum radius	210
8	Chassis radius	150
9	Cover radius	95
10	Gap between inside cover and gear	5

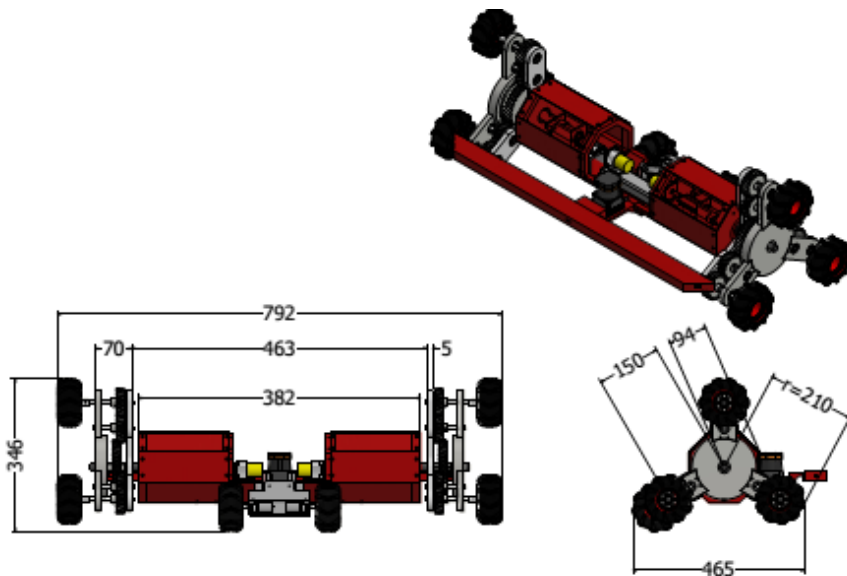


Figure 3.6 General Dimension of Design (in mm)

The total weight of the robotic system was also calculated as 10.5 kg with the sum of the unit weight of the components.

### 3.2.2. Material and Manufacturing of the Components

The mechanical components of the robotic system except commercially available, shafts and motor wheel connector were manufactured using a type of additive manufacturing technique FDM (Fused deposition modelling). The material of these components were selected as PLA (Polylactic acid) because of the advantages such as biodegradability, high printing speed and low layer height (Madani et al., 2015). The infill type and the density of the material were selected as hexagonal and 50%, respectively to decrease the time & material consumption and to increase the mechanical strength (Sucuoglu et al., 2018).

The shafts and motor wheel connectors were produced with S 235 steel using the machining processes as turning, milling and drilling.

Pololu off-road type of the wheels with the diameter of about 120 and width of 60 mm were selected for traction of the robotic system (Figure 3.7).



Figure 3.7 Off-Road Type Wheels (Pololu Robotics & Electronics, 2018 a)

The housing rubbers (Figure 3.8) were dedicated for housing of the main shaft at the rolling motion to climb the obstacles or ladders.

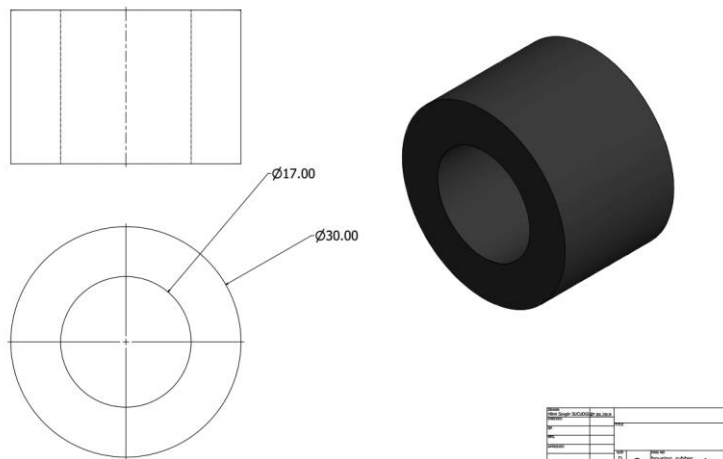


Figure 3.8 Housing Rubbers

The connection of the wheels and driven shafts were supplied with the connectors (Figure 3.9). For the fixing of these connectors, M5 (Metric five) bolts were used.

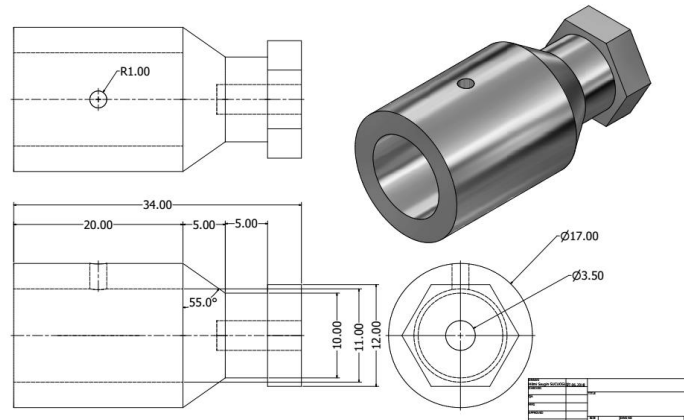


Figure 3.9 Connectors for Shaft to Wheel Connection

The Koyo 2RS bearings from Makparsan Company were used for housing of inner and outer gear shafts (Figure 3.10). The inner & outer diameters and width of bearings were 10.26 and 8 mm, respectively.

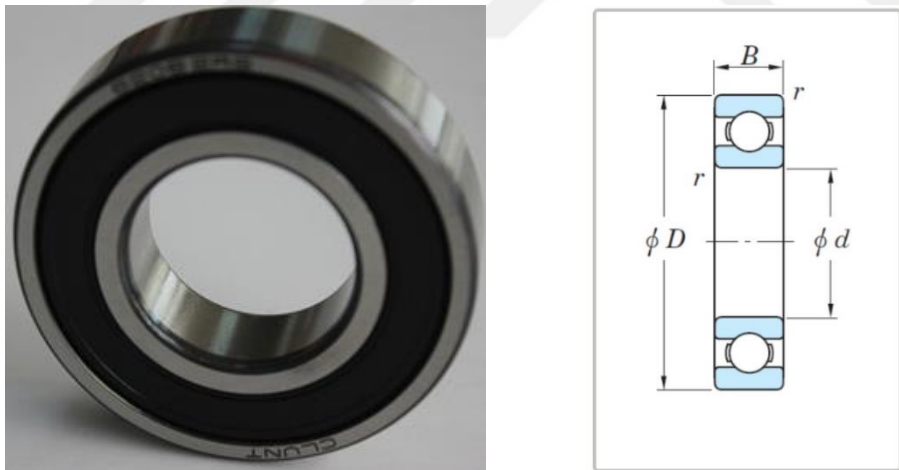


Figure 3.10 Bearings for inner and outer gear shafts housing (Makparsan, 2017 a)

The transmission system components (shafts, bearings and gears) were supported with inner and outer housing covers (Figure 3.11). The outer cover had also a part that the main gear fixed inside of it with rolling in the climbing motion.

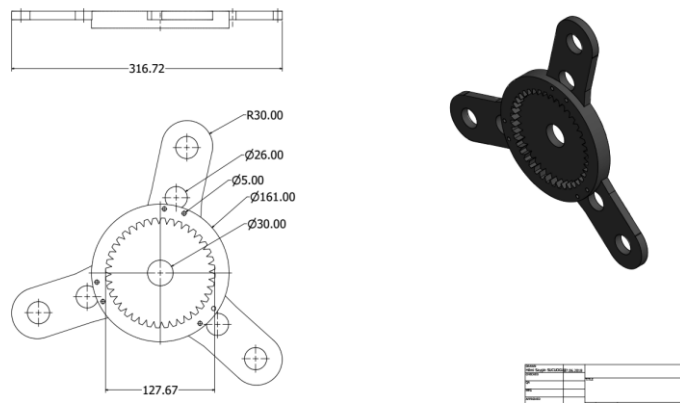


Figure 3.11 The Outer Cover

The inner and outer (driven) gear shafts were manufactured with S 235 steel (Figure 3.12). They were dedicated to transmit the motion from main gear to wheels through the inner and outer gears.

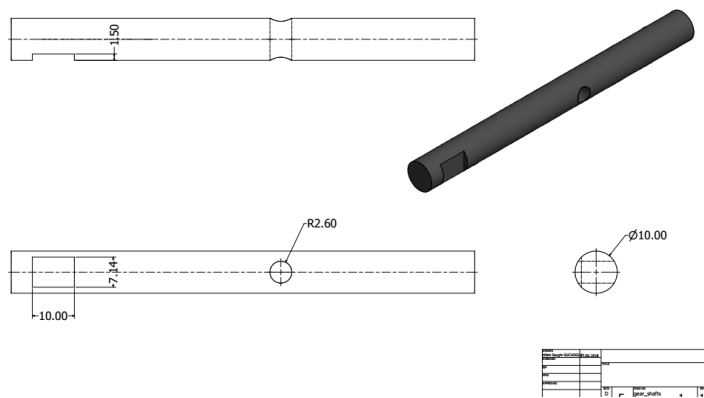


Figure 3.12 The Inner and Outer Gear Shafts

The gears were the main element of the transmission system. The required torque and angular velocity were distributed by the main gears (Figure 3.13). The inner and outer gears were used to transmit the motion from main gear to wheels in the linear motion.



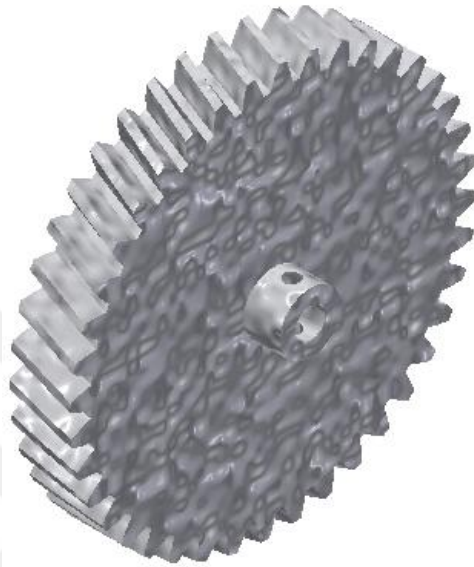


Figure 3.13 The Manufactured Helical Type Main Gear

The gears were produced with PLA material as helical type because of the reasons listed below (Sekercioglu, 2015):

1. It was easier to conduct the transition motion with helical gears, as the form of the pitch was more suitable to fit the covers and other gears.
2. It was observed that when the transmission system was operated with spur type gears, some cracks were occurred on the pitch as spur gear had lower strength to dynamic loads.

The dimensions of the manufactured main, inner and outer gears are given in Table 3.3 and Figure 3.14, respectively.

Table 3.3 The Dimensions of the Gears

Symbol	Parameter and Unit	Equation	Main Gear	Outer Gear
			Value	
$m_n$	Normal module, mm	$m = p/\pi$	3	3
$m_a$	Face module, mm	$m_a = m_n/\cos\beta$	3.19	3.19
$Z$	Number of tooth	$z = D/m$	40	20
$A$	Clutch angle, degree	-	20	20
$A_f$	Face clutch angle, degree	$\tan\alpha_f = \tan\alpha / \cos\beta$	43.2	43.2
$B$	Helix angle, degree	-	20	20
$D$	Pitch diameter, mm	$D = m_n z / \cos\beta$	127.7	63.85
$D_O$	Outside diameter, mm	$D_O = D + 2a$	133.7	69.85
$D_R$	Root diameter, mm	$D_R = D - 2b$	120.2	60.1
$p_n$	Normal pitch, mm	$p_n = m_n \pi$	9.42	9.42
$p_f$	Face pitch, mm	$p_f = m_f \pi$	10.05	10.05
$H$	Height of tooth, mm	$h = a + b = 2.25m$	6.75	6.75
$A$	Addendum, mm	$a = m$	3	3
$B$	Dedendum, mm	$b = 1.25m$	3.75	3.75
$T$	Thickness of tooth, mm	$t = p/2$	4.71	4.71
$S$	Space of tooth, mm	$s = p/2$	4.71	4.71
$B$	Face width, mm	-	2	2

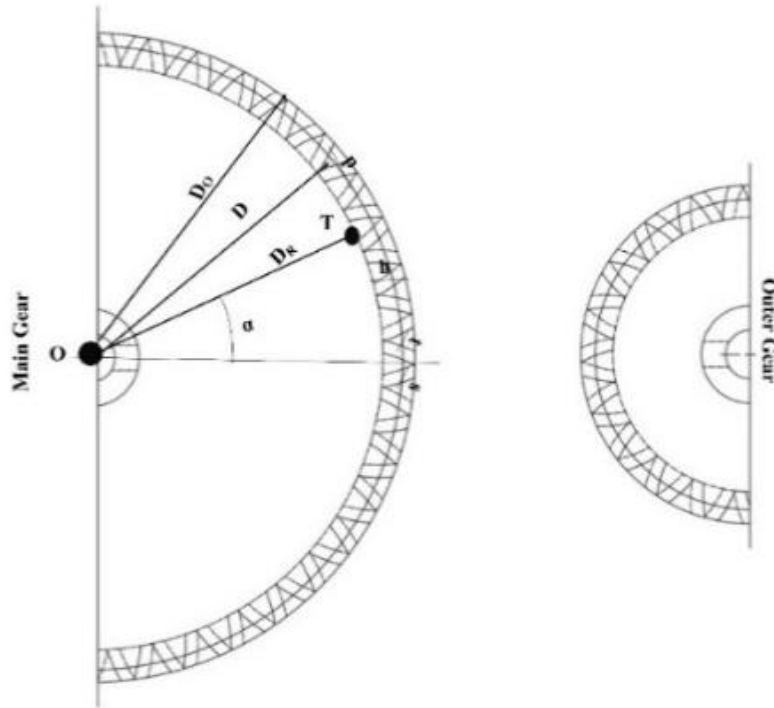


Figure 3.14 Technical Drawing of the Gears

The main gear shafts (Figure 3.15) were the important elements to distribute the torque and angular velocity to driven shafts and gears. The transition system was also activated through these shafts with the motions of push and pull.

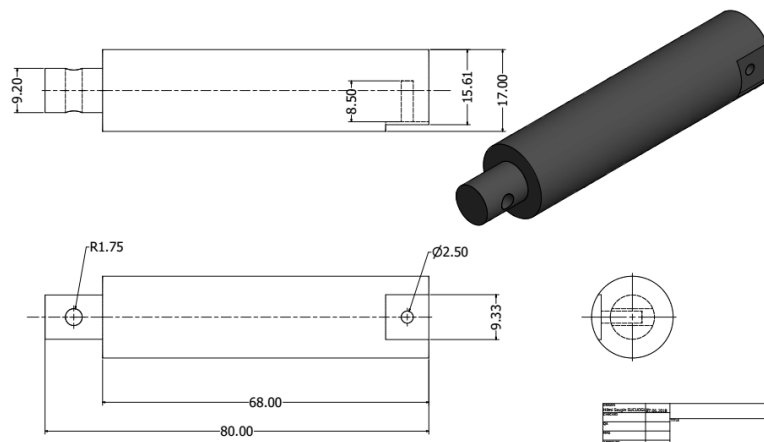


Figure 3.15 The Main Gear Shafts

The connector couplings (Figure 3.16) were used to connect the main shaft to DC motor. These couplings were also manufactured with S 235 Steel. Fixing between the main shaft and DC motor was conducted with M5 studs.

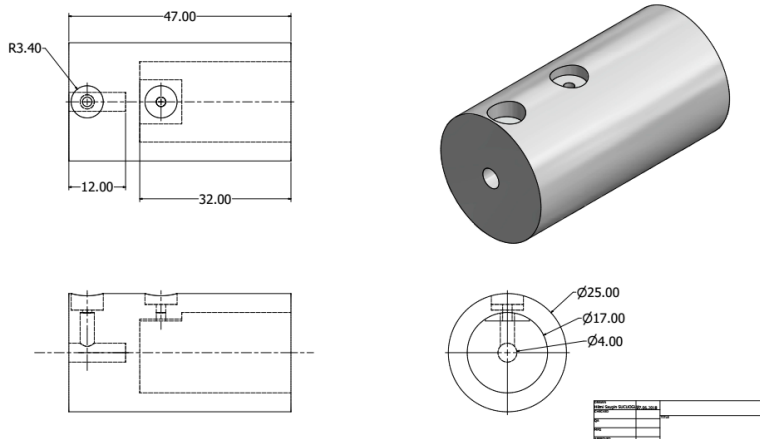
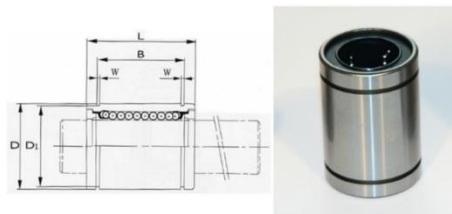


Figure 3.16 The Connector Couplings

LME 25 UU type linear motion ball bearings (Figure 3.17) were selected to provide both axial and radial motion capability to the main shaft to execute the transmission.



Part Number	d	D	L	B	W	Circuits	C	Co	Weight (g)
LM 8 UU	8	15	24	17.5	1.1	4	27	41	16
LM 10 UU	10	19	29	22	1.3	4	38	56	30
LM 12 UU	12	21	30	23	1.3	4	42	61	32
LM 13 UU	13	23	32	23	1.3	4	52	79	43
LM 16 UU	16	28	37	26.5	1.6	5	79	120	69
LM 20 UU	20	32	42	30.5	1.6	5	88	140	87
LM 25 UU	25	40	59	41	1.85	6	100	160	220

Figure 3.17 Linear Motion Ball Bearings (Sahin Rulman, 2017)

The Koyo 5114 axial ball bearing sets were used to thrust and house the transmission system in both linear and rolling motions. These bearings were selected with the inner & outer diameters and width of 70, 95 and 18 mm respectively (Figure 3.18).



Figure 3.18 Axial Ball Bearing Sets (Makparsan, 2017 b)

The side covers and the frames were used to connect the transmission system to main chassis. They were also used as the support material in the mechanical structure of the robotic system. The technical drawings of the side covers and frames are shown in Figure 3.19 and Figure 3.20, respectively.

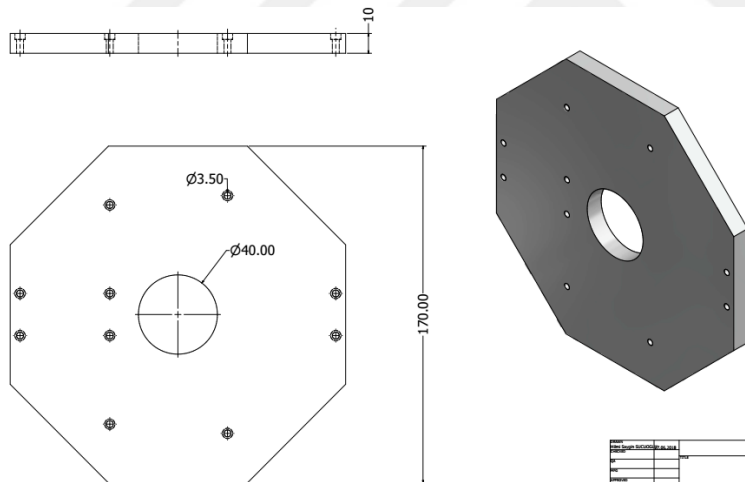


Figure 3.19 Side Covers

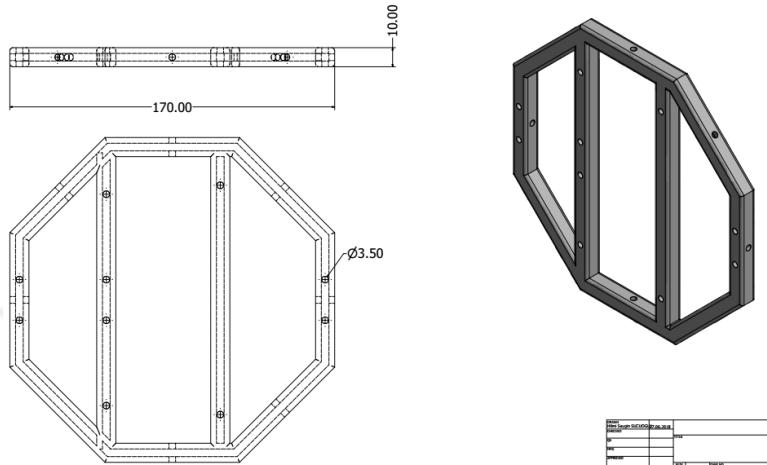


Figure 3.20 Side Frames

The DC motor cover were produced with PLA material to fix the DC motor and to protect the motors from the environmental conditions (Figure 3.21).

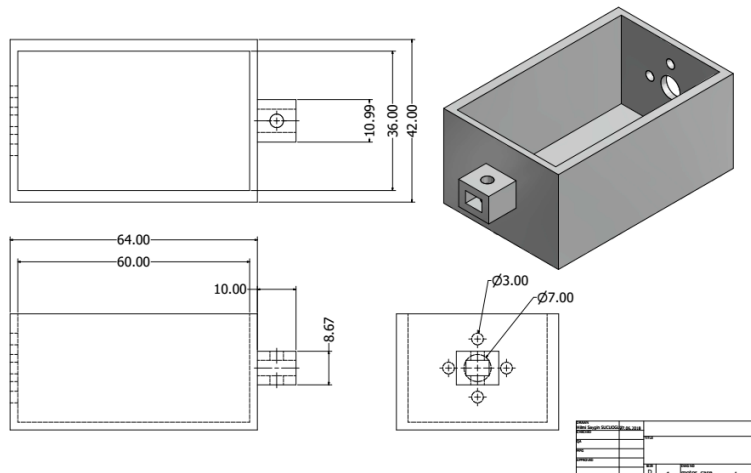


Figure 3.21 DC Motor Covers

The connection brackets were used to connect the chassis profiles to each other. They were also dedicated to main structure of the chassis of the robotic system (Figure 3.22).

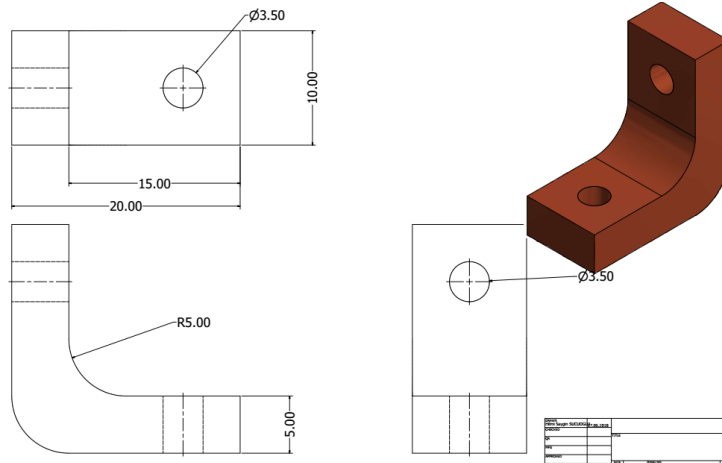


Figure 3.22 Connection Brackets

Linear actuation mechanisms (Figure 3.23) for two sides of the robotic system were manufactured instead of using a commercially available linear actuator. The transition from the motion types (from linear to rolling or rolling to linear) was conducted through the linear actuator push and pull motions. All parts were produced according of designation of DIN ISO 2768 general standards with the tolerance class of medium.

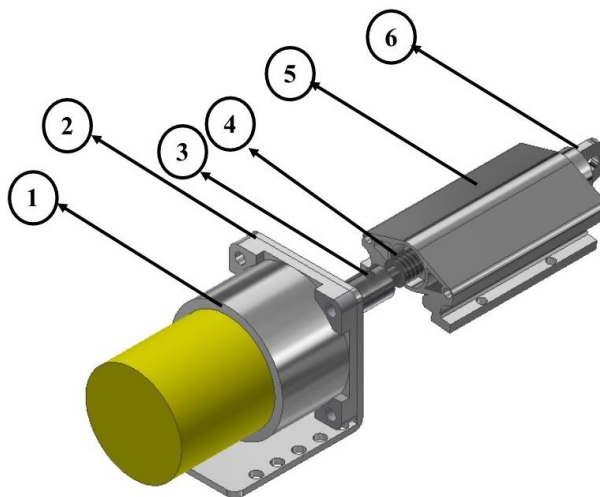


Figure 3.23 Designed Linear Actuation Mechanism

The DC motor was the first component of the linear actuation mechanism. DC motors with metal gears were selected with the power and the load capacity of

32 Watt and 450 N, respectively. The DC motors were fixed to chassis with the connection brackets. The connection bracket was stated as two in the design.

The threaded shafts (4) used to convert the rotational motion to linear were manufactured with the total length of 78 mm and M10 threads. The DC motor and threaded shaft was connected to the each other with produced steel coupling (3). The connection shaft (6) was used to connect the linear actuator to the DC motor cover of transmission system.

The covers (5) was dedicated to assembly of the connection and threaded shafts.

The motion systems were designed and produced to actuate the robotic system for linear and rolling motions. The design and the components of the motion systems are given in Table 3.4 and Figure 3.24, respectively.

Table 3.4 The Components of the Motion System

No	Name of the Component	Explanation
1	Linear rail and car	It was used as a support for DC motor cover push and pull motions that enable the transition motion.
2	DC motor case	The cover was manufactured as suitable for the linear actuator mechanism connection and fixing the DC motor.
3	DC motor	The motion was actuated with DC motor.
4	DC motor case cover	It was used to avoid the DC motor's uncontrolled body motion.
5	Connection coupling	The transmission and motion system were connected to each other with connection coupling.

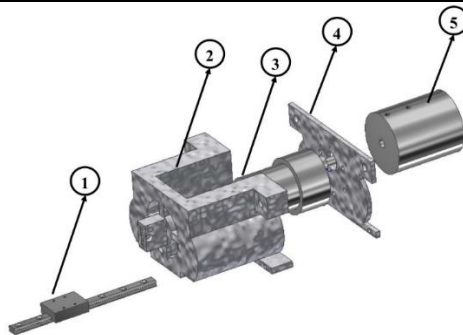


Figure 3.24 Designed Motion System



The designed and manufactured transmission systems were used for both linear and rolling motions of the robotic system. While the actuated main shaft provided the required rotation motion, the gears and the inner & outer shafts distributed the required torque and angular velocity to the wheels (Table 3.5 and Figure 3.25).

Table 3.5 The Components of the Transmission System

<b>No</b>	<b>Name of the Component</b>	<b>Explanation</b>
<b>1</b>	Linear motion ball Bearing	It was selected to provide both axial and radial motions to transmission system.
<b>2</b>	Bearing segment	It was used to fix the ball bearing and to make the motion and transmission system coaxial.
<b>3</b>	Inner housing covers	Transmission system elements were supported with outside-inside housing covers.
<b>4</b>	Main gear	The required torque and angular velocity were distributed by the main gears for linear and rolling motions.
<b>5</b>	Inner-outer gears	They were used to transmit the motion from main gear to wheels.
<b>6</b>	Ball bearings	They were used for housing of inner-outer gear shafts.
<b>7</b>	Shaft segment	It was used to fix the shafts and to avoid from axial misalignment.
<b>8</b>	Outer housing cover	Transmission system elements were supported with outside-inside housing covers.
<b>9</b>	Shaft to wheel connector	The connection of the wheels and shafts were provided with these components.
<b>10</b>	Off-road wheel	They were used for traction of the system.

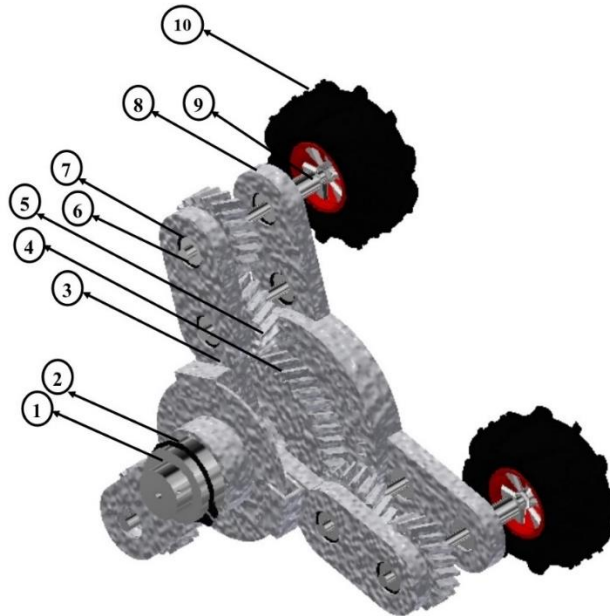


Figure 3.25 Designed and Manufactured Transmission System

### 3.3. Motion Calculations and Analyses

The motion calculations and analyses were conducted to find the occurred forces on the robotic system at the rolling and linear motions. According to the occurred forces torque, power and the angular velocity requirements were determined. The calculated torque, power and the angular velocity values were also used for DC motors and power supplier's selections.

For the climbing motion, the robotic system was assumed as a rolling down object on an inclined plane. The required calculations and analyses were conducted according to this assumption. The angle of slope of the inclined plane was determined using the dimensions (height and length) of step of ladder (Figure 3.26).

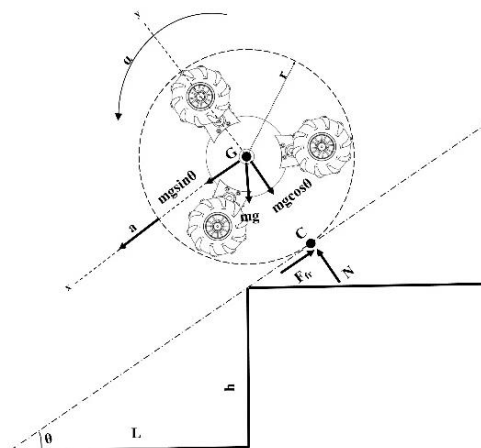


Figure 3.26 Rolling Down on an Inclined Plane Assumption

The  $h$  (height) and  $L$  (Length) values of the step were used as 150 and 300 mm, respectively. The angle of the slope " $\theta$ " was calculated as  $30^\circ$ . The affected forces, acceleration, required torque and power were calculated with the condition of equilibrium. According to the equilibrium assumption, if the calculated parameters were applied to the robotic system as reverse, it would be at the equilibrium and the rolling back was avoided (Figure 3.27).

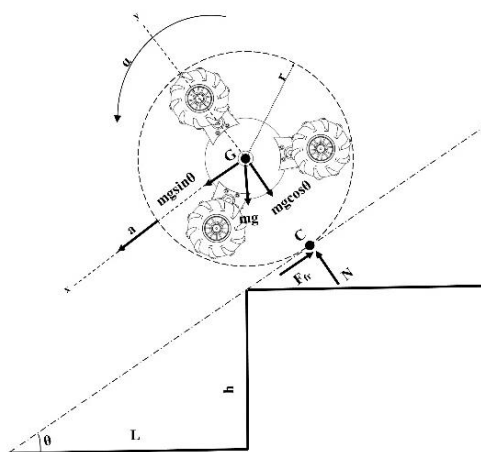


Figure 3.27 The Condition of Equilibrium

The forces:

$$\Sigma F_x = mg \sin \theta - F_{fr} = ma \quad (3.1)$$

$$N - mg \cos \theta = 0 \quad (3.2)$$

The moments:

$$\Sigma M_g = I\alpha \quad (3.3)$$

$$Fr = mr^2\alpha \quad (3.4)$$

In the calculations, if the maximum force derived from the mass and the maximum translational acceleration of the robotic system was larger than the slippage force caused from the relation of wheel and ground, the rolling motion occurred as desired.

$$F_{\max} = ma_m \quad (3.5)$$

$$mg \sin 30^\circ - F_{fr} = ma \quad (3.6)$$

$$T = I\alpha \quad (3.7)$$

$$F_{fr}r = \frac{1}{2}mr^2\frac{a}{r} \quad (3.8)$$

$$F_{fr} = \frac{1}{2}ma \quad (3.9)$$

$$mg \sin 30^\circ - \frac{1}{2}ma = ma \quad (3.10)$$

$$mg \sin \theta = \frac{3}{2}ma = F_{\max} \quad (3.11)$$

The maximum force was equal to as found in Equation 3.11.

$$T = I\alpha \quad (3.12)$$

$$F_{fr}r = \frac{1}{2}mr^2 \frac{a_m}{r} \quad (3.13)$$

$$F_{fr} = \frac{1}{2}ma_m \quad (3.14)$$

$$a_m = \frac{2F_{fr}}{m} = \frac{2mg \cos \theta \mu}{m} \quad (3.15)$$

$$a_m = 2g \cos \theta \mu \quad (3.16)$$

The maximum translational acceleration was found using the torque equations.

In the rolling torque calculations:

$$T = I\alpha \quad (3.17)$$

$$F_{fr}r = \frac{1}{2}mr^2 \frac{a}{r} \quad (3.18)$$

$$F_{fr} = \frac{1}{2}ma \quad (3.19)$$

$$mg \cos \theta \mu = \frac{1}{2}ma \quad (3.20)$$

$$g \cos \theta \mu = \frac{1}{2}a \quad (3.21)$$

$$\mu = \frac{a}{2g \cos \theta} \quad (3.22)$$

$$F = ma \quad (3.23)$$

$$mg \sin \theta - mg \cos \theta \mu = ma \quad (3.24)$$

$$g \sin \theta - g \cos \theta \mu = a \quad (3.25)$$

$$g \sin \theta - g \cos \theta \frac{a}{2g \cos \theta} = a \quad (3.26)$$

$$g \sin \theta = \frac{3}{2} a \quad (3.27)$$

$$a = 2/3 g \sin \theta \quad (3.28)$$

Then total rolling torque was the sum of the rotational and the translational moments:

$$\Sigma M = \frac{1}{2} m r^2 \frac{a}{r} + m a r \quad (3.29)$$

$$\Sigma M = \frac{1}{2} m a r + m a r \quad (3.30)$$

$$\Sigma M = 3/2 m a r \quad (3.31)$$

The rolling torque was equal to as found in Equation 3.31.

Where:

$\Sigma F_x$	Total translational force
$N$	Reaction force at the point of C
$I \alpha$	Torque
$F_{fr}$	Friction force
$a$	Translational acceleration
$a_m$	Maximum translational acceleration
$\alpha$	Angular acceleration
$F_s$	Slippage force
$\mu$	Static friction coefficient

The calculations of the maximum force, acceleration for rolling motion without slippage and rolling torque were conducted using Matlab Software as shown in (Figure 3.28).

The values used in the calculations:

m (total mass)	10.5 kg
$\mu$ (Static friction coefficient)	0.75
g (Standard earth gravity)	9.81 m/s <sup>2</sup>
r (Radius of the robotic system)	210 mm

The screenshot shows the MATLAB R2013a environment. The Editor window displays a script named 'rolling\_linear\_motion\_calculations.m' with the following code:

```

31
32 %Rolling Motion
33
34 %Acceleration and Torque
35
36 %T=I*alpha
37 %alpha=ar/r;
38 %Ffr=r/2*m*r^2*(ar/r);
39 %Ffr=1/2*m*ar;
40 %m*g*cos(thetal*pi/180)*M=1/2*m*ar
41 %M=ar/2*g*cos(thetal*pi/180);
42 %F=M*ar;
43 %m*ar=m*g*sin(thetal*pi/180)-m*g*cos(thetal*pi/180)*M;
44 %ar=2/3*g*sin(thetal*pi/180);
45
46 ar=2/3*g*sind(thetal);
47
48 %Torque Calculations for rolling motion
49 %Total Moment I*alpha+m*ar*r
50 %Meq=Total moment=T
51 %alpha=ar/r
52 %I=m*r^2
53 %I for cylinder 1/2*m*r^2
54 %M= (1/2*m*r^2) * (a/r) + (m*ar*r)
55 %T=M=3/2*(m*r*ar)
56
57 alpha=ar/r;           % rad/s^2,   angular acceleration
58 I=1/2*m*r^2;         % kg*m^2,   moment of inertia
59 Tr=3/2*(m*r*ar);    % Nm,       Total torque
60 Thr=Tr/2;           %           torque for one half

```

The Command Window displays the following output:

```

Rolling motion
check for rolling
amax = 12.74 m/s^2
Forces
Ffr = 66.90 N
Fs = 51.50 N
Fmax = 200.71 N
No slippage
ar = 3.27 m/s^2
Rolling torque
I = 0.23 kgm^2
Tr = 10.82 Nm
Thr = 5.41 Nm

```

Figure 3.28 The Torque and Acceleration Calculations for Rolling Motion

According to the slippage check, as the maximum force (200.7 N) was larger than the slippage (51.5 N) the rolling motion occurred as desired. The rolling acceleration and torque were also calculated as 3.27 m/s<sup>2</sup> and 10.8 Nm, respectively.

In calculation of angular velocity and power:

$$w = \frac{d\theta}{dt} = \dot{\theta} \quad (3.32)$$

$$\alpha = \frac{d\omega}{dt} = \dot{\omega} = \ddot{\theta} \quad (3.33)$$

$$V = \dot{r} = \omega r \quad (3.34)$$

$$P = T\omega \quad (3.35)$$

The required linear speed was determined as 1m/s for the calculation of “w” (Angular velocity). Then with the relationship between power and torque, the power requirement of the robotic system for rolling motion was calculated (Figure 3.29).The angular velocity and power requirement were found as about 46 rpm (Rotation per minute) and 52 Watt, respectively.

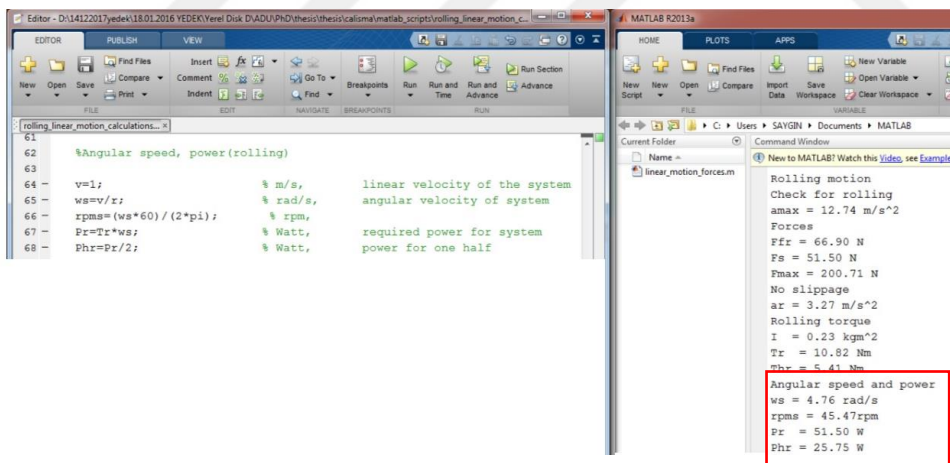


Figure 3.29 Angular Velocity and Power Calculations for Rolling Motion

In the linear motion, the angle of the slope was assumed as  $20^\circ$ . The linear speed was defined as 1 m/s. The resistance forces (gradient, acceleration and rolling), torque & power and angular velocity requirements were calculated (Figure 3.30).



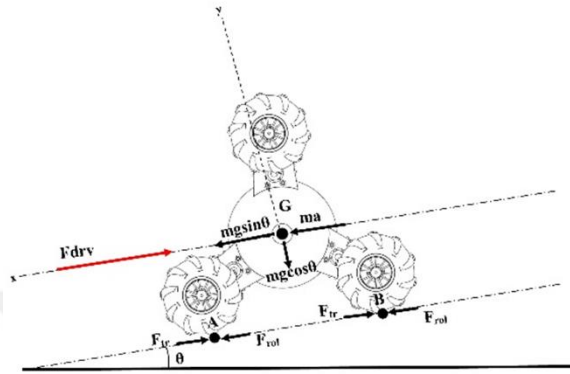


Figure 3.30 The Forces in Linear Motion

$$\Sigma F_x = 0 \quad (3.36)$$

$$mg \sin \theta - F_{rol} - ma + F_{drv} = 0 \quad (3.37)$$

$$F_{rol} = mg \cos \theta c_{rol} \quad (3.38)$$

$$F_{drv} = mg \sin \theta + F_{rol} + ma \quad (3.39)$$

$$T = mg(\sin \theta + F_{fr})r \quad (3.40)$$

$$P = \frac{F_{drv} v}{e} \quad (3.41)$$

$$w_o = v / r_o \quad (3.42)$$

$$w_m = w_o r_o / r_m \quad (3.43)$$

The rolling coefficient was selected as 0.02 and the efficiency of the gear system was used as 0.8 in the calculations. The  $r_o$  and  $r_m$  (radius of outer gear and main gear) were 30 and 60 mm, respectively. The calculation of angular speed, torque and power requirements in the linear motion is shown in Figure 3.31.

```

67 - P=P1*W2; % Watt, required power for system
68 - Pfr=P1/2; % Watt, power for one half
69
70 %Linear motion forces
71
72 - rw=0.06; % m, radius of wheel;
73 - a1=0; % m/s^2, translational acceleration;
74 - crol=0.02; % rolling coefficient;
75 - thetap=20; % degree, slope angle of path;
76 - Ftr=m*g*cosd(thetap); % traction force;
77 - Fgm=m*g*sind(thetap); % N, gradient resistance;
78 - Frol=m*g*crol; % N, rolling resistance;
79 - Fam=1; % N, acceleration resistance;
80 - Fdrv=Fg+Frol+Faz; % N, sum of the forces;
81
82 %Angular speed, power (linear)
83
84 - v=1; % m/s, linear velocity of the s
85 - w=v/rw; % rad/s, angular velocity of outer
86 - r=0.03; % m, radius of outer gear
87 - rm=0.06; % m, radius of main gear
88 - e= 0.8; %
89 - wo=w; % rad/s, angular velocity of outer
90 - wm=wo*r/rm; % rad/s, angular velocity of main
91 - rpm=(wm*60)/(2*pi); % rpm, total torque
92 - Tl=m*g*(sind(thetap)+(cfr))*rw; %Nm
93 - Tl=Tl/2; %Nm
94 - Pl=Fdrv*v/e; %Watt, required power for system
95 - Pfl=P1/2; %Watt, power for one half
96
97 %Power supply
98
99 - Ab=3; % Ah, amperehour
100 - Vb=14.5; % V, voltage

```

Output:

```

Name =
Gear_checks
Gear_checks
rolling_linear_motion_ca...
rolling_linear_motion_ca...
shaft_calculations.m
shaft_checks
Ffr = 66.90 N
Fs = 51.50 N
Fmax = 200.71 N
No slippage
ar = 3.27 m/s^2
Rolling torque
I = 0.23 kgm^2
Tr = 10.82 Nm
Thr = 5.41 Nm
Angular speed and power
ws = 4.76 rad/s
rpm = 45.47rpm
Pr = 51.50 W
Pfr = 25.75 W
Linear motion
Forces
Ftr = 96.79 N
Fg = 35.23 N
Frol = 2.06 N
Fa = 0.00 N
Fdrv = 37.29 N
No slippage
Angular speed and power
w = 16.67 rad/s
wo = 16.67 rad/s
wm = 8.33 rad/s
rpm = 79.58rpm
Tl = 6.75 Nm
Tl = 3.37 Nm
Pl = 46.61 W
Pfl = 23.31 W

```

Figure 3.31 Angular Velocity, Torque and Power Calculations for Linear Motion

The angular velocity, torque and power requirements were calculated as about 80 rpm, 6.8 Nm and 47 Watt, respectively.

The dynamic analyses with Ansys Rigid Body Dynamics tool was also conducted to compare and to check the results from the calculations for both climbing and linear motions. In the analysis of climbing motion, the required angular velocities were applied to DC motor for simulation (Figure 3.32).

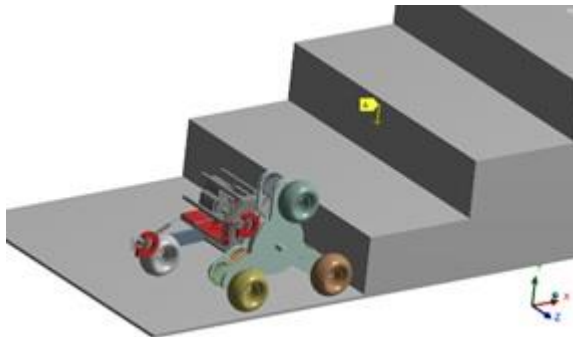


Figure 3.32 Simulation of Climbing Motion

The acceleration, torque and velocity values were obtained from the analysis environment as  $3.5 \text{ m/s}^2$ , 10.1 Nm and 0.97 m/s, respectively. The similar results were found with the calculations conducted in Matlab.

In the linear motion analysis (Figure 3.33), the sum of the acted forces and the velocity were found as 35 N and 0.91 m/s, respectively. These results were also similar with calculation results.

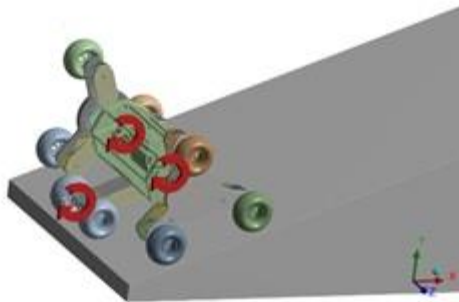


Figure 3.33 Simulation of Linear Motion

The DC motors and power supply system were selected according to the findings from the calculations and analyses. As the robotic system was actuated with three DC motors the required total torque and power values were divided by three. DC motors (Figure 3.34) were selected with the torque, angular velocity and power specifications of 4.2 Nm, 80 rpm and 40 W, respectively.



Figure 3.34 Selected DC Motor for Actuation (Robotistan, 2018 a)

For energy consumption of robotic system; Lipo type batteries with the voltage value of 14.8 V, current of 3 Ah (Ampere-hour) and total energy supply of 44 Wh (Watt hour). The robotic system could be operated about 1.5 hours with two pieces of selected batteries.



Figure 3.35 Lipo Batteries for Energy Supply (Robotistan, 2018 b)

### 3.4. Mechanical Strength Check of the Components

The strength check of the components was applied to determine the gear modules and to decide the minimum shaft and coupling pin diameters.

The minimum required shaft diameters for main, inner & outer shafts were calculated under dynamic loads. These calculations were conducted with the tangential, normal and radial forces affected on the gears (Figure 3.36).

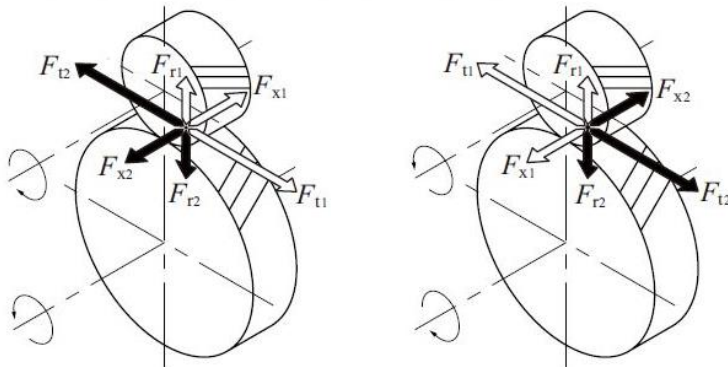


Figure 3.36 The Affected Forces on the Gears

$$Ft_1 = Ft_2 = \frac{2Mr_1}{d_1} \quad (3.44)$$

$$Fn_1 = \frac{Ft_1}{\cos \alpha} \quad (3.45)$$

$$Fr_1 = Fr_2 = Ft_1 \tan \alpha \quad (3.46)$$

The reaction forces at A & B bearings and bending moment at C (Center of the main gear) on X-Y and X-Z planes were calculated, respectively. The technical drawing of the main shaft with shear and moment diagram is shown in Figure 3.37.

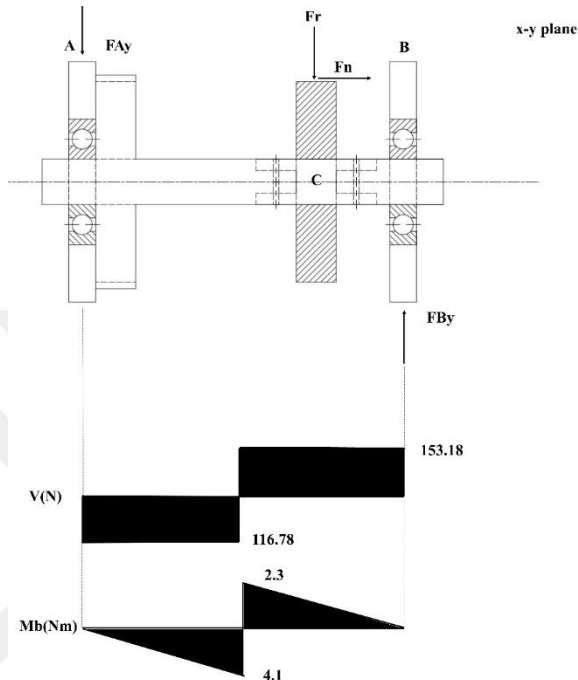


Figure 3.37 Technical Drawing of the Main Shaft with Shear and Moment Diagram

The occurred total moment was found as 10.5 Nm in the motion calculations. For the half side of the robotic system affected moment was used as 6 Nm more than the half of the calculated because of the safety. The clutch angle of the gear was selected as  $20^{\circ}$ . The resultant bending moment was found as about 4.2 Nm.

$$d = \sqrt[3]{\frac{32S}{\pi} \sqrt{\left(\frac{Me}{\sigma_{TDE}^*}\right)^2 + \frac{3}{4}\left(\frac{Md}{Re}\right)^2}} \quad (3.47)$$

$$\sigma_{TDE}^* = \frac{(Ky)(Kb)(\sigma_{TDE}^*)}{(Kc)} \quad (3.48)$$

The minimum shaft diameters were calculated using the equations of 3.47 and 3.48 (Sekercioglu, 2015).

Where:

$S$	Factor of safety	3
$\sigma_{TDE}$	Full variable stress strength of material	180 MPa
$Re$	Yield strength	235 MPa
$Kb$	Shape coefficient	0.95
$Ky$	Surface coefficient	0.9
$Kc$	Notch effect	1.2

The material of the shafts were selected as S 235 Steel. 6 and 2mm values were added to calculated minimum shaft values because of the shoulder and depth of the connection pin (Figure 3.38).

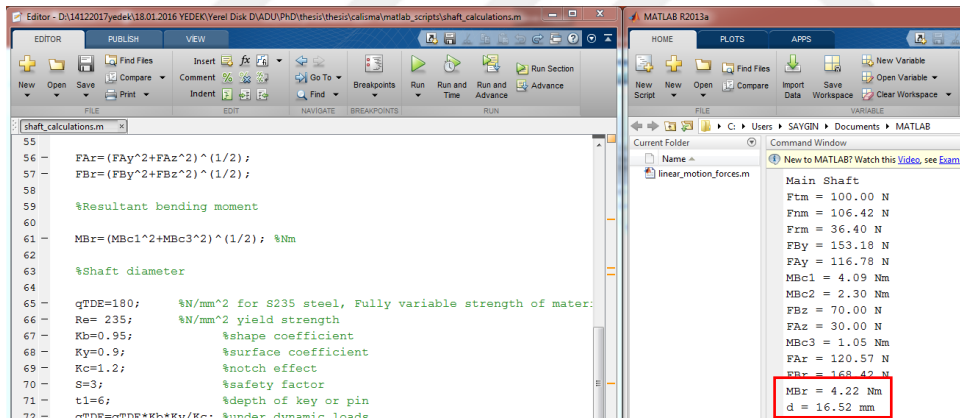


Figure 3.38 The Minimum Shaft Diameter Calculation of Main Shaft

The minimum diameter requirements were calculated as 17 and 10 mm for main and outer shafts, respectively.

The structural analyses were applied to the shafts with Ansys Static Structural Module to check the calculation results and to find the safety factor with selected diameter values (Figure 3.39). Six and three Nm torques were applied to the main and outer shafts, respectively. The safety factor values were found as 2.6 and 2.3. According to these 17 and 10 mm, diameters were accepted as safe and the shafts were manufactured with these diameter values.

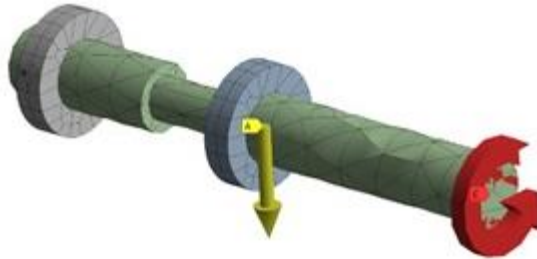


Figure 3.39 Structural Analysis of Main Shaft

The module of the gears were checked according to broken at tooth bottom and surface pressure. The module check calculations and analyses were applied to the outer gear as it was exposed to more rpm and load repeat.

The module check according to broken at tooth bottom (Sekercioglu, 2015):

$$\sigma_{sf} = \frac{(Kb)(Ky)(\sigma_{TD})}{(Kc)(S)} \quad (3.49)$$

$$\sigma_{TD} \cong 0.7\sigma_{FE} \quad (3.50)$$

$$m \geq \sqrt[3]{\frac{2(Mr)(Kf)(Ki)(Kv)(K\varepsilon)(Km)}{(\psi m)(z)(\sigma_{sf})}} \cos\beta \quad (3.51)$$

Where:

$\sigma_{sf}$	Allowable stress	4.375 MPa
S	Factor of safety	3
$\sigma_{TD}$	Full variable stress strength of material	21 MPa
FE	Continuous strength	30 MPa
Kb	Shape coefficient	1
Ky	Surface coefficient	0.75
Kc	Notch effect	1.2
Mr	Moment	3 Nm
Kf	Form coefficient	2.4

Ki	Operation coefficient	1.25
Kv	Speed coefficient	1
Kε	Clutch ratio coefficient	1
Km	Load distribution coefficient	1.2
ψm	Module width coefficient	15
z	Number of teeth	20

The continuous strength of the gears material was assumed equal to yield as they were manufactured with PLA, which was brittle.

The module check according to surface pressure (Sekercioglu, 2015):

$$P_{sf} = \frac{(\sigma_H)(Kp)}{S} \quad (3.51)$$

$$m \geq 0.7 \cos \beta^3 \sqrt{\frac{2(Mr)(Ki)(Kv)(Km)(K\varepsilon)E(P_{12} + 1)}{(\psi m)(z^2)(P_{sf}^2)(P_{12})}} \quad (3.52)$$

Where:

$P_{sf}$	Allowable pressure	10 MPa
$\sigma_H$	Continuous surface pressure strength	30 MPa
S	Factor of safety	3
Mr	Moment	3 Nm
Ki	Operation coefficient	1.25
Kv	Speed coefficient	1
Kε	Clutch ratio coefficient	1
Km	Load distribution coefficient	1.2
ψm	Module width coefficient	15
z	Number of teeth	20



E	Elastic Modulus	1,100 MPa
P <sub>12</sub>	Gear ratio	2

The normal and face module requirements were found as 2.74 and 2.91 mm, respectively according to the surface pressure. Thus, the gears were designed with the normal and face module values of 3 and 3.19 mm.

The structural analysis was applied to the gears to check the mechanical strength of the gears with the module value of three. In the analysis, 3 Nm torques were applied to the gear (3.40).

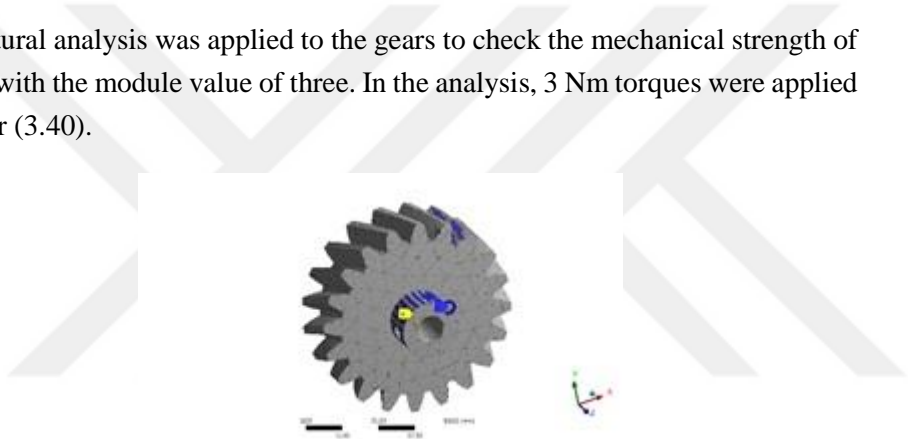


Figure 3.40 Analysis of the Gear

The safety factor was found as 2.8. According to these results, the helical gears with 3 mm modules were accepted as safe and they were produced with the designed dimensions.

The mechanical safety of the studs used as pin on the coupling for DC motor, main shaft connection was checked according to torsion stress.

$$\sigma_{sf} = \frac{Mr}{Wt} = \frac{(16)(F_t)(6)}{(\pi)(dp^3)} \leq \frac{Re}{S} \quad (3.53)$$

$$F_t = \frac{2Mr}{(z)(Do)} \quad (3.54)$$

Where:

$\sigma_{sf}$	Allowable stress	116.67 MPa
S	Factor of safety	3
Mr	Moment	6 Nm

$W_t$	Torsion strength moment	290.42 mm <sup>3</sup>
$F_t$	Tangential force	352.94 N
Re	Yield strength	350 MPa
z	Number of pin	2
Do	Inner diameter of coupling	17 mm
dp	Pin diameter	5 mm

The calculations for the coupling pin check were also conducted using Matlab Software. According to the calculation results, the studs were designed and produced with the diameter value of 5 mm. The calculations for coupling pin diameter with Matlab Software is shown in (Figure 3.41)

The screenshot shows the MATLAB R2013b environment. The left pane displays a script named 'stud\_check.m' with the following code:

```

1 - c1c
2 % Stud Diameter Check
3
4 Nr=6; %Nr
5 D=17; %Inner diameter
6 z=2; %Number of stud
7 Ft=(2*Nr*1000)/(z*D); %Tangential force
8
9 b=6; %Wall thickness of coupling
10 Rm=350; %Yield strength of stud material
11 S=3;
12 Mb=(16*Ft*b); %Bending moment
13
14 %Wb=(pi*dp^3) Bending moment strength mm^3
15 %qs=Mb*S/Wb*Rm Safe stress value
16
17 dp=[(8*Mb)/(pi*Rm)]^(1/3);
18 Wb=(pi*dp^3);
19 qs=Mb/Wb;
20 %As the minimum diameter was calculated 4.5 diameter was selected as 5.
21
22 fprintf('Stud Diameter Check\n');
23 fprintf('Ft = %4.2f N\n',Ft);
24 fprintf('Mb = %4.2f Nm\n',Mb);
25 fprintf('Wb = %4.2f mm^3\n',Wb);
26 fprintf('qs = %4.2f N/mm^2\n',qs);
27 fprintf('dp = %4.2f mm\n',dp);

```

The right pane shows the Command Window with the following output:

```

Stud Diameter Check
Ft = 352.94 N
Mb = 33882.35 Nm
Wb = 290.42 mm^3
qs = 116.67 N/mm^2
dp = 4.52 mm
>>

```

Figure 3.41 Coupling Pin Diameter Calculation

A motion and strength calculator software was also developed to conduct the required calculations. This software was designed in the Visual Studio Environment using the C# (C Sharp) programming language (Figure 3.42).

Figure 3.42 Motion and Strength Calculator

### 3.5. Electronic Systems and Hardware

There were two main electronic systems in the ladder climbing robot. The first one was designed and developed to control the motion through Lidar, ultrasonic and flame sensor data. The second was created for the fire detection via webcam and Raspberry pi. The circuit design created using Fritzing Software and the list of the components are given in Table 3.6 and Figure 3.43.

Table 3.6 Components of the Motion Control System

No	Name of Component	Pieces
1	Hokuyo UTM-30 LX Lidar	1
2	Raspberry pi 3b	1
3	Infrared Flame Sensors	3
4	Lipo Batteries	3
5	Ultrasonic Sensors	2
6	Power Bank	1
7	Arduino Mega 2560	1
8	Pololu Motor Drivers	3
9	DC Motors	3
10	Linear Actuators	2

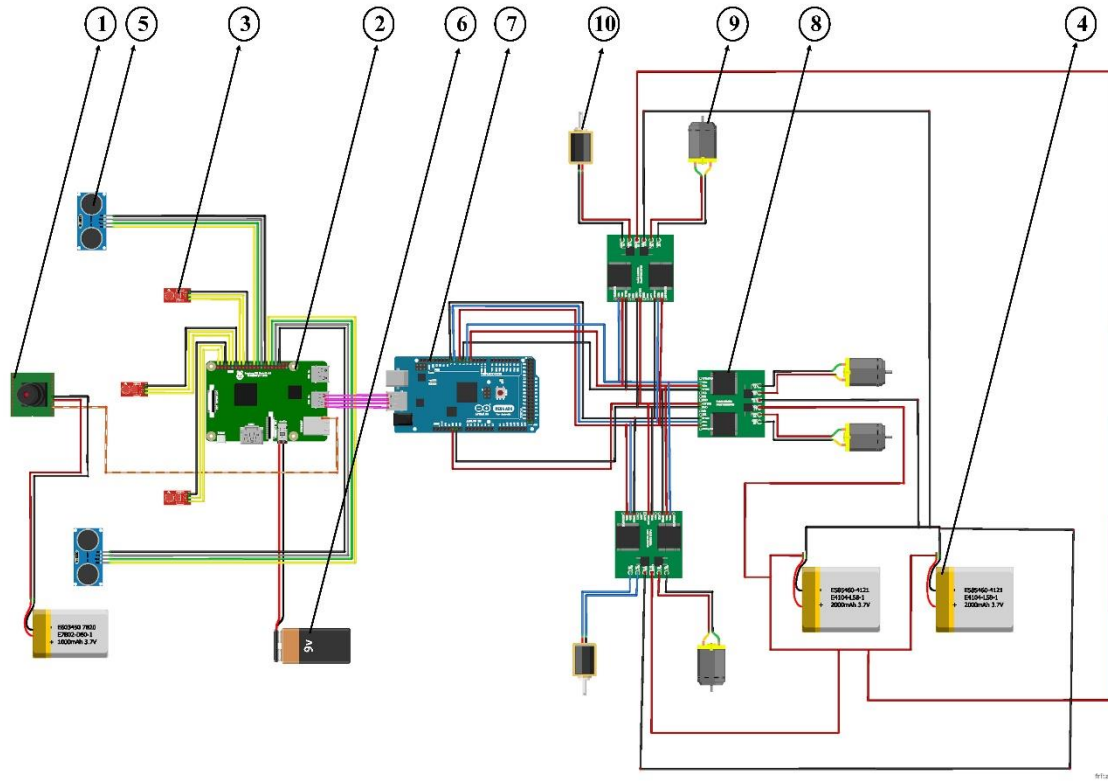


Figure 3.43 Circuit Design of the Motion Control System

The Hokuyo Lidar (Figure 3.44) was the main element of the motion control system. The 2D (Two-dimensional) distance data obtained from the Lidar. It was used to develop path planning and obstacle avoidance algorithms. The classification of the path (Corridor movement, fire seek etc.) was also conducted based on the information from the Lidar. The several technical specifications of the Hokuyo Lidar are given in Table 3.7.

Table 3.7 Technical Specifications of Hokuyo Lidar

<b>Specifications</b>	
<b>Power Source</b>	12 VDC
<b>Current</b>	0.7 A
<b>Detection Range</b>	0.1 to 30 m guaranteed, 60 m maximum
<b>Detection Object</b>	Minimum detectable width at 10m: 130mm
<b>Accuracy</b>	0.1 to 10m : $\pm 30\text{mm}$ , 10 to 30m : $\pm 50\text{mm}$
<b>Resolution</b>	0.1 to 10m : $< 10\text{ mm}$ , 10 to 30m : $< 30\text{mm}$
<b>Scan Angle</b>	$270^\circ$
<b>Scan Speed</b>	25 ms, 40 fps, (Motor speed : 2400rpm)
<b>Angular Resolution</b>	$0.25^\circ$
<b>Interface</b>	Ethernet
<b>Weight</b>	210g



Figure 3.44 Hokuyo 30-LX Lidar (Hokuyo, 2019)

The Hokuyo Lidar was connected to the Raspberry pi via Ethernet connection. The distance data were transferred to Raspberry pi via Ethernet port. The Hokuyo Lidar had  $270^{\circ}$  detection angle with the angular resolution of  $0.25^{\circ}$ . The full scan of  $270^{\circ}$  was 1081 steps. In the distance measurement process, Step “0” was the angle of “0” and Step “540” was the front of the Lidar with the angle value of “135”. The scanning process of the Hokuyo 30-LX Lidar is shown in Figure 3.45.

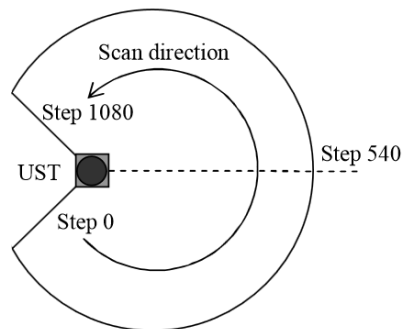


Figure 3.45 Scan Angle of Hokuyo 30-LX Lidar (Kazuya, 2015)

The Raspberry pi 3b was the main controller of the motion control electronic system (Figure 3.46). Raspbian Buster operating system was installed on Raspberry. The Lidar, infrared flame, ultrasonic distance sensors and Arduino were connected to the Raspberry. The algorithms of direction based angle calculation, fire search and find were run on the Raspberry pi using the sensor data of flame, ultrasonic and Lidar.

The communication between the Raspberry and Arduino was established via Serial communication protocol with the baud rate of 9600. The ultrasonic and flame sensor data were obtained with GPIO (General Purpose Input/Output) pins of the Raspberry. The power requirements of the sensors and Arduino were also satisfied from the 5V and ground pins and USB (Universal serial bus) of the Raspberry. A SD (Secure Digital) Memory card that all the software and algorithms for ladder climbing robot motion installed was put at the card slot. Several technical specification of Raspberry pi are given in Table 3.8.

Table 3.8 Technical Specifications of Raspberry pi 3b (Shinde and Mane, 2015)

<b>Specifications</b>	
<b>Power Source</b>	5 VDC
<b>Current</b>	2.5 A
<b>Storage</b>	Micro SD Card
<b>GPU (Graphical Processor Unit)</b>	Broadcom Video core-IV
<b>SoC (System on Chip)</b>	Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
<b>RAM (Random access memory)</b>	1 GB SDRAM
<b>USB</b>	Four pieces of USB 2.0
<b>Ethernet</b>	300 Mbps Giga Ethernet
<b>Expansion</b>	40 pins GPIO header
<b>Wi-Fi (Wireless fidelity)</b>	2.4 GHz and 5 GHz 802.11b/g/n/ac wireless LAN
<b>Bluetooth</b>	Bluetooth 4.2



Figure 3.46 Raspberry pi 3b (Robotistan, 2018 c)

Three pieces of the IR flame sensors (Figure 3.47) were connected to GPIO pins of the Raspberry to get the fire information from operation environment. These sensors were assembled to the front, front right and front left of the robotic system. According to the fire information (Digital output) the head direction of ladder climbing robot was adjusted.

The flame sensors could detect the fire source in the range of 760 to 1100 nm. They had ability to detect the fire sources in the maximum distance of 1,000 mm.



Figure 3.47 IR Flame Sensor (Robotistan, 2018 d)

Three pieces of Lipo batteries were used in the robotic system as power suppliers (Figure 3.48).



Figure 3.48 Lipo Battery for Lidar (Robotistan, 2018 e)

While two of them were dedicated for energy requirements of the DC motors, other was for supplying of Lidar. Lipo batteries for DC motors could provide 14.8 V



voltages, 3Ah current and totally 44 Wh energy. These batteries had four energy cells. The Lidar was supplied with a three cells Lipo battery with 11.1 V voltages, 1.35 Ah current rate and 15 Wh energy supplying capacity.

Two pieces of HC-SR04 Ultrasonic sensors were used for obstacle classification in the ladder climbing system. They could be worked with the 5 VDC and 15mA (Miliampere) operation voltages and current, respectively. They had the measurement range of 20 to 4,000 mm (Figure 3.50).



Figure 3.49 Ultrasonic Distance Sensor (Robotistan, 2018 f)

The distance sensors were dedicated to control the system to determine the required motion mode as climbing or linear. These sensors were positioned as up and down in the height distance of 150 mm to find the maximum sensing range.

A Xiaomi Power bank with the 20 Ah electric charge capacity was used in the system as a power source for the Raspberries (Figure 3.50).

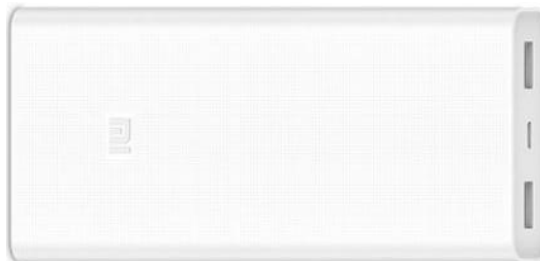


Figure 3.50 Xiaomi Power bank (Xiaomi, 2019)

This power bank had two pieces of USB type energy output slot. It could provide 5.1 V and 3.6 A outputs from the slots that sufficient to operate the two Raspberries for about 6 hours found (It was an experimental value obtained from operation tests).

An Arduino Mega 2560 was used in the system as a bridge between main controller Raspberry pi and motor controllers (Figure 3.51).



Figure 3.51 Arduino Mega 2560 (Robotistan, 2018 g)

It acquired the required data from Raspberry via serial connection and convey the required signals to motor control drivers. The on-off and direction control of DC motor for the stop, move forward and turning motions of robotic system were conducted with the signals obtained from the Arduino pins. The push and pull motions of the linear actuator were also activated with Arduino. The motion of the system were also adjusted with PWM (Pulse width modulation) pins of the Arduino.

Three pieces of Dual Pololu motor drivers were used in the system to drive the linear actuator mechanisms and DC motors with speed and direction control (Figure 3.52).

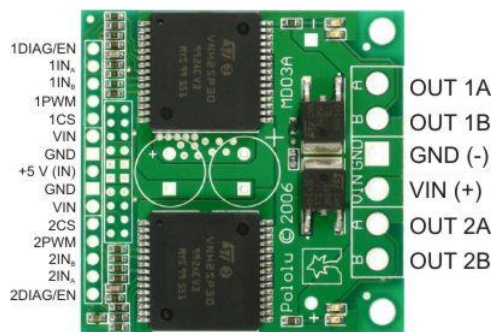


Figure 3.52 Pololu Dual Motor Driver (Pololu, 2019)

Each motor driver could control the motors with the voltage limits of 6 to 36 V and current for 9 amperes per channel.

As the DC motors for linear actuation mechanism and motion system could be operated with 12 VDC and 3.5 A maximum current, the selected drivers were sufficient for the requirements. The linear actuators and DC motors were categorized as right and left sides. Each side was established using one DC motor

and linear actuator. One motor control driver was used to control one side as it had dual control feature. The last driver was adopted to control of the support DC motors, which assembled to back side of the robotic system.

Three pieces of DC motors (Figure 3.53) were used in the motion system of the ladder climbing robot. Two of them were connected as right and left, other one was used as support that located the backside of the robot. Each DC motors could provide 4.2 Nm torque and 40 W power to the robotic system.



Figure 3.53 DC Motor of Motion System (Robotistan, 2018 a)

The other two DC motors (Figure 3.54) were also dedicated to linear actuation mechanism. These motors were selected with the power and load capacity of 32.4 W and 450 N, respectively.



Figure 3.54 DC Motor of Linear Actuation System (Robotistan, 2018 h)

The second electronic system was established via webcam and Raspberry pi 3b to apply the fire detection algorithm in real time application. The real time display of the operation environment obtained through webcam was acquired by the Raspberry. The algorithm of the fire detection was run at this Raspberry. According to the obtained display from the environment, the fire detection algorithm determined the fire probability. For this process, A4 Tech webcam (Figure 3.55) with the resolution of 1.3 MP (Megapixel) was connected to the Raspberry pi.



Figure 3.55 Fire Detection Webcam (A4 Tech, 2019)

Besides mentioned above the electronic systems provided real time observation and control opportunity to the operator via wireless connection network. The feature of the Raspberry for wireless connection network was used for this process. Using VNC (Virtual Network Computing) server the Raspberries were connected to the desktop PC.

### 3.6. Algorithms and Software

There were four main algorithms in the robotic system as “Motion Mode Decision”, “Direction Based Angle Calculation”, Fire search and Find” and “Fire Detection”. The first, motion mode decision algorithm determined about the motion type according to the measured height of the obstacle. The motion could be climbing or linear. The second, direction based angle calculation classified the operation path of the robot. In this approach, the situations were categorized into 17 main cases according to the location of the obstacles. The motion paths were defined according to their shapes and the rotation angles were assigned. The third algorithm was developed to find the location of the fire source through information from three flame sensors. The last was established using the Faster R-CNN deep learning model to determine the fire probability of the detected fire candidates.

#### 3.6.1. Motion Mode Decision

Motion mode decision was the first algorithm in the system to determine the motion mode of the robotic system. The ladder climbing robot had capability to move with linear and rolling motion types. It used the linear motion type for three purposes as:

1. Pass the negligible obstacles,
2. Move on the path defined according to the shape of the obstacles with shape bases angle calculation approach,
3. Try to reach to target (Fire source).

The robotic system activated the rolling motion to climb the obstacles or steps of the ladder. Two ultrasonic distance sensors as up and down with the height difference of 150 mm were positioned to chassis of the robotic system to activate climbing motion.

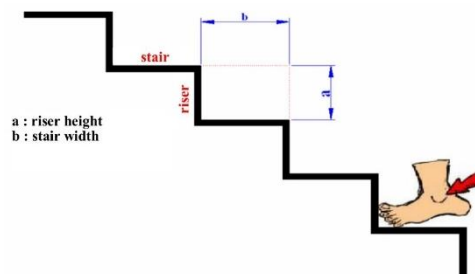


Figure 3.56 The Dimensions of Ladder Step (Megep, 2006)

The height and the width values of the ladder step were in the range of 120 to 200 and 260 to 320 mm, respectively (Figure 3.56). The height distance difference of the sensors were determined using this information. This difference value was also in the range of rolling capability of the robotic system, as the radius was 210 mm.

The flowchart of the motion type decision of robotic system is shown in Figure 3.57.

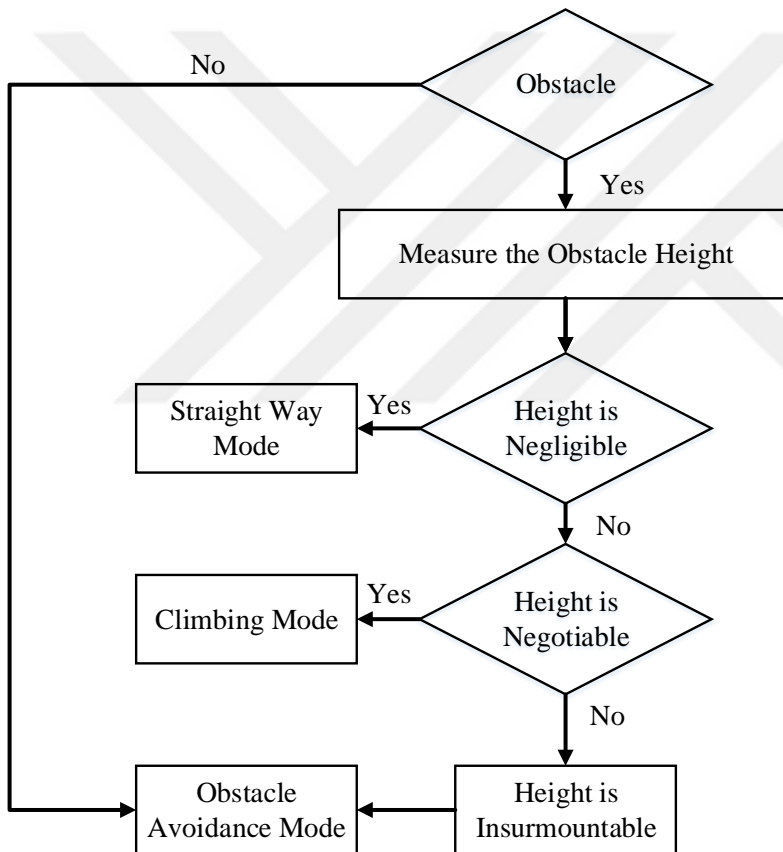


Figure 3.57 The Flowchart of the Motion Type Decision

According to the this algorithm if there is an obstacle in front of the robot;

- 1.If the obstacle was at a lower level than the down distance measurement sensor robotic system moved with the linear motion as the height of the obstacle was negligible.

2. If only down distance sensor took the measurement from the obstacle, it meant that the height of the obstacle was negotiable and the robotic system activated the rolling motion to pass obstacle.
3. If both up and down sensors took measurement from the obstacle, the type of the obstacle could be ladder or high wall. In this case, the difference of measurements from the distance sensors were important criteria. If the measurement difference of the distance sensors was in the range of 260 to 320 mm, it meant the type of the obstacle was ladder. The robotic system activated the rolling motion type and climbed the steps. Else, there was a wall type obstacle and the motion was determined according to the direction based angle calculation approach through information from Lidar.

### 3.6.2. Direction Based Angle Calculation

If the type of the motion was determined as obstacle avoidance, the robotic system tried to find the safe path to the target using the direction based angle calculation approach. In the direction based angle calculation approach, the field of view of the ladder climbing robot obtained via Lidar was divided into five regions (Figure 3.58) to determine about the shape of the obstacles and to decide the required motion types.

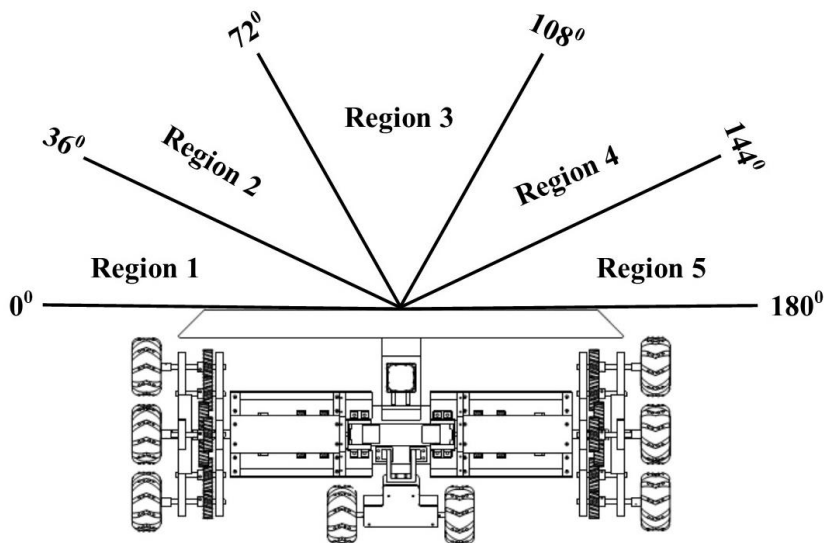


Figure 3.58 The Defined Regions in Direction Based Angle Calculation

The safe reaction distance of the system was determined as 1,600 mm twice of the maximum length of the robotic system. According to the occurrence region and distance of the obstacle, the types of the motion path (corner, corridor, trap and stop area) were decided.

Name of the Situation	Regions					Reaction of the System
	Region 1	Region 2	Region 3	Region 4	Region 5	
Fire Search						Try to Find the Fire Source
Front Obstacle			■			Turn Right (Smooth)
Front Left Obstacle		■				Turn Right (Smooth)
Left Obstacle	■					Turn Right (Smooth)
Front Left Obstacle	■	■				Turn Right (Smooth)
Front Right Obstacle				■		Turn Left (Smooth)
Right Obstacle					■	Turn Left (Smooth)
Front Right Obstacle				■	■	Turn Left (Smooth)
Left Corner		■	■			Turn Right (Sharp)
Right Corner			■	■		Turn Left (Sharp)
Obstacle		■	■	■		Turn Right (Sharp)
Left Corner	■	■	■			Turn Right (Sharp)
Right Corner			■	■	■	Turn Left (Sharp)
Corridor		■		■		Move Forward
Corridor	■				■	Move Forward
Corridor	■	■		■	■	Move Forward
Trap	■	■	■	■	■	Stop the Motion

Figure 3.59 The Situation and the Reaction of the System in Direction Based Angle Calculation Approach

As shown in the Figure 3.59, 17 different cases and reactions were defined according to the placement of the obstacle. In the “Fire Search”, case there was no obstacle in the field of view of and robotic system tried to find the fire source in the operation environment. In the “Front Obstacle” situation, the robotic system was turned to right smoothly with the angle of  $30^{\circ}$  as the default turning motion was defined as “Turn Right”. In the cases of “Front Left and Front Right Obstacle”, the



robotic system was turned to the side smoothly, which not contain obstacles to find the safe path. In the situations of “Right and Left Obstacle”, the robotic system moved forward but it didn’t try to search for fire source as the turning motion could lead to crash to obstacle. In the other case named as “Left and Right Corner” the robot turned to the side sharply, which was safe. The sharp rotations were defined as about  $90^0$  to pass the obstacle and adjust the motion according to the geometric shape of the corners. In the case of the “Obstacle”, the robotic system was turned to right sharply to pass the obstacle. The motion of the “Corridor” was defined as the move forward and not try to find the fire source until it finished. In the “Trap”, there was no gap for the motion and the robot was stopped. A different “Stop Area” was defined in the Direction based angle calculation approach. If there was an obstacle closer than the safe reaction distance the robotic system was stopped an waited until the obstacle removed from the path. Dynamic obstacle avoidance was also defined in the direction based angle calculation approach. If an obstacle placed on the path suddenly, the first target of the robot was changed as “Dynamic Obstacle Avoidance”. In this case, the robot tried to pass the dynamic obstacle according to location in the region zone. The linear speed of the robotic system was determined as 0.5 m/s in the direction based angle calculation approach with applied experimental tests.

### **3.6.3. Fire Search and Find**

The “Fire Search and Find” algorithm was developed to find the location of the fire source based on the data obtained from the IR flame sensors. The IR flame sensors were placed on the robotic system as front, right and left. The left and right sensors were positioned with the angle of  $45^0$  to the normal of the front. This algorithm was activated in the situation of “No Obstacle” in the field of view. If the robotic system was in the case of “Fire Seek” in the shape base angle calculation algorithm, it tried to reach to final target, fire source.

The safe distance to the fire source was determined as 800 mm. If any of the flame sensors detected a fire source in their detection distance the robot adjusted its head and direction of the motion according to the fire data. In the first step, the robot checked the front flame sensor data. If there was a fire source it stopped and waited to determine the fire probability of the candidate by the “Fire Detection” algorithm. If there was no fire source in front, the robotic system check the data of the right and left flame sensors, respectively. According to the location of the fire source,

such as it was on the right, the robot turned to right until it got into sight of the front flame sensor and then robot stopped and wait. The same process was applied for the left flame sensor detection. The flowchart of the Fire Search and Find Algorithm is shown in Figure 3.60.

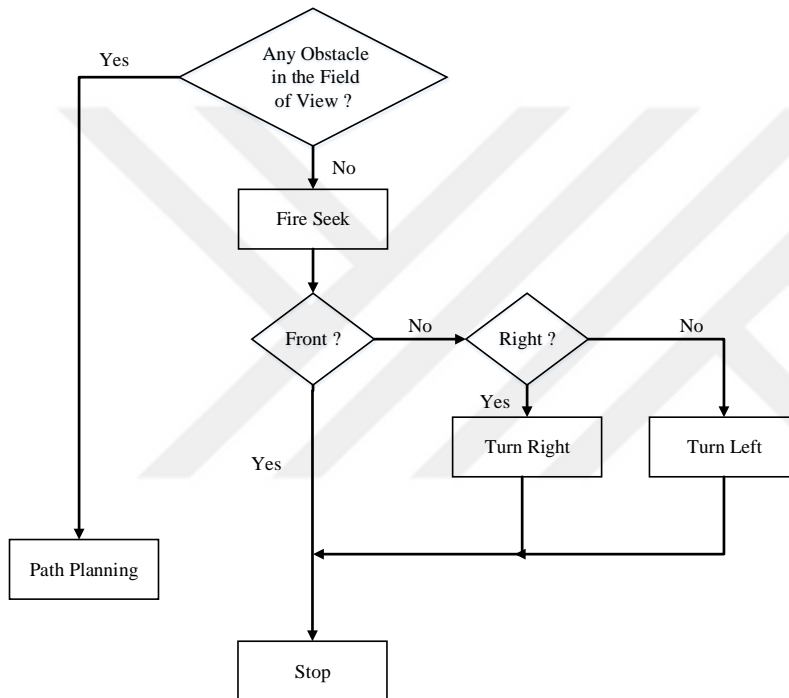


Figure 3.60 The Flowchart of the Fire Search and Find Algorithm

#### 3.6.4. Fire Detection

The “Fire Detection” algorithm was designed and developed to determine the fire probability of the fire candidates. The fire candidates were determined using the data from the Fire search and find algorithm. If any object in operation area was described as, “Fire Candidate” by the fire search and find algorithm, the fire detection algorithm was activated to find the probability of the candidate.

A fire detector software using the Faster R-CNN deep learning model was developed for the fire detection.

Faster R-CNN is an object detection model developed by Ross Girshick, Shaoqing Ren, Kaiming He and Jian Sun in 2015. Currently, this model are used commonly

for object detection applications. Faster R-CNN is consisted of three parts as convolution layers, region proposal network and classes & bounding box prediction.

CNN (Convolutional neural network) is one of the main thing to do image recognition and classification. CNN image classification takes an input image, process it and classify under certain categories. The input image is understood as an array of pixels. In CNN, each input image passes it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers. Then it applies Softmax function to classify an object with probabilistic values between “0” and “1”. The general structure of CNN is shown in Figure 3.61.

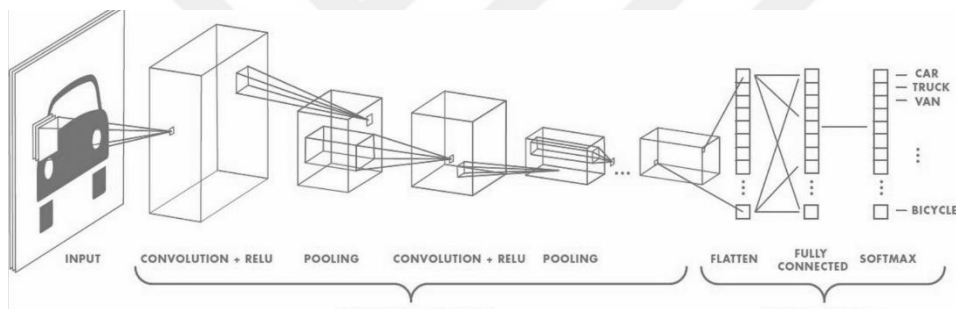


Figure 3.61 The Structure of CNN (Sucuoglu et al., 2019)

Convolution is the first footstep to extract features from an input image. Convolution holds the relationship between pixels via learning image features using small squares of input data. Pooling layers is used to reduce the number of parameters when the input images are too large. These layers can be different types as max pooling that takes the largest element from the rectified feature map, average pooling takes the average value and sum pooling used the sum of the values. In the fully connected layer, the matrix is flattened into the vector. The vector features are combined to create the new model. After all, an activation function e.g. the softmax is used to categorize the output. RPN (Region proposal network) is the second component of the Faster R-CNN structure. RPN is a neural network sliding on the last feature map of the convolution layers. It predicts that there is an object or not and bounding boxes of the objects (Sucuoglu et al., 2019).

In the learning process of the fire detector, the experiments were conducted using a system with the specifications of NVidia GeForce GTX 1070 Ti with 17 GB onboard memory. The required tools (Python 3.5, Tensorflow 1.13.1, OpenCV, CUDA-cuDNN toolkits) were installed on the computer with Anaconda Virtual

Environment. 1,000 images that include fire and non-fire scenes were used for the training process. Candle fire scenes were also used for training. 80 and 20 percent of these images were dedicated for training and validation, respectively. The images were classed and labelled as “Fire” and “No Fire” using LabelImg Software. The selected images with fire scene from training process are shown in Figure 3.62



Figure 3.62 The Selected Images from the Training Process (Personal Archive)

The initial learning process was applied as two stages. In the first learning, the number of steps was selected as 10,000 with the learning rate of 0.2. In the second, with 30,000 number of steps, the learning rate was increased to 0.1 to get more reliable results. Three epochs were applied to each of the steps. The softmax function was selected as the converter. The threshold for fire scenes was selected as 60% to increase the accuracy of the detection.

The Tensorboard toolkit was used to check the progress of the training and to obtain the metrics (Loss function and accuracy). The obtained graph from the Tensorboard is shown in Figure 3.63

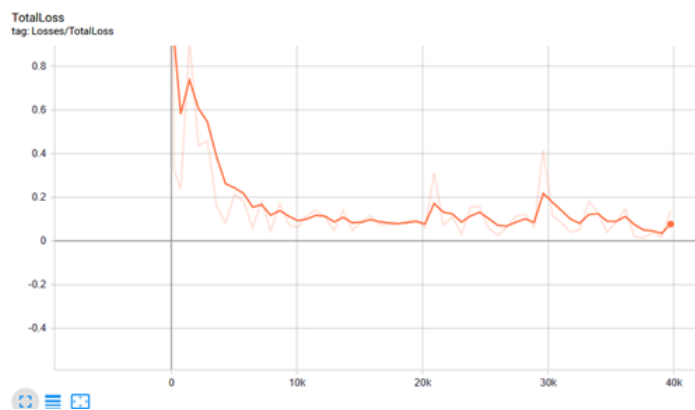


Figure 3.63 Total Loss Values in Training Process

It was obtained from the Total loss value graph that the total loss value was two at the starting of the training; it was decreased to the level of 0.02 at the end of the process. Therefore, the accuracy of the trained model was accepted as reliable and fire detector software was used in the robotic system. The flowchart of the fire detection with fire detector software is shown in Figure 3.64.

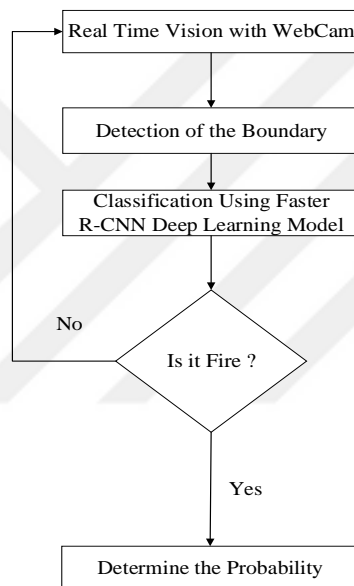


Figure 3.64 Implementation of Fire Detector Software

A “Serial Control Software” (Figure 3.65) was developed in the Visual Studio Environment using the C# programming language to check the performance of the transmission and motion system on the test carrier.

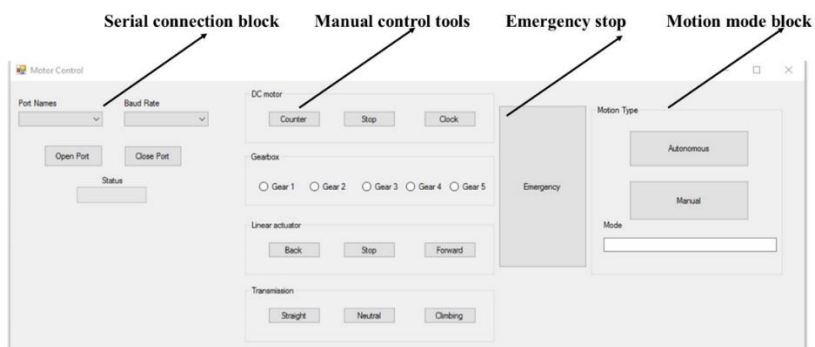


Figure 3.65 Serial Control Software

This software included the serial connection block for the communication between Arduino and computer, manual control tools to drive motion and linear actuation system with different speeds, motion mode block for transition between manual and autonomous control and emergency to stop the motion of the systems.

Ladder climbing robot control software (Suitable to run in Raspberry pi) was also developed. This control GUI (Graphical User Interface) was designed in PyQt 5 designer with the software of PyQt. Python Programming Language (Version of 3.7) was used to create the GUI, to obtain the sensor (flame and ultrasonic) data, read the Lidar information dynamically and to create the algorithms mentioned above.



Figure 3.66 Ladder Climbing Robot Control Software

The control software allowed to operator to control the robot with manual and autonomous motion modes. It also provided the information from the sensors and Lidar.

Besides mentioned above the developed software of “Ladder Climbing Control” and “Fire Detector” provided observation and control opportunities to the operator via wireless network. For this purpose, VNC server was installed on the remote controller computer. The two Raspberries were connected to the remote controller computer through VNC server. All of the functions of the software were controlled and applied from the remote controller computer.

## 4. RESULTS AND DISCUSSION

In this chapter of the thesis study, the applied tests to the robotic system and the obtained results were presented and explained.

### 4.1. Performances of Transmission and Motion Systems

The first experimental tests were applied to transmission and motion systems to observe performances. The robotic system was constructed with two elements of transmission and motion systems placed as right and the left sides. The performance test was applied to elements of one side. A wooden test carrier (Figure 4.1) was designed and manufactured to carry the robotic system components in performance test's stage. These components were the transmission and motion systems, electronic components (Arduino mega 2560, Pololu motor driver, cables and distance sensors). The transmission and motion systems were fixed to test board with L shape profiles.

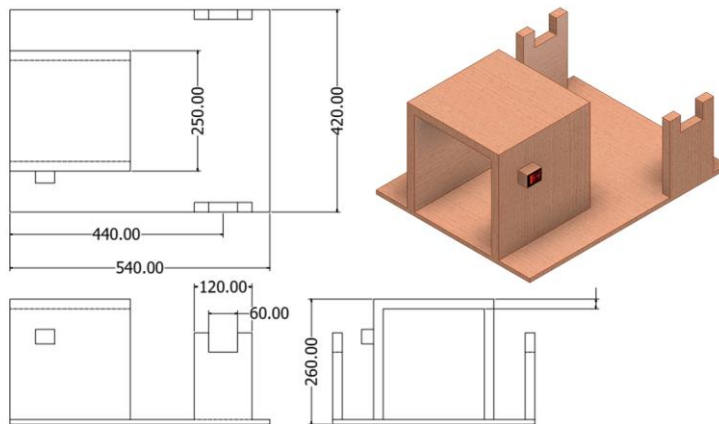


Figure 4.1 Wooden Test Carrier

The performance tests were applied for two motion modes as autonomous according to distance sensors data and manual. The test setup is shown in Figure 4.2.



Figure 4.2 General Test Setup (Personal Archive)

In the autonomous, the type of the motion was decided automatically by the motion decision algorithm according to the obtained data from distance sensors. In the manual, the commands to activate the motion and transmission systems were sent from control computer to Arduino via serial communication. The variables of “gear and gear\_t” were defined as PWM values to control the speed of DC motors. The integer of “s” was used for the initialization of the system. The variable of the “linear” was also dedicated to tell the position of the linear actuation mechanism to control system. The operation of the transmission system was conducted with a process as mentioned below:

If the value of the “s” was equal to zero the DC motor was rotated in counter clockwise direction and the linear actuator was pulled for 2 second period. The “s” values was assigned as “1”. Therefore, the transmission system was initialized and always started its operation from start position and stop situation.

When the sensors detected the obstacles in the range that cause to the climbing motion, if the linear actuator was at the start position the linear actuator mechanism was pushed for two seconds to activate the climbing motion. In other case, (linear actuator was at the end position) only DC motors of the motion system was driven for the climbing.



When the situation was linear motion, if the linear actuator was at the end position, it was pulled for two seconds to activate the linear motion. Else, only DC motor was driven.

In the manual mode test, the elements of the motion and transmission systems were tested manually using the serial control software. Five different speeds were defined with the gear values. The gears were between the values of “1” and “5” with the rotational velocity outputs of 40 to 80 rpm. For the transmission system test the DC motors of the linear actuator mechanism and motion system were driven together to prevent from the gear stripping.

The performance tests of the transmission and motion systems for real linear and climbing motions simulations are shown in Figures 4.3 and 4.4, respectively.



Figure 4.3 Linear Motion Test (Personal Archive)



Figure 4.4 Climbing Motion Test (Personal Archive)

A Plate established with styrofoam and wood was used as a ground in the linear motion test. The gear values, forward, backward and stop functions of the motion system was tested. It was observed from these tests that the power, torque and rotational velocity values of the selected DC motors were sufficient for the linear operation. In the climbing motion test, the transmission system performance for transition was tested. It was also observed from this test that the tolerance values, shape and the mechanical strength of the helical gears and the force property of the DC motor of the linear actuation mechanism were sufficient for transition. The power and the torque feature of the motion system's DC motors were also satisfactory for the climbing. The reaction of the system in the autonomous mode was also tested. It was also observed in the test that the transmission system activated the climbing and linear motions successfully when the required situations were occurred.

#### **4.2. Lidar Interface**

A Lidar data test interface was designed to classification of the path according to the location of the obstacle. This interface was developed in the Visual Studio Environment using the C# language. The communication between the Lidar and computer was established via Ethernet connection with the IP (Internet Protocol)

address of the Hokuyo Lidar (192.168.0.10). The distance to obstacle data was obtained with UrgBenri Software in real time (The official GUI of the Hokuyo Lidar). In the UrgBenri Software, the gaps (places without obstacle) were indicated with the green color. When an obstacle was occurred in the field of view, the free motion areas were restricted according to the location and the size of the obstacle. This software also gave opportunity to record the measured distance data as CSV file dynamically. The distance data was recorded as CSV file to write into the developed interface. The dynamic data stream between the interface and CSV file was created with the embedded codes of the interface. The required conditions of the shape base angle calculation for path planning such as stop area, fire seek and front right obstacle were established and the reactions of the interface were tested.

For the stop area condition, an obstacle was placed on a distance closer than the safe reaction distance of the robotic system (Figure 4.5).



Figure 4.5 The Condition of the Stop Area (Personal Archive)

It was observed from both developed interface and UrgBenri Software that the system reaction was “There is an obstacle in the stop area” as shown in Figures 4.6 and 4.7, respectively.

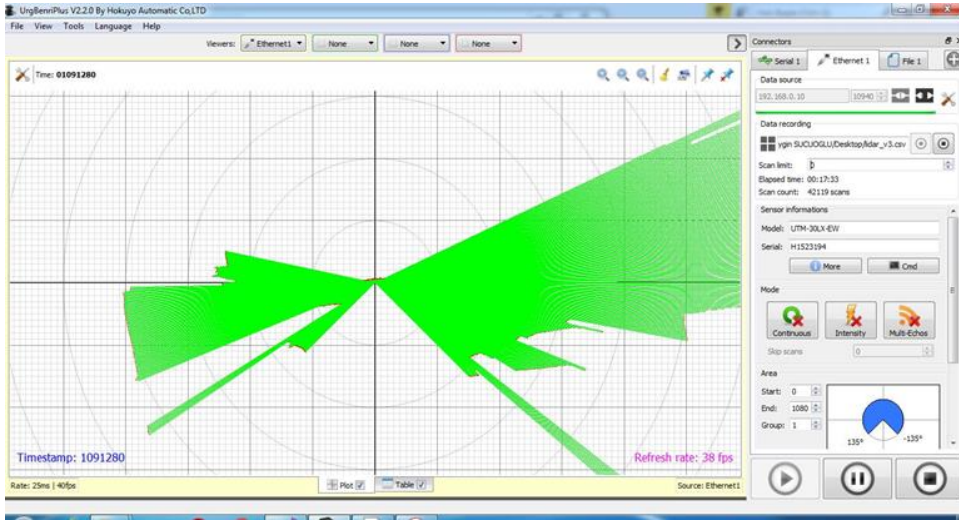


Figure 4.6 The Distance Measurement in UrgBenri Software

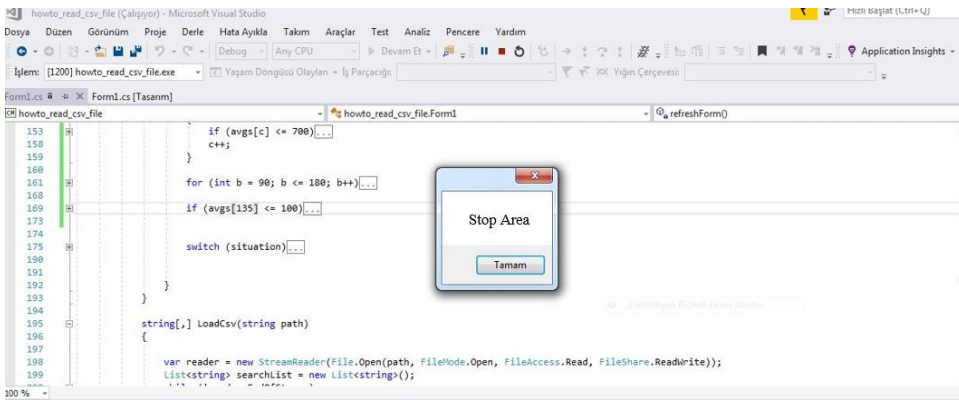


Figure 4.7 Stop Area Reaction

In the other condition, an obstacle was placed on the front right side of the Lidar. The reaction of the interface was tested for this front right obstacle case of the direction based angle calculation approach. The placement of the obstacle and the reaction of the interface are shown in Figures 4.8 and 4.9, respectively.



Figure 4.8 Front Right Obstacle Condition (Personal Archive)

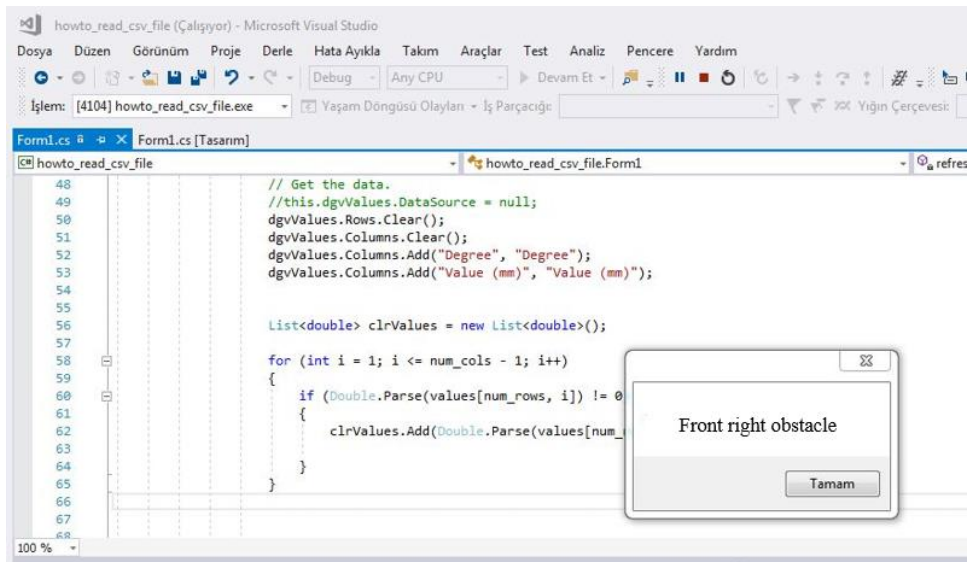


Figure 4.9 Front Right Obstacle Reaction

As understood from tests results the reactions of the developed interface were correct under the established conditions. As the developed interface could give the required reactions to the conditions using Hokuyo Lidar, the algorithms of the

direction based angle calculation approach was created in Raspberry pi using Python language, which was suitable to operate in the developed robotic system.

### 4.3. Fire Detector

The fire detector software developed with fire detection algorithm using Faster R-CNN deep learning model was first tested in computer environment before adopted to robotic system. Anaconda Virtual Environments and Python version of 3.5 were installed on the test computer. A webcam was connected to computer through USB port to acquire the environmental vision. The fire detection software was initialized with the Console of the Anaconda. The fire detection software was triggered and the reaction of the software was observed as shown in Figure 4.10.



Figure 4.10 Implementation of Fire Detector Software (Personal Archive)

The fire detector software was tested with lighter and candidate flames in different light conditions. It was observed from the test results that the developed software could classify the fire source with the score of 99%. When this detection score combined with the metrics obtained from the Tensorboard tool (0.02 loss value after the learning process that include three epochs with 40,000 number of steps), the fire detector software accepted as reliable for fire detection task.

### 4.4. Tests of the Algorithms

In this section of the thesis study the developed algorithms (Direction based angle calculation, motion type decision, fire search and find and fire detection) were tested. In the first stage of tests of algorithms, a smaller size robotic system named



as “Test Robot” was used (Figure 4.11). Test Robot was equipped with the same hardware of the ladder climbing robot fire detection robot.



Figure 4.11 Test Robot (Personal Archive)

The technical drawing of the test robot with the general dimensions is shown in Figure 4.12.

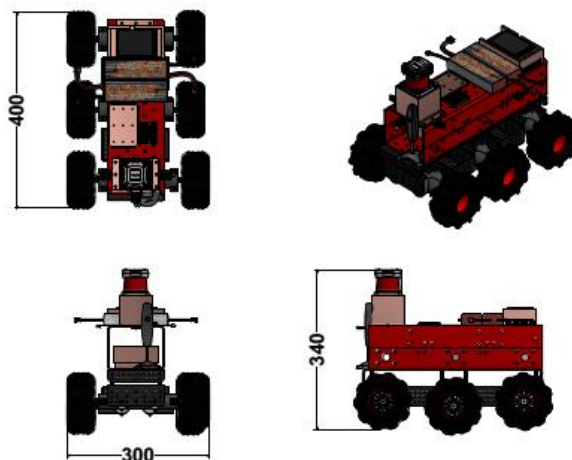


Figure 4.12 General Dimensions of Test Robot(Personal Archive)

#### 4.4.1. Direction Based Angle Calculation

The first tested algorithm was the direction based angle calculation. In this test, the test beds were designed for different conditions contain the corridor, right & left corners, front and front right & left obstacles and dynamic obstacles. All these mazes were used to evaluate the capability of the direction based angle calculation approach for path planning and obstacle avoidance.

The first test environment created like a maze included the front left obstacle, front right obstacle and left corner. The plan drawing of the first test environment is shown in Figure 4.13.

The total wide width and the length of the maze were 2.5 and 4.4 m, respectively. In the maze, the first front left type obstacle with 1.2 m wide width was located to 1m distance from the starting point. The second obstacle was front right type with 0.9 m wide width was put the 3.4 m distance from starting point (Figure 4.13).

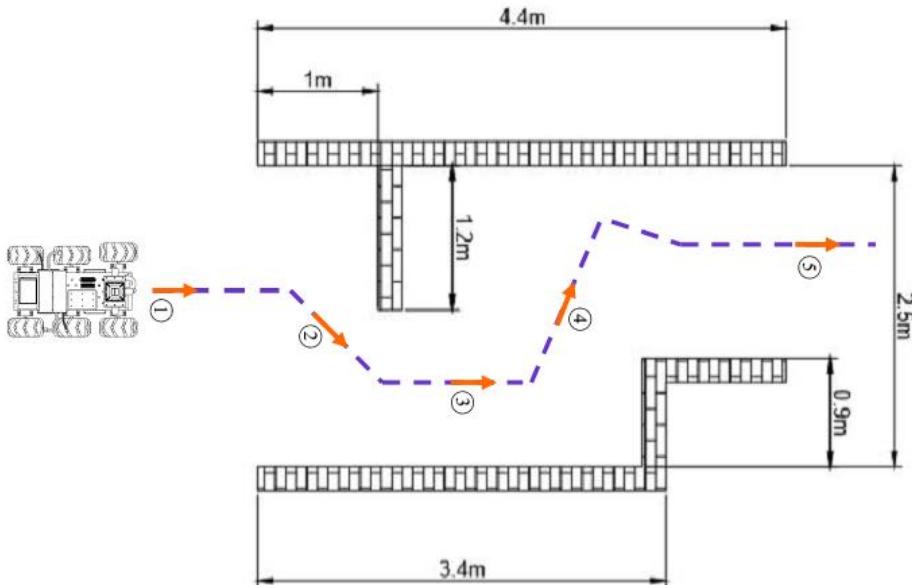


Figure 4.13 Plan Drawing of the First Test Bed

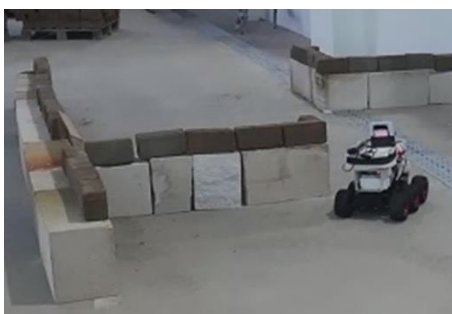


The reactions of the robotic system were expected as starting with corridor movement, smooth turning to right, corridor movement, smooth turning to left, sharp turning to right and finishing the maze with corridor movement.

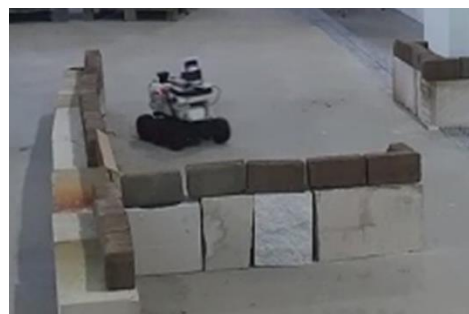


Figure 4.14 Test Bed with Front Left & Right Obstacles and Left Corner (Personal Archive)

As expected the test robot started its motion with move forward movement. Then, it adjusted the head of direction and turned right smoothly. In a short time, the test robot moved forward as it detected the corridor. When it detected the front right obstacle, it switched the motion as turning left smoothly until reached to left corner. In the left corner, it turned right sharply. Then with last move forward motion, the test robot finished the maze. The experimental results from the first test bed environment is shown in Figure 4.15.



(a)



(b)

Figure 4.15 Experimental Results from Test Bed One (a) Front Left Obstacle, (b) Left Corner (Personal Archive)

The second test bed was designed with right corner, front right obstacle, left corner and right obstacle. The sketch plan drawing of the second test bed is shown in Figure 4.16.

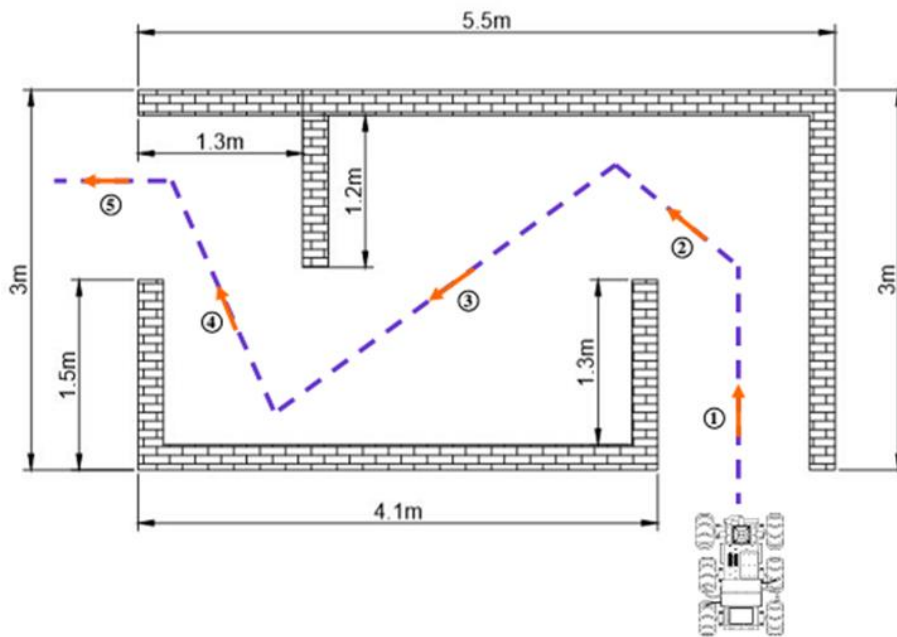


Figure 4.16 Plan Drawing of the Second Test Bed

The total width and the length of the path were 3 and 5.5 m, respectively. This path was started with a corridor with the length of 1.3 m. It was a right corner at the end of the corridor. Then, a front right type obstacle with width value of 1.2 m was located on a distance from the finish point of 1.3 m. The maze was finished with left corner and right obstacle, respectively (Figure 4.17).



Figure 4.17 Test Bed with Right & Left Corners and Right Obstacle  
(Personal Archive)

The reactions of the robotic system in the second test bed were expected as starting with corridor movement, sharp turning to left, smooth turning to left, corridor movement, sharp turning to right and finishing the maze with smooth left turning and moving forward.



Figure 4.18 Experimental Results from Test Bed Two  
(a) Right Corner, (b) Corridor Movement (Personal Archive)

As expected, the robotic system started the motion with move forward as it was in the corridor. Then, suddenly it switched the motion type to sharp turning left because of the right corner. When it detected the front right obstacle, it turned to left smoothly and continued with move forward. Then, because of the left corner, it turned right sharply and it finished the maze with smooth left turning and move forward motions. The experimental test results from the second test bed are shown in Figure 4.18.

In the third test bed for a difference from the second, one more obstacle with the sizes of 350x550 mm was added to robot path when it was at the corridor movement. This test maze included right corner, front right obstacle, front left obstacle, left corner and right obstacle. The plan drawing of the third test maze is shown in Figure 4.19.

The third path had the same width and the length values with the second. This path was started with a corridor with the length of 1.3 m. It was a right corner at the end of the corridor. Then, a front right type obstacle with width value of 1.2 m was located on a distance from the finish point of 1.3 m. A Front left type obstacle was also put the front direction of the robotic system. The path was finished with left corner and right obstacle, respectively (Figure 4.20).

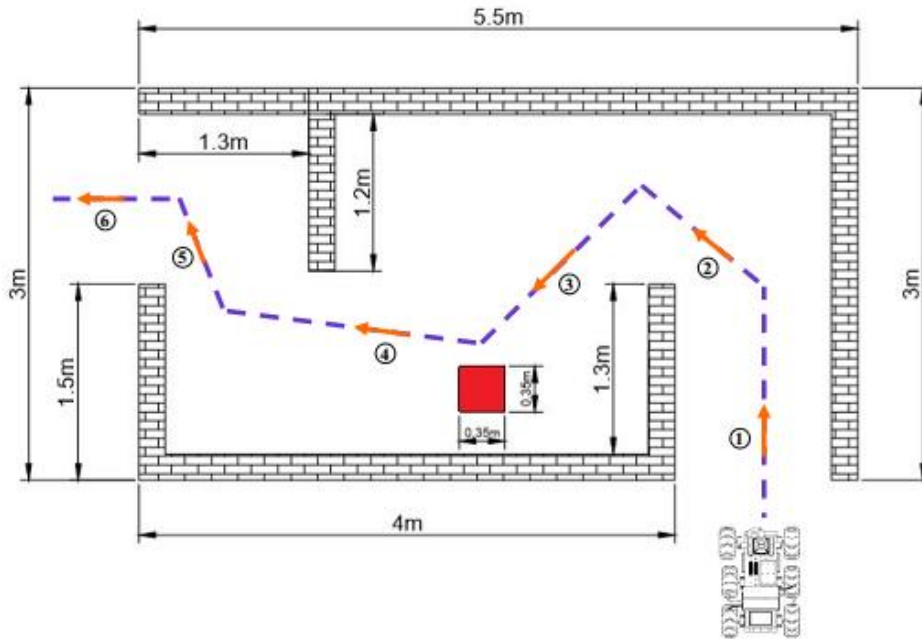


Figure 4.19 Plan Drawing of the Third Test Bed

The reactions of the robotic system in the third test bed were expected as starting with corridor movement, sharp turning to left, smooth turning to left, smooth turning to right because of the front left obstacle, corridor movement, sharp turning to right and finishing the maze with smooth left turning and moving forward.



Figure 4.20 Test Bed with Right & Left Corners and Front Right & Left Obstacles (Personal Archive)

As expected, the robotic system started the motion with move forward as it was in the corridor, similar to second test bed. Then, it changed the motion type to turning

left as the occurrence of the right corner. It turned to left smoothly because of the right obstacle and moved forward. When it detected the front left obstacle, it turned right smoothly and moved forward. Then, because of the front right obstacle, it turned right smoothly one more time and moved forward until reached to the left corner. It passed the left corner with sharp right turning and finished the maze with smooth left turning and move forward motions. The experimental test results from the third test bed are shown in Figure 4.21.



Figure 4.21 Experimental Results from Test Bed Three  
(a) Front Left Obstacle, (b) Front Right Obstacle (Personal Archive)

The fourth test bed was designed looking like the capital letter “S”. This test bed was created to check especially sudden reaction capability of the robotic system to changed obstacle types. This test bed contained three curves with the radius values of 3.5, 1.2 and 2 m, respectively. The total length values were created as 6.7 m at the X-axis and 5.5 m at the Y-axis. The wide width was 1.2 m. The fourth test bed also included front right & front obstacles and left corner numbered as 2, 3 and 4, respectively as shown in Figure 4.22. As a difference from the formers the distance among the obstacles were so close to each other (Figure 4.23). Therefore, the robotic system had limited time to determine and to switch the required movement type.

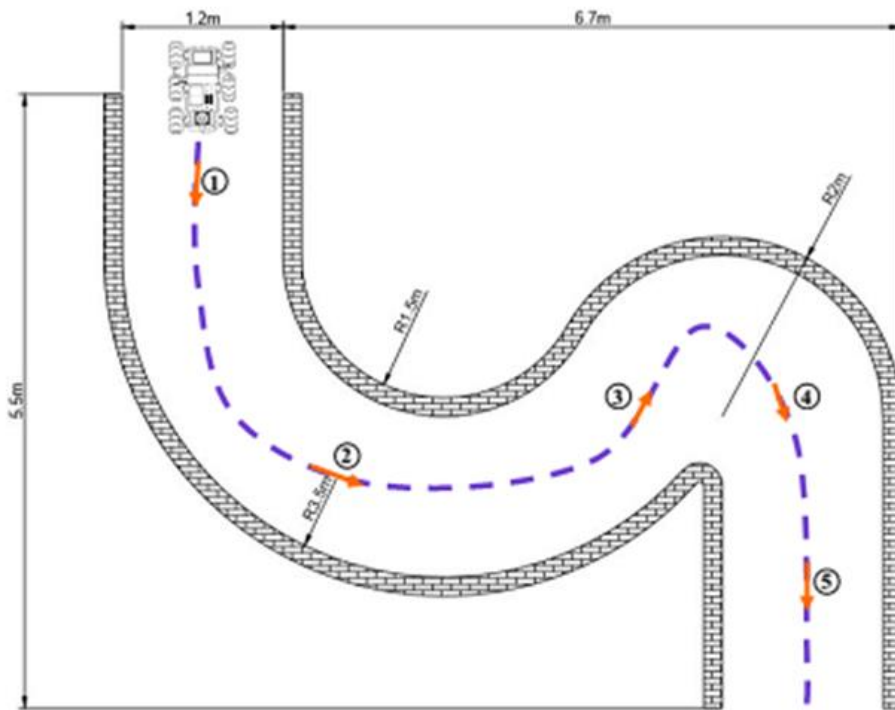


Figure 4.22 Plan Drawing of the S Type Test Bed

The reactions of the robotic system in the S test bed were expected as starting with corridor movement, smooth turning to left, smooth turning to right, sharp right turning and finishing the path with moving forward. In the S type test bed the robotic system had to determine and change the movement type in short time intervals to prevent from the crash and to finish the path.



Figure 4.23 S Type Test Bed (Personal Archive)

In the S type test bed, the robotic system started the motion with move forward as it detected a corridor movement. Then, with the entrance of the first curve, it found a

front right obstacle and suddenly changed the motion type to smooth turning left. It continued the motion with move forward until reached to the curve with the radius value of 2m. It detected the curve as front obstacle because of the motion direction and turned right smoothly. After it passed to curve, it switched the motion as sharp turn right to pass the left corner and finished the maze with the motion of move forward. The experimental test results from the S type test bed is shown in Figure 4.24.



(a)



(b)

Figure 4.24 Experimental Results from the S Type Test Bed  
(a) Front Right Obstacle, (b) Front Obstacle (Personal Archive)

In the fifth step of the direction based angle calculation algorithm test, a complicated environment was created with six obstacles. These obstacles was located in a rectangular type boundary with the length and width values of 3 and 2 m, respectively. The test environment with complicated obstacles is shown in Figure 4.25.



Figure 4.25 Test Bed with Complicated Obstacles (Personal Archive)

In the complicated test environment, the robotic system first adjusted its head to pass obstacles from mid of them. Then, it detected front left obstacle so turned smoothly right and because of right corner type obstacle it turned left sharply. When it moved forward, it found a front right obstacle and turned left smoothly. Then, it



finished the test environment with move forward motions. The experimental test results obtained from complicated obstacles environment is shown in Figure 4.26.



Figure 4.26 Test Bed with Complicated Obstacles  
(a) Right Corner, (b) Front Right Obstacle (Personal Archive)

The performance of the direction based angle calculation approach was also evaluated in an open environment where the obstacles were placed dynamically on the path of the robotic system.

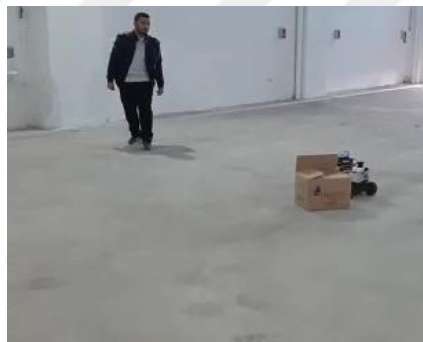


Figure 4.27 First Case of Dynamic Obstacle Avoidance (Personal Archive)

As shown in Figure 4.27 a box was thrown suddenly to the motion path of the robotic system. The robotic system detected the obstacle as front right type. It stopped the motion for a short time period (0.1 second), and then passed the obstacle with a smooth turning left.

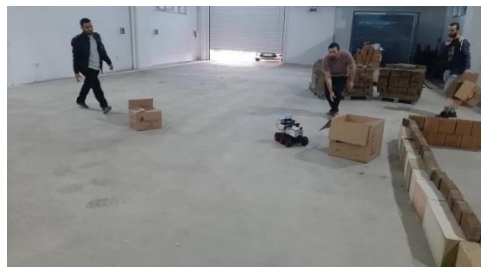


Figure 4.28 Second Case of Dynamic Obstacle Avoidance (Personal Archive)



Then as presented in Figure 4.28, another box was thrown the front of the robotic system. It determined the obstacle type as front left waited and passed the obstacle with smooth turning right motion.

The performance of the developed direction based algorithm was tested in five different mazes. These mazes were created with different types of the obstacles such as front, left corner and front right, etc. The robotic system had to conduct the path plan and obstacle avoidance when it moved on the created mazes.

In the direction based angle calculation approach, the types of the obstacles were classified and defined to reduce the computation time and to prevent from the burden. It was observed from the tests that the robotic system determined to correct motion type in short periods and it passed the obstacles without crash. It also gave the required reactions according to the shape of the obstacles and planned the movement. Especially, in the S type maze the time interval among the different required motion types were so limited. However, the robot could pass the obstacles and it finished the maze, successfully.

The reaction time of the robotic system was also tested in the complicated and dynamic obstacle environments. It was observed from these test sets that the robotic system could switch the motion type sequentially and overcome the obstacles. According to the results of the mentioned above, the direction based angle calculation algorithm provided the satisfactory path planning and obstacle avoidance capability to the robotic system. This approach can be also used in both outdoor and indoor operation environments due to its complicated and dynamic obstacles avoidance abilities.

#### **4.4.2. Fire Search and Find**

The second tested function of the developed robotic system was the fire search and find algorithm. This algorithm was designed and developed to find the location of the fire source in the operation environment. This algorithm was structured based on fused data obtained from three IR sensors. The IR sensors were placed on the front of the robotic system as front, right and left. The left and right sensors were assembled with the angle of  $45^{\circ}$  to the normal (Figure 4.29).

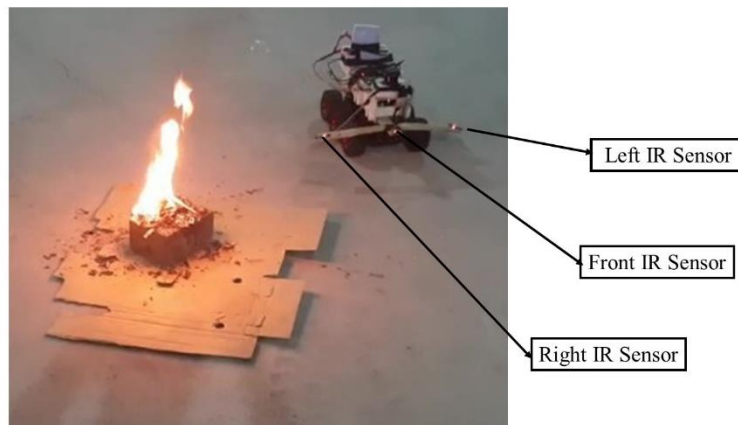


Figure 4.29 Placement of the IR Sensors (Personal Archive)

The fire search and find algorithm was activated when the direction based angle calculation determined the situation as “Fire Seek” mean of no obstacle in the field of view. The safe distance to the fire source in this algorithm was determined as 800 mm to prevent the robotic system from the heat of the fire.

The robotic system tried to find the fire source according to the fused data from the IR sensors. If the front sensor detected a fire source it stopped, waited and activated the fire detection algorithm to determine the fire probability of the candidate. In the situation of no fire source in front, the robotic system checked the data from right and left sensors, respectively. According to the right and left sensors data the robotic system switched its motion direction to the fire source by turning right or left. After that as the fire candidate was located in the line sight of the front sensor it stopped, waited and activated the fire detection algorithm. If there was no fire data from the sensors the robotic system followed on the commands from the direction based angle calculation algorithm

The first test was conducted for the situation of front fire source detection. The steps of this condition are shown in Figure 4.30.



(a)



(b)

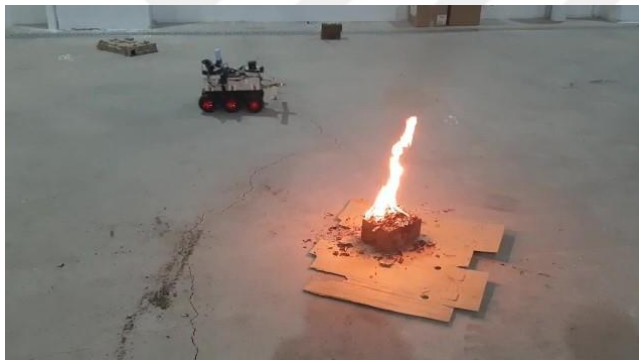


(c)

Figure 4.30 Front Fire Source Detection Process (a) Placement of the Fire Source, (b) Detection of the Fire Source in Front of the Robotic System, (c) The motion of the Robotic System After Fire Source is Removed (Personal Archive)

In the first step, the fire source was located in front of the robotic system. After a short moving forward motion, the robotic system was detected the fire source. When it detected the fire source, the robotic system activated the fire detection algorithm and determine the fire probability. Then, the fire source was removed from the path of the robotic system. As there was no fire source in front, the robotic system followed on the commands from the direction based angle calculation approach and moved forward.

The second test condition was created by placement of the fire source to the right side according to the robotic system motion path. The steps of this case are shown in Figure 4.31.



(a)

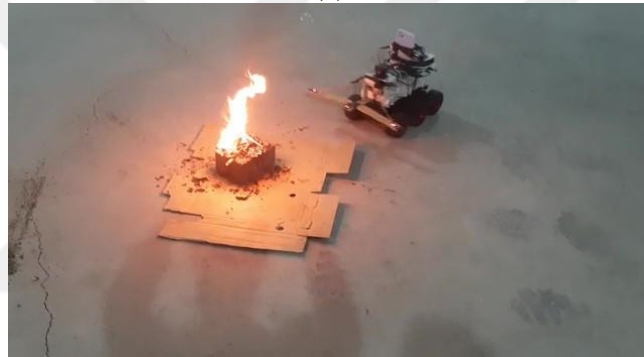


(b)

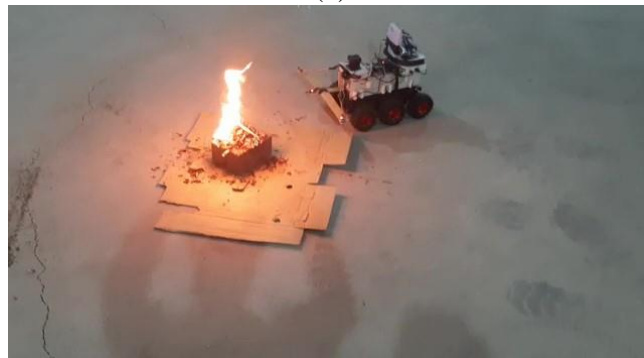
Figure 4.31 Right Side Fire Source Detection Process (Continued)



(c)



(d)



(e)

Figure 4.31 Right Side Fire Source Detection Process (a) Placement of the Fire Source at the Right Side, (b) Right Turning Motion to Detect the Fire Source, (c) Moving Forward Action, (d) Second Right Turning to Find the Fire Source, (e) Detection of the Fire Source in Front (Personal Archive)

In the first step, the fire source was located on right side according to the motion path. When the robotic system detected the fire source, it turned right because of the fire data from the right IR sensor. Then, it moved forward as the fire source was not at the sight of view of front IR. After a short move forward action, the right flame sensor was detected fire source and turned the robotic system one more time. By

this reaction, the fire source was located at the field of view of the front sensor. The algorithm stopped and waited the robotic system to activate the fire detection algorithm.

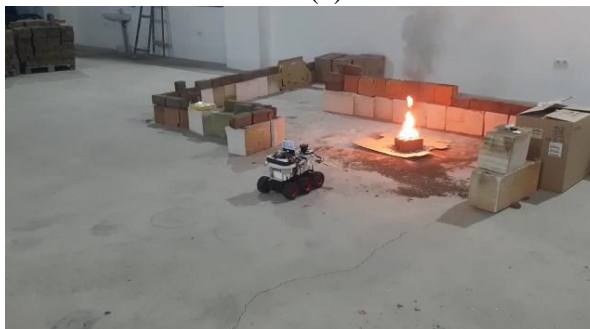
In the last test condition, the fire source was placed on a maze located at the left side of the robotic system motion path. The steps of this case are shown in Figure 4.32.



(a)



(b)



(c)

Figure 4.32 Left Side Fire Source Detection Process (a) Placement of Fire Source at Left Side Maze, (b) Left Turning Motion to Detect the Fire Source, (c) Detection of the Fire Source in the Maze (Personal Archive)

In the first step, the fire source was located on a maze placed on the left side of the robotic system path. When fire source was detected by the left flame sensor, the robotic system turned left with a large angle value to put the fire source in the field of view of front sensor. Then, the front IR sensor detected the fire source and activated the fire detection algorithm.

The performance of the developed fire search and find was evaluated for three different cases. In the first case, the fire source was located in front of the robotic system. As expected, the robotic system stopped its motion and activated the fire detection algorithm when front IR detected the fire source.

In the second test, the fire source was placed on the right side of the robotic system. It was observed that the algorithm turned the robot to the right side until it detected the fire source.

In the last test, the fire source was put in a maze positioned at the left of the robot. It was a hard condition as the front path was free gap and the fire source was at the maze. However, the robotic system detected the fire source by the data of the left IR and turned left until it reached to the fire source. By the means of the results mentioned above it could be concluded that the developed fire search and find algorithm based on the fused data of three IR sensors was satisfactory for fire detection operations.

#### **4.4.3. Fire Detection**

The fire detection algorithm was developed to determine the fire probability of the detected candidates by the fire search and find algorithm. According to the operation process if any object was defined as a fire source by the fire search and find algorithm the fire detection algorithm was activated to decide the probability.

The fire detection algorithm was developed using the Faster R-CNN deep learning model. The performance of the algorithm was evaluated and the obtained results were presented in this section of the thesis study.

The fire detection algorithm was operated in a Raspberry pi 3b. A LCD (Liquid crystal display) touch screen was connected to Raspberry. A wireless network was also established to observe the fire detection data between the Raspberry and remote

controller computer. The obtained fire probability data was saved to the CSV file in the Raspberry to evaluate the performance of the algorithm in mobile application.

The performance tests were applied for three different fire sources as candle, wood and complex. Complex fire was the combination of candle, paper and wood (Figure 4.33).



(a)



(b)

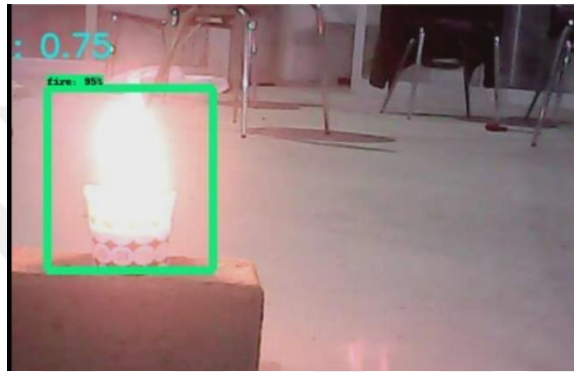


(c)

Figure 4.33 Different Fire Sources Used in Fire Detection Test  
(a) Candle Fire, (b) Wood Fire, (c) Complex Fire (Personal Archive)



The data from the algorithm were saved in a CSV file for 10 minutes for each of the fire source. The time interval was determined as one second. Therefore, totally 600 data were obtained for each fire source detection. The image scenes obtained from the fire detection process are presented in Figure 4.34.



(a)



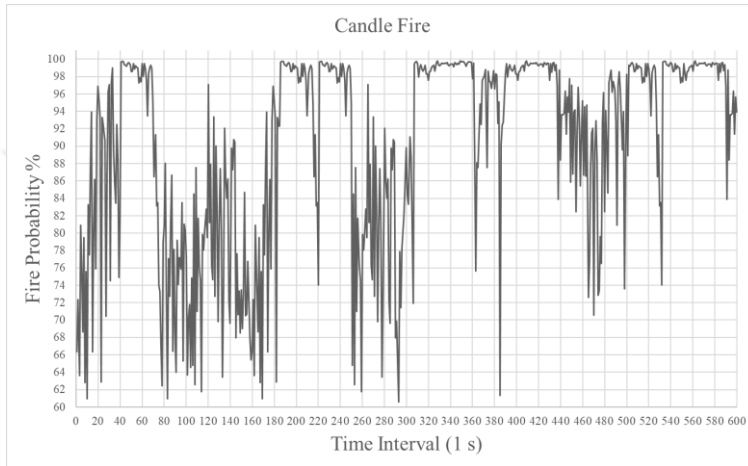
(b)



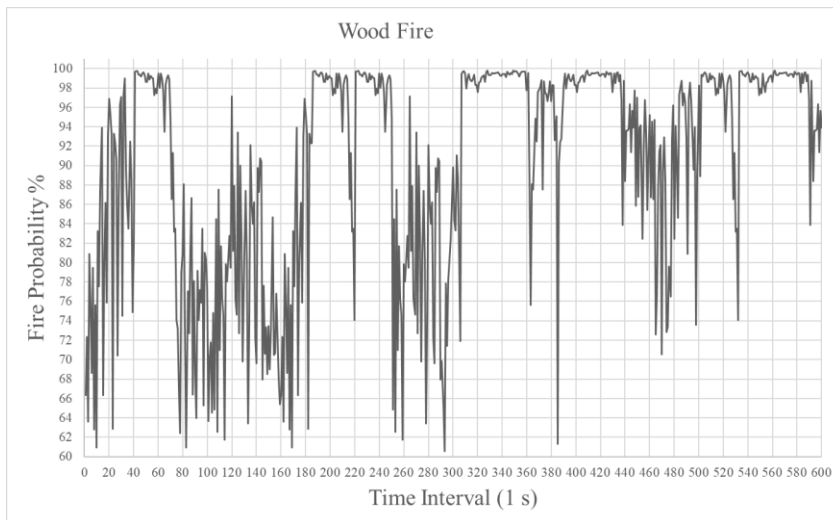
(c)

Figure 4.34 Detection of the Fire Sources with Fire Detection Algorithm  
(a) Detection of Candle, (b) Detection of Wood, (c) Detection of Complex Fire Source (Personal Archive)

As shown in the Figure 4.34 the detection algorithm created a box boundary on the determined fire source. The threshold value of the probability to create the detection box was selected as 60% percent to increase the accuracy.



(a)



(b)

Figure 4.35 Obtained Fire Probabilities in Detection Process (Continued)

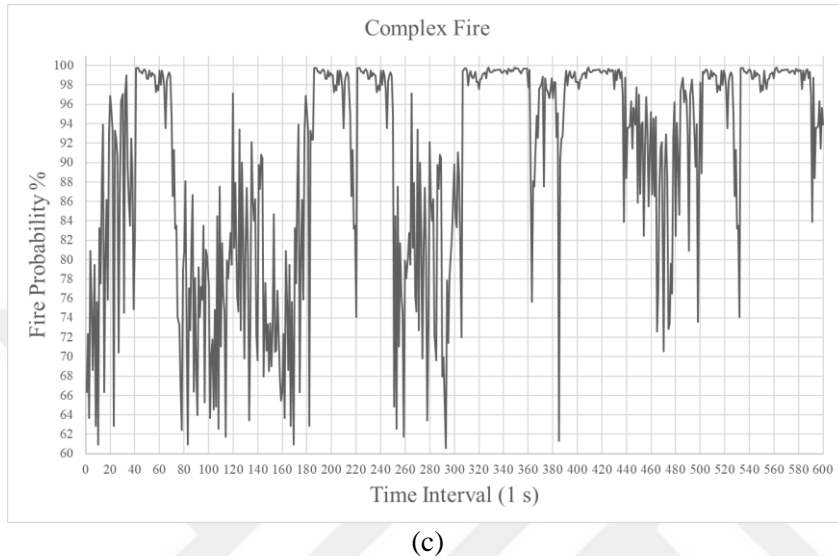


Figure 4.35 Obtained Fire Probabilities in Detection Process  
 (a) Candle Fire (b) Wood Fire, (c) Complex Fire

The determined fire probabilities of the different fire sources are shown in Figure 4.35. The range of the graphs were drawn in the range of 60 to 100 as the minimum detection threshold was selected as 60%. The mean of the obtained detection values for candle, wood and complex fires were calculated as 86, 90 and 93%, respectively. The minimum detection accuracy was seen at the candle fire detection process. It could be explained with the smaller size of the flame and the maximum number of the flickering scenes in the real time detection operation. The accuracy of the detection increased to 90% for wood and 93% for the complex fire detection. As the fire detection was conducted in the visible range; the sizes, color and flickering of the flames were the important criteria for the learning and detection capabilities. As understood from the fire type images (Figure 4.33) the minimum flickering and maximum sizes were occurred on the complex fire. The scene of the complex fire was the more similar to fire images used for learning of the deep learning model.

As the results of the mentioned above;

1. The fire detector developed using fire detection algorithm had capability to detect the fire in the range of 86 to 93% for candle, wood and complex fire sources.

2. The accuracy of the 93% was acceptable result, as the fire detection algorithm was created with compressed form of the Faster R-CNN suitable to run in mobile devices such as Raspberry pi.
3. Different from the studies of the Frizzi et al. (2016) and Zhong et al. (2018) the fire detection system developed using the Faster R-CNN model instead of CNN and the performance was tested in real time applications.

#### 4.5. Tests of Ladder Climbing Robot

The obstacle avoidance, local path planning and ladder climbing capability of the robotic system were tested. In the first stage, the obstacle avoidance and path planning abilities of the robot were tested. The obstacles were placed as front left, front and left corner, respectively (Figure 4.36).



(a)



(b)



(c)

Figure 4.36 Obstacle Avoidance and Local Path Planning Tests (a) Front Left Obstacle, (b) Front Obstacle, (c) Left Corner (Personal Archive)

In the second stage, the ladder climbing test was applied to the robotic system (Figure 4.37).



Figure 4.37 Ladder Climbing Test (Personal Archive)

It was observed from the tests that the ladder climbing robotic system passed the obstacles without crash. The robotic system could switch the motion type sequentially and overcome the obstacles. It also could climb the ladder as expected and desired.

## 5. CONCLUSIONS

In this thesis study a robotic system with the capabilities of ladder climbing and fire detection was developed. The robotic system was designed and fabricated as adaptable for both indoor and outdoor operation environments. An adaptive three wheel legged hybrid locomotion mechanism with a transition system to provide the motion of ladder climbing and ground drive was developed. The mathematical models were derived for the hybrid locomotion system. A path planning and obstacle avoidance approach named as “Direction Based Angle Calculation” was proposed and the required experimental tests were applied to check the performance of the robotic system with developed approach. A motion decision algorithm to classify the obstacles as insurmountable, negotiable or negligible and to activate the climbing or straight motion according to the shape and height of the obstacle was established. Fire search and find and fire detection algorithms and models were also constructed to find the fire candidates and to determine the probabilities.

Several experimental tests were applied to the mechanical systems and algorithms of the ladder climbing fire detection robot. The obtained results from the applied tests were presented in the section of “Results and Discussion”. According to the obtained results, the listed are concluded:

1. The transmission and motion systems are capable for both ladder climbing and straight way motions.
2. The motion decision algorithm can determine the required motion type successfully according to the situation of the obstacle.
3. The direction based angle calculation approach is suitable and satisfactory for local path planning and obstacle avoidance applications.
4. The fire search and find algorithm can detect the fire source placed on the front, right and left sides of the robotic system and activate the fire detection algorithm successfully.
5. The developed fire detection algorithm using the Faster R-CNN deep learning model determines the probability of the detected fire sources with the accuracy of the 93%.

## 6. RECOMMENDATIONS AND FUTURE WORKS

By further developments and improvements, the listed are recommended:

- 1.The detection accuracy can be increased with the usage of thermal camera and thermal images for the training process of the deep learning model and for the fire probability determination of the detection algorithm.
- 2.A global path planning can be added to develop for the usage of the robotic system in prescribed environments.
- 3.Different deep learning model such as reinforcement deep learning can be used for path planning and fire detection algorithms.

As future research directions:

- 1.It is planned to create a hybrid fire detection algorithm using the thermal and visible images in the training and detection processes.
- 2.It is also planned to use the ROS (Robotic Operation System) for the mapping, localization and path planning applications of the robotic system.

## REFERENCES

- A4Tech. 2019. Fire Detection Webcam (A4 Tech, 2019) [Online]. Available: <https://www.a4tech.com/> Access: 14.02.2019
- Alamdari, A., Hérin, R., Krovi, V. N. 2013. Quantitative kinematic performance comparison of reconfigurable leg-wheeled vehicles. Nature-Inspired Mobile Robotics. In Proceedings of the 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, (Waldron, K. J., Tokhi, M. O., Virk, G. S.), pp. 585-592. University of Technology Sydney, Australia.
- Aliff, M., Yusof, M. I., Sani, N. S., Zainal, A. 2019. Development of Fire Fighting Robot (QRob). **International Journal of Advanced Computer Science and Applications**, 10(1): 142-147.
- Alsaab, A. 2015. Behavioural Strategy for Indoor Mobile Robot Navigation in Dynamic Environments. School of Mechanical and Systems Engineering Newcastle University, United Kingdom, Ph.D. Thesis, Newcastle.
- Babinec, A., Duchoň, F., Dekan, M., Pászto, P., Kelemen, M. 2014. VFH\* TDT (VFH\* with Time Dependent Tree): A New Laser Rangefinder Based Obstacle Avoidance Method Designed for Environment with Non-Static Obstacles. **Robotics and Autonomous Systems**, 62(8): 1098-1115.
- Behnke, S. 2008. Humanoid Robots from Fiction to Reality. **KI-Zeitschrift**, 4(8): 5-9.
- Bertram, C., Evans, M. H., Javaid, M., Stafford, T., Prescott, T. 2013. Sensory Augmentation with Distal Touch: The Tactile Helmet Project. Biomimetic and Biohybrid Systems, In Proceedings of Second International Conference, Living Machines, pp. 24-35. London, United Kingdom.
- Borges, P.V.K., Mayer, J., Izquierdo, E. 2008. Efficient Visual Fire Detection Applied for Video Retrieval. In Proceedings of 16th European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland.



- Bruzzone, L., Fanghella, P. 2014. Mantis: Hybrid Leg-Wheel Ground Mobile Robot. **Industrial Robot: An International Journal**, 41(1): 26-36.
- Bruzzone, L., Quaglia, G. 2012. Locomotion systems for ground mobile robots in unstructured environments. **Mechanical Sciences**, 3: 49-62.
- Calderara, S., Piccinini, P., Cucchiara, V. 2008. Smoke Detection in Video Surveillance: A Mog Model in the Wavelet Domain. In Proceedings of 6th International Conference in Computer Vision Systems (ICVS), pp. 119–128, Santorini, Greece.
- Cao, J. 2016. Robot Global Path Planning Based on an Improved Ant Colony Algorithm. **Journal of Computer and Communications**. 4(2): 11-19.
- Ceccarelli, M., Carbone, G. 2005. Legged Robotic Systems. In: Cutting Edge Robotics (Vedran, K., Aleksandar, L., Munir, M.), InTech, pp. 553-576, Germany.
- Celik, T., Demirel, H. 2009. Fire Detection in Video Sequences Using a Generic Color Model. **Fire Safety Journal**, 44(2): 147-158.
- Celik, T., Ma, K. K. 2008. Computer Vision Based Fire Detection in Color Images. In Proceedings of 2008 IEEE Conference on Soft Computing in Industrial Applications, pp. 258-263, Muroran, Japan
- Çetin, A. E., Dimitropoulos, K., Gouverneur, B., Grammalidis, N., Günay, O., Habiboğlu, Y. H., Töreyn, B. U., Verstockt, S. 2013. Video Fire Detection–Review. **Digital Signal Processing**, 23(6): 1827-1843.
- Chang, P.H., Kang, Y.H., Cho, G.R., Kim, J.H., Jin, M., Lee, J. 2006. Control Architecture Design for a Fire Searching Robot using Task Oriented Design Methodology. SICE-ICASE International Joint Conference, IEEE, pp.3126 - 3131, Busan, South Korea.
- Chean S. L., Doria, M., Chan, J., Parker S., Goatley, S., 2002. Path Planning and High Level Control of an Unmanned Aerial Vehicle. University of Sydney, Ph.D. Thesis, Sydney.

- Chen, J., He, Y., Wang, J. 2010. Multi-Feature Fusion Based Fast Video Flame Detection, **Building and Environment**. 45(5): 1113–1122.
- Chen, T.-H., Wu, P.-H., Chiou, Y.-C. 2004. An Early Fire-Detection Method Based on Image Processing. In Proceedings of IEEE International Conference on Image Processing (ICIP), pp. 1707–1710, Singapore.
- Chen, X.Q., Chen, Y.Q., Chase, J.G. 2009. Mobiles Robots – Past Present and Future. In: Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions, (Chen, X. Q., Eds.), pp.1-32, InTech, Croatia.
- Cheng, P., LaValle, S. M. 2001. Reducing Metric Sensitivity in Randomized Trajectory Design. In Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Expanding the Societal Role of Robotics in the Next Millennium (Cat. No.01CH37180), pp. 43-48, Maui, HI, USA.
- Choset, H. M., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., Thrun, S. 2005. Principles of Robot Motion: Theory, Algorithms, and Implementation. Robotics and Control Systems, MIT press, 2005.
- Christensen, C. N., Zainchkovskyy, Y., Barrera-Figueroa, S., Torras-Rosell, A., Marinelli, G., Sommerlund-Thorsen, K. Kleven, J., Kleven, K., Voll, E., Petersen, J. C., Lassen, M. 2019. Simple and Robust Speckle Detection Method for Fire and Heat Detection in Harsh Environments. **Applied Optics**, 58(28): 7760-7765.
- Coste-Manière, E., Simmons, R. 2000. Architecture, the Backbone of Robotic Systems. In Proceedings of the 2000 IEEE International Conference on Robotics & Automation, pp. 67-72, San Francisco, CA.
- Dalvand, M., Moghadam, M. 2006. Stair climber smart mobile robot (MSRox). **Autonomous Robots**, 20(1): 3-14.
- Demir, M., Sezer, V. 2017. Improved Follow the Gap Method for Obstacle Avoidance. In 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1435-1440, Munich, Germany.

- Doretto, G. Chiuso, A., Wu, Y. N., Soatto, S. 2003. Dynamic textures, **International Journal of Computer Vision**, 51(2): 91–109.
- Durán, B., Thill, S. 2012. Rob's Robot: Current and Future Challenges for Humanoid Robots. In: *The Future of Humanoid Robots - Research and Applications* (Zaier, R., Eds.), pp. 280-300, InTech.
- Edlinger, R., Nuechter, A. 2019. MARC-Modular Autonomous Adaptable Robot Concept. In *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1-7, Würzburg, Germany.
- Fiorini, P., Shiller, Z. 1998. Motion Planning in Dynamic Environments Using Velocity Obstacles. **The International Journal of Robotics Research**, 17(7): 760-772.
- FLIR. 2019. FLIR Neuro Technology: Automate Complex Decisions Faster with Deep Learning. [Online]. Available: <https://www.flir.com/discover/iis/machine-vision/flir-neuro-technology-automate-complex-decisions-faster-with-deep-learning/>. Access: 11.21.2019.
- Floreano, D., Godjevac, J., Martinoli, A., Mondada, F., Nicoud, J. D. 1999. Design, Control, and Applications of Autonomous Mobile Robots. In: *Advances in Intelligent Autonomous Systems*, Springer, pp. 159-186, Netherlands.
- Fox, D., W. Burgard and Thrun, S. 1997. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics and Automation Magazine*, 4(1): 23-33.
- Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Moreau, E., Fnaiech, F. 2016. Convolutional Neural Network for Video Fire and Smoke Detection. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 877-882, Florence, Italy.
- Garcia, E., De Santos, P. G. 2006. On the improvement of walking performance in natural environments by a compliant adaptive gait. **IEEE Transactions on Robotics**, 22(6): 1240-1253.

- Garcia, E., Jimenez, M., De Santos, P., Armada, M. 2007. The Evolution of Robotics Research. **IEEE Robotics & Automation Magazine**, 14(1): 90-103.
- Geppert, L. 2014. The Robot That Could. **IEEE Spectrum**, 41(5): 34-37.
- Gonzalez R. A., Gonzalez R. A., Rea, P. 2011. A New Articulated Leg for Mobile Robots. **Industrial Robot**, 38(5): 521-532.
- Hahnel, D., Triebel, R., Burgard, W., Thrun, S. 2003. Map Building with Mobile Robots in Dynamic Environments. In Proceedings of the 2003 IEEE International Conference on Robotics & Automation, pp. 1557 – 1563, Taipei, Taiwan.
- Hamme, D. V., Veelaert, P., Philips, W., Teelen, K. 2010. Fire Detection in Color Images Using Markov Random Fields. In Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) Part II, pp. 88–97. Sydney, Australia.
- Haukur, I., Heimo, T., Anders, L. 2010. Industrial fires An Overview, Brandforsk Project, SP Report 2010:17, Sp Technical Research Institute of Sweden, Boras.
- Hobbelen, D.G.E., Wisse, M. 2007. Limit Cycle Walking. In: Humanoid Robots, Human-like Machines (Matthias, H.), pp.278-294, I-Tech Education and Publishing, Vienna, Austria.
- HOKUYO. 2019. Distance Data Output/UTM-30LX-EW. [Online]. Available: <https://www.hokuyo-aut.jp/search/single.php?serial=170>. Access: 10.07.2019.
- Hong, J. H., Min, B. C., Taylor, J. M., Raskin, V., Matson, E. T. 2012. NL-Based Communication with Firefighting Robots. In IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1461-1466, Seoul, Korea.
- Jasika, N., Alispahic, N., Elma, A., Ilvana, K., Elma, L., Nosovic, N. 2012. Dijkstra's Shortest Path Algorithm Serial and Parallel Execution Performance Analysis. In Proceedings of the 35th International Convention MIPRO, pp. 1811-1815, Opatija, Croatia.

- Jia, Y. Z., Li, J. S., Guo, N., Jia, Q. S., Du, B. F., Chen, C. Y. 2018. Design and Research of Small Crawler Fire Fighting Robot. In 2018 Chinese Automation Congress (CAC), pp. 4120-4123, Xi'an, China.
- Kamil, F., Tang, S., Khaksar, W., Zulkifli, N., Ahmad, S. A. 2015. A Review on Motion Planning and Obstacle Avoidance Approaches in Dynamic Environments. **Advances in Robotics & Automation**, 4(2): 134-142.
- Kamon, I., Rivlin, E., Rimon, E. 1996. A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots. In Proceedings of IEEE International Conference on Robotics and Automation, pp. 429-435, Minneapolis, USA.
- Kazuya, Y. 2015. Using the Hokuyo Sensor with ROS on the Raspberry Pi. Field Robotics Group, Tohoku University. Available: [http://www.ryugakujapan.com/uploads/2/9/8/0/29803303/hokuyo\\_sensor\\_with\\_ros\\_tutorial.pptx](http://www.ryugakujapan.com/uploads/2/9/8/0/29803303/hokuyo_sensor_with_ros_tutorial.pptx). Access: 23.07.2019.
- Khatib, O., 1985. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In Proceedings of 1985 IEEE International Conference on Robotics and Automation, pp. 500–505, St. Louis, MO, USA.
- Khoon, T.N., Sebastian, P., Saman, A.B.S. 2012. Autonomous Fire Fighting Mobile Platform. **Procedia Engineering**, 41:1145-1153.
- Kim, J. H., Starr, J. W., Lattimer, B. Y. 2015. Firefighting Robot Stereo Infrared Vision and Radar Sensor Fusion for Imaging through Smoke. **Fire Technology**, 51(4): 823-845.
- Kim, J. H. 2014. Autonomous Navigation, Perception and Probabilistic Fire Location for an Intelligent Firefighting Robot. Dissertation Submitted to the Faculty of the Virginia Polytechnic Institute and State University, Ph.D. Thesis in Mechanical Engineering, Blacksburg, USA.
- Kim, Y. D., Kim, Y. G., Lee, S-H., Kang, J-H., An, J. 2009. Portable Fire Evacuation Guide Robot System. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2789 - 2794, St. Louis, USA.

- Kluge, B. 2004. Recursive Agent Modeling with Probabilistic Velocity Obstacles for Mobile Robot Navigation among Humans. In Proceedings of the 2003 IEEWRSJ International Conference on Intelligent Robots and Systems, pp. 371-376, Las Vegas, Nevada, USA.
- Kluge, B., Prassler, E. 2007. Recursive Agent Modeling with Probabilistic Velocity Obstacles for Mobile Robot Navigation among Humans. In Autonomous Navigation in Dynamic Environments (Laugier, C., Chatila, R. Eds.), pp. 121-134, Springer, Berlin, Heidelberg.
- Ko, B.C., Cheong, K.H., Nam, J.Y. 2010. Early Fire Detection Algorithm Based on Irregular Patterns of Flames and Hierarchical Bayesian Networks. **Fire Safety Journal**, 45(4): 262–270.
- Koditschek, D. E., Full, R. J., Buehler, M. 2004. Mechanical Aspects of Legged Locomotion Control. **Arthropod Structure & Development**, 33(3): 251-272.
- Kong, S. G., Jin, D., Li, S., Kim, H. 2016. Fast Fire Flame Detection in Surveillance Video Using Logistic Regression and Temporal Smoothing. **Fire Safety Journal**, 79: 37-43.
- Koren, Y., Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings of 1991 IEEE International Conference on Robotics and Automation, pp. 1398-1404, Sacramento, CA, USA.
- Kunchev, V., Jain, L., Ivancevic, V., Finn, A. 2006. Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 537-544), Springer, Berlin, Heidelberg.
- Lakshmi, S. V. R., Babu, C. A. 2018. Blaze Hostile Portable Machine: Formal of the Sculpture and Modern Improvement. **International Journal of Research**, 5(15): 2598-2603.

- Laumond, J. P. 1993. Singularities and Topological Aspects in Nonholonomic Motion Planning. In *Nonholonomic Motion Planning* (Li, Z., Canny, J. F. Eds.), pp. 149-199, Springer, Boston, MA.
- Laumond, J. P., Risler, J. J. 1996. Nonholonomic Systems: Controllability and Complexity. **Theoretical Computer Science**, 157(1): 101-114.
- Levendis, Y. A., Delichatsios, M. A. 2011. Pool Fire Extinction by Remotely Controlled Application of Liquid Nitrogen. **Process Safety Progress**, 30(2): 164-167.
- Li, X., Choi, B. J. 2013. Design of Obstacle Avoidance System for Mobile Robot Using Fuzzy Logic Systems. **International Journal of Smart Home**. 7(3): 321-328.
- Liljebäck, P., Stavadahl, O., Beitnes, A. 2006. Snakefighter-Development of a Water Hydraulic Fire Fighting Snake Robot. In *9th International Conference on Control, Automation, Robotics and Vision*, pp. 1-6, Singapore.
- Lim, H., Takanishi, A. 2006. Mechanism and Control of Anthropomorphic Biped Robots. In: *Mobile Robots, Moving Intelligence* (Bucli, J., Eds.), InTech, pp. 308-324, Vienna, Austria.
- Lumelsky, V. J., Skewis, T., 1990. Incorporating Range Sensing In the Robot Navigation Function. **IEEE Transactions on Systems, Man, and Cybernetics**. 20(5): 1058–1069.
- Lumelsky, V., Stepanov, A. 1987. Path Planning Strategies for a Point Mobile Automation Moving Among Unknown Obstacles of Arbitrary Shape. **Algorithmica**, 2:403-430.
- Luo, R.C., Su, K.L. 2007. Autonomous Fire-Detection System Using Adaptive Sensory Fusion for Intelligent Security Robot. **IEEE/ASME Transactions on Mechatronics**, 12(3): 274-281.

- Madani, R. S., Baines, E., Moroz, A., Makled, B. 2015. Evaluation of Suitability of Rapid Prototyping Techniques for Use by Children Evaluation. **Journal of Multidisciplinary Engineering Science and Technology (JMEST)**, 2(1): 261-266.
- MAK-PAR-SAN. 2017a. 608 2RS C3 - Koyo Rulman. [Online]. Available: <http://www.makparsan.com/urunDetay.aspx?id=59347>. Access: 14.08.2017.
- MAK-PAR-SAN. 2017b. 51114 - Koyo Rulman. [Online]. Available: <http://www.makparsan.com/urunDetay.aspx?id=59422>. Access: 14.08.2017.
- Manchester, I.R., Mettin, U., Iida, F., Tedrake, R., 2011. Stable Dynamic Walking Over Uneven Terrain. **International Journal of Robotics Research**, 30(3): 265-279.
- Martinson, E., Lawson, W., Blisard, S., Harrison, A., Trafton, G. 2012. Fighting Fires with Human Robot Teams. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2682 - 2683, Vilamoura, Portugal.
- MEGEP (Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi). 2006. Merdiven Planı ve Donatı Çizimi. Millî Eğitim Bakanlığı, Ankara.
- Minguez, J., Montano, L. 2004. Nearness Diagram Navigation (ND): Collision Avoidance in Troublesome Scenarios. **IEEE Transactions on Robotics and Automation**, 20(1): 45-59.
- Mishra, K. B., Wehrstedt, K.D., Krebs, H. 2013. Lessons Learned From Recent Fuel Storage Fires. **Fuel Processing Technology**, 107: 166-172.
- Miui. 2019. Xiaomi Power Bank [Online]. Available: <https://market.miuiturkiye.net/powerbank>. Access: 14.02.2019
- Miyazawa, K. 2002. Fire robots developed by the Tokyo Fire Department. **Advanced Robotics**, 16(6): 553-556.



- Molinos, E., Llamazares, A., Ocaña, M., Herranz, F. 2014. Dynamic obstacle avoidance based on curvature arcs. In *Proceeding of 2014 IEEE/SICE International Symposium on System Integration*, (pp. 186-191), Tokyo, Japan.
- Narukawa, T., Yokoyama, K., Takahashi, M., Yoshida, K. 2010. An Experimental Study of Three Dimensional Passive Dynamic Walking With Flat Feet and Ankle Springs. In: *Cutting Edge Robotics* (Vedran, K., Eds.), InTech, pp.132-144, Vienna, Austria.
- Nguyen, H. T., Le, H. X. 2016. Path Planning and Obstacle Avoidance Approaches for Mobile Robot. **IJCSI International Journal of Computer Science Issues**, 13(4): 1-10.
- Nikitin, V. S., Golubin, S. A., Belov, R. B., Andrianov, N. V., Shurtakov, V. V. 2019. Multiagent Robotic System Application in Conditions of Forest Firefighting. **International Journal of Civil Engineering and Technology**, 10(2): 144-151.
- Nonami, K., Barai, R. K., Irawan, A., Daud, M. R. 2014. Historical and Modern Perspective of Walking Robots. In *Hydraulically Actuated Hexapod Robots* (Nonami, K., Barai, R. K., Irawan, A., Daud, M. R., Eds.), pp. 19-40, Springer, Tokyo, Japan.
- Ohki, T., Nagatani, K., Yoshida, K. 2012. Local Path Planner for Mobile Robot in Dynamic Environment Based on Distance Time Transform Method. **Advanced Robotics**. 26(14): 1623-1647.
- Oroko, J., Nyakoe, G. N. 2014. Obstacle Avoidance and Path Planning Schemes for Autonomous Navigation of a Mobile Robot: A Review. In *Proceedings of the 2012 Mechanical Engineering Conference on Sustainable Research and Innovation*, pp. 314-318, Juja, Kenya.
- Patnaik, S. 2007. *Cybernetic View of Robot Cognition and Perception*. In: *Robot Cognition and Navigation*, Springer, pp 10-20, New York, USA.

- Perumal, K. A., Ali, M. A., Yahya, Z. H. 2019. Fire Fighter Robot with Night Vision Camera. In IEEE 15th International Colloquium on Signal Processing & Its Applications (CSPA), pp. 270-274, Penang, Malaysia.
- Phillips, W., Shah, M., da Vitoria Lobo, N. 2002. Flame Recognition in Video, **Pattern Recognition Letters**. 23(1-3): 319-327.
- Pololu Robotics & Electronics. 2018a. Dagu Wild Thumper Wheel 120x60mm Pair with 4mm Shaft Adapters - Metallic Red [Online]. Available: <https://www.pololu.com/product/1555>. Access: 20.06.2018.
- Pololu Robotics & Electronics. 2018b. Dagu Wild Thumper Wheel 120x60mm Pair with 4mm Shaft Adapters - Chrome [Online]. Available: <https://www.pololu.com/product/1557/>. Access: 20.06.2018.
- Poobalan, K., Liew, S. 2015. Fire Detection Algorithm Using Image Processing Techniques. In Proceedings of the 3rd International Conference on Artificial Intelligence and Computer Science (AICS2015), pp. 160-168, Penang, Malaysia.
- Qi, X., Ebert, J. 2009. A Computer Vision Based Method for Fire Detection in Color Videos. **International Journal of Imaging**, 2(9): 22–34.
- Raibert, M.H. 1986. Legged Robots That Balance, MIT Press, Cambridge, MA, USA.
- Rakib, T., Sarkar, M. R. 2016. Design and Fabrication of an Autonomous Fire Fighting Robot with Multisensor Fire Detection Using PID Controller. In the Proceedings of the 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp. 909-914, Dhaka, Bangladesh.
- Robotistan 2018 e. Lipo Battery for Lidar. [Online]. Available: <https://www.robotistan.com/111v-lipo-batarya-1350mah-25c>  
Access: 11.07.2018
- Robotistan. 2018 a. 12 V 37 mm 500 RPM Enkoderli 19:1 Redüktörlü DC Motor. [Online]. Available: <https://www.robotistan.com/12v-37mm-500-rpm-enkoderli-191-reduktorlu-dc-motor>. Access: 11.07.2018.

- Robotistan. 2018 b. 14.8 V 4S Lipo Batarya 3000 mAh 25C [Online]. Available: <https://www.robotistan.com/148v-lipo-batarya-3000mah-25c>  
Access: 11.07.2018.
- Robotistan. 2018 d. IR Flame Sensor [Online]. Available: <https://www.robotistan.com/ates-algilayici-sensor-karti-flame-sensor>  
Access: 11.07.2018.
- Robotistan. 2018 f. Ultrasonic Distance Sensor [Online]. Available: <https://www.robotistan.com/hc-sr04-ultrasonik-mesafe-sensoru> Access: 11.07.2018
- Robotistan. 2018 g. Arduino Mega 2560 [Online]. Available: <https://www.robotistan.com/orjinal-arduino-mega-2560-r3-yeni-versiyon-1sensoru> Access: 11.07.2018
- Robotistan. 2018 h. 12 V 42 mm 200 RPM Redüktörlü DC Motor. [Online]. Available: <https://www.robotistan.com/12v-42mm-200rpm-reduktorlu-dc-motor>. Access: 11.07.2018.
- Robotistan. 2018c. Raspberry pi 3b [Online]. Available: <https://www.robotistan.com/raspberry-pi-3> Access: 11.07.2018.
- Ronchi, E., Gwynne, S., Purser, D., Colonna, P. 2013. Representation of the impact of smoke on agent walking speeds in evacuation models. **Fire Technology**, 49(2): 411-431.
- Sahin Rulman. 2017. Lineer Rulman LME 25 UU. [Online]. Available: <https://www.sahinrulman.com/lineer-rulman-lme-25-uu>.  
Access: 14.08.2017.
- Seeni, A., Schäfer, B., Hirzinger, G. 2010. Robot Mobility Systems for Planetary Surface Exploration—State-Of-The-Art and Future Outlook: A Literature Survey. **Aerospace Technologies Advancements**, 189-208.
- Sekercioglu, T. 2015. Makine Elemanları: Hesap Şekillendirme. Birsen Kitabevi.

- Šelek, A., Jurić, D., Čirjak, A., Marić, F., Seder, M., Marković, I., Petrović, I. 2019. Control Architecture of a Remotely Controlled Vehicle in Extreme CBRNE Conditions. In Proceedings of International Conference on Electrical Drives & Power Electronics (EDPE), pp. 273-278, The High Tatras, Slovakia.
- Shinde, P. A., Mane, Y. B. 2015. Advanced vehicle monitoring and tracking system based on Raspberry Pi. In IEEE 9th International Conference on Intelligent Systems and Control (ISCO), pp. 1-6, Coimbatore, India.
- Siegwart, R., Lamon, P., Estier, T., Lauria, M., & Piguët, R. 2002. Innovative Design for Wheeled Locomotion in Rough Terrain. **Robotics and Autonomous Systems**, 40(2-3): 151-162.
- Siegwart, R., Lauria, M., Maesli, P. Van Winnendael, M. 1998. Design And Implementation Of An Innovative Micro-Rover, In Proceedings of 3rd ASCE Speciality Conference on Robotics for Challenging Environments (Robotics 98), Albuquerque, New Mexico, USA.
- Silva, M. F., Machado, J. T. 2012. A Literature Review on the Optimization of Legged Robots. **Journal of Vibration and Control**, 18(12): 1753-1767.
- Simmons, R. 1996. The Curvature-Velocity Method for Local Obstacle Avoidance. In Proceedings of IEEE International Conference on Robotics and Automation, pp. 3375-3382, Minneapolis, USA.
- Subramanian, S., George, T, Thondiyath, A. 2012. Obstacle Avoidance Using Multi-Point Potential Field Approach for an Underactuated Flat- Fish Type AUV in Dynamic Environment. In Proceedings of International Conference on Intelligent Robotics, Automation, and Manufacturing, IRAM 2012: Trends in Intelligent Robotics, Automation, and Manufacturing, pp. 20-27, Kuala Lumpur, Malaysia.
- Sucuoglu, H. S. 2015. The development of fire detection robot. Aydin Adnan Menderes University, Master Thesis on Institute of Science. Aydın.
- Sucuoglu, H. S., Bogrekci, I., Demircioglu, P. 2018. Development of Mobile Robot with Sensor Fusion Fire Detection Unit. **IFAC-Papers on Line**, 51(30): 430-435.

- Sucuoglu, H. S., Bogrekci, I., Demircioglu, P. 2019. Real Time Fire Detection Using FASTER R-CNN Model. *International Journal of 3D Printing Technologies and Digital Industry*. 3(3).
- Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Gultekin, A. 2018. The Effect of Three Dimensional Printed Infill Pattern on Structural Strength. ***El-Cezeri Journal of Science and Engineering***. 5(3): 785-796.
- Tadakuma, K., Tadakuma, R., Maruyama, A., Rohmer, E., Nagatani, K., Yoshida, K. Ming, A., Shimojo, M., Higashimori, M., Kaneko, M. 2010. Mechanical Design of the Wheel-Leg Hybrid Mobile Robot to Realize a Large Wheel Diameter. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 3358-3365, Taipei, Taiwan.
- Tan, C.F., Liew, S.M., Alkahari, M.R., Ranjit, S.S.S., Said, M.R., Chen W., Rauterberg, G.W.M., Sivarao D.S. 2013. Fire Fighting Mobile Robot: State of the Art and Recent Development. ***Australian Journal of Basic and Applied Sciences***, 10: 220-230.
- Toreyin, B. U., Cetin, A. E. 2008. Computer Vision Based Fire Detection Software. In *IEEE 16<sup>th</sup> Signal Processing, Communication and Applications Conference*. Aydin, Turkey.
- Toreyin, B. U., Dedeoglu, Y., Cetin, A. E., Fazekas, S., Chetverikov, D., Amiaz, T., Kiryati, N. 2007. Dynamic Texture Detection, Segmentation and Analysis. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007*, pp. 131-134, Amsterdam, Netherlands.
- Valgren, C. 2007. Incremental Spectral Clustering and Its Application to Topological Mapping. In *Proceedings of IEEE International Conference on Robotics and Automation, IEEE*, pp. 4283-4288, Roma, Italy.
- Verstockt, S., Vanoosthuysse, A., Van Hoecke, S., Lambert, P., Van de Walle, R. 2010. Multi-Sensor Fire Detection by Fusing Visual and Non-Visual Flame Features. In *Proceedings of the 4th International Conference on Image and Signal Processing (ICISP)*, pp. 333 –341, Québec, Canada.

- Vidoni, R., Bietresato, M., Gasparetto, A., Mazzetto, F. 2015. Evaluation and Stability Comparison of Different Vehicle Configurations for Robotic Agricultural Operations on Side-Slopes. **Biosystems Engineering**, 129: 197-211.
- Vignesh, R., Venkatesh, D., Bhaskar, K. 2012. Design of a Small Mobile Robot Using an Efficient Heuristic Approach for Reduced Travel Time Avoiding Obstacles. In 2012 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-8, Coimbatore, India.
- Vukobratovic, M., Borovac B. 2004. Zero Moment Point Thirty-Five Years of Its Life. **International Journal of Humanoid Robotics**, 1: 157-173.
- Wagoner, A., Jagadish, A., Matson, E. T., EunSeop, L., Nah, Y., Tae, K. K., Lee, D. H., Joeng, J. E. 2015. Humanoid Robots Rescuing Humans and Extinguishing Fires for Cooperative Fire Security System Using HARMS. In 2015 6th International Conference on Automation, Robotics and Applications (ICARA), pp. 411-415, Queenstown, New Zealand.
- Westervelt, E. R., Grizzle, J. W., Chevallereau, C., Choi, J. H., Morris, B. 2018. Feedback Control of Dynamic Bipedal Robot Locomotion. CRC press.
- Wright, M., Cook, G., Webber, G. 2001. The Effects of Smoke on People's Walking Speeds Using Overhead Lighting and Way guidance Provision. In Proceedings of the 2nd International Symposium on Human Behaviour in Fire. pp. 275-284, MIT, Boston.
- Xiong, Z, Caballero, R., Wang, H., Finn, A.M., Lelic, M.A., Peng, P.-Y. 2007. Video-Based Smoke Detection: Possibilities, Techniques, and Challenges. In Proceedings of Suppression and Detection Research and Applications (SUPDET) – A Technical Working Conference, Orlando, FL.
- Yang, X., Voyles, R. M., Li, K., Povilus, S. 2009. Experimental Comparison of Robotics Locomotion with Passive Tether and Active Tether. In Proceedings of IEEE International Workshop on Safety, Security & Rescue Robotics (SSRR 2009), pp. 1-6, Denver, CO, USA.

- Yuan, F. 2008. A Fast Accumulative Motion Orientation Model Based on Integral Image for Video Smoke Detection. **Pattern Recognition Letters**, 29(7): 925–932.
- Yuan, F. 2011. Video-Based Smoke Detection with Histogram Sequence of LBP and LBPV Pyramids. **Fire Safety Journal**, 46 (3): 132–139.
- Zeng, W., Church, R. L. 2009. Finding Shortest Paths on Real Road Networks: The Case for A. **International Journal of Geographical Information Science**, 23(4): 531-543.
- Zhong, X., Peng, X., Zhou, J. 2011. Dynamic Collision Avoidance of Mobile Robot Based on Velocity Obstacles. In Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE). pp. 2410-2413, Changchun, China.
- Zhong, Z., Wang, M., Shi, Y., Gao, W. 2018. A Convolutional Neural Network-Based Flame Detection Method in Video Sequence. **Signal, Image and Video Processing**, 12(8), 1619-1627.
- Zidane, I. M., Ibrahim, K. 2017. Wavefront and A-Star Algorithms for Mobile Robot Path Planning. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, pp. 69-80, Springer, Cham.
- Zielinska, T. 2004. Development of a Walking Machine: Mechanical Design and Control Problems. **Mechatronics**, 12(5): 737-754.

## APPENDICES

### Appendix 1 (Matlab Scripts)

#### Motion Calculations

```

%Acceleration and Force Calculations%

%Check for rolling

%F=m*a;

%m*a=m*g*sin(thetal*pi/180)-Ffr;

%T=I*alpha

%alpha=a/r;

%Ffr*r=1/2*m*r^2*(a/r);

%Ffr=1/2*m*a;

%m*a=m*g*sin(thetal*pi/180)-1/2*m*a;

%a=2*Ffr/m;

%a=2*g*cos(thetal*pi/180)*M;

%Fmax-Ffr=m*a;

%Fmax-1/2*(m*a)=m*a;

%Fmax=3/2*(m*a);

%Fs=m*g*sin(thetal*pi/180);

%If Fmax>Fs; rolling is ok there is no slippage;

m=10.5; % kg, weight of total system

cfr=0.75; % cfr, coefficient of static friction

```



```

g=9.81; % m/s^2, standard earth gravity

thetar=30; % degree, slope angle of ladder

r= 0.21; % m, radius of total system

amax=2*g*cosd(thetar)*cfr; % m/s^2, translational acceleration

Ffr=1/2*(m*amax); % N, friction force

Fs=m*g*sind(thetar); % N, slippage force

Fmax=3/2*(m*amax); % N,

%Rolling Motion

%Acceleration and Torque

%T=I*alpha

%alpha=ar/r;

%Ffr*r=1/2*m*r^2*(ar/r);

%Ffr=1/2*m*ar;

m*g*cos(thetal*pi/180)*M=1/2*m*ar

%M=ar/2*g*cos(thetal*pi/180);

%F=m*ar;

m*ar=m*g*sin(thetal*pi/180)-m*g*cos(thetal*pi/180)*M;

%ar=2/3*g*sin(thetal*pi/180);

ar=2/3*g*sind(thetar);

%Torque Calculations for rolling motion

%Total Moment I*alpha+m*ar*r

```

148

%Meq=Total moment=T

%alpha=ar/r

%I=m\*r^2

%I for cylinder 1/2\*m\*r^2

%M=(1/2\*m\*r^2)\*(a/r)+(m\*ar\*r)

%T=M=3/2\*(m\*r\*ar)

alpha=ar/r; % rad/s^2, angular acceleration

I=1/2\*m\*r^2; % kg\*m^2, moment of inertia

Tr=3/2\*(m\*r\*ar); % Nm, Total torque

Thr=Tr/2; % torque for one half

% Angular speed, power(rolling)

v=1; % m/s, linear velocity of the system

ws=v/r; % rad/s, angular velocity of system

rpms=(ws\*60)/(2\*pi); % rpm,

Pr=Tr\*ws; % Watt, required power for system

Phr=Pr/2; % Watt, power for one half

%Linear motion forces

rw=0.06; % m, radius of wheel;

al=0; % m/s^2, translational acceleration;

crol=0.02; % rolling coefficient;

thetap=20 ; % degree, slope angle of path;

```

Ftr=m*g*cosd(thetap); % traction force;

Fg=m*g*sind(thetap); % N, gradient resistance;

Frol=m*g*crol; % N, rolling resistance;

Fa=m*al; % N, acceleration resistance;

Fdrv=Fg+Frol+Fa; % N, sum of the forces;

% Angular speed, power (linear)

v=1; % m/s, linear velocity of the system

w=v/rw; % rad/s, angular velocity of outer gear

ro=0.03; % m, radius of outer gear

rm=0.06; % m, radius of main gear

e= 0.8; %

wo=w; % rad/s, angular velocity of outer gear

wm=wo*ro/rm; % rad/s, angular velocity of main gear

rpm=(wm*60)/(2*pi); % rpm,

Tl=m*g*(sind(thetap)+(cfr))*rw; %Nm total torque

Thl=Tl/2; %Nm torque for one half

Pl=Fdrv*v/e; % Watt, required power for system

Phl=Pl/2; % Watt, power for one half

%Power supply

Ab=3; % Ah, ampere hour

Vb=14.8; % V, voltage

```

150

Pb=Ab\*Vb; % Wh, Energy

fprintf('Rolling motion\n');

fprintf('Check for rolling\n');

fprintf('amax = %4.2f m/s^2\n',amax);

fprintf('Forces\n');

fprintf('Ffr = %4.2f N\n',Ffr);

fprintf('Fs = %4.2f N\n',Fs);

fprintf('Fmax = %4.2f N\n',Fmax);

if Fmax > Fs

fprintf('No slippage\n');

else

fprintf('slippage\n');

end

fprintf('ar = %4.2f m/s^2\n',ar);

fprintf('Rolling torque\n');

fprintf('I = %4.2f kgm^2\n',I);

fprintf('Tr = %4.2f Nm\n',Tr);

fprintf('Thr = %4.2f Nm\n',Thr);

fprintf('Angular speed and power\n');

fprintf('ws = %4.2f rad/s\n',ws);

fprintf('rpms = %4.2frpm\n',rpms);

```
fprintf('Pr = %4.2f W\n',Pr);  
  
fprintf('Phr = %4.2f W\n',Phr);  
  
fprintf('Linear motion\n');  
  
fprintf('Forces\n');  
  
fprintf('Ftr = %4.2f N\n',Ftr);  
  
fprintf('Fg = %4.2f N\n',Fg);  
  
fprintf('Frol = %4.2f N\n',Frol);  
  
fprintf('Fa = %4.2f N\n',Fa);  
  
fprintf('Fdrv = %4.2f N\n',Fdrv);  
  
if Ftr > Fdrv  
  
    fprintf('No slippage\n');  
  
else  
  
    fprintf('slippage\n');  
  
end  
  
fprintf('Angular speed and power\n');  
  
fprintf('w = %4.2f rad/s\n',w);  
  
fprintf('wo = %4.2f rad/s\n',wo);  
  
fprintf('wm = %4.2f rad/s\n',wm);  
  
fprintf('rpm = %4.2frpm\n',rpm);  
  
fprintf('Tl = %4.2f Nm\n',Tl);  
  
fprintf('Thl = %4.2f Nm\n',Thl);
```

152

```
fprintf('Pl = %4.2f W\n',Pl);
```

```
fprintf('Phl = %4.2f W\n',Phl);
```

```
fprintf('Power supply\n');
```

```
fprintf('Pb = %4.2f Wh\n',Pb);
```

### **Mechanical Strength Check of the Components**

```
clc
```

```
%Main shaft minimum diameter%
```

```
%the affected forces on the gear Ftm, Fnm, Frm
```

```
%alpha is the clutch angle 20 degree
```

```
%dm is diameter of main gear
```

```
%Mr is rotation moment
```

```
Mr=6; %Nm
```

```
alpha=20; %degree
```

```
dm=0.120; %m
```

```
Ftm=(2*Mr/dm);
```

```
Fnm=Ftm/cosd(alpha);
```

```
Frm=Ftm*tand(alpha);
```

```
%Forces act on the xy plane and bearings
```

```
%AB,AC and BC are the distances on the shaft
```

```
% sum moment at A=0
```

```
% Fnm*dm/2+Frm*AC-Fby*AB=0
```

AB=50; %mm

AC=35; %mm

BC=15; %mm

FBy=(Fnm\*dm\*1000/2+Frm\*AC)/AB; %N

%FAy+FBy=Frm

if Frm > FBy

FAy=(Frm-FBy);

else

FAy=(Frm-FBy)\*-1; %because of the direction of the forces

end

%Bending Moment at C point for xy

MBc1=(FAy\*AC)/1000;

if (FAy\*AC/1000)>(Fnm\*dm/2)

MBc2=[(FAy\*AC/1000)-(Fnm\*dm/2)];

else

MBc2=[(FAy\*AC/1000)-(Fnm\*dm/2)]\*-1;

end

%Forces act on the xz plane and bearings

% sum moment at A=0

%Ftm\*AC-FBz\*50=0

FBz=Ftm\*AC/50;

154

FAz=Ftm-FBz;

%Bending Moment at C point for xz

MBc3=(FAz\*AC)/1000;

%Resultant forces at A and B bearings

FAr=(FAy^2+FAz^2)^(1/2);

FBr=(FBy^2+FBz^2)^(1/2);

%Resultant bending moment

MBr=(MBc1^2+MBc3^2)^(1/2); %Nm

%Shaft diameter

qTDE=180; %N/mm^2 for S235 steel, Fully variable strength of material

Re= 235; %N/mm^2 yield strength

Kb=0.95; %shape coefficient

Ky=0.9; %surface coefficient

Kc=1.2; %notch effect

S=3; %safety factor

t1=6; %depth of key or pin

qTDE=qTDE\*Kb\*Ky/Kc; %under dynamic loads

d=[32\*S/pi\*((MBr\*1000/qTDE)^2+(3/4\*Mr\*1000/Re)^2)^(1/2)]^(1/3);

d=d+t1;

fprintf('Main Shaft\n');

fprintf('Ftm = %4.2f N\n',Ftm);



```

fprintf('Fnm = %4.2f N\n',Fnm);

fprintf('Frm = %4.2f N\n',Frm);

fprintf('FBy = %4.2f N\n',FBy);

fprintf('FAy = %4.2f N\n',FAy);

fprintf('MBc1 = %4.2f Nm\n',MBc1);

fprintf('MBc2 = %4.2f Nm\n',MBc2);

fprintf('FBz = %4.2f N\n',FBz);

fprintf('FAz = %4.2f N\n',FAz);

fprintf('MBc3 = %4.2f Nm\n',MBc3);

fprintf('FAr = %4.2f N\n',FAr);

fprintf('FBr = %4.2f N\n',FBr);

fprintf('MBr = %4.2f Nm\n',MBr);

fprintf('d = %3.2f mm\n',d);

%outer shaft minium diameter%

do=0.060; %m

Mr=Mr*(do/dm); %Nm

Fto=(2*Mr/do);

Fno=Fto/cosd(alpha);

Fro=Fto*tand(alpha);

%Forces act on the xy plane and bearings

%AB,AC and BC are the distances on the shaft

```

156

% sum moment at A=0

%  $F_{ni} \cdot d_i/2 + F_{ri} \cdot AC - F_{by} \cdot AB = 0$

AB=50; % mm

AC=35; % mm

BC=15; % mm

FBy=(Fno\*do\*1000/2+Fro\*AC)/AB; % N

%FAy+FBy=Frm

if Fro > FBy

FAy=(Fro-FBy);

else

FAy=(Fro-FBy)\*-1; % because of the direction of the forces

end

% Bending Moment at C point for xy

MBc1=(FAy\*AC)/1000; % Nm

if (FAy\*AC/1000)>(Fno\*do/2)

MBc2=[(FAy\*AC/1000)-(Fno\*do/2)];

else

MBc2=[(FAy\*AC/1000)-(Fno\*do/2)]\*-1;

end

% Forces act on the xz plane and bearings

% sum moment at A=0

```
%Ftm*AC-FBz*50=0
```

```
% Bending Moment at C point for xy
```

```
FBz=Fto*AC/50;
```

```
FAz=Fto-FBz;
```

```
% Bending Moment at C point for xz
```

```
MBc3=(FAz*AC)/1000;
```

```
% Resultant forces at A and B bearings
```

```
FAr=(FAy^2+FAz^2)^(1/2);
```

```
FBr=(FBy^2+FBz^2)^(1/2);
```

```
% Resultant bending moment
```

```
MBr=(Mbc1^2+Mbc3^2)^(1/2); %Nm
```

```
d=[32*S/pi*((MBr*1000/qTDE)^2+(3/4*Mr*1000/Re)^2)^(1/2)]^(1/3);
```

```
t2=2;
```

```
d=d+t2;
```

```
fprintf('Outer Shaft\n');
```

```
fprintf('Fto = %4.2f N\n',Fto);
```

```
fprintf('Fno = %4.2f N\n',Fno);
```

```
fprintf('Fro = %4.2f N\n',Fro);
```

```
fprintf('FBy = %4.2f N\n',FBy);
```

```
fprintf('FAy = %4.2f N\n',FAy);
```

```
fprintf('Mbc1 = %4.2f Nm\n',Mbc1);
```

158

```
fprintf('MBc2 = %4.2f Nm\n',MBc2);
```

```
fprintf('FBz = %4.2f N\n',FBz);
```

```
fprintf('FAz = %4.2f N\n',FAz);
```

```
fprintf('MBc3 = %4.2f Nm\n',MBc3);
```

```
fprintf('FAr = %4.2f N\n',FAr);
```

```
fprintf('FBr = %4.2f N\n',FBr);
```

```
fprintf('MBr = %4.2f Nm\n',MBr);
```

```
fprintf('d = %3.2f mm\n',d);
```

```
clc
```

```
%Gear main dimensions
```

```
%p=pitch
```

```
%d=pitch diameter
```

```
%da=major diameter
```

```
%s=pitch thickness
```

```
%b=pitch width
```

```
%m=module =p/pi
```

```
%z= number of teeth
```

```
%alpha=clutch angle generally 20 degree
```

```
%minimum number of teeth is theoretically 17
```

```
%Basic equations
```

$\%d=m*z$

$\%p=m*pi$

$\%$  Check according to broken at pitch bottom

$R_m=35$ ;  $\%$  UTS MPa

$R_e=30$ ;  $\%$  Yield MPa

$K_b=1$ ;  $\%$  shape coefficient

$K_y=0.75$ ;  $\%$  surface coefficient

$K_c=1.2$ ;  $\%$  notch effect

$S=3$ ;  $\%$  safety factor

$\%$  As the PLA brittle plastic material, continuous strength of the material was assumed equal to  $R_e$ .

$\%q_{TD}$ = pure variable strength

$q_{Fe}=R_e$ ;

$q_{TD}=R_e*0.7$ ;

$q_s=(K_b*K_y*q_{TD})/(K_c*S)$ ;

$\%$  Module Check (m)

$M_r=3$ ;  $\%$  Nm

$K_f=2.4$ ;  $\%$  form coefficient

$K_i=1.25$ ;  $\%$  operation coefficient

$K_v=1$ ;  $\%$  speed coefficient

$K_e=1$ ;  $\%$  clutch ratio coefficient

160

$K_m=1.2$ ; %Load dist. coefficient

$S=3$ ; %safety factor

$w_m=15$ ; %module width coefficient it can selected between 10-20

$z=20$ ; %number of teeth

$\beta=20$ ; %beta

$m_n = [(2 * M_r * 1000 * K_f * K_i * K_v * K_e * K_m) / (w_m * z * q_s * \cos(\beta))]^{1/3}$ ;

$m_a = [m_n / \cos(\beta)]$ ;

fprintf('Gear check\n');

fprintf('Check according to broken at pitch bottom \n');

fprintf('q<sub>s</sub>= %4.2f N/mm<sup>2</sup>\n',q<sub>s</sub>);

fprintf('m<sub>n</sub>= %4.2f mm\n',m<sub>n</sub>);

fprintf('m<sub>a</sub>= %4.2f mm\n',m<sub>a</sub>);

% as the calculated value was found as 2.49 module was determined as 3.

%Check according to surface pressure

%q<sub>H</sub>= continous surface pressure strength

%K<sub>p</sub>= pressure coefficient

q<sub>H</sub>=Re;

K<sub>p</sub>=1;

P<sub>s</sub>=q<sub>H</sub>\*K<sub>p</sub>/S;

% Module Check

E=1100; %Elastic modulus of material MPa

```

i12=2; %rotation ratio

mn=[2*Mr*1000*Ki*Kv*Km*Ke*E*(i12+1)/(wm*z^2*Ps^2*i12)]^(1/3)*(cosd(b
eta));

ma=[mn/cosd(beta)];

fprintf('Check according to surface pressure \n');

fprintf('Ps= %4.2f N/mm^2\n',Ps);

fprintf('mn= %4.2f mm\n',mn);

fprintf('ma= %4.2f mm\n',ma);

% as the calculated value was found as 2.74 module was determined as ok.

% Pitch thickness check

m=3;

p=pi*m;

d=m*z;

dr=d+2*m;

s=p/2;

alpha=20; %degree

cosr= d*cosd(alpha)/dr; %clutch angle at related angle

alphan=acosd(cosr);

invr = tand(alphan)-alphan*pi/180;

inv20=0.01490; %inv value for 20 degree

sr=dr*[s/d+(0.01490-invr)] ;

```

162

```
srmin= 0.4*m;
```

```
fprintf('tooth thickness check \n');
```

```
fprintf('sr= %4.2f mm\n',sr);
```

```
fprintf('srmin= %4.2f mm\n',srmin);
```

```
% as the calculated sr is more than srmin gear set is ok.
```

```
clc
```

```
% Stud Diameter Check
```

```
Mr=6; %Nm
```

```
D0=17; % Inner diameter
```

```
z=2; %Number of stud
```

```
Ft=(2*Mr*1000)/(z*D0); % Tangential force
```

```
b=6; % Wall thickness of coupling
```

```
Rs=350; % Yield strength of stud material
```

```
S=3;
```

```
Mb=(16*Ft*b); % Bending moment
```

```
% Wb=(pi*dp^3) Bending moment strength mm^3
```

```
% qs=Mb*S/Wb*Rs Safe stress value
```

```
dp=[(S*Mb)/(pi*Rs)]^(1/3);
```

```
Wb=(pi*dp^3);
```

```
qs=Mb/Wb;
```

```
% as the minimum diameter was calculated 4.5 diameter was selected as 5.
```

```
fprintf('Stud Diameter Check\n');
```

```
fprintf('Ft = %4.2f N\n',Ft);
```

```
fprintf('Mb = %4.2f Nm\n',Mb);
```

```
fprintf('Wb = %4.2f mm^3\n',Wb);
```

```
fprintf('qs = %4.2f N/mm^2\n',qs);
```

```
fprintf('dp = %4.2f mm\n',dp);
```



## Appendix 2 (C# Codes)

### Serial Control Software

```
namespace serial_control
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed;
        otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
        }
    }
}
```

```
        base.Dispose(disposing);

    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.components = new System.ComponentModel.Container();
        this.comboBox1 = new System.Windows.Forms.ComboBox();
        this.comboBox2 = new System.Windows.Forms.ComboBox();
        this.progressBar1 = new System.Windows.Forms.ProgressBar();
        this.groupBox1 = new System.Windows.Forms.GroupBox();
        this.button1 = new System.Windows.Forms.Button();
        this.textBox1 = new System.Windows.Forms.TextBox();
        this.groupBox2 = new System.Windows.Forms.GroupBox();
        this.listBox1 = new System.Windows.Forms.ListBox();
        this.label4 = new System.Windows.Forms.Label();
        this.button2 = new System.Windows.Forms.Button();
        this.button3 = new System.Windows.Forms.Button();
    }
}

```

```
this.button4 = new System.Windows.Forms.Button();  
  
this.label1 = new System.Windows.Forms.Label();  
  
this.label2 = new System.Windows.Forms.Label();  
  
this.label3 = new System.Windows.Forms.Label();  
  
this.serialPort1 = new System.IO.Ports.SerialPort(this.components);  
  
this.button5 = new System.Windows.Forms.Button();  
  
this.button6 = new System.Windows.Forms.Button();  
  
this.button7 = new System.Windows.Forms.Button();  
  
this.button8 = new System.Windows.Forms.Button();  
  
this.button9 = new System.Windows.Forms.Button();  
  
this.radioButton1 = new System.Windows.Forms.RadioButton();  
  
this.radioButton2 = new System.Windows.Forms.RadioButton();  
  
this.radioButton3 = new System.Windows.Forms.RadioButton();  
  
this.radioButton4 = new System.Windows.Forms.RadioButton();  
  
this.radioButton5 = new System.Windows.Forms.RadioButton();  
  
this.groupBox3 = new System.Windows.Forms.GroupBox();  
  
this.button10 = new System.Windows.Forms.Button();  
  
this.trackBar1 = new System.Windows.Forms.TrackBar();  
  
this.vScrollBar1 = new System.Windows.Forms.VScrollBar();  
  
this.label5 = new System.Windows.Forms.Label();  
  
this.label6 = new System.Windows.Forms.Label();
```

```
this.label7 = new System.Windows.Forms.Label();

this.groupBox4 = new System.Windows.Forms.GroupBox();

this.button11 = new System.Windows.Forms.Button();

this.radioButton6 = new System.Windows.Forms.RadioButton();

this.radioButton7 = new System.Windows.Forms.RadioButton();

this.radioButton8 = new System.Windows.Forms.RadioButton();

this.groupBox6 = new System.Windows.Forms.GroupBox();

this.radioButton9 = new System.Windows.Forms.RadioButton();

this.radioButton10 = new System.Windows.Forms.RadioButton();

this.radioButton11 = new System.Windows.Forms.RadioButton();

this.button13 = new System.Windows.Forms.Button();

this.radioButton12 = new System.Windows.Forms.RadioButton();

this.radioButton13 = new System.Windows.Forms.RadioButton();

this.radioButton14 = new System.Windows.Forms.RadioButton();

this.listBox2 = new System.Windows.Forms.ListBox();

this.listBox3 = new System.Windows.Forms.ListBox();

this.groupBox1.SuspendLayout();

this.groupBox2.SuspendLayout();

this.groupBox3.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.trackBar1)).BeginInit();
```

```
this.groupBox4.SuspendLayout();

this.groupBox6.SuspendLayout();

this.SuspendLayout();

//

// comboBox1

//

this.comboBox1.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;

this.comboBox1.FormattingEnabled = true;

this.comboBox1.Location = new System.Drawing.Point(34, 32);

this.comboBox1.Name = "comboBox1";

this.comboBox1.Size = new System.Drawing.Size(121, 21);

this.comboBox1.TabIndex = 0;

this.comboBox1.SelectedIndexChanged +=
new
System.EventHandler(this.comboBox1_SelectedIndexChanged);

//

// comboBox2

//

this.comboBox2.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;

this.comboBox2.FormattingEnabled = true;

this.comboBox2.Items.AddRange(new object[] {
```

```
"9600",  
"115200"});  
  
this.comboBox2.Location = new System.Drawing.Point(204, 32);  
  
this.comboBox2.Name = "comboBox2";  
  
this.comboBox2.Size = new System.Drawing.Size(121, 21);  
  
this.comboBox2.TabIndex = 1;  
  
//  
  
// progressBar1  
  
//  
  
this.progressBar1.Location = new System.Drawing.Point(73, 283);  
  
this.progressBar1.Name = "progressBar1";  
  
this.progressBar1.Size = new System.Drawing.Size(100, 23);  
  
this.progressBar1.TabIndex = 2;  
  
//  
  
// groupBox1  
  
//  
  
this.groupBox1.Controls.Add(this.button1);  
  
this.groupBox1.Controls.Add(this.textBox1);  
  
this.groupBox1.Location = new System.Drawing.Point(25, 68);  
  
this.groupBox1.Name = "groupBox1";  
  
this.groupBox1.Size = new System.Drawing.Size(190, 122);
```

```
this.groupBox1.TabIndex = 3;

this.groupBox1.TabStop = false;

this.groupBox1.Text = "Send data";

//

// button1

//

this.button1.Enabled = false;

this.button1.Location = new System.Drawing.Point(55, 93);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(75, 23);

this.button1.TabIndex = 4;

this.button1.Text = "Send";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1_Click);

//

// textBox1

//

this.textBox1.Enabled = false;

this.textBox1.Location = new System.Drawing.Point(22, 19);

this.textBox1.Multiline = true;

this.textBox1.Name = "textBox1";
```

```
this.textBox1.Size = new System.Drawing.Size(144, 52);

this.textBox1.TabIndex = 4;

//

// groupBox2

//

this.groupBox2.Controls.Add(this.listBox1);

this.groupBox2.Controls.Add(this.label4);

this.groupBox2.Location = new System.Drawing.Point(245, 68);

this.groupBox2.Name = "groupBox2";

this.groupBox2.Size = new System.Drawing.Size(195, 122);

this.groupBox2.TabIndex = 5;

this.groupBox2.TabStop = false;

this.groupBox2.Text = "Receive data";

//

// listBox1

//

this.listBox1.FormattingEnabled = true;

this.listBox1.Location = new System.Drawing.Point(24, 21);

this.listBox1.Name = "listBox1";

this.listBox1.Size = new System.Drawing.Size(154, 95);

this.listBox1.TabIndex = 6;
```



```
//  
  
// label4  
  
//  
  
this.label4.AutoSize = true;  
  
this.label4.Location = new System.Drawing.Point(80, 49);  
  
this.label4.Name = "label4";  
  
this.label4.Size = new System.Drawing.Size(35, 13);  
  
this.label4.TabIndex = 22;  
  
this.label4.Text = "label4";  
  
//  
  
// button2  
  
//  
  
this.button2.Location = new System.Drawing.Point(250, 208);  
  
this.button2.Name = "button2";  
  
this.button2.Size = new System.Drawing.Size(75, 23);  
  
this.button2.TabIndex = 5;  
  
this.button2.Text = "read";  
  
this.button2.UseVisualStyleBackColor = true;  
  
this.button2.Click += new System.EventHandler(this.button2_Click);  
  
//  
  
// button3
```

```
//  
  
this.button3.Location = new System.Drawing.Point(34, 217);  
  
this.button3.Name = "button3";  
  
this.button3.Size = new System.Drawing.Size(84, 31);  
  
this.button3.TabIndex = 6;  
  
this.button3.Text = "Open Port";  
  
this.button3.UseVisualStyleBackColor = true;  
  
this.button3.Click += new System.EventHandler(this.button3_Click);  
  
//  
  
// button4  
  
//  
  
this.button4.Location = new System.Drawing.Point(132, 217);  
  
this.button4.Name = "button4";  
  
this.button4.Size = new System.Drawing.Size(83, 30);  
  
this.button4.TabIndex = 7;  
  
this.button4.Text = "Close Port";  
  
this.button4.UseVisualStyleBackColor = true;  
  
this.button4.Click += new System.EventHandler(this.button4_Click);  
  
//  
  
// label1  
  
//
```

```
this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(77, 9);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(62, 13);

this.label1.TabIndex = 8;

this.label1.Text = "Port Names";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(228, 9);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(58, 13);

this.label2.TabIndex = 9;

this.label2.Text = "Baud Rate";

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(102, 262);

this.label3.Name = "label3";
```

```
this.label3.Size = new System.Drawing.Size(37, 13);

this.label3.TabIndex = 10;

this.label3.Text = "Status";

//

// button5

//

this.button5.ImageKey = "(none)";

this.button5.Location = new System.Drawing.Point(624, 68);

this.button5.Name = "button5";

this.button5.Size = new System.Drawing.Size(75, 23);

this.button5.TabIndex = 11;

this.button5.Text = "forward";

this.button5.UseVisualStyleBackColor = true;

this.button5.Click += new System.EventHandler(this.button5_Click);

//

// button6

//

this.button6.Location = new System.Drawing.Point(624, 148);

this.button6.Name = "button6";

this.button6.Size = new System.Drawing.Size(75, 23);

this.button6.TabIndex = 12;
```

```
this.button6.Text = "back";

this.button6.UseVisualStyleBackColor = true;

this.button6.Click += new System.EventHandler(this.button6_Click);

//

// button7

//

this.button7.Location = new System.Drawing.Point(531, 107);

this.button7.Name = "button7";

this.button7.Size = new System.Drawing.Size(75, 23);

this.button7.TabIndex = 13;

this.button7.Text = "left";

this.button7.UseVisualStyleBackColor = true;

this.button7.Click += new System.EventHandler(this.button7_Click);

//

// button8

//

this.button8.Location = new System.Drawing.Point(722, 107);

this.button8.Name = "button8";

this.button8.Size = new System.Drawing.Size(75, 23);

this.button8.TabIndex = 14;

this.button8.Text = "right";
```

```
this.button8.UseVisualStyleBackColor = true;

this.button8.Click += new System.EventHandler(this.button8_Click);

//

// button9

//

this.button9.Location = new System.Drawing.Point(624, 107);

this.button9.Name = "button9";

this.button9.Size = new System.Drawing.Size(75, 23);

this.button9.TabIndex = 15;

this.button9.Text = "stop";

this.button9.UseVisualStyleBackColor = true;

this.button9.Click += new System.EventHandler(this.button9_Click);

//

// radioButton1

//

this.radioButton1.AutoSize = true;

this.radioButton1.Location = new System.Drawing.Point(20, 41);

this.radioButton1.Name = "radioButton1";

this.radioButton1.Size = new System.Drawing.Size(57, 17);

this.radioButton1.TabIndex = 16;

this.radioButton1.TabStop = true;
```

```
this.radioButton1.Text = "Gear 1";

this.radioButton1.UseVisualStyleBackColor = true;

this.radioButton1.CheckedChanged += new
System.EventHandler(this.radioButton1_CheckedChanged);

//
// radioButton2
//

this.radioButton2.AutoSize = true;

this.radioButton2.Location = new System.Drawing.Point(93, 41);

this.radioButton2.Name = "radioButton2";

this.radioButton2.Size = new System.Drawing.Size(57, 17);

this.radioButton2.TabIndex = 17;

this.radioButton2.TabStop = true;

this.radioButton2.Text = "Gear 2";

this.radioButton2.UseVisualStyleBackColor = true;

//
// radioButton3
//

this.radioButton3.AutoSize = true;

this.radioButton3.Location = new System.Drawing.Point(168, 41);

this.radioButton3.Name = "radioButton3";
```

```
this.radioButton3.Size = new System.Drawing.Size(57, 17);

this.radioButton3.TabIndex = 18;

this.radioButton3.TabStop = true;

this.radioButton3.Text = "Gear 3";

this.radioButton3.UseVisualStyleBackColor = true;

//

// radioButton4

//

this.radioButton4.AutoSize = true;

this.radioButton4.Location = new System.Drawing.Point(231, 41);

this.radioButton4.Name = "radioButton4";

this.radioButton4.Size = new System.Drawing.Size(57, 17);

this.radioButton4.TabIndex = 19;

this.radioButton4.TabStop = true;

this.radioButton4.Text = "Gear 4";

this.radioButton4.UseVisualStyleBackColor = true;

//

// radioButton5

//

this.radioButton5.AutoSize = true;

this.radioButton5.Location = new System.Drawing.Point(294, 41);
```



```
this.radioButton5.Name = "radioButton5";

this.radioButton5.Size = new System.Drawing.Size(57, 17);

this.radioButton5.TabIndex = 20;

this.radioButton5.TabStop = true;

this.radioButton5.Text = "Gear 5";

this.radioButton5.UseVisualStyleBackColor = true;

//

// groupBox3

//

this.groupBox3.Controls.Add(this.radioButton1);

this.groupBox3.Controls.Add(this.radioButton5);

this.groupBox3.Controls.Add(this.radioButton2);

this.groupBox3.Controls.Add(this.radioButton4);

this.groupBox3.Controls.Add(this.radioButton3);

this.groupBox3.Location = new System.Drawing.Point(498, 217);

this.groupBox3.Name = "groupBox3";

this.groupBox3.Size = new System.Drawing.Size(358, 78);

this.groupBox3.TabIndex = 21;

this.groupBox3.TabStop = false;

this.groupBox3.Text = "Gearbox";

//
```

```
// button10

//

this.button10.Location = new System.Drawing.Point(348, 208);

this.button10.Name = "button10";

this.button10.Size = new System.Drawing.Size(75, 23);

this.button10.TabIndex = 22;

this.button10.Text = "clear";

this.button10.UseVisualStyleBackColor = true;

this.button10.Click += new System.EventHandler(this.button10_Click);

//

// trackBar1

//

this.trackBar1.Location = new System.Drawing.Point(498, 24);

this.trackBar1.Maximum = 40;

this.trackBar1.Name = "trackBar1";

this.trackBar1.Size = new System.Drawing.Size(351, 45);

this.trackBar1.TabIndex = 23;

this.trackBar1.Value = 4;

//

// vScrollBar1

//
```

```
this.vScrollBar1.Location = new System.Drawing.Point(831, 135);

this.vScrollBar1.Name = "vScrollBar1";

this.vScrollBar1.Size = new System.Drawing.Size(17, 80);

this.vScrollBar1.TabIndex = 24;

//

// label5

//

this.label5.AutoSize = true;

this.label5.Location = new System.Drawing.Point(655, 9);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(44, 13);

this.label5.TabIndex = 25;

this.label5.Text = "Velocity";

this.label5.Click += new System.EventHandler(this.label5_Click);

//

// label6

//

this.label6.AutoSize = true;

this.label6.Location = new System.Drawing.Point(470, 26);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(34, 13);
```

```
this.label6.TabIndex = 26;

this.label6.Text = "0 m/s";

//

// label7

//

this.label7.AutoSize = true;

this.label7.Location = new System.Drawing.Point(845, 24);

this.label7.Name = "label7";

this.label7.Size = new System.Drawing.Size(34, 13);

this.label7.TabIndex = 27;

this.label7.Text = "4 m/s";

//

// groupBox4

//

this.groupBox4.Controls.Add(this.listBox2);

this.groupBox4.Controls.Add(this.radioButton12);

this.groupBox4.Controls.Add(this.radioButton13);

this.groupBox4.Controls.Add(this.radioButton8);

this.groupBox4.Controls.Add(this.radioButton7);

this.groupBox4.Controls.Add(this.radioButton6);

this.groupBox4.Controls.Add(this.button11);
```

```
this.groupBox4.Location = new System.Drawing.Point(885, 12);

this.groupBox4.Name = "groupBox4";

this.groupBox4.Size = new System.Drawing.Size(115, 283);

this.groupBox4.TabIndex = 30;

this.groupBox4.TabStop = false;

this.groupBox4.Text = "Sensor 1";

//

// button11

//

this.button11.Location = new System.Drawing.Point(14, 123);

this.button11.Name = "button11";

this.button11.Size = new System.Drawing.Size(75, 23);

this.button11.TabIndex = 32;

this.button11.Text = "Apply";

this.button11.UseVisualStyleBackColor = true;

//

// radioButton6

//

this.radioButton6.AutoSize = true;

this.radioButton6.Location = new System.Drawing.Point(14, 24);

this.radioButton6.Name = "radioButton6";
```

```
this.radioButton6.Size = new System.Drawing.Size(55, 17);

this.radioButton6.TabIndex = 21;

this.radioButton6.TabStop = true;

this.radioButton6.Text = "LM 35";

this.radioButton6.UseVisualStyleBackColor = true;

this.radioButton6.CheckedChanged += new
System.EventHandler(this.radioButton6_CheckedChanged);

//

// radioButton7

//

this.radioButton7.AutoSize = true;

this.radioButton7.Location = new System.Drawing.Point(14, 47);

this.radioButton7.Name = "radioButton7";

this.radioButton7.Size = new System.Drawing.Size(63, 17);

this.radioButton7.TabIndex = 33;

this.radioButton7.TabStop = true;

this.radioButton7.Text = "DHT 21";

this.radioButton7.UseVisualStyleBackColor = true;

//

// radioButton8

//
```

```
this.radioButton8.AutoSize = true;

this.radioButton8.Location = new System.Drawing.Point(14, 70);

this.radioButton8.Name = "radioButton8";

this.radioButton8.Size = new System.Drawing.Size(89, 17);

this.radioButton8.TabIndex = 34;

this.radioButton8.TabStop = true;

this.radioButton8.Text = "Flame Sensor";

this.radioButton8.UseVisualStyleBackColor = true;

//

// groupBox6

//

this.groupBox6.Controls.Add(this.listBox3);

this.groupBox6.Controls.Add(this.radioButton14);

this.groupBox6.Controls.Add(this.radioButton9);

this.groupBox6.Controls.Add(this.radioButton10);

this.groupBox6.Controls.Add(this.radioButton11);

this.groupBox6.Controls.Add(this.button13);

this.groupBox6.Location = new System.Drawing.Point(1006, 12);

this.groupBox6.Name = "groupBox6";

this.groupBox6.Size = new System.Drawing.Size(115, 283);

this.groupBox6.TabIndex = 35;
```

```
this.groupBox6.TabStop = false;

this.groupBox6.Text = "Sensor 2";

//

// radioButton9

//

this.radioButton9.AutoSize = true;

this.radioButton9.Location = new System.Drawing.Point(14, 70);

this.radioButton9.Name = "radioButton9";

this.radioButton9.Size = new System.Drawing.Size(89, 17);

this.radioButton9.TabIndex = 34;

this.radioButton9.TabStop = true;

this.radioButton9.Text = "Flame Sensor";

this.radioButton9.UseVisualStyleBackColor = true;

//

// radioButton10

//

this.radioButton10.AutoSize = true;

this.radioButton10.Location = new System.Drawing.Point(14, 47);

this.radioButton10.Name = "radioButton10";

this.radioButton10.Size = new System.Drawing.Size(63, 17);

this.radioButton10.TabIndex = 33;
```



```
this.radioButton10.TabStop = true;

this.radioButton10.Text = "DHT 21";

this.radioButton10.UseVisualStyleBackColor = true;

//

// radioButton11

//

this.radioButton11.AutoSize = true;

this.radioButton11.Location = new System.Drawing.Point(14, 24);

this.radioButton11.Name = "radioButton11";

this.radioButton11.Size = new System.Drawing.Size(55, 17);

this.radioButton11.TabIndex = 21;

this.radioButton11.TabStop = true;

this.radioButton11.Text = "LM 35";

this.radioButton11.UseVisualStyleBackColor = true;

//

// button13

//

this.button13.Location = new System.Drawing.Point(14, 123);

this.button13.Name = "button13";

this.button13.Size = new System.Drawing.Size(75, 23);

this.button13.TabIndex = 32;
```

```
this.button13.Text = "Apply";

this.button13.UseVisualStyleBackColor = true;

//

// radioButton12

//

this.radioButton12.AutoSize = true;

this.radioButton12.Location = new System.Drawing.Point(14, 93);

this.radioButton12.Name = "radioButton12";

this.radioButton12.Size = new System.Drawing.Size(73, 17);

this.radioButton12.TabIndex = 35;

this.radioButton12.TabStop = true;

this.radioButton12.Text = "HCSR501";

this.radioButton12.UseVisualStyleBackColor = true;

//

// radioButton13

//

this.radioButton13.AutoSize = true;

this.radioButton13.Location = new System.Drawing.Point(14, 93);

this.radioButton13.Name = "radioButton13";

this.radioButton13.Size = new System.Drawing.Size(73, 17);

this.radioButton13.TabIndex = 36;
```

```
this.radioButton13.TabStop = true;

this.radioButton13.Text = "HCSR501";

this.radioButton13.UseVisualStyleBackColor = true;

//

// radioButton14

//

this.radioButton14.AutoSize = true;

this.radioButton14.Location = new System.Drawing.Point(14, 93);

this.radioButton14.Name = "radioButton14";

this.radioButton14.Size = new System.Drawing.Size(73, 17);

this.radioButton14.TabIndex = 37;

this.radioButton14.TabStop = true;

this.radioButton14.Text = "HCSR501";

this.radioButton14.UseVisualStyleBackColor = true;

//

// listBox2

//

this.listBox2.FormattingEnabled = true;

this.listBox2.Location = new System.Drawing.Point(14, 181);

this.listBox2.Name = "listBox2";

this.listBox2.Size = new System.Drawing.Size(89, 82);
```

```
this.listBox2.TabIndex = 23;

//

// listBox3

//

this.listBox3.FormattingEnabled = true;

this.listBox3.Location = new System.Drawing.Point(14, 181);

this.listBox3.Name = "listBox3";

this.listBox3.Size = new System.Drawing.Size(89, 82);

this.listBox3.TabIndex = 37;

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(1127, 352);

this.Controls.Add(this.groupBox6);

this.Controls.Add(this.groupBox4);

this.Controls.Add(this.label7);

this.Controls.Add(this.label6);

this.Controls.Add(this.label5);

this.Controls.Add(this.vScrollBar1);
```

```
this.Controls.Add(this.trackBar1);  
  
this.Controls.Add(this.button10);  
  
this.Controls.Add(this.groupBox3);  
  
this.Controls.Add(this.button2);  
  
this.Controls.Add(this.button9);  
  
this.Controls.Add(this.button8);  
  
this.Controls.Add(this.button7);  
  
this.Controls.Add(this.button6);  
  
this.Controls.Add(this.button5);  
  
this.Controls.Add(this.label3);  
  
this.Controls.Add(this.label2);  
  
this.Controls.Add(this.label1);  
  
this.Controls.Add(this.button4);  
  
this.Controls.Add(this.button3);  
  
this.Controls.Add(this.groupBox2);  
  
this.Controls.Add(this.groupBox1);  
  
this.Controls.Add(this.progressBar1);  
  
this.Controls.Add(this.comboBox2);  
  
this.Controls.Add(this.comboBox1);  
  
this.Name = "Form1";  
  
this.Text = "Serial Control";
```

```
this.groupBox1.ResumeLayout(false);

this.groupBox1.PerformLayout();

this.groupBox2.ResumeLayout(false);

this.groupBox2.PerformLayout();

this.groupBox3.ResumeLayout(false);

this.groupBox3.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.trackBar1)).EndInit();

this.groupBox4.ResumeLayout(false);

this.groupBox4.PerformLayout();

this.groupBox6.ResumeLayout(false);

this.groupBox6.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.ComboBox comboBox1;

private System.Windows.Forms.ComboBox comboBox2;

private System.Windows.Forms.ProgressBar progressBar1;

private System.Windows.Forms.GroupBox groupBox1;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.TextBox textBox1;
```

```
private System.Windows.Forms.GroupBox groupBox2;

private System.Windows.Forms.Button button3;

private System.Windows.Forms.Button button4;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Label label3;

private System.IO.Ports.SerialPort serialPort1;

private System.Windows.Forms.Button button5;

private System.Windows.Forms.Button button6;

private System.Windows.Forms.Button button7;

private System.Windows.Forms.Button button8;

private System.Windows.Forms.Button button9;

private System.Windows.Forms.RadioButton radioButton1;

private System.Windows.Forms.RadioButton radioButton2;

private System.Windows.Forms.RadioButton radioButton3;

private System.Windows.Forms.RadioButton radioButton4;

private System.Windows.Forms.RadioButton radioButton5;

private System.Windows.Forms.GroupBox groupBox3;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.ListBox listBox1;

private System.Windows.Forms.Label label4;
```

```
private System.Windows.Forms.Button button10;

private System.Windows.Forms.TrackBar trackBar1;

private System.Windows.Forms.VScrollBar vScrollBar1;

private System.Windows.Forms.Label label5;

private System.Windows.Forms.Label label6;

private System.Windows.Forms.Label label7;

private System.Windows.Forms.GroupBox groupBox4;

private System.Windows.Forms.RadioButton radioButton6;

private System.Windows.Forms.Button button11;

private System.Windows.Forms.ListBox listBox2;

private System.Windows.Forms.RadioButton radioButton12;

private System.Windows.Forms.RadioButton radioButton13;

private System.Windows.Forms.RadioButton radioButton8;

private System.Windows.Forms.RadioButton radioButton7;

private System.Windows.Forms.GroupBox groupBox6;

private System.Windows.Forms.ListBox listBox3;

private System.Windows.Forms.RadioButton radioButton14;

private System.Windows.Forms.RadioButton radioButton9;

private System.Windows.Forms.RadioButton radioButton10;

private System.Windows.Forms.RadioButton radioButton11;

private System.Windows.Forms.Button button13;
```



```

    }
}

```

### Appendix 3 (Arduino Codes)

#### Control of the Transmission and Motion Systems

```

int in7 = 7; //DC A (RIGHT SIDE CONTROL)

int in8 = 8; //DC B

int in9 = 9; //DC PWM

int in4 = 4; // IINEAR A (LEFT SIDE CONTROL)

int in5 = 5; // IINEAR B

int in6 = 6; // IINEAR PWM

int gear;

int gear_t;

int linear=0;

int s=0;

int cons1=0;

int cons2=0;

void setup() {

    Serial.begin(9600);

    pinMode(in4, OUTPUT);

    pinMode(in5, OUTPUT);

    pinMode(in6, OUTPUT);

```

196

```
pinMode(in7, OUTPUT);

pinMode(in8, OUTPUT);

pinMode(in9, OUTPUT);

pinMode(10, INPUT);

pinMode(11, INPUT);
}

void loop()
{
  if (Serial.available() > 0)
  {
    gear=Serial.read();
  }

  if (gear == 'X') //Emergency
  {
    gear = 0;

    gear_t= 0;

    analogWrite(in9, gear);

    analogWrite(in6, gear_t);

    digitalWrite(in7, LOW);

    digitalWrite(in8, LOW);

    digitalWrite(in4, LOW);
```

```
digitalWrite(in5, LOW);

cons1=0;

cons2=0;

}

if (gear == 'r') //reset
{
  gear = 0;
  gear_t= 0;
  analogWrite(in9, gear);
  analogWrite(in6, gear_t);
  digitalWrite(in7, LOW);
  digitalWrite(in8, LOW);
  digitalWrite(in4, LOW);
  digitalWrite(in5, LOW);
  cons1=0;
  cons2=0;
}

if(gear=='a')// autonomous mode activated

//Linear Actuator Initialization

if (s==0)

{
```

198

```
    gear = 255;

    gear_t = 255;

    analogWrite(in9, gear);

    analogWrite(in6, gear_t);

    digitalWrite(in7, LOW);

    digitalWrite(in8, HIGH);

    digitalWrite(in4, LOW);

    digitalWrite(in5, HIGH);

    delay(2000);

    digitalWrite(in7, LOW);

    digitalWrite(in8, LOW);

    digitalWrite(in4, LOW);

    digitalWrite(in5, LOW);

    s=1;

}

else{

if ((digitalRead(10)==LOW) && (digitalRead(11)==LOW))

{

else

{

    gear = 255;
```

```
gear_t = 255;

analogWrite(in9, gear);

analogWrite(in6, gear_t);

digitalWrite(in7, HIGH);

digitalWrite(in8, LOW);

digitalWrite(in4, LOW);

digitalWrite(in5, LOW);

delay(2000);
}

if ((digitalRead(10)==LOW) && (digitalRead(11)==LOW))

{

gear = 255;

gear_t = 255;

analogWrite(in9, gear);

analogWrite(in6, gear_t);

digitalWrite(in7, HIGH);

digitalWrite(in8, LOW);

if( linear==0)

{digitalWrite(in4, HIGH); //Linear actuator

digitalWrite(in5, LOW); //Linear actuator

delay(2000);
```

200

```
    linear=1;}

else

    {digitalWrite(in4, LOW);

    digitalWrite(in5, LOW);}

}

else{

    gear = 255;

    gear_t = 255;

    analogWrite(in9, gear);

    analogWrite(in6, gear_t);

    digitalWrite(in7, LOW);

    digitalWrite(in8, HIGH);

    if(linear==1)

    {

        digitalWrite(in4, LOW);

        digitalWrite(in5, HIGH);

        delay(2000);

        linear=0;}

    else{

        digitalWrite(in4, LOW);

        digitalWrite(in5, LOW);
```

```
digitalWrite(in7, HIGH);

digitalWrite(in8, LOW);}

}

}

//manual activated

if(gear=='m' || gear=='r' || gear=='5' || gear=='4' || gear=='3' || gear=='2' || gear=='1' ||
gear=='0' || gear=='E' || gear=='D' || gear=='C' || gear=='B' || gear=='A' || gear=='W' ||
gear=='Z' || gear=='F' || gear=='S' || gear=='N' || gear=='L' || gear=='X')

{

digitalWrite(in5, LOW);

digitalWrite(in7, LOW);

digitalWrite(in8, LOW);

analogWrite(in9, 0);

analogWrite(in6, 0);

//DC CONTROL

if (gear == '1')

{

gear = 150;

analogWrite(in9, gear);

digitalWrite(in7, HIGH);

digitalWrite(in8, LOW);

}
```

202

```
if (gear == '2')
{
    gear = 175;
    analogWrite(in9, gear);
    digitalWrite(in7, HIGH);
    digitalWrite(in8, LOW);
}
if (gear == '3')
{
    gear =200 ;
    analogWrite(in9, gear);
    digitalWrite(in7, HIGH);
    digitalWrite(in8, LOW);
}
if (gear == '4')
{
    gear = 225;
    analogWrite(in9, gear);
    digitalWrite(in7, HIGH);
    digitalWrite(in8, LOW);
}
```



```
if (gear == '5')
{
    gear = 250;
    analogWrite(in9, gear);
    digitalWrite(in7, HIGH);
    digitalWrite(in8, LOW);
}
if (gear == 'A')
{
    gear = 150;
    analogWrite(in9, gear);
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);
}
if (gear == 'B')
{
    gear = 175;
    analogWrite(in9, gear);
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);
}
```

204

```
if (gear == 'C')
{
    gear = 200;
    analogWrite(in9, gear);
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);
}
if (gear == 'D')
{
    gear = 225;
    analogWrite(in9, gear);
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);
}
if (gear == 'E')
{
    gear = 250;
    analogWrite(in9, gear);
    digitalWrite(in7, LOW);
    digitalWrite(in8, HIGH);
}
```

```
if (gear == '0')
{
    gear = 0;
    analogWrite(in9, gear);
    digitalWrite(in7, LOW);
    digitalWrite(in8, LOW);
}
//LINEAR ACTUATOR
if (gear == 'F')
{
    gear = 150;
    analogWrite(in6, gear);
    digitalWrite(in4, HIGH);
    digitalWrite(in5, LOW);
}
if (gear == 'W')
{
    gear = 150;
    analogWrite(in6, gear);
    digitalWrite(in4, LOW);
    digitalWrite(in5, HIGH);
```

```
    }  
    if (gear == 'Z')  
    {  
        gear = 0;  
        analogWrite(in6, 0);  
        digitalWrite(in4, LOW);  
        digitalWrite(in5, LOW);  
    }  
    //TRANSMISSION  
    if (gear == 'L')  
    {  
        if (cons2<1)  
        {  
            gear = 255;  
            gear_t = 255;  
            analogWrite(in9, gear);  
            analogWrite(in6, gear_t);  
            digitalWrite(in4, HIGH);  
            digitalWrite(in5, LOW);  
            digitalWrite(in7, HIGH);  
            digitalWrite(in8, LOW);
```

```
    delay(2000);

    cons2=1;

}

else

{

    gear = 255;

    gear_t = 0;

    analogWrite(in9, gear);

    analogWrite(in6, gear_t);

    digitalWrite(in4, LOW);

    digitalWrite(in5, LOW);

    digitalWrite(in7, HIGH);

    digitalWrite(in8, LOW);

    cons1=0;

}

}

if (gear == 'S')

{

    if (cons1<1)

    {

        gear = 255;
```

```
gear_t = 255;

analogWrite(in9, gear);

analogWrite(in6, gear_t);

digitalWrite(in4, LOW);

digitalWrite(in5, HIGH);

digitalWrite(in7, LOW);

digitalWrite(in8, HIGH);

delay(2000);

cons1=1;
}

else

{

gear = 255;

gear_t = 0;

analogWrite(in9, gear);

analogWrite(in6, gear_t);

digitalWrite(in4, LOW);

digitalWrite(in5, LOW);

digitalWrite(in7, HIGH);

digitalWrite(in8, LOW);

cons2=0;
```

```
    }  
  }  
  if (gear == 'N')  
  {  
    gear = 0;  
    gear_t=0;  
    analogWrite(in9, gear);  
    analogWrite(in6, gear);  
    digitalWrite(in4, LOW);  
    digitalWrite(in5, LOW);  
    digitalWrite(in7, LOW);  
    digitalWrite(in8, LOW);  
    cons1=0;  
    cons2=0;  
  }  
}  
}
```

## **Appendix 4 (Faster R-CNN)**

### **Training**

```
model {  
  faster_rcnn {
```

210

```
num_classes: 2

image_resizer {

  keep_aspect_ratio_resizer {

    min_dimension: 600

    max_dimension: 1024

  }

}

feature_extractor {

  type: 'faster_rcnn_inception_v2'

  first_stage_features_stride: 16

}

first_stage_anchor_generator {

  grid_anchor_generator {

    scales: [0.25, 0.5, 1.0, 2.0]

    aspect_ratios: [0.5, 1.0, 2.0]

    height_stride: 16

    width_stride: 16

  }

}

first_stage_box_predictor_conv_hyperparams {

  op: CONV
```



```
regularizer {  
  l2_regularizer {  
    weight: 0.0  
  }  
}  
  
initializer {  
  truncated_normal_initializer {  
    stddev: 0.01  
  }  
}  
  
}  
  
first_stage_nms_score_threshold: 0.0  
first_stage_nms_iou_threshold: 0.7  
first_stage_max_proposals: 300  
first_stage_localization_loss_weight: 2.0  
first_stage_objectness_loss_weight: 1.0  
initial_crop_size: 14  
maxpool_kernel_size: 2  
maxpool_stride: 2  
second_stage_box_predictor {  
  mask_rcnn_box_predictor {
```

```
use_dropout: false

dropout_keep_probability: 1.0

fc_hyperparams {
  op: FC
  regularizer {
    l2_regularizer {
      weight: 0.0
    }
  }
  initializer {
    variance_scaling_initializer {
      factor: 1.0
      uniform: true
      mode: FAN_AVG
    }
  }
}

second_stage_post_processing {
  batch_non_max_suppression {
```

```
score_threshold: 0.0

iou_threshold: 0.6

max_detections_per_class: 100

max_total_detections: 300
}
score_converter: SOFTMAX
}
second_stage_localization_loss_weight: 2.0
second_stage_classification_loss_weight: 1.0
}
}

train_config: {
  batch_size: 1

  optimizer {
    momentum_optimizer: {
      learning_rate: {
        manual_step_learning_rate {
          initial_learning_rate: 0.0002

          schedule {
            step: 900000

            learning_rate: .00002
```

```
    }  
  
    schedule {  
  
        step: 1200000  
  
        learning_rate: .000002  
  
    }  
  
    momentum_optimizer_value: 0.9  
  
    }  
  
    use_moving_average: false  
  
    }  
  
    gradient_clipping_by_norm: 10.0  
  
    fine_tune_checkpoint:  
    "C:/tensorflow1/models/research/object_detection/faster_rcnn_inception_v2_coco  
_2018_01_28/model.ckpt"  
  
    from_detection_checkpoint: true  
  
    load_all_detection_checkpoint_vars: true  
  
    num_steps: 40000  
  
    data_augmentation_options {  
  
        random_horizontal_flip {  
  
        }  
  
    }  
  
    }  
  
    train_input_reader: {  
  
        tf_record_input_reader {
```

```

    input_path: "C:/tensorflow1/models/research/object_detection/train.record"
  }

  label_map_path:
"C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"
}

eval_config: {
  metrics_set: "coco_detection_metrics"
  num_examples: 90
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "C:/tensorflow1/models/research/object_detection/test.record"
  }

  label_map_path:
"C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"

  shuffle: false

  num_readers: 1
}

```

### **Implementation**

```

import os

import cv2

import numpy as np

```

216

```
import tensorflow as tf

import sys

sys.path.append("..")

# Import utilites

from utils import label_map_util

from utils import visualization_utils as vis_util

# Name of the directory

MODEL_NAME = 'inference_graph'

# Grab path CWD_PATH = os.getcwd()

# Path to frozen detection graph

PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,'frozen_inference_graph.pb')

# Path to label map

PATH_TO_LABELS = os.path.join(CWD_PATH,'training','labelmap.pbtxt')

NUM_CLASSES = 1

label_map = label_map_util.load_labelmap(PATH_TO_LABELS)

categories=label_map_util.convert_label_map_to_categories(label_map,
max_num_classes= NUM_CLASSES, use_display_name=True)

category_index = label_map_util.create_category_index(categories)

# Load the Tensorflow model

detection_graph = tf.Graph()

with detection_graph.as_default():
```

```
od_graph_def = tf.GraphDef()

with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:

    serialized_graph = fid.read()

    od_graph_def.ParseFromString(serialized_graph)

    tf.import_graph_def(od_graph_def, name='')

sess = tf.Session(graph=detection_graph)

# Input tensor is the image
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')

# Output tensors are the detection boxes, scores, and classes
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')
num_detections = detection_graph.get_tensor_by_name('num_detections:0')

video = cv2.VideoCapture(0)

ret = video.set(3,1280)

ret = video.set(4,720)

while(True):

    value

    ret, frame = video.read()

    frame_expanded = np.expand_dims(frame, axis=0)
```

```
(boxes, scores, classes, num) = sess.run(
    [detection_boxes, detection_scores, detection_classes, num_detections],
    feed_dict={image_tensor: frame_expanded})

vis_util.visualize_boxes_and_labels_on_image_array(
    frame,
    np.squeeze(boxes),
    np.squeeze(classes).astype(np.int32),
    np.squeeze(scores),
    category_index,
    use_normalized_coordinates=True,
    line_thickness=8,
    min_score_thresh=0.60)

cv2.imshow('Object detector', frame)

# Press 'q' to quit
if cv2.waitKey(1) == ord('q'):
    break

# Clean up
video.release()

cv2.destroyAllWindows()
```



## Appendix 5 (Python Codes)

### Robot Control Software

```
import sys

from PyQt4 import QtCore, QtGui, uic

from hokuyoLX import HokuyoLX

import numpy as np

import time

import serial

laser = HokuyoLX()

ser = serial.Serial('/dev/ttyACM0', 9600)

form_class = uic.loadUiType("interface_2.ui")[0]

import RPi.GPIO as GPIO

import time

GPIO_TRIGGER1 = 23

GPIO_ECHO1 = 24

GPIO_TRIGGER2 = 20

GPIO_ECHO2 = 21

flame1=25

flame2=8

flame3=7

GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(GPIO_TRIGGER1, GPIO.OUT)

GPIO.setup(GPIO_ECHO1, GPIO.IN)

GPIO.setup(GPIO_TRIGGER2, GPIO.OUT)

GPIO.setup(GPIO_ECHO2, GPIO.IN)

GPIO.setup(flame1, GPIO.IN)

GPIO.setup(flame2, GPIO.IN)

GPIO.setup(flame3, GPIO.IN)

class MyWindowClass(QtGui.QMainWindow, form_class):

    def __init__(self, parent=None):

        QtGui.QMainWindow.__init__(self, parent)

        self.setupUi(self)

        QtGui.QApplication.setStyle('cleanlooks')

        self.timer_meas = QtCore.QTimer(self)

        self.timer_meas.timeout.connect(self.top_sonar)

        self.timer_meas3 = QtCore.QTimer(self)

        self.timer_meas3.timeout.connect(self.bottom_sonar)

        self.timer_meas2 = QtCore.QTimer(self)

        self.timer_meas2.timeout.connect(self.autonom)

        self.timer_meas4 = QtCore.QTimer(self)

        self.timer_meas4.timeout.connect(self.flame_starting)

        self.sonar_start.clicked.connect(self.top_sonar_start)
```

```
self.sonar_start.clicked.connect(self.bottom_sonar_start)

self.sonar_stop.clicked.connect(self.top_sonar_stop)

self.sonar_stop.clicked.connect(self.bottom_sonar_stop)

self.sonar_start.setStyleSheet("background-color:white")

self.sonar_stop.setStyleSheet("background-color:white")

self.flame_start.clicked.connect(self.flame_starting)

self.flame_stop.clicked.connect(self.flame_stopping)

self.autonomous_start.setStyleSheet("background-color:white")

self.autonomous_start.clicked.connect(self.autonom_start)

self.autonomous_stop.clicked.connect(self.autonom_stop)

self.autonomous_start.setStyleSheet("background-color:white")

self.autonomous_stop.setStyleSheet("background-color:white")

self.linear.clicked.connect(self.linear_motion)

self.linear.setStyleSheet("background-color:white")

self.neutral.clicked.connect(self.neutral_motion)

self.neutral.setStyleSheet("background-color:white")

self.rolling.clicked.connect(self.rolling_motion)

self.rolling.setStyleSheet("background-color:white")

self.forward.clicked.connect(self.move_forward)

self.forward.setStyleSheet("background-color:white")

self.gear_1.setChecked(True)
```

```
self.gear_2.setChecked(True)

self.gear_3.setChecked(True)

self.gear_4.setChecked(True)

self.gear_5.setChecked(True)

self.backward.clicked.connect(self.move_backward)

self.backward.setStyleSheet("background-color:white")

self.right.clicked.connect(self.move_right)

self.right.setStyleSheet("background-color:white")

self.left.clicked.connect(self.move_left)

self.left.setStyleSheet("background-color:white")

self.stop.clicked.connect(self.move_stop)

self.stop.setStyleSheet("background-color:white")

self.emergency.clicked.connect(self.emergency_situation)

self.emergency.setStyleSheet("background-color:red")

def emergency_situation(self):

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="x"

    string_encode = string.encode()
```

```
ser.write(string_encode)

self.manual.setText(str('EMERGENCY'))

self.motion_type.setText(str('EMERGENCY'))

self.reaction.setText(str('EMERGENCY'))

self.flame_1.setText(str('EMERGENCY'))

self.flame_2.setText(str('EMERGENCY'))

self.flame_3.setText(str('EMERGENCY'))

self.flame_situation.setText(str('EMERGENCY'))

self.top_lcd.display(str('0'))

self.bottom_lcd.display(str('0'))

self.region_1.display(str('0'))

self.region_2r.display(str('0'))

self.region_2l.display(str('0'))

self.region_3.display(str('0'))

self.timer_meas.stop()

self.timer_meas2.stop()

self.reaction.setText(str('Autonomous_Stopped'))

self.region_1.display(str('0'))

self.region_2r.display(str('0'))

self.region_2l.display(str('0'))

self.region_3.display(str('0'))
```

```
self.timer_meas3.stop()

self.bottom_lcd.display(str('0'))

self.timer_meas4.stop()

self.flame_1.setText(str('Flame_Stopped'))

self.flame_2.setText(str('Flame_Stopped'))

self.flame_3.setText(str('Flame_Stopped'))

def move_forward (self):

    self.manual.setText(str('Move Forward'))

    self.backward.setChecked(False)

    self.right.setChecked(False)

    self.left.setChecked(False)

    self.stop.setChecked(False)

    if self.gear_1.isChecked() == True:

        string="s"

        string_encode = string.encode()

        ser.write(string_encode)

        time.sleep(0.0001)

        string="1"

        string_encode = string.encode()

        ser.write(string_encode)

    if self.gear_2.isChecked() == True:
```

```
string="s"

string_encode = string.encode()

ser.write(string_encode)

time.sleep(0.0001)

string="2"

string_encode = string.encode()

ser.write(string_encode)

if self.gear_3.isChecked()==True:

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="3"

    string_encode = string.encode()

    ser.write(string_encode)

if self.gear_4.isChecked()==True:

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="4"
```

```
string_encode = string.encode()

ser.write(string_encode)

if self.gear_5.isChecked()==True:

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="5"

    string_encode = string.encode()

    ser.write(string_encode)

def move_backward (self):

    print("Move_Backward")

    self.manual.setText(str('Move Backward'))

    self.forward.setChecked(False)

    self.right.setChecked(False)

    self.left.setChecked(False)

    self.stop.setChecked(False)

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)
```



```
string="b"

string_encode = string.encode()

ser.write(string_encode)

def move_right (self):

    print("Turn_Right")

    self.manual.setText(str('Move Right'))

    self.forward.setChecked(False)

    self.backward.setChecked(False)

    self.left.setChecked(False)

    self.stop.setChecked(False)

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="r"

    string_encode = string.encode()

    ser.write(string_encode)

def move_left (self):

    print("Turn_Left")

    self.manual.setText(str('Move Left'))

    self.forward.setChecked(False)
```

```
self.backward.setChecked(False)

self.right.setChecked(False)

self.stop.setChecked(False)

string="s"

string_encode = string.encode()

ser.write(string_encode)

time.sleep(0.0001)

string="I"

string_encode = string.encode()

ser.write(string_encode)

def move_stop (self):

    print("Stop")

    self.manual.setText(str('Stop'))

    self.forward.setChecked(False)

    self.backward.setChecked(False)

    self.right.setChecked(False)

    self.stop.setChecked(False)

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)
```

```
string="s"

string_encode = string.encode()

ser.write(string_encode)

def linear_motion(self):

    print("Linear_Motion")

    self.motion_type.setText(str('Linear Activated'))

    self.neutral.setChecked(False)

    self.rolling.setChecked(False)

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="L" #linear_motion

    string_encode = string.encode()

    ser.write(string_encode)

def neutral_motion(self):

    print("Neutral_Motion")

    self.motion_type.setText(str('Neutral Activated'))

    self.neutral.setChecked(False)

    self.rolling.setChecked(False)

    string="s"
```

```
string_encode = string.encode()

ser.write(string_encode)

time.sleep(0.0001)

string="n"

string_encode = string.encode()

ser.write(string_encode)

def rolling_motion(self):

    print("Climbing_Motion")

    self.motion_type.setText(str(' Climbing Activated'))

    self.neutral.setChecked(False)

    self.rolling.setChecked(False)

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="c" #climbing

    string_encode = string.encode()

    ser.write(string_encode)

def top_sonar_start(self):

    if self.sonar_start.isChecked():

        self.timer_meas.start(500)
```

```
else:

    self.timer_meas.stop()

def top_sonar_stop(self):

    self.sonar_start.setChecked(False)

    self.timer_meas.stop()

    self.top_lcd.display(str('0'))

def bottom_sonar_start(self):

    if self.sonar_start.isChecked():

        self.timer_meas3.start(500)

    else:

        self.timer_meas3.stop()

def bottom_sonar_stop(self):

    self.sonar_start.setChecked(False)

    self.timer_meas3.stop()

    self.bottom_lcd.display(str('0'))

def autonom_start(self):

    if self.autonomous_start.isChecked():

        self.timer_meas2.start(500)

    else:

        self.timer_meas2.stop()

def autonom_stop(self):
```

```
self.autonomous_start.setChecked(False)

self.timer_meas2.stop()

self.reaction.setText(str('Autonomous_Stopped'))

self.region_1.display(str('0'))

self.region_2r.display(str('0'))

self.region_2l.display(str('0'))

self.region_3.display(str('0'))

def autonom(self):

    lid=self.run()

    print (lid)

    self.reaction.setText(lid)

def flame_starting (self):

    if self.flame_start.isChecked():

        self.timer_meas4.start(500)

        if GPIO.input(flame1) == 0:

            print("Fire")

            self.flame_1.setText(str('Fire'))

            self.flame_situation.setText(str('Fire Source in Right'))

        if GPIO.input(flame1) == 1:

            print("No_Fire")

            self.flame_situation.setText(str('Clear'))
```

```
        self.flame_1.setText(str('No_Fire'))

    if GPIO.input(flame2) == 0:

        print("Fire")

        self.flame_situation.setText(str('Fire Source in Front'))

        self.flame_2.setText(str('Fire'))

    if GPIO.input(flame2) == 1:

        print("No_Fire")

        self.flame_situation.setText(str('Clear'))

        self.flame_2.setText(str('No_Fire'))

    if GPIO.input(flame3) == 0:

        print("Fire")

        self.flame_3.setText(str('Fire'))

        self.flame_situation.setText(str('Fire Source in Left'))

    if GPIO.input(flame3) == 1:

        print("No_Fire")

        self.flame_3.setText(str('No_Fire'))

        self.flame_situation.setText(str('Clear'))

    else:

        self.timer_meas4.stop()

def flame_stoping(self):

    self.flame_start.setChecked(False)
```

```
self.timer_meas4.stop()

self.flame_1.setText(str('Flame_Stopped'))

self.flame_2.setText(str('Flame_Stopped'))

self.flame_3.setText(str('Flame_Stopped'))

self.flame_situation.setText(str('Flame_Stopped'))

def top_sonar(self):

    dist1=self.measure1()

    print (dist1)

    self.top_lcd.display(str(round(dist1,1)))

def bottom_sonar(self):

    dist2=self.measure2()

    print (dist2)

    self.bottom_lcd.display(str(round(dist2,1)))

def measure1(self):

    try:

        GPIO.output(GPIO_TRIGGER1, True)

        time.sleep(0.00001)

        GPIO.output(GPIO_TRIGGER1, False)

        time.sleep(0.00006)

        start = time.time()
```



```
while GPIO.input(GPIO_ECHO1)==0:

    start = time.time()

stop = time.time()

while GPIO.input(GPIO_ECHO1)==1:

    stop = time.time()

    elapsed = stop-start

    distance1 = (elapsed * 343000)/2

    return distance1

except KeyboardInterrupt:

    print("Measurement stopped by User")

    GPIO.cleanup()

def measure2(self):

    try:

        GPIO.output(GPIO_TRIGGER2, True)

        time.sleep(0.00001)

        GPIO.output(GPIO_TRIGGER2, False)

        time.sleep(0.00006)

        start = time.time()

        while GPIO.input(GPIO_ECHO2)==0:

            start = time.time()

        stop = time.time()
```

```
while GPIO.input(GPIO_ECHO2)==1:

    stop = time.time()

elapsed = stop-start

distance2 = (elapsed * 343000)/2

return distance2

except KeyboardInterrupt:

    print("Measurement stopped by User")

    GPIO.cleanup()

def run(self):

    situation="fire_seek"

    region_1 = laser.get_dist(start=180,end=372)

    region_1_array=region_1[1]

    min_distance_1=min(region_1_array)

    self.region_1.display(str(round(min_distance_1,1)))

    region_2 = laser.get_dist(start=376,end=708)

    region_2_array=region_2[1]

    min_distance_2=min(region_2_array)

    region_2r = laser.get_dist(start=376,end=542)

    region_2r_array=region_2r[1]

    min_distance_2r=min(region_2r_array)

    self.region_2r.display(str(round(min_distance_2r,1)))
```

```
region_2l=laser.get_dist(start=543,end=708)

region_2l_array=region_2l[1]

min_distance_2l=min(region_2l_array)

self.region_2l.display(str(round(min_distance_2l,1)))

region_3 = laser.get_dist(start=712,end=900)

region_3_array=region_3[1]

min_distance_3=min(region_3_array)

self.region_3.display(str(round(min_distance_3,1)))

if min_distance_1 > 1600 and min_distance_2 > 1600 and min_distance_3
> 1600:

    print("fire_seek")

else:

    print("path_plan")

if min_distance_1 >1600 and min_distance_2 < 1600 and min_distance_3 <
1600:

    print("left_corner")

    situation="left_corner"

if min_distance_1 <1600 and min_distance_2 < 1600 and min_distance_3 >
1600:

    print("right_corner")

    situation="right_corner"

if min_distance_1 <1600 and min_distance_2 < 1600 and min_distance_3 <
1600:
```

```
print("trap_mode")

situation="trap_mode"

if min_distance_1 >1600 and min_distance_2 < 1600 and min_distance_3 >
1600:

    print("obstacle")
    situation="obstacle"

if min_distance_1 <1600 and min_distance_2 > 1600 and min_distance_3 <
1600:

    print("coridor")
    situation="coridor"

if situation=="obstacle":

    if min_distance_2r > min_distance_2l:

        print("turn_right")

        string="s"

        string_encode = string.encode()

        ser.write(string_encode)

        time.sleep(0.0001)

        string="r"

        string_encode = string.encode()

        ser.write(string_encode)

    elif min_distance_2r < min_distance_2l:

        print("turn_left")
```

```
string="s"

string_encode = string.encode()

ser.write(string_encode)

time.sleep(0.0001)

string="l"

string_encode = string.encode()

ser.write(string_encode)

else:

if min_distance_1 < min_distance_3:

    print("turn_left")

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="l"

    string_encode = string.encode()

    ser.write(string_encode)

else:

    print("turn_right")

    string="s"

    string_encode = string.encode()
```

```
        ser.write(string_encode)

        time.sleep(0.0001)

        string="r"

        string_encode = string.encode()

        ser.write(string_encode)
if situation=="left_corner":

    print("turn_right")

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="r"

    string_encode = string.encode()

    ser.write(string_encode)
if situation=="right_corner":

    print("turn_left")

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="l"
```

```
string_encode = string.encode()

ser.write(string_encode)

if situation=="coridor":

    print("move_forward")

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="f"

    string_encode = string.encode()

    ser.write(string_encode)

if situation=="fire_seek":

    print("fire_seek")

    string="s"

    string_encode = string.encode()

    ser.write(string_encode)

    time.sleep(0.0001)

    string="f" # flame e gore fire seek

    string_encode = string.encode()

    ser.write(string_encode)

if situation=="trap_mode":
```

```
print("rotate_backwards")

string="s"

string_encode = string.encode()

ser.write(string_encode)

time.sleep(0.0001)

string="b" #tam geri dönüş tanımla

string_encode = string.encode()

ser.write(string_encode)

return(situation)

app = QtGui.QApplication(sys.argv)

myWindow = MyWindowClass(None)

myWindow.show()

app.exec_()
```



## RESUME

### Personel Information

Name Surname : Hilmi Saygın SUCUOĞLU

Place and Date of Birth : 16.07.1987

### Education

Undergraduate : B.Sc. degree at Uşak University

Post Graduate : M.Sc. degree at Aydın Adnan Menderes University

Foreign Languages : English

### Scientific Activities

#### A) Papers in Referred Journals

1. Bogrekci, I., Demircioglu, P., Sucuoglu, H. S., Turhanlar, O. (2019). The Effect Of The Infill Type And Density On Hardness Of 3d Printed Parts. **International Journal of 3D Printing Technologies and Digital Industry**, 3(3), 212-219.
2. Sucuoğlu, H. S., Bogrekci, I., Demircioğlu, P. (2019) Real Time Fire Detection Using Faster R-Cnn Model. **International Journal of 3D Printing Technologies and Digital Industry**, 3(3), 220-226.
3. Bogrekci, I., Sucuoglu, H. S., Demircioglu, P., Turhanlar, O. (2018). Structural Analysis of Corn Silo by Computing Vertical and Lateral Forces. **Acta Technica Napocensis Series-Applied Mathematics Mechanics and Engineering**, 61(4), 423-428. (Publication No: 5493763).
4. Bogrekci, I., Sucuoglu, H. S., Demircioglu, P., Turhanlar, O., Gultekin A. (2018). Structural and Discrete Element Analysis of Coal Mine Conveyor System. **Acta Technica Napocensis Series-Applied Mathematics Mechanics and Engineering**, 61(4), 429-434. (Publication No: 5493846).

5. Bogrekci, I., Demircioglu, P., Sucuoglu, H. S., Gultekin A., Guven, E. (2018). Computational Fluid Dynamic Analyses of Wind Turbines for Soke Region. **Acta Technica Napocensis Series-Applied Mathematics Mechanics and Engineering**, 61(4), 631-636. (Publication No: 4746937).
6. Sucuoglu, H. S., Bogrekci, I., Gultekin A., Demircioglu, P. (2018). Design, Analysis and Development of Mobile Robot with Flip-Flop Motion Ability. **Elsevier**, 51(30), 436-440. (Publication No: 4529118).
7. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P. (2018). Development of Mobile Robot with Sensor Fusion Fire Detection Unit. **Elsevier**, 51(30), 430-435. (Publication No: 4529098).
8. Sucuoglu, H. S., Bogrekci, I., Gültekin A., Demircioglu, P., Turhanlar, O. (2018). Development of Hybrid Pattern for Three Dimensional Printing Optimization. **Sigma**, 36(3) (Publication No: 4507713).
9. Demircioglu, P., Bogrekci, I., Sucuoglu, H. S., Hacıyusufoglu, A. F. (2017). Determining Surface Topography for Cylinder Liner Surfaces Using 2D Fast Fourier Transform. **Sigma** (Publication No: 3586113).
10. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Turhanlar, O. (2016). Analysis of Suspension System for 3d Printed Mobile Robot. *International Journal of Applied Mathematics*, **Electronics and Computers** (Publication No: 3277593).

#### B) Proceedings

1. Demircioglu, P., Bogrekci, I., Sucuoglu, H. S., Demir N. (2019). The Effect of Extrusion Speed on Surface Roughness for 3d Printed Parts. 4th International Congress on 3d Printing (Additive Manufacturing) Technologies and Digital Industry.
2. Demircioglu, P., Bogrekci, I., Sucuoglu, H. S., Gultekin A. (2019). The Effect of Contact and the Raft on the Surface Roughness of 3d Printed Object. 4th International Congress on 3d Printing (Additive Manufacturing) Technologies and Digital Industry.
3. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P. (2019). The Effect of Shape on the Dimensional Accuracy for 3d Printed Parts. 4th International Congress on 3d Printing (Additive Manufacturing) Technologies and Digital Industry.
4. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Gultekin A. (2017). Analysis and Optimization of Mini Excavator with Hydraulic System. 8th International Advance Technologies Symposium.

5. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Gultekin A. (2017). Analysis and Optimization of Mini Excavator with Hydraulic System. 8th International Advanced Technologies Symposium (IATS 2017).
6. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Turhanlar, O. (2017). Design, Analysis and Manufacturing of Multipurpose Cylindrical Shape Robot. 6th International Conference on Advanced Technology Sciences.
7. Bogrekci, I., Sucuoglu, H. S., Demircioglu, P., Kanlı, O. (2017). Development and Performance Analysis of Mobile Explosive Combustible Gas Leakage Detection Robot. ISPR 2017 – International Symposium for Production Research Transition to Industry 4.0.
8. Demircioglu, P., Bogrekci, I., Sucuoglu, H. S., Hacıyusufoglu, A. F. (2017). Intelligent Surface Finish Model for Industry 4.0 and Roughness Measurement For Cylinder Liner Surfaces Based on Experimental Learning Based Knowledge System. ISPR 2017 – International Symposium for Production Research Transition to Industry 4.0.
9. Demircioglu, P., Bogrekci, I., Sucuoglu, H. S., Karabulut G. (2017). Thermal and Structural Analysis of Firefighting Robot. WORLD ELECTRO MOBILITY CONFERENCE.
10. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Turhanlar, O. (2016). Analysis of Suspension System for 3d Printed Mobile Robot. International Conference on Advanced Technology Sciences, ICAT'16. In: Proceedings of the 3th International Conference on Advanced Technology Sciences.
11. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P., Acel, M. (2016). Optimization of Robotic Arm Structure for Human Rescue. International Conference on Production Research - Regional Conference Africa, Europe and the Middle East and 4th International Conference on Quality and Innovation in Engineering and Management (ICPR-AEM 2016), s. 228-232, 228-232.
12. Demircioglu, P., Bogrekci, I., Elaff Ihab A. I. M., Sucuoglu, H. S (2014). Determination of Tool Wear Using Thermal Camera. 2014 International Conference on Production Research - Regional Conference Africa, Europe and The Middle East and 3rd International Conference on Quality and Innovation in Engineering and Management (ICPR-AEM 2014), 131-134.

## C) Book Chapters

1. Bogrekci I., Demircioğlu P., Sucuoğlu H. S., Guven E., Demir N., Lecture Notes In Mechanical Engineering, Structural And Modal Analyses Of Naca 66-206 Aircraft Wing Model (2020). **Springer**, Editing by :Durakbasa N., Gencyilmaz M, Basım Sayısı:1, ISBN:978-3-030-31343-2.
2. Sucuoğlu H. S., Bogrekci I., Demircioğlu P., Turhanlar O. Advanced Mechatronics Solutions,(Design & FEA and Multi Body System Analysis of Human Rescue Robot Arm) (2016). **Springer International Publishing**, Editing by:Ryszard Jabłoński, Tomas Brezina, ISBN:978-3-319-23921-7.

## D) Projects

1. Güvenlik Ve Arama Kurtarma Amaçlı Çoklu Mobil Robot Sisteminin Geliştirilmesi, Yükseköğretim Kurumları Tarafından Destekli Bilimsel Araştırma Projesi, Yürütücü: Demircioğlu Pinar, Araştırmacı: Bögrekci İsmail, Araştırmacı: Sucuoğlu Hilmi Saygin, Araştırmacı: Gültekin Aslı, Araştırmacı: Turhanlar Oğulcan, Araştırmacı: Emanet Cihat Göktuğ, 10/03/2017 (Devam Ediyor) (ULUSAL).
2. Binalarda Enerji Verimliliği Amaçlı Kullanılacak İnsansız Hava Aracı (İHA) Sisteminin Geliştirilmesi, Yükseköğretim Kurumları Tarafından Destekli Bilimsel Araştırma Projesi, Yürütücü: Bögrekci İsmail, Araştırmacı: Demircioğlu Pinar, Araştırmacı: Akkoyun Fatih, Araştırmacı: Sucuoğlu Hilmi Saygin, Araştırmacı: Gültekin Aslı, Araştırmacı: Turhanlar Oğulcan, 10/03/2017 (Devam Ediyor) (ULUSAL).
3. Nanoteknoloji Ölçüm Laboratuvarının Altyapısının Oluşturulması Ve Teknik Teknik Olmayan Biyolojik Yüzeylerin Karakterizasyonlarının Yapılması, Yükseköğretim Kurumları Tarafından Destekli Bilimsel Araştırma Projesi, Araştırmacı, 24/03/2014 - 24/03/2015 (ULUSAL).
4. Sera Bitki Ve Ürünlerinin Kalitelerini Belirleyen Hiperspektral Görüntüleme Sisteminin Geliştirilmesi, Yükseköğretim Kurumları Tarafından Destekli Bilimsel Araştırma Projesi, Yürütücü: Aydoğan Yüksel, Yürütücü: Demircioğlu Pinar, Araştırmacı: Bögrekci İsmail, Araştırmacı: Sucuoğlu Hilmi Saygin, 16/02/2018 (Devam Ediyor) (ULUSAL).
5. Havacılık Uzay Ve Hızlı Ulaşım Araç Yapılarında Kullanılan Kompozit Malzemelerin Titreşim Yüklü Altındaki Yorulma Ömürlerinin Ve Sönümleme Değerlerinin İncelenmesi, Yükseköğretim Kurumları Tarafından Destekli Bilimsel Araştırma Projesi, Araştırmacı, 24/03/2014 - 24/03/2015 (ULUSAL).

**Contact**

E-mail : [hilmisucuoglu@adu.edu.tr](mailto:hilmisucuoglu@adu.edu.tr)

Date :13/01/2020

