DEVELOPMENT OF AN AUTONOMOUS LAWN MOWER WITH MINIMALIST HARDWARE APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SERKAN ÇİÇEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

FEBRUARY 2014

Approval of the thesis:

DEVELOPMENT OF AN AUTONOMOUS LAWN MOWER WITH MINIMALIST HARDWARE APPROACH

submitted by **SERKAN ÇİÇEK** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Süha ORAL
Head of Department, **Mechanical Engineering** ______________

Assoc. Prof. Dr. E. İlhan KONUKSEVEN
Supervisor, **Mechanical Engineering Dept., METU** ______________

Assist. Prof. Dr. A. Buğra KOKU
Co-supervisor, **Mechanical Engineering Dept., METU** ______________

Examining Committee Members:

Prof. Dr. Tuna BALKAN
Mechanical Engineering Department, METU ______________

Assoc. Prof. Dr. E. İlhan KONUKSEVEN
Mechanical Engineering Department, METU ______________

Assist. Prof. Dr. A. Buğra KOKU
Mechanical Engineering Department, METU ______________

Assist. Prof. Dr. Kıvanç AZGIN
Mechanical Engineering Department, METU ______________

Mechanical Engineer Dr. Ali Emre TURGUT
The Katholieke Universiteit Leuven ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Serkan ÇİÇEK

Signature :

ABSTRACT

DEVELOPMENT OF AN AUTONOMOUS LAWN MOWER WITH MINIMALIST HARDWARE APPROACH

ÇİÇEK, Serkan
M.S., Department of Mechanical Engineering
Supervisor : Assoc. Prof. Dr. E. İlhan KONUKSEVEN
Co-Supervisor : Assist. Prof. Dr. A. Buğra KOKU

February 2014, 138 pages

Starting from early 2000's domestic robots have been taking their place in our daily lives. Today, numerous products are globally available on the domestic robotics market. Among many domestic robot types, robotic cleaners and lawn mowers take the lead in this competition, with their success for reducing undesired house chores.

Like in every other product, the price-performance ratio is the most significant evaluation metric for a consumer while buying an autonomous robotic cleaner or an autonomous lawn mower (ALM). The "performance" in this context, mainly denotes the navigation success of the robot. Therefore, it is becoming more and more crucial for developers to offer best navigation performance with the lowest price in the market.

In this thesis, ALMs are chosen as the field of study. A minimalist, low-cost navigation approach has been proposed in order to conform customer and market requirements. For this scope, a differentially driven ALM, operating with predefined geometrical coverage patterns has been designed, developed, improved and manufactured. All of these phases are given in detail in this study.

The use of the localization methods that demand expensive hardware is avoided. In order to present a low-cost product, dead-reckoning method is used with only wheel encoders. In addition, an enhanced position correction technique especially for ALMs is developed. In this technique, odometric error compensation is applied, based on the identification of mowed and non-mowed lawn areas. The main objective of this technique is to provide an improvement for the coverage performance of the ALM.

After completion of the design, manufacturing and system integration tasks, performance tests have also been performed. These physical indoor and outdoor tests have been performed on different terrains with various coverage patterns. These tests revealed both the navigation characteristics of the ALM and the coverage performance improvement of the proposed position correction technique.

Test results showed that the proposed position correction technique achieved a significant improvement over sole odometry for outdoor coverage. It is concluded that the proposed technique can be an alternative compared to relatively expensive outdoor navigation methods.

Keywords: Autonomous Lawn Mower, Autonomous Mobility, Autonomous Navigation, Autonomous Outdoor Coverage, Autonomous Coverage Performance, Dead-reckoning, Slippage-based Error Compensation, Trajectory Tracking, Low-cost Localization Technique

ÖZ

MİNİMALİST DONANIM YAKLAŞIMI İLE OTONOM ÇİM BİÇME ROBOTU GELİŞTİRİLMESİ

ÇİÇEK, Serkan
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. E. İlhan KONUKSEVEN
Ortak Tez Yöneticisi : Yrd. Doç. Dr. A. Buğra KOKU

Şubat 2014 , 138 sayfa

Evsel otonom robotlar, 2000'li yılların başından itibaren günlük yaşantımızdaki yerlerini almaya başlamışlardır. Günümüzde, evsel robot pazarı içerisinde sayısız ürün bulunmaktadır. Tüm bu çeşitlilik içerisinde, otonom temizlik ve çim biçme robotları, ev işlerini hafifletmekteki başarıları sebebi ile pazarda en ön sırada gelmektedirler.

Otonom temizlik veya çim biçme robotu satın alma aşamasındaki bir tüketici için en önemli değerlendirme kriteri, her ticari üründe olduğu gibi fiyat-performans oranıdır. Buradaki "performans" kavramı, evsel robotun seyrüsefer başarısını göstermektedir. Dolayısıyla tüm üreticiler için en iyi seyrüsefer performansını en uygun fiyata sunmak son derece önemlidir ve gün geçtikçe önem kazanmaktadır.

Tez kapsamında, çalışma alanı olarak otonom çim biçme robotları seçilmiş, tüketici ve pazar ihtiyaçlarına cevap verebilecek bir minimalist seyrüsefer yaklaşımı önerilmiştir. Bu kapsamda, ön-tanımlı geometrik desenler ile hareket edecek, diferansiyel sürüş kabiliyetine sahip bir otonom çim biçme robotu tasarlanmış, geliştirilmiş ve üretilmiştir. Tüm bu süreçler, tez kapsamında detaylı olarak sunulmuştur.

Pahalı donanımlar gerektiren seyrüsefer tekniklerinin kullanımından kaçınılmıştır. Uygun fiyat kriterini sağlamak adına, tekerlek enkoderleri ile odometrik seyrüsefer gerçekleştirecek bir yönteme karar verilmiştir. Ayrıca, otonom çim biçme robotlarına özel bir pozisyon bilgisi düzeltme tekniği geliştirilmiştir. Odometrik seyrüsefer kaynaklı pozisyon hatasını azaltmayı amaçlayan bu teknik, biçilmiş ve biçilmemiş çim alanların tespit edilmesine dayanmaktadır. Tekniğin temel amacı, otonom çim biçme robotunun operasyonel tarama performansını arttırmak adına bir iyileştirme sağlamaktır.

Tasarım, imalat ve sistem entegrasyon görevlerinin tamamlanmasının ardından, performans testleri de gerçekleştirilmiştir. Bu iç ve dış ortam fiziksel testleri, çeşitli zemin şartları ve çeşitli hareket desenleri için gerçekleştirilmiştir. Bu testler ile birlikte, otonom çim biçme robotunun seyrüsefer karakteristiği ve geliştirilen pozisyon hatası düzeltme tekniğinin tarama performansı üzerindeki etkisi incelenmiştir.

Yapılan testler sonucunda, dış ortam tarama oranının arttırılması için geliştirilmiş pozisyon bilgisi düzeltme tekniğinin, odometrik seyrüsefer performansını iyileştirdiği gözlemlenmiştir. Sonuç olarak, tez kapsamında önerilen bu tekniğin, görece pahalı dış ortam seyrüsefer yöntemlerine bir alternatif olabileceğine karar verilmiştir.

Anahtar Kelimeler: Otonom Çim Biçme Robotu, Otonom Seyrüsefer, Otonom Dış Ortam Taraması, Otonom Ortam Tarama Performansı, Hesabi Seyrüsefer, Kayma-Tabanlı-Hata Düzeltme, Güzergah İzleme, Düşük Maliyetli Lokalizasyon Tekniği

To each and every person that makes me who I am...

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisor Assoc. Prof. Dr. E. İlhan KONUKSEVEN and my co-supervisor Assist. Prof. Dr. A. Buğra KOKU for their guidance, advice, criticism, encouragement and insights throughout the research.

I would also like to thank my dear friend and colleague Ardıç KAROL for all of his support and self-sacrifice. The technical assistance of Salih Can ÇAMDERE is gratefully acknowledged. I would also like to thank Yelda DEMİR for her assistance on grammar and writing.

Finally, I would like to thank my lovely family, all my friends, and my love Tuba for their understanding, help, encouragement and faith in me.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

c	Speed of wave propagation
C_r	Rolling resistance
d	Distance travelled
D	Drive wheel diameter
e_p	Position error
e_θ	Heading error
h	Desired height of the area to be mowed
l	Length parameter of the Line command
L	Distance between two traction wheels
m	Mass
N	Number of ticks per revolution
r	Radius
r_d	Radius parameter of the Arc command
R	Radius of a traction wheel
$R_{Abs_{Des}}$	Absolute desired ratio
t	Time of flight
v_l	Velocity of the left motor
v_r	Velocity of the right motor
$v_{max,flat}$	Maximum robot velocity for inclined surface
w	Desired width of the area to be mowed
W	Weight
x	X-coordinate
$\dot{x}$	Velocity in x-direction
y	Y-coordinate
$\dot{y}$	Velocity in y-direction
z	Z-coordinate
$\triangle\alpha$	Sweep Angle parameter of the Arc command
$\triangle\theta$	Rotation Angle parameter of the Rotate command
$\triangle v$	Delta velocity
θ	Heading angle
θ_{inc}	Maximum inclination
$\dot{\theta}$	Change of heading angle by the time

LIST OF ABBREVIATIONS

AFRL	Air Force Research Laboratory
ALM	Autonomous Lawn Mower
CAD	Computer Aided Design
CAN	Controller Area Network
CCW	Counter-Clockwise
CMOS	Complementary Metal–Oxide–Semiconductor
CNC	Computer Numerical Control
CPR	Counts Per Revolution
CPU	Central Processing Unit
CW	Clockwise
DC	Direct Current
dGPS	Differential Global Positioning System
DC	Direct Current
DMA	Direct Memory Access
DR	Dead Reckoning
FOV	Field of View
GLONASS	Global Navigation Satellite System
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
ICE	Internal Combustion Engine
IFR	International Federation of Robotics
IMU	Inertial Measurement Unit
ION	Institute of Navigation
IR	Infrared
IRED	Infrared Emitting Diode
LIDAR	Light Detection and Ranging
LI-ION	Lithium - ION
LI-PO	Lithium - Polymer
MBD	Multi Body Dynamics
MEMS	Microelectromechanical Systems
NI-MH	Nickel - Metal Hydride
PD	Photo Diode

RADAR	Radio Detection and Ranging
RF	Radio Frequency
RTK	Real-Time Kinematics
SLAM	Simultaneous Localization and Mapping
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
TOF	Time of Flight
TTL	Transistor-Transistor Logic
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

CHAPTER 1

INTRODUCTION

1.1 Brief History of Autonomous Robots

Autonomous mobile robot concept is a relatively-new topic when whole robotics is considered. The term "robota" was first used in a play by a Czech author Karel Capek in 1921. Ethymologically, the word "robota" means "servant" in Czech language. The word "robot" was first introduced in English by Rossum's Universal Robots. They conceived a future of fully-automated workers.

Considering the historical examples of autonomous mobile robots, it is necessary to mention Grey Walter's mobile robot, "Tortoise". Grey Walter (University of Bristol, UK) was building three-wheeled mobile robots in 1950. Tortoise uses two basic sensors like a light and a touch sensor. It has propulsion motors, a steering motor for actuation, and a two vacuum tube analog computer for computation. By using such a simplistic approach, Walter has shown that those robots can exhibit complex behaviors. Those robots are later called "Machina Speculatrix" because of their tendency on exploring environment. [6]

The first mobile robot controlled by vision is the Shakey robot. It was built in Stanford University in 1969. There is a TV camera, a range finder sensor, and bumping detectors on this robot. There are two electric motors for traction, and one caster wheel for balance. They were using an on-board computer for computation. [17] [14]

After the Shakey project, Stanford Cart project began in 1967. After many of

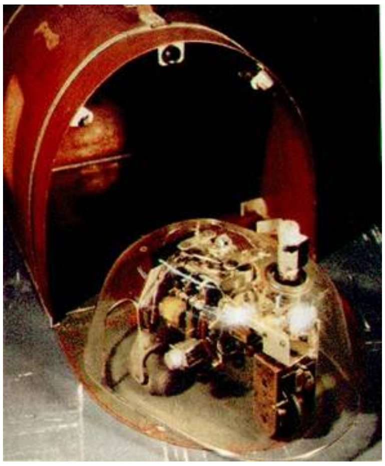


Figure 1.1: Elsie Robot (Machina Speculatrix)

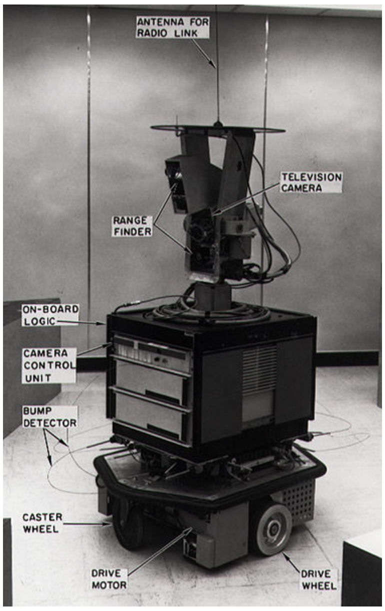


Figure 1.2: Shakey Robot

revisions, the robot had the final shape with stereo vision by Hans Moravec in 1977. It was successfully crossed a chair-filled room without any human intervention in 1979. The underlying stereo-vision technology was taking pictures of the environment by the use of a linear rail on top of the cart and relaying them to a computer. The computer gauged the distance in this manner.

Hilare Mobile Robots created a breaktrough in mobile robotics. The Hilare family was developed in LAAS in years 1977 to 1992. First Hilare robot, Hilare-I was developed in 1977. Two driving wheels and a caster wheel are used in actuation. Multibus is used as the bus in that robot with four Intel 802286 processors. In communication, a 9600 baud serial radio modem is used. The robot had odometer, 16 US sensors, and a laser range finder as the sensor equipment.

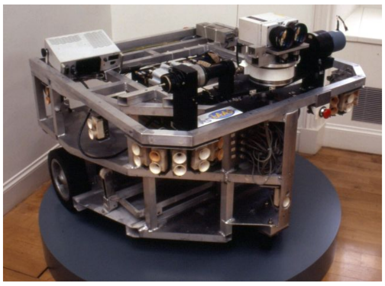


Figure 1.3: Hilare-I mobile Robot

After Hilare I, LAAS was developed Hilare II robot with new actuation, processor, and operating system approaches. They used two drive wheels, and four idle wheels in the new robot. Four processors used in this robot are Motorola 68040 and Motorola PPC750. This robot has VxWorks 5.3.1 operating system. For sensing, odometric sensors are used in addition to 32 sonar range sensors, 2D laser range finder, a pan/tilt/zoom color camera and two pan/tilt/zoom black and white cameras.

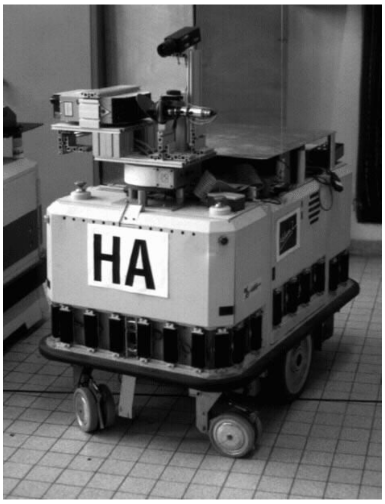


Figure 1.4: Hilare-II Mobile Robot

In 1992, Hilare II-bis was developed. In that robot, a robotic manipulator is added while reducing the weight to 400 kg. There are no major changes except the cart added and the rotary encoder attached to the connection in order to sense the rotations effectively.

After 1992, there are many robots have been developed. However, Hilare robots

Table 1.1: Taxonomy of Domestic Robots

Categories of Domestic Robots	**Types**	**Examples**
Entertainment Robots	-	LEGO Mindstorms, Robosapien v2
Security Robots	-	WowWee, Spykee
Collaborative Robots	Vacuum Cleaner Robots	Roomba, Navibot
	Lawn Mower Robots	Automower, Robomower
	Window Cleaning Robots	Windoro
	Pool Cleaner Robots	Verro, Aquabot
	Ironing Robots	Dressman
Personal Robots	Mobile Robots	AIBO
	Physical Assistant Robots	Wakamuru, Care-O-bot

are still representing a model for nonholonomic device control and navigation.

1.2 Domestic Robots

Within a vast robotic ecosystem, domestic robot can be defined as an autonomous robot, used mainly for house chores or entertainment. Although the first commercial products are launched in 80's, domestic robots have seriously been taking their place in our daily lives. Many types of domestic robots such as vacuum cleaners, lawn mowers, gutter cleaners, window cleaners, pool cleaners, etc. are available in the market today. Taxonomy of these domestic robots is given in Table 1.1. [13]

1.2.1 Entertainment Robots

Entertainment robots which are also called "Toy Robots" are specially designed for multiple levels of interaction to human and objects. As the main aspect of these type of robots, children can enjoy and learn with these robots at the same time.

Figure 1.5: Mindstorms, LEGO

Figure 1.6: Robosapien v2, WowWee

1.2.2 Security Robots

Main aspect of these type of robots is home surveillance. They continuously gather information from inside or outside of the house, check for any invasion or unauthorized entry situation, and report back to user or the police by using 802.11 wireless network.

Figure 1.7: Rovio, WowWee

1.2.3 Collaborative Robots

"Collaborative" context in this classification is different from the well-known "collaborative robotics" concept. In this definition, collaborative type of domestic robots help and assist people in their daily chores. This type of domestic robots is the globally the most popular one. Vacuum cleaners, lawn mowers, window cleaners, pool cleaners, and even ironing robots take their place in this segment. There are some other subtypes like gutter cleaners which are not mentioned in this work. [7]

Figure 1.8: Roomba, iRobot

Figure 1.9: Robomower, Friendly Robotics

Figure 1.10: Windoro, ALM Robotics

Figure 1.11: Aquabot, Aquaproducts

Figure 1.12: Dressman, Siemens

1.2.4 Personal Robots

This type of robots mainly performs aiding or caring actions. They can assist especially to elder or disabled people for walking, feeding, load lifting, shop listing, etc. They also provide companionship by talking or by music. Some of them are used for therapeutic purposes.

Figure 1.13: AIBO

It is always a dream of everyone for robots to do undesired house chores like cleaning, dish washing, ironing, etc. Some part of society thinks this progress of

Figure 1.14: Wakamaru

robotic development was inevitable and the other part refuses it. But it seems like the statistics verifies the thoughts of optimist futurists.

The leading company in robotic vacuum cleaner market, iRobot, got into the market in 2002. They have sold over 8 million units of Roomba and Scooba worldwide till 2012. [19]

Regarding International Federation of Robotics (IFR), about 2.5 million service robots for personal and domestic use were sold in 2011, which is 15% more than the previous year's number. Considering the sales characteristics, IFR projected that the sales of all types of domestic robots could reach almost 11 million units in the period of 2012-2015, with an estimated value of 4.8 billion USD. [1]

Greatful to continuously developing technology, today, domestic robots can think, evaluate, manage, decide and execute their missions very successfully. They can even interact with people. From now on, it is not a science-fiction thought that robots help, assist and care about all possible needs of mankind in the close future.

1.3 Scope of the Thesis

The main scope of this thesis is to build an affordable autonomous lawn mower with limited hardware. The study mainly presents the design, development, manufacturing and test phases for a real ALM. It is intended for the end product (i.e. the physical working prototype) to have a potential to be turned into a commercial product after some modifications.

The primary objective is to determine and enhance robot's outdoor coverage performance for varying geometrical and environmental parameters, despite the lack of any internal or external position correction references.

Within the content of this thesis, an ALM is intended to be designed, developed and manufactured. All of the mechanical design, software development and system integration tasks are aimed to be done from scratch, and aimed to be presented in detail in the scope of this thesis.

In order to make an improvement on outdoor coverage performance of the robot, a novel enhancement technique for position correction is proposed. It is aimed to be used for odometric error compensation. The key idea in this method is to determine the mowed and non-mowed areas. In the scope of this thesis, the evaluation of this proposed technique and the rest of whole system is also aimed to be done by indoor and outdoor tests.

1.4 Outline of the Thesis

First chapter is the introduction to the subject.

In Chapter 2, literature survey is presented and common techniques for autonomous land navigation are revealed. Main commercial ALMs, their specifications and capabilities are also discussed comprehensively. Moreover, ALMs are investigated from an academic point of view.

Chapter 3 contains the main work and reveals it in system development stages hierarchically. It starts with the problem definition and continues with the systematical approaches performed to determine the system requirements, which eventually shaped conceptual design of the ALM. Critical design considerations, hardware selections, electrical design, mechanical design, software architecture, mathematical modelling and corresponding calculations are also given in detail.

In Chapter 4, manufacturing details of the ALM has been presented. In addition, a brief information about the system integration is given.

Chapter 5 introduces the test scenarios and aspects, gives details about the test setup and results. Comparisons for physical indoor and outdoor test results are also given. A detailed conclusion that evaluates test outcomes is also given in this chapter.

Finally, Chapter 6 summarizes and concludes work done within the thesis. Some possibilities for future work are also discussed.

CHAPTER 2

LITERATURE SURVEY

2.1 Techniques for Autonomous Land Navigation

From a general perspective, navigation is known as a field of study that focuses on the process of monitoring and controlling the movement of a craft or vehicle from one place to another. The field of navigation includes four general categories: land navigation, marine navigation, aeronautic navigation, and space navigation. All navigation techniques involve locating the navigator's position compared to the known locations or patterns. Therefore, it can be said that navigation is based on localization.

Regarding that this study is based on a mobile ground robot, only land navigation and corresponding techniques are investigated is this section. Before going deep into detail about navigation techniques, it is suitable to discuss the types of sensors used in mobile ground robots. [11] [9]

2.1.1 Sensor Classification

Sensors can be classified by using two functional axes, which are: proprioceptive/exterceptive and passive/active. [17]

Proprioceptive sensors measure the internal values of the system, which is the ALM in our case. These internal values can be counted as motor speed, wheel load, voltage, current, etc.

Exteroceptive sensors collect the environmental information by measuring the parameters like distance, light intensity, etc. These measurements must be interpreted or converted to get meaningful information.

Passive sensors measure the environment by collecting the information getting through the sensor. To examplify, some specific types of cameras, microphones or temperature sensors can be counted as passive sensors.

Active sensors are the best sensor segment in terms of controllability. They emit energy and collect the reaction of the environment. However, these sensors must be used professionally since they have energy bounds. Signals emitted by the environment can affect and disturb these type of sensors.

A taxonomy done on sensors is given in Table 2.1. [17] [5]

Investigating the main sensor types which can be used in this study gives the conclusions as follows.

2.1.1.1 Wheel/Motor Sensors

These sensors are used to measure the internal state of the system. By the aid of this type of sensors, it is possible to find the dynamic and kinematic characteristics of the system.

Optical encoders are widely used in wheeled mobile robots to calculate the position, velocity and acceleration. Since optical encoders are proprioceptive sensors, their measurements are in vehicle carried reference frame. Because of this, using these sensors in robot localization problems requires much labor for calibration and disturbance rejection. There are two significant problems in wheeled autonomous mobile robots, which are the drift and the slippage. Therefore, using these sensors in such robots requires a global reference for position correction.

Considering from mechanical aspect, it is seen that this sensor is basically a light chopper which counts the discontinuities on the sensor by emitting and receiving the light beams. The changes on the receiver side are counted in operation and this count gives the rotation in a discrete time interval. Resolution of these

Table 2.1: Taxonomy of Sensors

General Classification **(typical use)**	**Sensor** **Sensor System**	**PC or EC**	**A or P**
Tactile Sensors	Contact Switches, bumpers	EC	P
	Optical barriers	EC	A
	Proximity sensors	EC	A
Wheel/motor sensors	Brush encoders	PC	P
	Potentiometers	PC	P
(wheel/motor speed and position)	Synchros, resolvers	PC	A
	Optical encoders	PC	A
	Magnetic encoders	PC	A
	Inductive encoders	PC	A
	Capacitive encoders	PC	A
Heading sensors	Compass	EC	P
	Gyroscope	PC	A
	Inclinometers	EC	P/A
Grond-based beacons	GPS	EC	A
	Active optical or RF beacons	EC	A
	Active US beacons	EC	A
	Reflective beacons	EC	A
Active ranging	Reflectivity sensors	EC	A
	US sensor	EC	A
	LIDAR	EC	A
	Optical triangulation (1D)	EC	A
	Structured light (2D)	EC	A
Motion/speed sensors	Doppler radar	EC	A
	Doppler sound	EC	A
Vision-based sensors	CCD/CMOS camera	EC	P
	Visual ranging packages		
	Object tracking packages		

sensors is measured in terms of counts per revolution (CPR). Multiplication of the counts with the value of CPR gives the angular displacement within a specific time interval.

Quadrature encoders are mostly used in robotics. This type of encoders give both the direction and the number of counts since they produce two square-waves with an appropriate phase difference. This logic is schematically expressed in Figure 2.1.

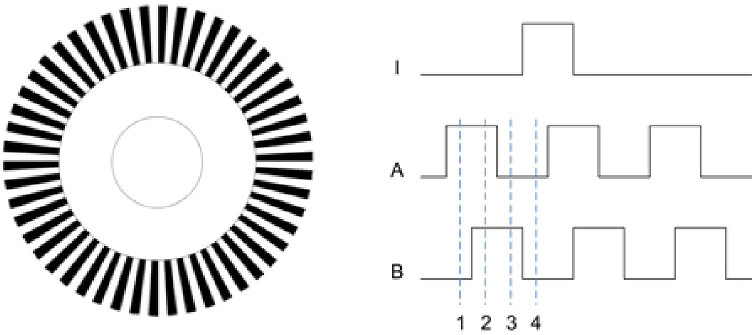


Figure 2.1: Optical Encoder Working Principle

Knowing the channel characteristics, the direction of revolution can be known by counting both channels. Counting ticks, the angular speed of the joint attached to the encoder can be calculated. The combinations of this logic is given in Table 2.2.

Table 2.2: Encoder State Table

State	**Channel A**	**Channel B**
s_1	High	Low
s_2	High	High
s_3	Low	High
s_4	Low	Low

2.1.1.2 Heading and Orientation Sensors

Heading and orientation sensors can be both proprioceptive and exterceptive. Inertial Measurement Units (IMU) are used in robotics for heading and orientation sensing. They consist of an accelerometer, a gyroscope and in some types, a compass. Navigation done by using these type of sensors is called dead-reckoning.

Accelerometers are used to determine the acceleration in a specified axis. In kinematics control point of view, this tool is convenient to use.

Gyroscopes are used to determine the heading of the device by measuring the rotation in a specified axis. These tools can be found as 1D, 2D or 3D types. By using this tool, both angular displacement, velocity and acceleration can easily be calculated.

Compasses are exteroceptive sensors. They are used to determine the localization of the system with respect to a predefined point by using the magnetic flux generated by that station. The sensor determines the direction of the magnetic flux and then by observing the change of flux, it senses the vehicle state.

There are two main compass types which are flux-gate and hall-effect compasses. Hall-effect compasses can be used in applications where a wire or a semiconductor is carrying a current. Due to the magnetic flux change caused by that semiconductor, there will be a voltage difference and hall-effect sensors give that difference as a result of the change in reading. Although these sensors are inexpensive and the hall-effect sensor is accurate to determine the localization, they have range and noise problems. Even the magnetic disturbance caused by electrical circuits may lead to undesired results. Flux gate compasses on the other hand, have a completely different working principle. They determine the change of the direction of the alternating current as well as the strength of that alternating magnetic field. Although this sensor type has an improved precision and accuracy, they are not strictly reliable and robust due to the disturbance caused by the magnetic field of the Earth.

2.1.1.3 Ground-Based Beacons and Satellites

In order to solve the navigation problems of outdoor autonomy and mobility, using beacons is an efficient and effective approach. In this manner, using sensors on robot and beacons localized in the environment leads to localize the vehicle precisely and accurately. In addition, this localization type can see the natural landmarks and can easily calibrate itself. It increases the accuracy of the localization and mapping. By using efficient sensors while constructing the beacons, localization can be done within five centimetres diameter in a few kilometres square of a field.

By using satellites (i.e. the Global Positioning System (GPS)), navigation of a system can be done globally. This sensor type has the same working principle with the ground-based beacon systems. It measures the position by transceiving the signals. Since the signals in GPS travel long distances compared to the ground-based beacons, localization accuracy reduces and the precision may reduce to several metres of diameter range. However, this method can still be an efficient way of localization especially for large vehicles.

2.1.1.4 Active Ranging Sensors

Active ranging methods are becoming the most popular sensing methods, due to the ease of application and relatively lower costs. One of the major benefit is these sensors provide direct measurements. This leads to estimate the positions of the objects in the sensor skirt instantaneously. Most mobile robots rely on these active ranging sensors. However, these sensors are used as a part of the sensor set since they are sensing the instantaneous obstacles.

There are two different active ranging methods which are mostly used. One of them is the laser range finder type of sensors and the other on is ultrasonic sensors. [4]

Time-of-flight active ranging makes use of the speed of sound or the electromagnetic wave propagation. In general, the travel distance of the electromagnetic wave of the sound is given by;

$$d = c \times t \tag{2.1}$$

where;
d : travel distance (usually round-trip)

c : speed of wave propagation
t : time of flight

The main criteria of the time-of-flight active ranging quality is measured by the velocity of the emitted signal from the sensor. The reality here underlies the common knowledge of the quality difference between laser range finders and ultrasonic sensors.

2.1.2 Classification of Navigation Techniques for Ground Robots

Since most of the concepts and techniques used for autonomous ground robot navigation are interlocked and nested, it is not easy to classify them separately. However, a general classification can be made for ground robots. There are three main navigation techniques; Trilateration/Multilateration, Dead-Reckoning and Landmark-Based (Vision-Based) Techniques.

2.1.2.1 Trilateration / Multi-Lateration Techniques

In geometry, trilateration is the process of determining absolute or relative locations of points by measuring distances, and using the geometry of circles, spheres or triangles. On contrary to triangulation, it only consists of distance measurements, and does not involve the measurement of angles. If the coordinates of a point can be found using three geometrical objects, this technique is called trilateration. If references are more than three, it is called multilateration.

The coincidence of three circles algorithm used for the trilateration procedure can be shown in Figure 2.2.

The derivation of trilateration is represented in 3D as follows;

$$r_1^2 = x^2 + y^2 + z^2 \tag{2.2}$$

$$r_2^2 = (x-d)^2 + y^2 + z^2 \tag{2.3}$$

$$r_3^2 = (x-i)^2 + (y-j)^2 + z^2 \tag{2.4}$$

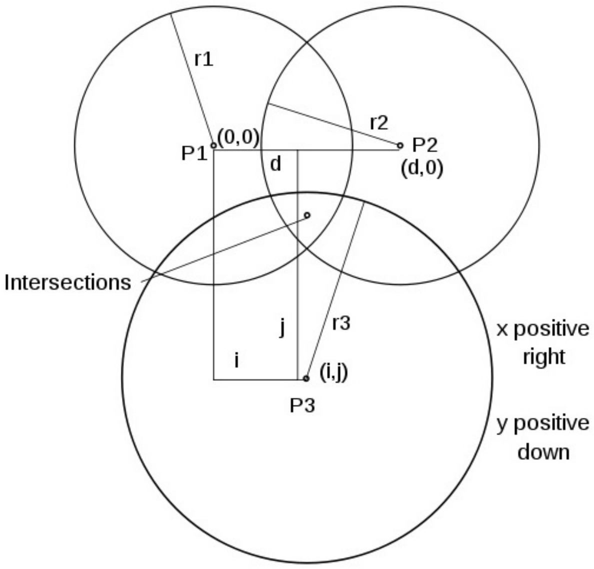


Figure 2.2: Trilateration Circles

After the subtraction is done for x;

$$x = \frac{(r_1^2 - r_2^2 + d^2)}{2d} \tag{2.5}$$

Having found the x coordinate, y coordinate is found by

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x - i)^2 + j^2}{2j} \tag{2.6}$$

Although, the only two-dimensional planar case is considered in this study, the elevation is found by using the formula below;

$$z = \sqrt{r_1^2 - x^2 - y^2} \tag{2.7}$$

There are two major localization methods for ground robots, using trilateration/multilateration technique; Satellite and Beacon localization.

Satellite Localization is a well-known method used in Global Positioning Systems (GPS and GLONASS). GPS devices localize an earth-fixed reference position by using four or more satellites. In this method, the GPS device acts as a receiver and the satellites as transmitters. Each satellite continually transmits messages that include the time that the message was transmitted and satellite position at time of message transmission. By using this information, GPS device computes its position within a bounded accuracy. The accuracy depends on the parameters such as weather condition, obstacle existence around the GPS device, and the number of transmitting satellites.

Figure 2.3: A typical GPS Receiver

The utilization of the GPS is easy and this method is comparably inexpensive. However, it determines one's position within a range of 10-15 m radius. As mentioned above, this accuracy also depends on the temperature and the humidity conditions of the environment.

The use of sole GPS localization can be meaningful in very large regions that the defined accuracy can be applicable. However, in many robotic applications, an accuracy of 10 m is far away from the desired limits.

This accuracy can be enhanced by using an improved method called Differential GPS (dGPS). DGPS uses a network of fixed, ground-based reference stations to broadcast the difference between the positions indicated by the satellite systems and the known fixed positions. These broadcasted signals are called correction signals. The illustration of dGPS method is given in Figure 2.4.

By using one fixed beacon, whose earth-fixed position is exactly known, the positional accuracy can be improved to 1-5 meters. The disadvantage of this method is that it uses commercial beaconing, where the correction signals are supplied from. Therefore the operation of dGPS is very costly. It is mainly used for tracking freighters.

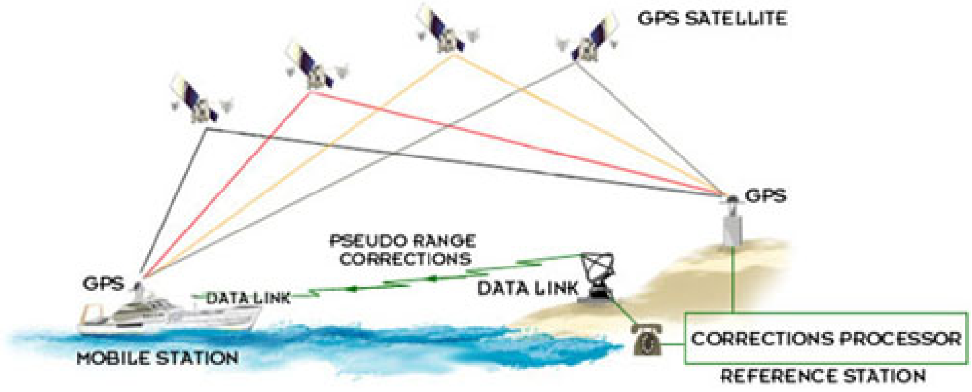


Figure 2.4: Schematics for dGPS Method

A further improvement has been developed for dGPS which is called dGPS with RTK (Real Time Kinematics). In this technique, many fixed beacons send real time correction signals to the receiver. Therefore, positioning accuracy is improved to 1-80 cm, depending on the weather conditions and existing obstacles.

Figure 2.5: dGPS with RTK Receiver

Even though the dGPS with RTK is very accurate, use of RTK technique is very expensive. The initial hardware costs are about 15.000 USD. Additionally, 1.500 USD must be paid annually in order to get real time correction services.

Beacon Localization is another method which uses triangulation or trilateration techniques. Similar to satellite localization, this method uses at least three transmitting fixed-beacons to determine the position of the receiver. As the positions of beacons are known, relative position of the receiver within a beacon frame can be calculated.

Among others, Radio Beaconing and Ultrasonic Beaconing are the most common methods, used for mobile robot localization. As it can be understood from the names, electromagnetic waves are sent and received in radio beaconing, whereas high speed sound waves are used in ultrasonic beaconing.

In both methods, the time passed between departure from transmitter and arrival to receiver is measured. The electromagnetic radio signals travel at the speed of light and ultrasonic waves travel at the speed of sound. Measuring the time between departure and arrival, multiplication with constant velocity leads as the distance. This is known as time-of-flight (TOF) computation.

This technique seems reasonable but it also has some limitations. This method is also affected by weather conditions. In addition, increasing the distance between transmitters and receiver, the accuracy changes adversely. Therefore, it is widely used as an indoor navigation method for the ground robots. A schematic for this method is given in Figure 2.6.

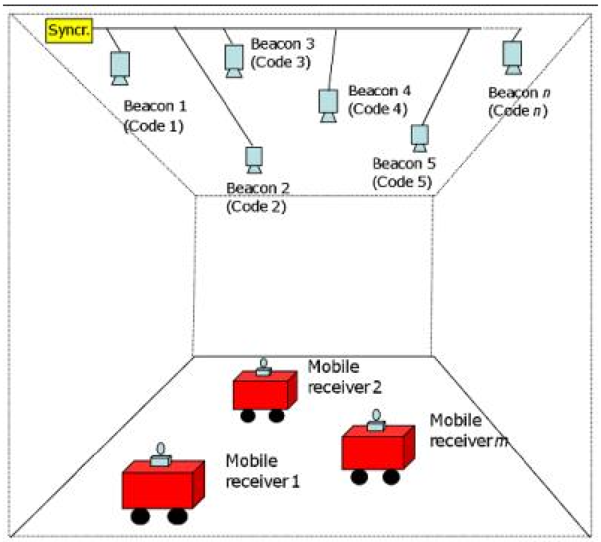


Figure 2.6: Schematics for Beacon Localization

The advantage of this method is the indoor accuracy, which is about 30 cm. However, it brought some difficulties for beacon identification and the investment costs can be high depending on the number of active beacons.

2.1.2.2 Dead-Reckoning Techniques

Dead-Reckoning (DR) is the method of calculating current position from a previously determined position by using velocity and time measurements. DR is widely used in almost all navigational purposes. In robotics, there are two main sub classifications of this technique.

Odometric Navigation utilizes odometry. The word is composed of the Greek words hodos (travel) and metron (measure) etymologically. Odometry is the use of data from moving sensors to estimate the change in position over time. Most common odometric sensors are rotary encoders. By mounting rotary encoders to the drive wheels of a mobile robot, and counting the encoder "ticks" during operation, the relative position of the robot can be calculated.

A major disadvantage of the odometric navigation is that it is inherently erroneous. For a wheeled ground vehicle, there will always be slippage and drift. As each estimation of position is based on the previous one, errors are cumulatively added. On the other hand, odometric sensors are very cheap.

Inertial Navigation for robots benefit from accelerometers and gyroscopes to continuously calculate the position, orientation, and the velocity of a moving object by dead-reckoning method without the need for any external reference. Therefore, they are comparably precise and they are not erroneous. They can be used for correction mechanisms for many robotic applications.

With the recent advances in micro-electromechanical systems (MEMS), small and lightweight inertial navigation systems are available for robotic applications today. However, they are still a little bit expensive.

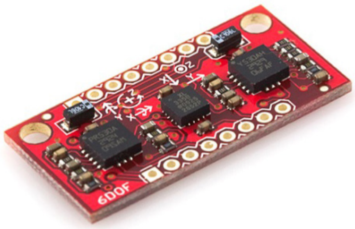


Figure 2.7: MEMS Inertial Navigation Hardware

2.1.2.3 Landmark-Based (Vision-Based) Navigation Techniques

Landmark-Based Navigation techniques determine robots position relative to an external reference. They are widely used in mobile robots for mapping and obstacle avoidance. As a general disadvantage, it is hard to use only this technique for localization, since the exact positions of external references must be known for accurate localization. They also serve as perfect correction mechanisms for DR procedures.

Landmark-Based Navigation may also called Vision-Based Navigation, since it needs a vision for determining its position along references. This vision can be supplied by various optical or magnetic sensors.

In this technique, a robot can follow a prior path (path guidance) or creates its own path with its visionary sensors.

The simplest and well-known application as the example usage of landmark-based navigation is line-following robots. By using the optical, infrared or magnetic sensors, robots can follow a prior path within a decent accuracy.

More sophisticated robots use LIDARs (Light Detection and Ranging), RADARs (Radio Detection and Ranging), Scanning Range Finders or Cameras for distance measurements with external references. This external reference can even be the real environment.

Figure 2.8: A Laser Scanning Range Finder

RADARs and LIDARs are extremely expensive hardware which also requires a great computational power to process the gathered data. Therefore, they are mainly used for mapping purposes instead of localization in mobile robotics field.

2.2 Recent Autonomous Lawn Mowers

Until today, million units of domestic robots have already been sold in the market. One of the most popular products in house chores is ALM. ALMs take the second biggest share after autonomous vacuum cleaners globally. In this section, commercial ALMs are compared in various aspects.

2.2.1 Commercial Autonomous Lawn Mowers

Starting from the mid 90's, a necessity arise for robots to take care of the house chores as the time spent on those chores decreases inevitably. First commercial autonomous products sold globally for house chores are lawn mowers. Major commercial ALMs are investigated in this section from a broad perspective. Navigation techniques, used equipment and specifications of these robots are compared in detail.

Approximately 10 main ALM manufacturers, with more than 25 products, are available on the global market today. A detailed comparison that reveals the top products of the top 5 manufacturers are given in Table 2.3.

All of these commercial ALMs are internal battery powered. Almost all of them have 2 DC motors for differential drive and 2 additional castor wheels for balance. In order to lower the manufacturing costs, almost all of the body components are made by plastic materials with injection molding or vacuum forming processes.

All of the products use mulching technique for lawn mowing. In mulching, grass is cut into very small clippings (about a millimeter) that are buried in the roots of the lawn, where they decompose and act like a natural fertilizer. This technique results in a healthier and better-looking lawn, and major advantage is that it eliminates the need to collect and remove the clippings.

On contrary to the conventional man-powered lawn mowers, most of the ALMs do not have a single large cutting blade. Instead, they use multiple smaller blades for this purpose.

Table 2.3: Products Comparison

Company	Husqvarna	Friendly Robotics	Kyodo America	Bosch	John Deere
Model	Automower - 220 AC	Robomower - RL 2000	Lawnbott - LB 2150	Indego	Tango E5
Coverage $[m^2]$	1800	2000	2250	1000	1800
Image					
Max. Inclination [°]	19	15	27	19	19
Battery Type	NiMH	Lead - Acid	LI-ION	LI-ION	LI-ION
Charging Time	45 min	20 h	3 h	50 min	90 min
Mowing Time	45 min	2.5 h	3 h	50 min	60 min
Cutting System	3 Razor Blades	3 Blades	4 Blades	3 Centrifugal Blades	1 x 4 Sided Blade
Cutting Height $[mm]$	20 - 61	25 - 82.5	19 - 70	20 - 60	19 - 102
Cutting Width $[mm]$	221	214	305	260	N/A
Sound $[dB]$	64	90	N/A	75	69
Dimensions $[mm]$	711 x 533 x 305	889 x 660 x 318	560 x 420x 254	700 x 520 x 300	775 x 535 x 360
Weight $[kg]$	10	39	10.5	11.5	15.3
Self-Charging	Yes	Yes	Yes	Yes	Yes
Coverage Pattern	Random	Random	Random	Par. Swath and Random	Random and Spiral
Bordering	Perimeter Wire	Perimeter Wire	Perimeter Wire	Perimeter Wire	Perimeter Wire
Obstacle Avoidance	Yes	Yes	Yes	Yes	Yes
Online Price	USD 2400	USD 2500	USD 2600	USD 2100	USD 3000

The cutting height is manually adjustable in all ALMs. In average, lawn height can be kept around 20 100 mm, depending on the ALM model.

The navigation techniques of all the commercial ALMs are quite similar. They all have a charging station from where they start their operations. There is a limited IR communication between the charging station and the robot used for the robot to find charge station when the batteries have consumed. They have no expensive visionary sensors like LIDARs or cameras to inspect the environment. Even though some of them use inertial sensors (i.e. accelerometers or digital gyroscopes) to make the localization more accurate, they all operate "blind" with their odometric wheel encoders (i.e. they count encoder ticks for calculating traveled distances). This lack of "wisdom" causes the robots to operate mainly on random patterns instead of operating on predefined geometrical patterns.

In order to work within the desired area, all of the commercial ALMs use the same type of method. This method consists of a perimeter wire and an inductive sensor. A perimeter wire is simply a harmless low-current passing wire. The user connects both ends of this wire to the charging station to obtain a closed current loop. The wire is buried only a few millimeters deep in the ground along the perimeter of the desired area. During their operations, robots sense the borders with the aid of the inductive sensors located at the bottom of their body. When they arrive to a border, they act with respect to their predefined motion algorithms.

Figure 2.9: Perimeter Wire

Static or dynamic obstacle avoidance is achieved by using ultrasonic (US) sensors or bumper switches in all commercial products.

The average online price of an ALM is about 2.500 USD without shipping fees and taxes. Regarding that the average lifetime of a commercial ALM is 3-5 years, this investment seems to be unnecessary for many potential customers.

ALM technology has proceeded within the last decade, but there is still a lot to do in order to make these robots more intelligent and more affordable.

2.2.2 Autonomous Lawn Mowers from Academic Perspective

Only the commercial ALM attributes and capabilities have discussed in the previous section. Besides their commercial potentials, ALMs are perfect research platforms for students and academic staff, when outdoor navigation is considered. Mapping of unknown environments, accurate instant localization, path-planning and obstacle avoidance are the major topics for these types of researches.

ION (Institute of Navigation) Satellite Division and AFRL (Air Force Research Laboratory), organizes an ALM competition in US annually. The purpose of this competition is to design and operate an autonomous, unmanned lawn mower using the art and science of navigation to rapidly and accurately mow a field of grass. [15]

ION Autonomous Lawn Mower Competition is held in two classes; basic and advanced mowing competitions. Both competitions are aimed to award the winner who performs the "best cut". The "best cut" definition here stands for a minimum 75% grass-cut of the given area.

Basic mowing competition field is a rectangle with one static obstacle. The advanced mowing competition field has a non-square shape and it contains both static and dynamic obstacles as well as a border such as a fence.

As mentioned before, the purpose of this competition is to use art and science of navigation. Therefore, participant robots capabilities are different compared to the conventional ALMs. As mentioned in the previous section, all of the commercial ALMs use perimeter wire in order not to pass to neighbor's yard or not to ruin any flowerbed. Almost all of them operate with random coverage patterns. None of them use any external correction reference. Many of them have just bumb-switches for obstacle avoidance.

All participant robots for ION competition use wheel encoders for dead-reckoning. Many of them use expensive LIDAR (Light Detection and Ranging) equipment for mapping and obstacle avoidance challenges. Perimeter wire type bordering is not allowed in this competition, so the robots must calculate border positions throughout its operation; otherwise the team will get a penalty.

All participant robots use GPS for position correction. As it was mentioned in previous sections, this is an unreliable method unless the dGPS system is used. Moreover, the teams also use visionary sensors (i.e. cameras and/or inertial sensors) for corrections.

With this expensive hardware usage, the average budget of a team is about 20,000 USD. Most of the budget is spent for hardware in this competition.

Some of this ION competition rules and descriptions are taken as a reference in this study. It has been stated for all ALMs that;

- Lawnmowers shall be autonomous and unmanned and shall not be remotely controlled during the competition.
- Lawnmowers shall have a maximum speed of 10 km/hour for safety reasons.
- Lawnmowers shall not exceed 2 meters in any dimension.
- Lawnmower movement shall be accomplished through direct contact with the ground.
- Lawnmowers shall demonstrate the ability to mow a predetermined path void of any obstacles.
- Teams shall have a maximum of 20 minutes to cut the field.
- Mowers shall be designed to operate in any weather condition.

in the competition. [19]

Table 2.4: Sensor Comparison

Sensor Hardware Description	**Price [USD]**
dGPS Unit	30,000.00
Encoder	170.00
IMU	2,400.00
LIDAR	6,000.00 (estimated)

As mentioned before, there are two types of fields that correspond for basic and advanced mowing competitions.

The basic field is 10 x 15 m^2 with 2 m buffer zone in each side. Field contains a static obstacle such as a standard size plastic container. The drawing of the field is given in Figure 2.10.

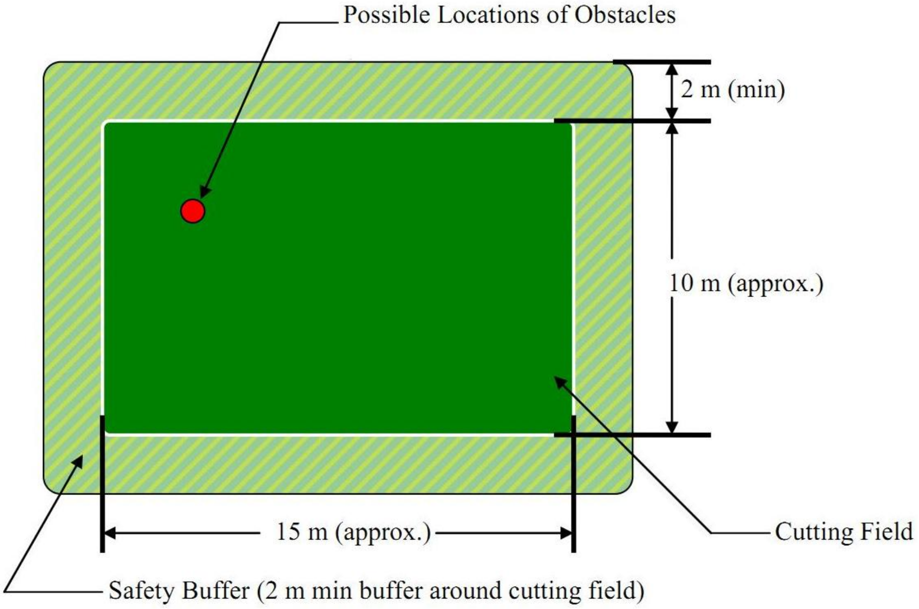

Possible Locations of Obstacles
2 m (min)
10 m (approx.)
15 m (approx.)
Cutting Field
Safety Buffer (2 m min buffer around cutting field)

Figure 2.10: Mowing simulation on the basic field

The advanced field is a little bit tricky. It has an irregular shape with at least one non-perpendicular side, a flowerbed and a dynamic obstacle. The drawing of the advanced field is given in Figure 2.11.

The major sensory hardware configuration for autonomous navigation of the winner team in 2012 ION Autonomous Lawn Mower Competition is given in Table 2.4. [18]

Even for a sponsored organization, the prices are too expensive as seen. Table 2.4

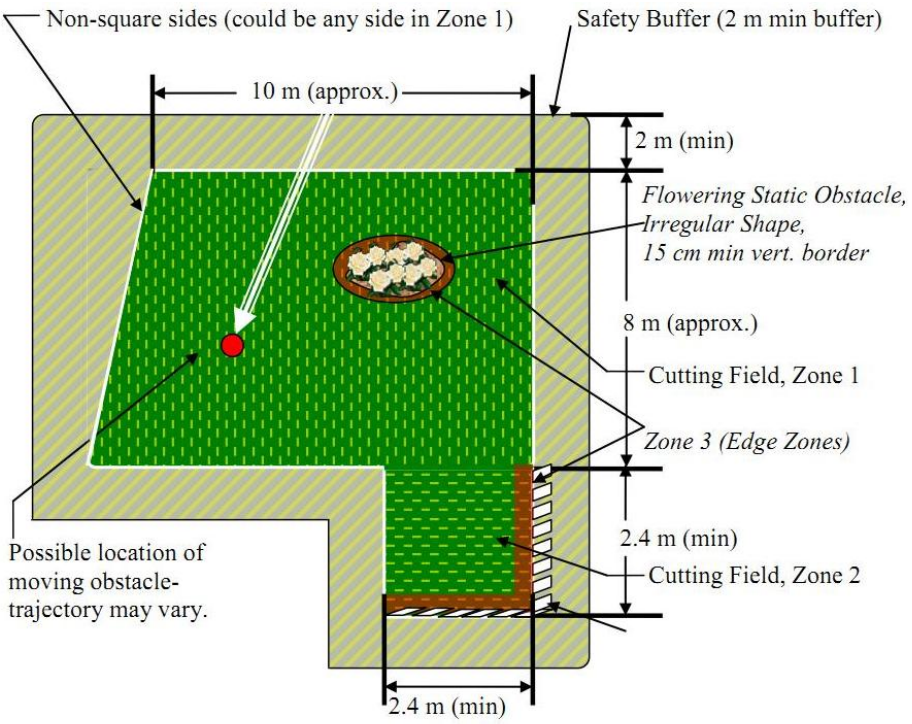


Figure 2.11: Mowing simulation on the advanced field

is nearly the standard hardware today for SLAM (Simultaneous Localization and Mapping) research applications for ground robots. At the end, it is impossible to convert this system to a commercial product.

The details of the low-cost outdoor navigation method proposed in the scope of this thesis is discussed in further chapters.

CHAPTER 3

SYSTEM DEVELOPMENT

Section 2 presents different localization and mapping techniques that can be used for ground robots while navigating in an unknown outdoor environment. In this section, these techniques are comprehensively discussed, and the proposed autonomous navigation technique is revealed. Moreover, concept selections for other general requirements have also been mentioned.

3.1 Problem Definition

This study mainly presents the design, development, manufacturing and test phases for a real ALM. All these design phases are completed and a physical working prototype is obtained, which has a potential to be turned into a commercial product after some modifications.

A real product must satisfy the real environmental and economic considerations within the sector. Generally, seven main properties can be expected from an ALM in a customer's perspective. These properties can be listed in the order of priority: an ALM must mow efficiently with high quality cutting, be mobile, be autonomous, be affordable, move in the desired area, avoid obstacles, and visually has to be attractive. These properties are the requirements and considered as the design criteria in this study.

Mowing efficiently with quality and being visually attractive are comparably easy-to-do tasks. The remaining requirements can be combined and defined as affordable autonomous mobility. The main evaluation metric for autonomous mobility is the navigation performance. Consequently, the primary problem in this work can be defined as autonomous outdoor navigation.

3.2 Concept Selection

3.2.1 Lawn Mowing Technique

All of the commercial ALMs use mulching technique, which is an agricultural term for lawn mowing. In mulching, the grass is cut into very small clippings that are buried in the roots of the lawn. They decompose and act like a natural fertilizer in time. This technique results in a healthier and better-looking lawn. The major advantage of this technique is that it eliminates the need to collect and remove the lawn clippings.

In a conventional man-powered lawn mower that does not use mulching, lawn clippings are collected in a basket. The volume of a conventional basket is about 40 L. It can carry the clippings of an approximately 50 m2 lawn yard. After mowing this area, the user must empty the basket. The mass of a basket full of lawn clippings is about 6 kg. It is found inconvenient for an ALM to collect lawn clippings since they add extra weight to the system, which means a decrease on the operation time inefficiently. Moreover, the main idea of autonomous lawn mowing is regaining the time spent on lawn mowing to the user. This goal cannot be achieved if the user periodically empties the lawn basket of a robot.

Therefore, mulching technique is selected for lawn mowing in this study. For grass cutting, a single cutting blade with multiple cutting edges is used for energy efficiency and easy-cleaning purposes.

3.2.2 Mobility

Most of the conventional lawn mowers use internal combustion engines (ICE) for their cutting systems. ICEs can create large amounts of energy, however they are inefficient and they have high running costs for this type of relatively-low-power-required applications. The user must also refill the fuel almost before every operation, which is an undesired task.

Man-powered lawn mowers that use electricity for "mower motor" usually do not use internal batteries. Electric is supplied from an external source by using a power cable. This cable always rambles to the mower or to the user, which is also undesired.

Considering mobility, using an internal source of energy is almost compulsory. Since the desired mowing area is limited, internal batteries are decided to be used in the proposed design. Given that the mowing system will be electrically powered, the driving mechanism is also preferred to be fed by the same source. Ni Mh batteries are used since they are affordable compared to LI ION or LI PO batteries.

3.2.3 Autonomous Outdoor Navigation Technique

Almost all the commercial ALMs in the market operate randomly. The major disadvantage of operating with random patterns is that the robot's coverage performance is random. Considering ALMs, a randomly operating type robot may not be able to mow some specific regions of the garden even if the garden is geometrically too simple to cover. The user must intervene in the motion of the robot by a remote controller or manually move it to a non-mowed area. It also behaves differently in every single operation, so the grass quality can be harmed. In a random operation, the robot may cross over the same region again and again during its operation. Operating with this type of a coverage algorithm is not energy efficient. Regarding these disadvantages, instant localization is found to be compulsory, even if the robot operates randomly.

Considering GPS as a localization technique, it is relatively inaccurate. Since some improved techniques like dGPS and dGPS-RTK can locate the position of the receiver accurately, they are extremely expensive for such a product. Moreover, all of them are affected by weather conditions and signal strength. Therefore, satellite navigation is also found to be disadvantageous and expensive for such a system.

Another trilateration/multilateration technique, beacon localization can be considered, which performs well indoors. However, it is not suitable for outdoor applications due to the complexity of the application of this system for comparably large regions. It is inconvenient to install beacons for every garden, even if the required number of beacons is low. This means the method becomes more expensive and besides needs too much effort. Consequently, beacon localization has certain disadvantages.

Landmark-Based navigation brings some difficulties with itself inherently. For sole localization, exact positions of the external references must be known. It is almost impossible to apply this method to every single garden. LIDARs can be very useful for such an application, however commercial LIDARs are extremely expensive.

Path guidance methods are also found useless, since it is nearly impossible to create prior paths for every garden.

In this study, in order to maintain the simplicity, odometric dead-reckoning method is used for localization. Although the odometric dead-reckoning method is vulnerable to inevitable slippage-based error deposition, it is cheap and simple. In our case, coverage performance is more important than localization accuracy. In order to increase the coverage performance of the robot, a novel enhancement technique for position correction is proposed. The key idea in this method is to determine the mowed and non-mowed areas using IR switch sensors. Other details will be explained in further sections.

3.2.4 Bordering Technique

Considering the gardens as the field of operation, all of them are unique regarding their geometry and fencing. Therefore, it is found to be necessary to utilize a generic bordering technique for the ALM operation. Using the perimeter wire seems to be logical in this manner. In this technique, a low current passing wire is buried only a few millimeters deep in the ground. The current (i.e. the magnetic field) on the wire can be detected by the inductive sensors located at the bottom of the robot. This technique is decided to be used for bordering unknown environments (i.e. operation fields) in this work.

3.2.5 Obstacle Avoidance Technique

Every single garden may have many static or dynamic obstacles. In order not to collide with these obstacles, some "sensing" equipment such as infrared (IR) sensors, US sensors, LIDARs, RADARs, etc. can be used. LIDAR and RADAR sensors are highly accurate, but they are extremely expensive for our purpose. IR sensors are quite cheap but they are incapable of detecting transparent obstacles, since the infrared beams can pass through and cannot be reflected back from the medium. In this study, ultrasonic sensors, which detect transparent and non-transparent mediums, are decided to be used for obstacle avoidance.

3.2.6 Visual Attraction

All commercial ALMs are machine-looking robots that many people can find unattractive. In this study, esthetic concerns played an important role in the design procedure. It is aimed to produce a visually appealing and attractive robot. Since the ALM is planned to be operate mainly in domestic gardens, a pet-like shape is given to the robot. In order to emphasize its fast operation, and due to its best fitting to a grass area, the robot is decided to be rabbit-shaped.

3.3 Hardware Architecture

3.3.1 Hardware Selection

Selection of major hardware components of the ALM and their selection criteria are briefly presented in this section. These selections are important since they directly affect all design phases and the resultant performance of the system. Functionality, availability and price are chosen as the main considerations for

Table 3.1: Parameters Used for Traction Motor Selection

Parameter	**Value**
Mass, m $[kg]$	30.0
Rolling Resistance, C_r	0.1
Maximum Robot Velocity for Inclined Surface $v_{max,flat}$	3.6
Maximum Inclination θ_{inc}	10.0
Drive Wheel Diameter, D $[mm]$	155.5

all hardware selections.

3.3.1.1 Selection of Drive Motors

It has been stated that two differentially driven DC motors are decided to be used for robot's motion. For selection of these "drive motors"; torque, power and speed requirements have been calculated in advance. Figure 3.1 represents the free body diagram of the system.

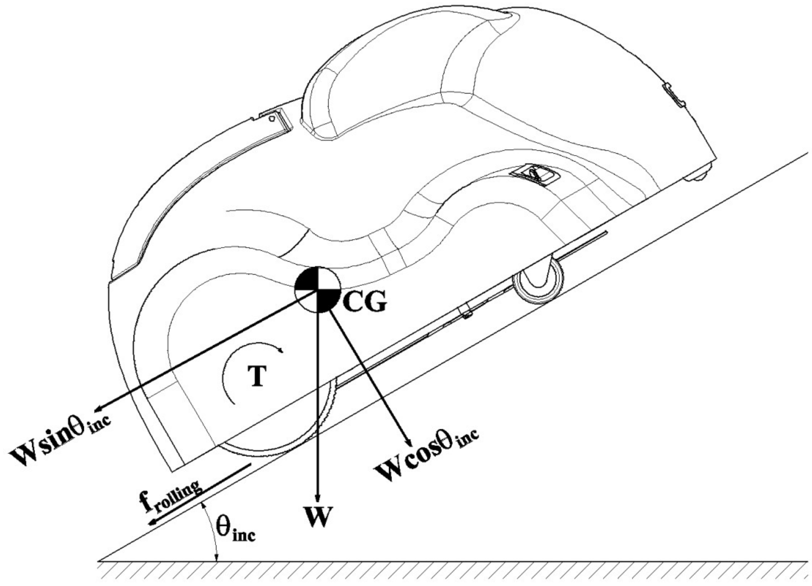


Figure 3.1: Free Body Diagram of the Autonomous Lawn Mower

Assuming that there is no slippage at the beginning of the motion, the following calculations are made for a single wheel to satisfy the requirements considering the two-wheeled system. Assumed parameters for initial calculations are also given in Table 3.1.

The net force acting on the direction of motion is;

$$F_{net} = \frac{W \times \sin\theta}{2} + f_{rolling} \tag{3.1}$$

where $f_{rolling}$ is the rolling force acting on a single wheel. It can be represented as;

$$f_{rolling} = \frac{W \times \cos\theta \times C_r}{2} \tag{3.2}$$

The minimum torque and power requirements of a single motor is;

$$T_{min} = F_{net}\frac{D}{2} \tag{3.3}$$

$$P_{min} = T_{min} \times \omega \tag{3.4}$$

where ω is the desired angular velocity of the drive wheel for maximum robot speed.

For the given parameters in Table 3.1; the minimum torque, power, and angular velocity requirements are calculated as;

$$T_{min} = 3.1Nm \tag{3.5}$$

$$P_{min} = 33.37W \tag{3.6}$$

$$\omega_{min} = 123.28rpm \tag{3.7}$$

Due to relatively high torque requirement, a gear-motor is decided to be used for the driveline. The major properties of the selected drive gear-motor are given in Table 3.2.

Table 3.2: Properties of Selected Traction Motor

Model	Intecno EC035.240 + P42.2.25
Gearbox Type	Planetary
Gear Ratio	1:25
Voltage [V]	24 VDC
Rated Current [A]	2.6
Peak Current [A]	4
Rated Power [W]	35
Rated Rotational Speed [rpm]	140
Rated Torque [Nm]	3.38

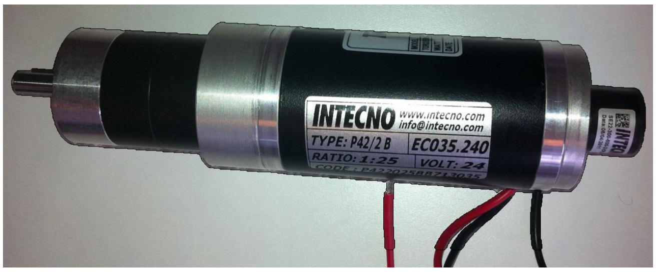


Figure 3.2: Selected Drive Motor

3.3.1.2 Selection of Mower Motor

It is hard to determine the required power and torque for a lawn mowing process since the shear strength of a grassy area depends on various parameters. For that purpose, some product comparisons have been made on both conventional and autonomous lawn mowers. At the end, it is concluded that a minimum rotational speed of 2500 rpm is sufficient for lawn mowing operation.

The properties of the selected "mower motor" are given in Table 3.3.

Table 3.3: Properties of Selected Mower Motor

Model	Kormas 671-143-23
Voltage [V]	24 VDC
Rated Current [A]	3.0
Peak Current [A]	7.0
Rated Power [W]	140
Rated Rotational Speed [rpm]	2800

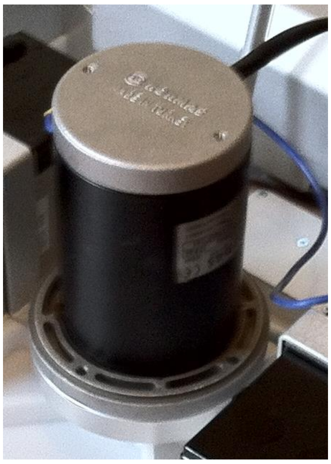


Figure 3.3: Selected Mower Motor

Table 3.4: Properties for Selected Gear-motor Driver

Model	Dimension Engineering Sabertooth 2 x 12A
Input Voltage [V]	6-24 VDC
Output Current [A]	12 A Per Motor (Continuous)

3.3.1.3 Selection of Motor Drivers

Two "drive motors" and one "mower motor" with their drivers are used in the system. The selected 24 V DC "drive motors" can consume up to 4 A per motor. A sophisticated and cheap driver is chosen to drive both "drive motors". Properties of the selected motor driver are given in Table 3.4.

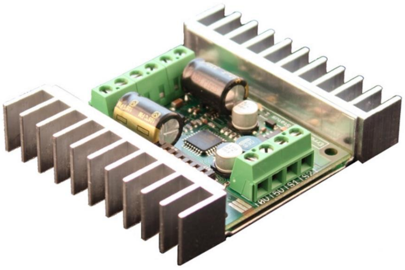


Figure 3.4: Selected Dual Traction Motor Driver

Similarly, a reliable, low cost motor driver has been chosen to drive the "mower motor". Its properties are given in Table 3.5.

Table 3.5: Properties for Selected Mower Motor Driver

Model	Pololu 24V12
Input Voltage [V]	24 VDC
Output Current [A]	12 (Continuous)

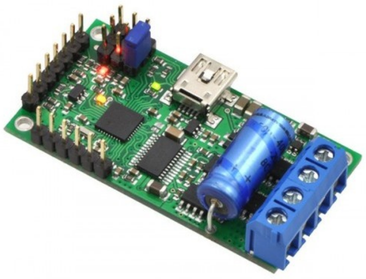


Figure 3.5: Selected Mower Motor Driver

3.3.1.4 Selection of Internal Batteries

The desired operation time for an ALM is thought to be about 1 hour. Except the "drive" and "mower" motors, the power consumption is found to be negligible for the determination of the power budget.

The current to-be-consumed by the "drive motors" is;

$$I_{rated} = 2 \times 2.6Ah \quad (3.8)$$

$$= 5.2Ah \quad (3.9)$$

The current to-be-consumed by the "mower motor" is;

$$I_{rated} = 3.0Ah \quad (3.10)$$

Therefore for all motors, a total of 8.2 A supplement is required for 1 hour operation. It can be concluded that a current value of 6.15 A is required for a 45 minute operation.

As presented through the Sections 3.3.1.1 to 3.3.1.3, the supply voltage for all of the major motion components is selected as 24 V DC. Therefore the battery output voltage must be equal to 24 V DC.

Table 3.6: Properties for Selected Internal Battery

Model	Yuasa NP 7-12
Battery Type	Ni-Mh
Output Voltage [V]	12 VDC
Output Current [Ah]	7 (Continuous)

Table 3.7: Properties for Selected Wheel Encoders

Model	Intecno se22
Type	Optical, Hollow Shaft
Output Channels	Quadrature + 1 Index Channel
Encoder Resolution [CPR]	300
Input Voltage [V]	5 VDC
Frequency [Hz]	60

In order to achieve 24 V DC and 6.15 Ah output, Ni-Mh batteries given in Figure 3.6 are concluded to be connected in parallel. Properties for selected internal batteries are as follows.

While two parallel connected batteries can supply the desired 24 V DC, they can also propagate the operation to 45 minutes (7 Ah > 6.15 Ah).

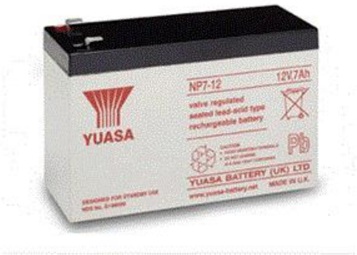


Figure 3.6: Selected Battery

3.3.1.5 Selection of Sensors

Encoders One of the major reasons for the selection of specified "drive motor" is its integrated encoder, which is used for odometric calculations. Two identical optical, hallow-shaft type, quadrature wheel encoders are used in the system. The device provides one, two or two plus index incremental encoding square wave signal outputs. The component is very suitable for speed and incremental position detection. Properties of the selected encoder are presented in Table 3.7.

Figure 3.7: Selected Encoder

Table 3.8: Properties for Selected Ultrasonic Sensors

Model	Parallax PING
Communication	Positive TTL Pulse
Range	2 cm - 3 m
Input Trigger [νs]	2 - 5
Echo Pulse	5 115 νs - 18.5 ms
Input Voltage [V]	5 VDC
Rated Current [mA]	30

Ultrasonic Sensors For obstacle avoidance, US sensor is used since it is relatively inexpensive and accurate. The selected sensor is a US distance sensor, which provides precise, non contact distance measurements from about 2 cm to 3 m. The PING sensor works by transmitting an ultrasonic burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated. It is very easy to connect to microcontrollers requiring only one I/O pin for both TX and RX. The properties for the selected ultrasonic sensor are given in Table 3.8.

Figure 3.8: Selected Ultrasonic Proximity Sensor

Table 3.9: Properties for Selected IR Sensors

Model	Sharp GP2Y0D810Z0F
Type	Digital Output
Range [mm]	5 - 100
Input Voltage [V]	2.7-6.2 VDC
Rated Current [mA]	5

Table 3.10: Properties for Selected IR Switches

Model	Sharp GP1A57HRJ00F
Type	Optical IC (OPIC) Output
Gap [mm]	10
Input Voltage [V]	5 VDC
Rated Current [mA]	15

Infrared Sensors Infrared sensors are decided to be used for the fall-off and the tilt-over detection. For this purpose, Sharp GP2Y0D810Z0F model IR sensor is chosen. It is a distance measuring sensor unit composed of an integrated combination of PD (photo diode), IRED (infrared emitting diode) and signal processing circuits. The properties for the selected ultrasonic sensor are given in Table 3.9.

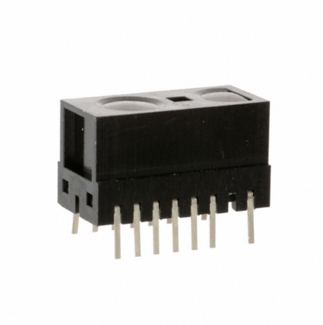


Figure 3.9: Selected Infrared Proximity Sensor

Infrared Switches IR switches are decided to be used for identification of mowed and non mowed areas within the garden. For this purpose the SHARP GP1A57HRJ00F model IR switch sensor is chosen. It is a standard transmissive photo-interrupter with an opposing emitter and detector, providing non-contact sensing. The emitter and detectors are inserted in a case, resulting in a through-hole design, which is suitable for this type of application.

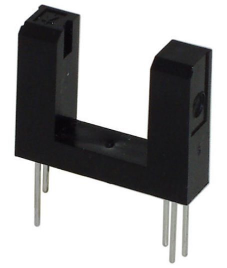


Figure 3.10: Selected Infrared Switch

3.3.1.6 Selection of Controllers

The entire control algorithm developed for this prototype robot is run on the Arduino Due. Arduino is a single-board microcontroller used for building electronics projects. It consists of a physical programmable circuit board and an Integrated Development Environment (IDE), used to write and upload the code to the physical board.

Arduino Due hardware consists of an open-source hardware board designed around a 32-bit Atmel ARM. Pre-programmed into the on-board microcontroller chip is a boot loader that allows uploading programs into the microcontroller memory without needing a chip programmer. The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is a 32-bit ARM core that can outperform typical 8-bit microcontroller boards. A 32-bit core, that allows operations on 4 bytes wide data within a single CPU clock. It has a CPU clock speed of 84 Mhz, 96 KBytes of SRAM, 512 KBytes of Flash memory for code, and a a DMA controller that can relieve the CPU from doing memory intensive tasks.

Using a single controller, both timer and encoder interrupts are handled, however it is realized that nested interrupts occur in the system which cause missing ticks from the encoders. The software architecture needs a main timer interrupt routine for all the process to run successfully. Therefore, it is planned to handle encoder tick -hardware- interrupts by a separate controller.

One of those two identical controllers is used just for counting encoder ticks.

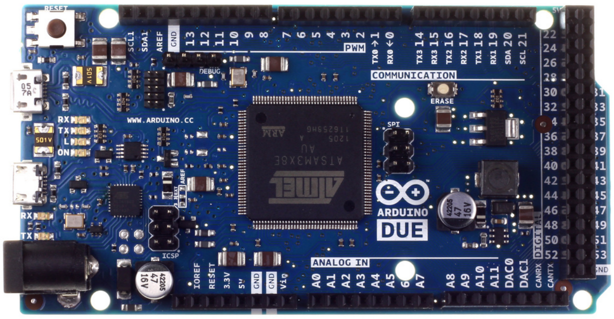


Figure 3.11: Arduino Due - Front View

Actually, it has an additional task such as transmitting encoder tick count information to the other controller so that it can calculate the heading angle, position and wheel velocities on each and every time interval. For communication purposes, Arduino Due has a number of facilities for communicating with a computer, another Arduino or other microcontrollers, and different devices like phones, tablets, cameras and so on. The SAM3X provides one hardware UART and three hardware USARTs for TTL (3.3V) serial communication. For communicating two Arduino Due's, options like Hardware Serial Ports, SPI Communication Ports, CAN Communication Protocol (not yet supported by Arduino APIs), TWI Communication Ports are available. Serial Port is not fast enough for our application since we have pretty much higher pulse rates for encoders. Within the scope of the thesis, TWI Communication Protocol is preferred because of its readily available, easy-to-use "Wire Library" for Arduino.

3.3.2 Electrical Design

First of all, a power management circuit is available in the system. It is simply managing the charging system which uses an external adaptor to charge internal batteries. On the user control panel, there is a power switch that has a third (Charge) position in addition to main ON/OFF function. Charge level can be seen on the external adaptor, at least for this version of the robot. Additionally, power management circuit distributes the main power to two motor drivers and two controller boards. All other circuits and components on the system work with TTL level. The main circuit schematic of the electrical system can be seen in Figure 3.12.

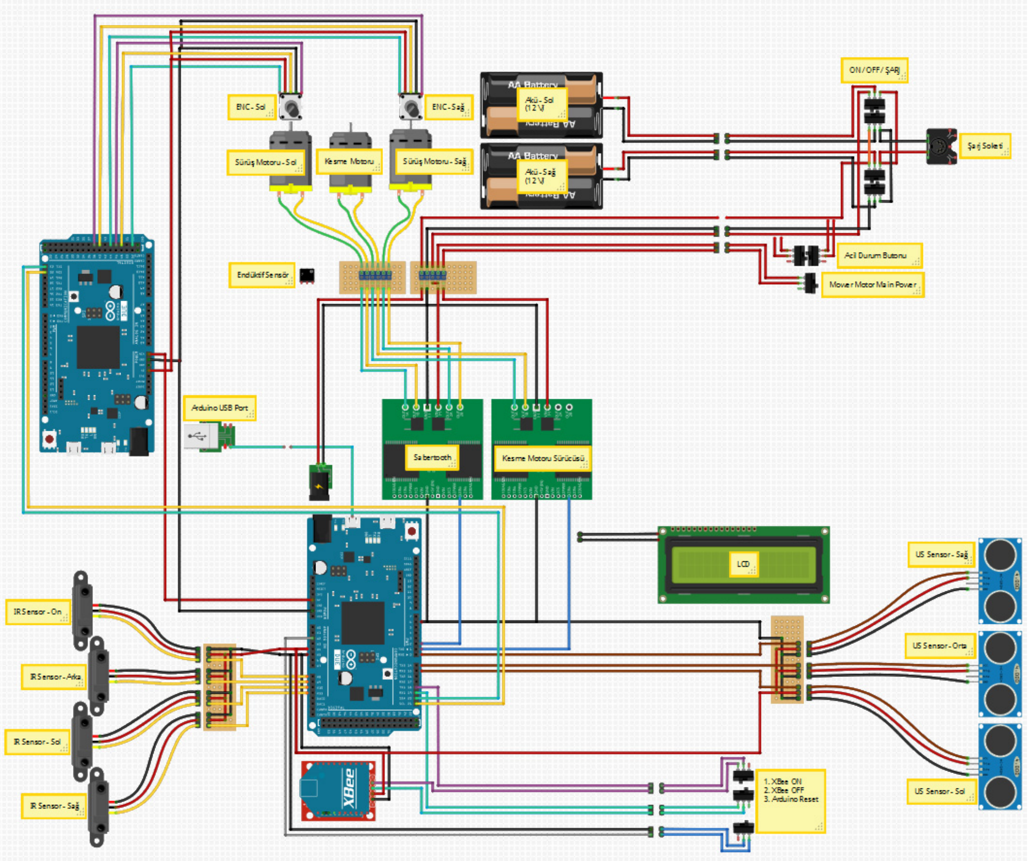


Figure 3.12: Main Circuit Schematic of the Electrical System

Unlike other Arduino boards, the recently released Arduino model, Due, runs at 3.3V. In other words, the maximum voltage that the I/O pins can tolerate is 3.3V. Since we have other components working at 5V like motor drivers, encoders, sensors, etc. there must be a logic level converter circuit in between 3.3V controllers and other 5V devices. A logic level converter performs level shifting/conversion to protect the 3.3V device from 5V, and operate steadily. Although one can use resistors to make a divider, for high speed transfers just like we have in our system, the resistors can add a lot of slew and cause havoc that is tough to debug. For that reason, using a Logic Level Converter to perform proper level shifting is beneficial. Adafruit's TXB0108 model LLC chip performs bi-directional level shifting from pretty much any voltage to any voltage and will auto-detect the direction. It has operating times about 10nsecs that is suitable for our application.

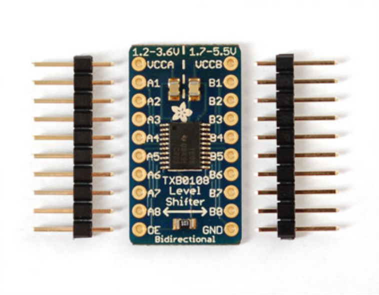


Figure 3.13: Bi-directional Logic Level Converter

Although harness has not taken into account sufficiently on this prototype platform, there are some points that require to be concerned. A mainboard-like circuit is designed so that all cables can easily be plugged and unplugged tons of times all through the prototype tests period long. The other part that the harness needs attention to be paid is the control panel which is on top of the outer cover. All components on that control panel like switches, USB port and charging socket is plugged with qualified connectors so that the cover can be completely removed, again specially for that first prototype.

3.3.3 Mechanical Design

This section of the study reveals the mechanical design, which is finalized after numerous revisions. The key components and subsystems are explained in detail below.

3.3.3.1 Overview and General Dimensions

Functionality and visual perfection are the main considerations taken during the mechanical design process of the ALM. It is also intended to minimize bounding box as much as possible without affecting the desired functionality and visuality.

As a result, the outer skin, which acts as a cover for the main body and interior components, is designed as a rabbit shape. All the outer dimensions of the ALM are intended to be kept lower from the rivals. The finalized mechanical design overview and the major dimensions are given in figures below.

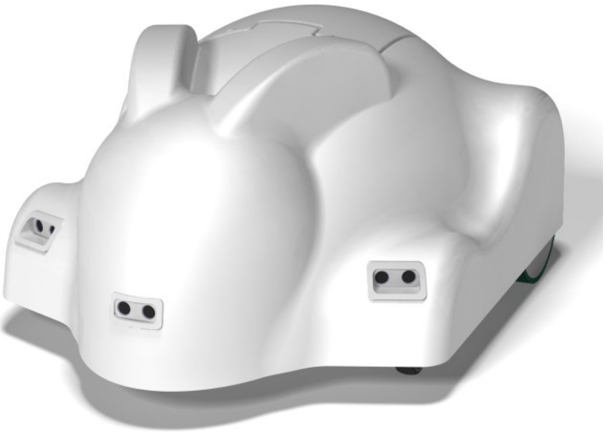


Figure 3.14: Overview of Mechanical Design - Isometric View (Front)

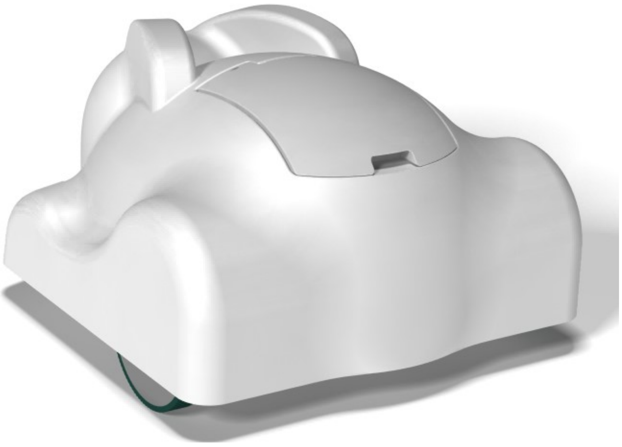


Figure 3.15: Overview of Mechanical Design - Isometric View (Rear)

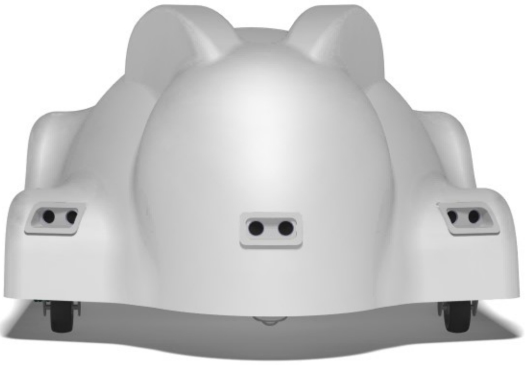


Figure 3.16: Overview of Mechanical Design - Front View

The main dimensions for the ALM are also given in Figure 3.18.

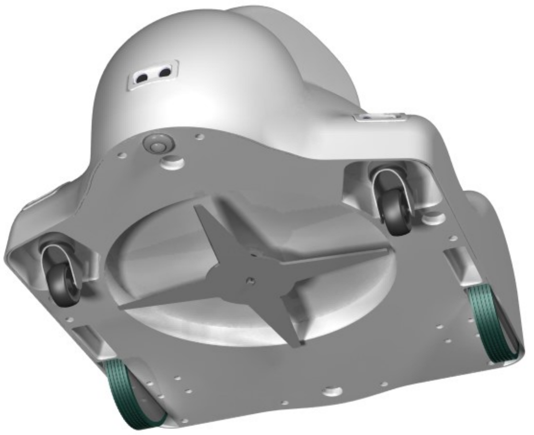


Figure 3.17: Overview of Mechanical Design - Bottom View

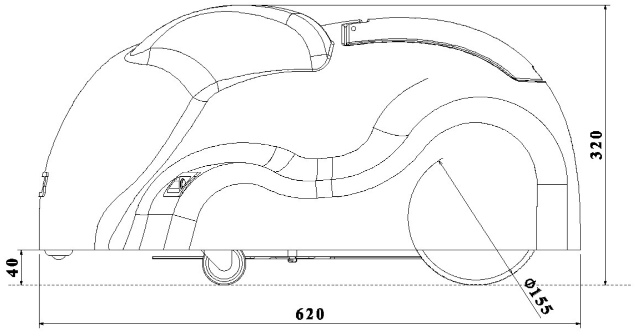


Figure 3.18: Main Dimensions of the Autonomous Lawn Mower (Side View)

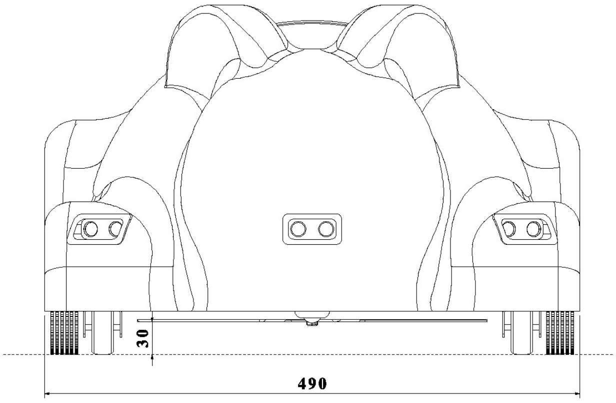


Figure 3.19: Main Dimensions of the Autonomous Lawn Mower (Front View)

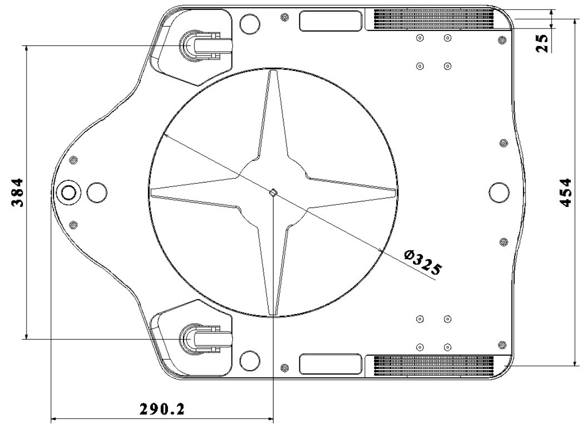


Figure 3.20: Main Dimensions of the Autonomous Lawn Mower (Bottom View)

3.3.3.2 External Design and Layout

The external view of the ALM is intended to be kept as lean as possible. The layout of external components can be seen in Figure 3.21.

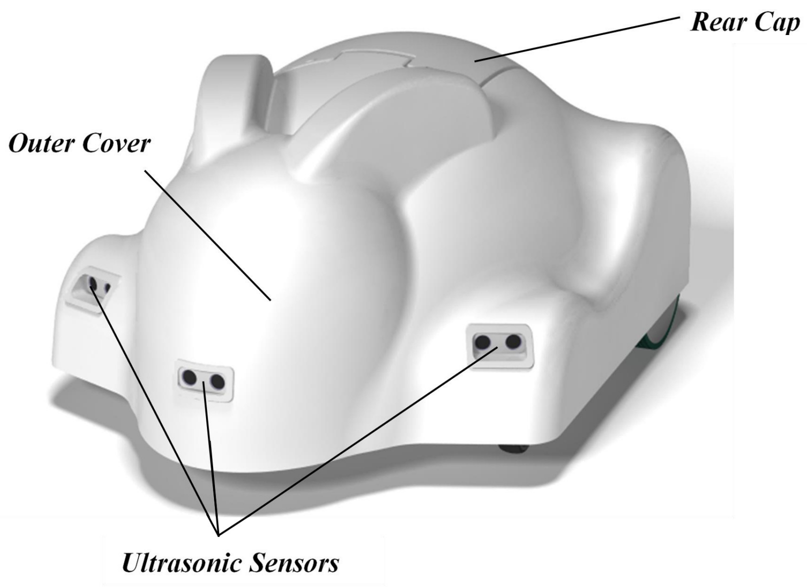


Figure 3.21: External Layout of Autonomous Lawn Mower

The outer cover mainly acts as a water and dust proof cover for internal components. In order to lower the manufacturing costs, this cover is decided to be made via vacuum forming process using polyethylene (PE) plastic material. Therefore, no negative slopes exist in any curvature of the design. The top and the bottom views of the outer cover is presented in Figure 3.22.

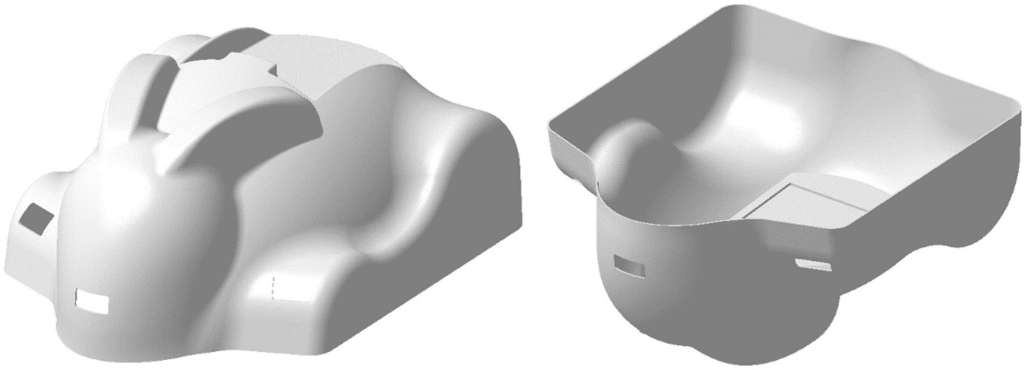


Figure 3.22: Outer Cover of Autonomous Lawn Mower

The US sensors are assembled on the outer cover using their own enclosures. Also a hinged rear-cap is designed to protect the robot-human physical interface from water and dust. Rear cap and US sensor enclosures are made by a lightweight plastic, Delrin.

3.3.3.3 Wheel Design and Selection

The wheel diameter plays an important role for the trade-off between torque and drive speed to satisfy the desired design requirements. The outer diameter for the drive wheels is chosen as 155 mm, which satisfies both requirements at the same time.

The main threat for an odometry-based robot is the slippage. In order to decrease the potential slippage, drive wheels are coated with a synthetic rubber material, which increases the friction coefficient between wheels and the ground.

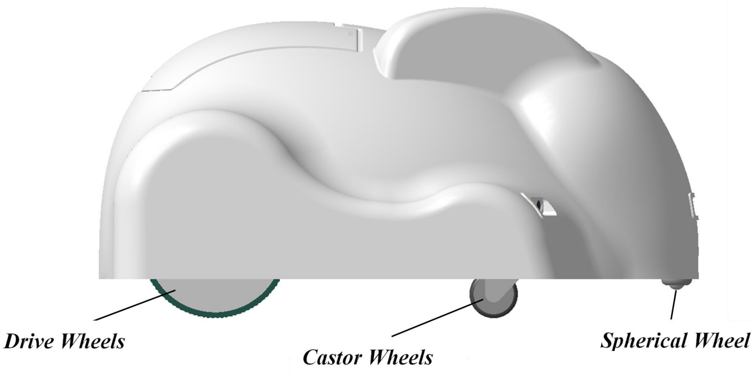


Figure 3.23: Wheel Design of Autonomous Lawn Mower

In order to balance the weight distribution, two standard castor wheels are placed in front of the robot. Even if the castor wheels introduce a disturbance to the robot inherently, their usage is found to be essential for outdoor navigation. Moreover, a spherical wheel is placed in front of the robot in order to provide robot's first ground contact through this wheel.

3.3.3.4 Internal Design and Layout

The internal layout of the ALM is presented in Figure 3.24. The main body, which acts as a base for all of the components to be assembled, is also designed to be made from a lightweight, durable material, Delrin. The main body design is suited for conventional milling process.

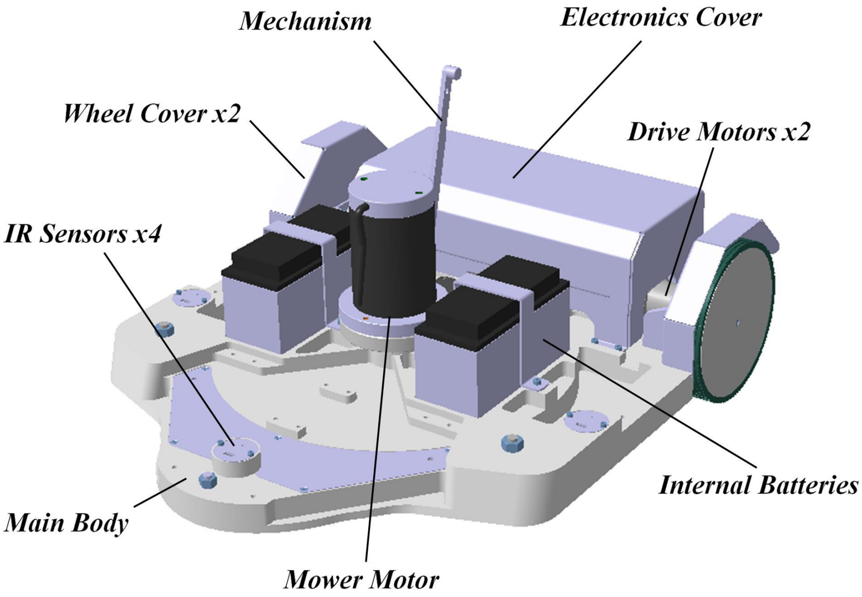


Figure 3.24: Internal Layout of Autonomous Lawn Mower

The gear-motors are assembled using a bracket, which is placed on the rear of the main body and attached to the drive wheels without any additional transmission. Sheet metal wheel covers have also been placed on both wheels in order to prevent the particles to enter the robot interior. Both of the wheel covers and "drive motor" brackets are designed to be aluminum.

All electronics are located on the plate above the "drive motors" in order to use this dead volume efficiently. An enclosure with a cap is designed to protect electronics from dust. This electronics cover is also thought to be aluminum.

"Mower motor" and the cutting blades are placed as close as possible to the front region of the main body to make the mowing operation more efficient. The details of mowing system will be presented in the following section.

In order to move the center of mass to the rear side of the robot the internal Ni-Mh batteries are placed besides the "mower motor" using aluminum brackets.

Four infrared sensors are placed as a "plus" sign, at the bottom of the body. These IR sensors are used for fall-off and tilt-over detection as mentioned before.

The rest of the interior zone (i.e. front region) has been left empty for harness details and for possible future requirements.

3.3.3.5 Mowing System Design

The design of the mowing system and its components is critical since the actual purpose of the robot is lawn mowing. In this section, mower blade, housing of components and cutting height adjustment mechanism designs are presented in detail. A clipped view of the mowing system is given in Figure 3.26.

The mowing system is mainly composed of a "mower motor" and a cutting blade. A comparably bigger, single blade has been decided to be used. The cutting blade has four cutting edges sharpened from both sides. Steel (St37) is chosen as the blade material for strength purposes.

There is no reduction or transmission between the "mower motor" and the cutting blade since the torque requirement is negligible compared to the angular velocity requirement. The motor and the blade is aligned by using a shaft. This blade shaft is mounted in a motor holder by using a ball bearing and a linear bearing. The blade shaft material is chosen as stainless steel.

The reason for using linear bearing comes from the requirement of cutting height adjustment for lawn mowers, whether they are autonomous or not. As given in

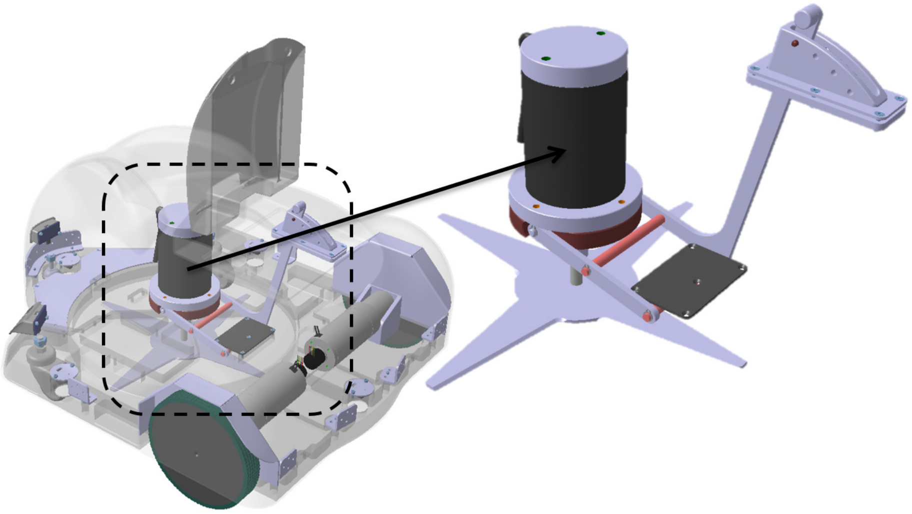


Figure 3.25: Overview of Mowing System Design

Figure 3.26, a lever arm is designed and hinged onto the "mower motor" holder to position whole mowing mechanism to the desired cutting height. An existing housing for the lever arm is assembled on the outer cover.

It is intended to move whole height adjustment mechanism by a finger movement. Therefore a finger handle is designed and mounted on the top of the lever arm. It actually stays out of the outer cover, inside the rear cap. Also a balance arm and torsion bars have been designed for distributing mowing system weights evenly, so that the required pulling force is decreased. The mechanism parts are made of aluminum.

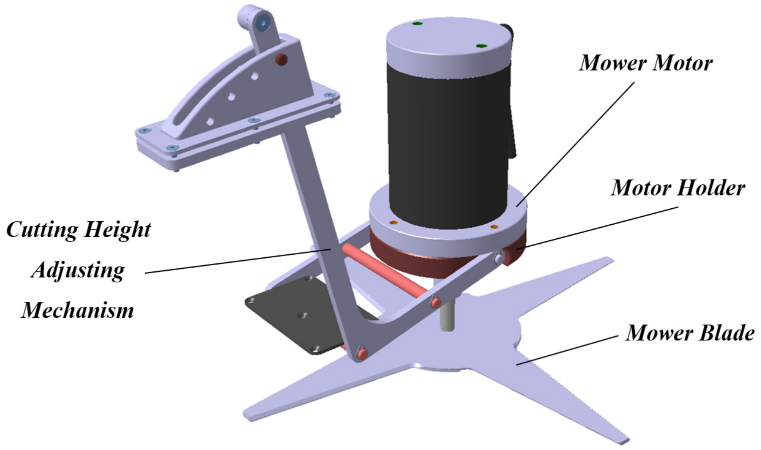


Figure 3.26: Detailed View of Mowing System Design

With the help of the finger handle and the mechanism, the cutting height (i.e.

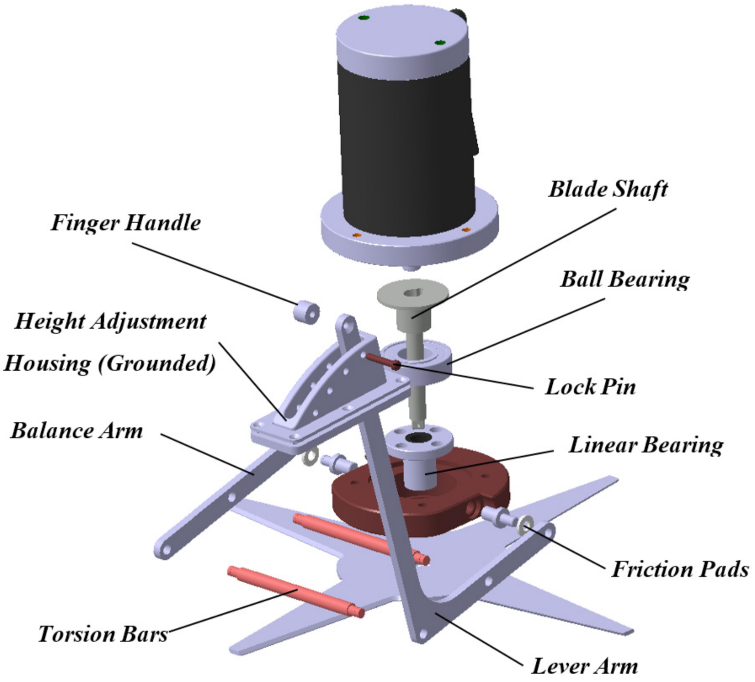


Figure 3.27: Detailed View of Mowing System Design

the height between the cutting blade and the ground surface) can be adjusted from 30 mm to 60 mm. Adjustable cutting height range (30 mm) is divided into 4 segments. In the first stage the lock-pin is attached to the first hole of the height adjustment housing, which provides 30 mm cutting height. In the last stage, where cutting height is modified to 60 mm, the lock-pin must be attached to the fourth hole of the height adjustment housing. All of the cutting height adjustment stages are shown in Figure 3.28.

The use of finger handle for adjusting the mechanism has an important necessity i.e. the whole mechanism must be designed in such a way that a standard user can easily generate the amount of force required to pull the handle by his/her finger. For this requirement, a Multi Body Dynamics analysis was performed. A simulation model was prepared using the key components of the height adjustment mechanism. Actual weights of the simulation components are defined and the rest is modelled by a force, acting on the mower motor holder. Components used in this simulation model are given in Figure 3.29.

Total weight of components (mower motor, mower blade, blade shaft, etc.) carried by the mower motor holder is calculated as 2.6 kg. The required finger force

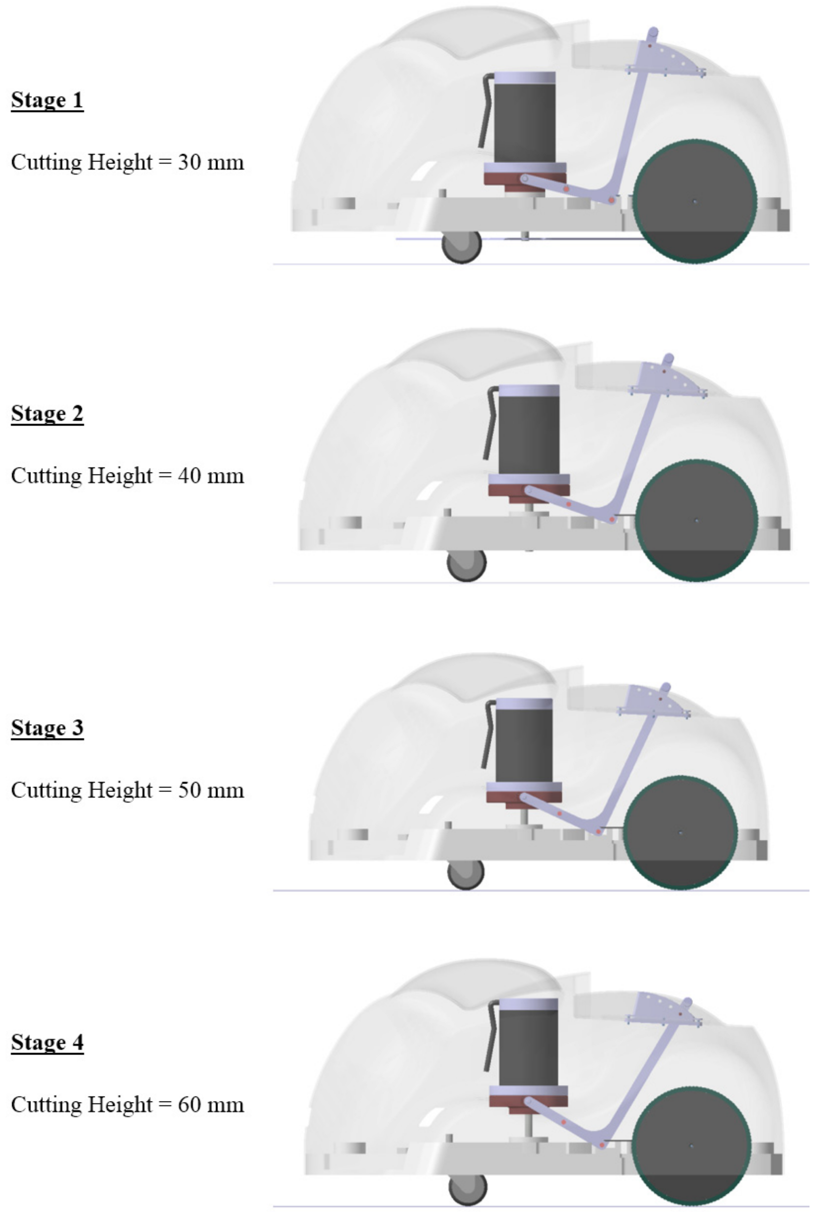


Figure 3.28: Stages of Cutting Height Adjustment Mechanism

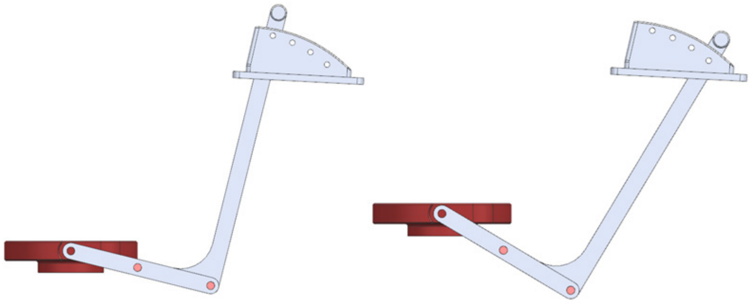


Figure 3.29: Simulation Model for Cutting Height Adjustment Mechanism

to lift this weight 30 mm above from the lowest level is given in Figure 3.30.

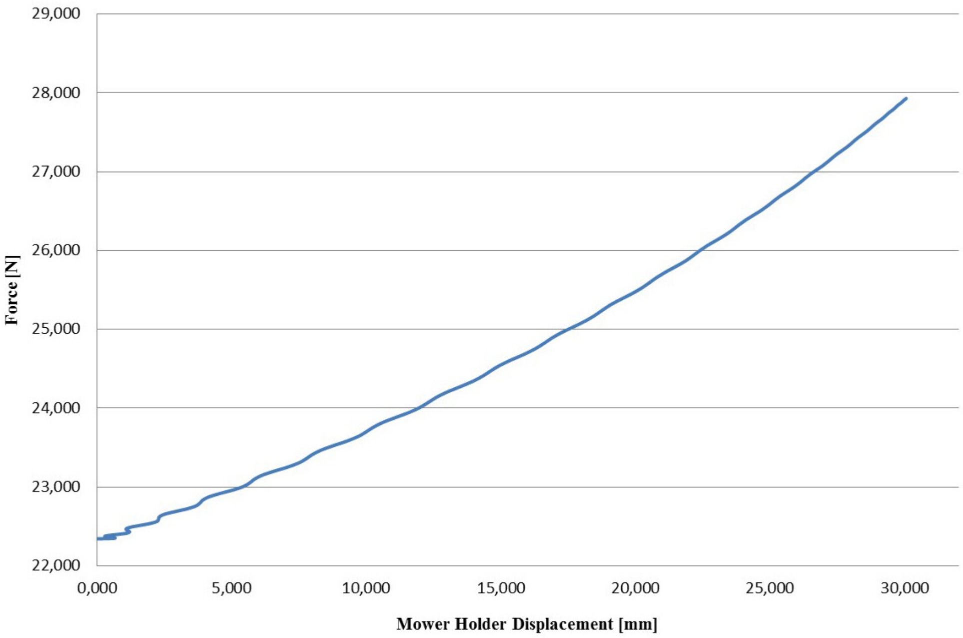


Figure 3.30: Force Requirement for Cutting Height Adjustment

It is calculated that a minimum of 22.3 N pulling force can trigger the mechanism and 27.9 N is required to lift whole mowing system 30 mm above (i.e. to insert the lock-pin to the last hole). Finally, it was concluded that 27.9 N force can easily be generated by a standard user's finger and thereby the mowing system mechanical design was successfully completed.

3.3.3.6 Infrared Switch Layout

IR switches are placed to the bottom of the mower body near to the mower blade. As explained in the mathematical modelling section, their places are quite important when the controllability and the performance issues are considered. Actually, considering a commercial product, these IR switches need to be protected from external effects such as stones, water or dust. However, these IR switches are not designed to be an end-user product component, rather their purpose is to test the performance of proposed method and the IR switches themselves.

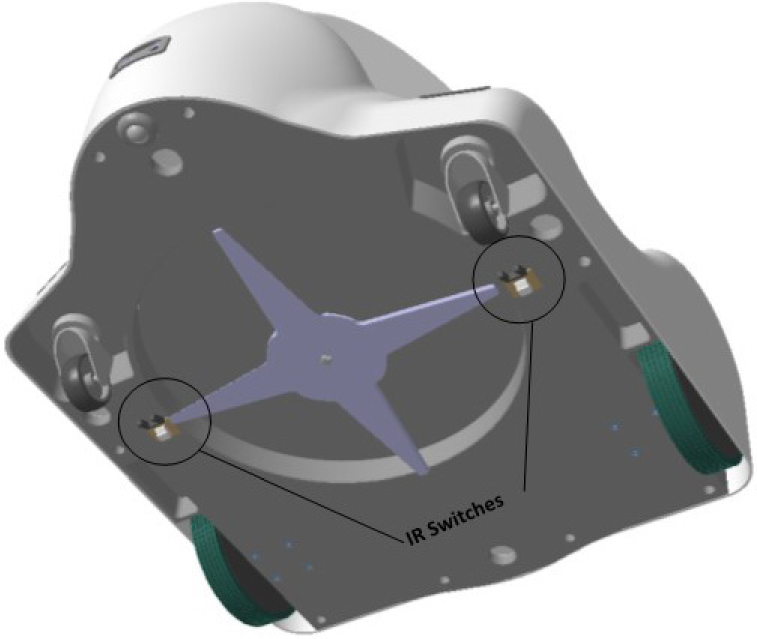


Figure 3.31: Infrared Switch Layout

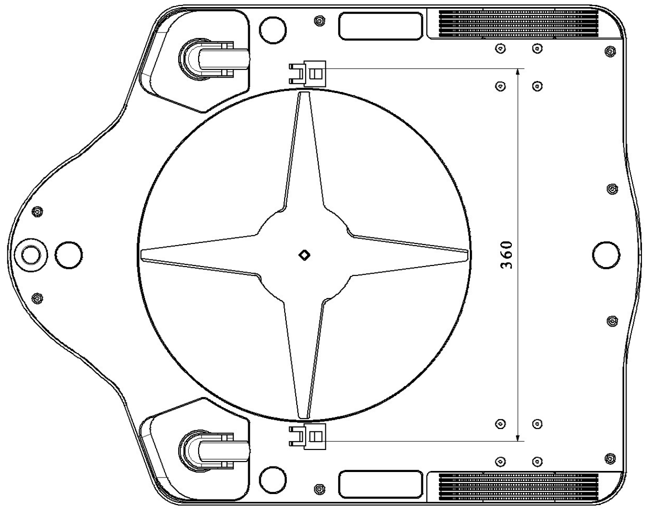


Figure 3.32: Infrared Switch Positions

3.4 Mathematical Modelling

3.4.1 Introduction

Mathematical modelling of mobile robots is a keystone of designing the appropriate controller and choosing the appropriate sensing method as well as planning the robot trajectory. Literature says that there is no need for dynamic modelling for the vehicles running under 10km/h speed. Below this speed levels, dynamic modelling and kinematic modelling gives almost the same results without any significant loss. Additionally, we have no significant disturbances/forces on our system that need to be taken into account as a dynamic feature. Due to those reasons and basically for simplicity, there is no dynamic models used for this project. Kinematic models will be explained in detail in this section. [2]

For two traction wheeled mobile robots, there are two main modelling methods which are differential drive robot model and unicycle robot model. [4] Concerning the core, these two methods are similar with minor differences.

Differential Drive Robot Model Differential Drive Robot Model deals with the modelling the behavior of each wheel to determine the robot motion in translational and rotational axes.

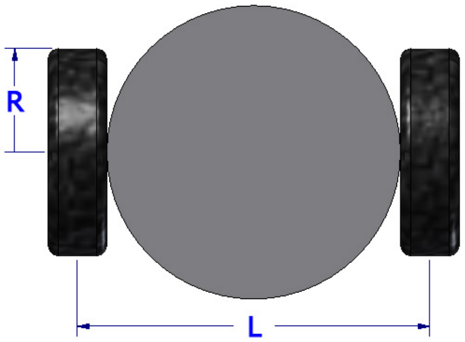


Figure 3.33: Differential Drive Robot Model

For a two-wheeled differentially driven mobile robot represented in the figure, determining the robot behaviour from wheel behaviours directly gives that;

$$\dot{x} = \frac{R}{2}(v_r + v_l)\cos(\theta) \tag{3.11}$$

$$\dot{y} = \frac{R}{2}(v_r + v_l)\sin(\theta) \tag{3.12}$$

$$\dot{\theta} = \frac{R}{L}(v_r - v_l) \tag{3.13}$$

Unicycle Robot Model Unicycle Robot Model is like treating the robot as a point mass. In this method, the control inputs are linear and angular velocity values. After separating these control inputs into their components, this model enables the control action in terms of position and heading.

Figure 3.34 shows the unicycle robot approach and shows the position and orientation notation.

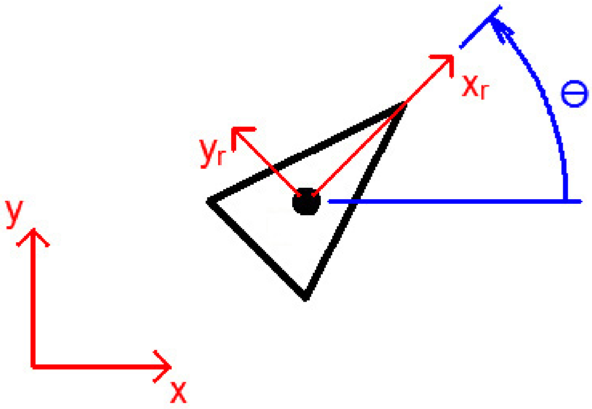


Figure 3.34: Unicycle Robot Model

Unicycle robot approach gets two control inputs and two outputs by considering the non-holonomic constraints. The planar kinematics of the system is;

$$\dot{x} = v\cos\theta \tag{3.14}$$

$$\dot{y} = v\sin\theta \tag{3.15}$$

$$\dot{\theta} = \omega \tag{3.16}$$

3.4.2 Implementation

Differential drive robot model and unicycle robot model were compared and their pro's and con's were considered from almost all aspects. At the end, although the differential drive robot model has complexities and difficulties to implement, it is chosen for the mower robot due to its flexibility for creating specific commands and operating-system-like software architecture.

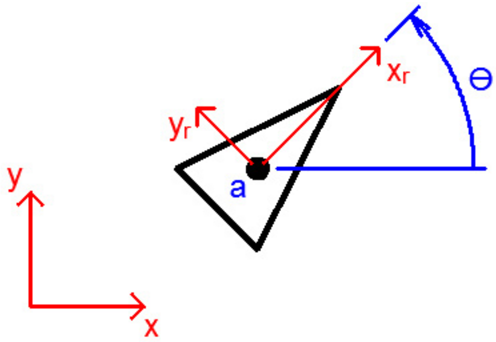


Figure 3.35: The robot just before a command execution

First, we have a global reference frame and we must know the position (x,y) of the rotation center and the heading angle (θ) of our robot. The differential drive robot model deals with modelling the behavior of each wheel to determine the robot motion. By using its equations, we can calculate the change of position and heading angle values of the robot. In order to determine the desired velocities of each wheel, we need to derive some equations to calculate the error values while the robot is going on its desired trajectory. For this purpose, mathematical error values of line and rotate commands should be derived. Mathematical representations of Line and Arc shaped trajectories to be followed by the robot for Line and Arc commands, and Point representation to be followed by the robot for Rotate command are given in detail below.

3.4.2.1 Line Command

For a straight line for the robot to follow, we have to know the initial (start) point of that line, know the initial (desired) heading angle for the robot, and calculate the end (target) point of that line at the begining of that command.

Additionally, we have to check if our robot is exactly on the desired trajectory and has the desired heading angle all along the command is running. For that purpose, we have to know the position error and the heading angle error at any time during the command is running. At the same time, we have to set the absolute value of desired wheel speeds ratio required for the controllers.

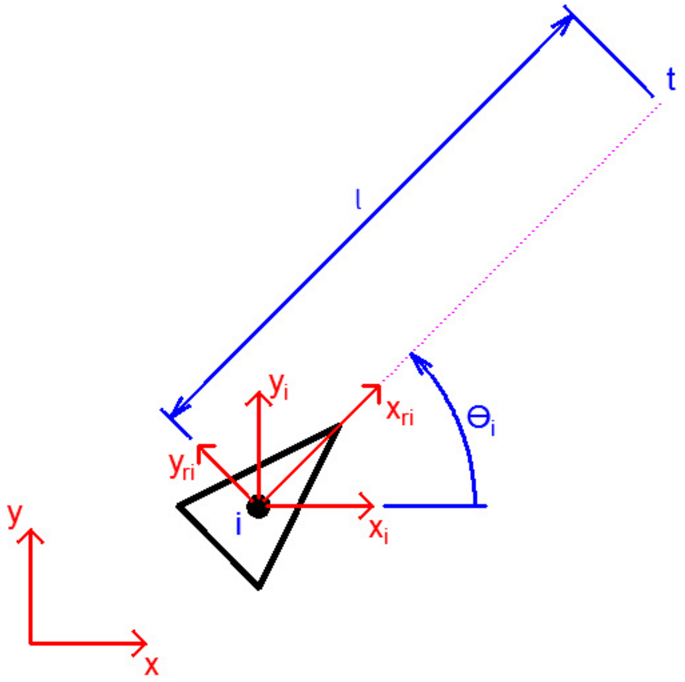


Figure 3.36: Target Point of the Line Command

Absolute Desired Wheel Speeds Ratio Both wheels should have exactly the same speed for line command. Their absolute ratio should be simply;

$$R_{Abs_{Des}} = 1 \tag{3.17}$$

Target Point Calculations Translating the origin of the global coordinate axes $x - y$ to the command-initiation point i gives a new coordinate system called $x_i - y_i$.

Rotating the coordinate axes $x_i - y_i$ through the angle θ_i about the origin gives a new coordinate system $x_{r_i} - y_{r_i}$.

Now that the coordinate system $x_{r_i} - y_{r_i}$ is exactly on the robot body-fixed-frame $x_r - y_r$, it can be said for the coordinates of target point t, that;

$$\begin{bmatrix} x_{r_i} \\ y_{r_i} \end{bmatrix} = \begin{bmatrix} l \\ 0 \end{bmatrix} \tag{3.18}$$

Transforming the coordinate system $x_{r_i} - y_{r_i}$ back to the coordinate system $x_i - y_i$, coordinates of the target point t becomes;

$$\begin{bmatrix} x_{i_t} \\ y_{i_t} \end{bmatrix} = \begin{bmatrix} \cos(-\theta_i) & \sin(-\theta_i) \\ -\sin(-\theta_i) & \cos(-\theta_i) \end{bmatrix} \begin{bmatrix} x_{ri_t} \\ y_{ri_t} \end{bmatrix} \tag{3.19}$$

Substituting x_{ri_t} and y_{ri_t} into Equation 3.19 gives;

$$\begin{bmatrix} x_{i_t} \\ y_{i_t} \end{bmatrix} = \begin{bmatrix} \cos(-\theta_i)l \\ -\sin(-\theta_i)l \end{bmatrix} \tag{3.20}$$

Translating the coordinate system $x_i - y_i$ back to the global coordinate system $x - y$, coordinates of target point t becomes;

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_{i_t} + x_i \\ y_{i_t} + y_i \end{bmatrix} \tag{3.21}$$

Substituting x_{i_t} and y_{i_t} into Equation 3.21 gives;

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \cos(-\theta_i)l + x_i \\ -\sin(-\theta_i)l + y_i \end{bmatrix} \tag{3.22}$$

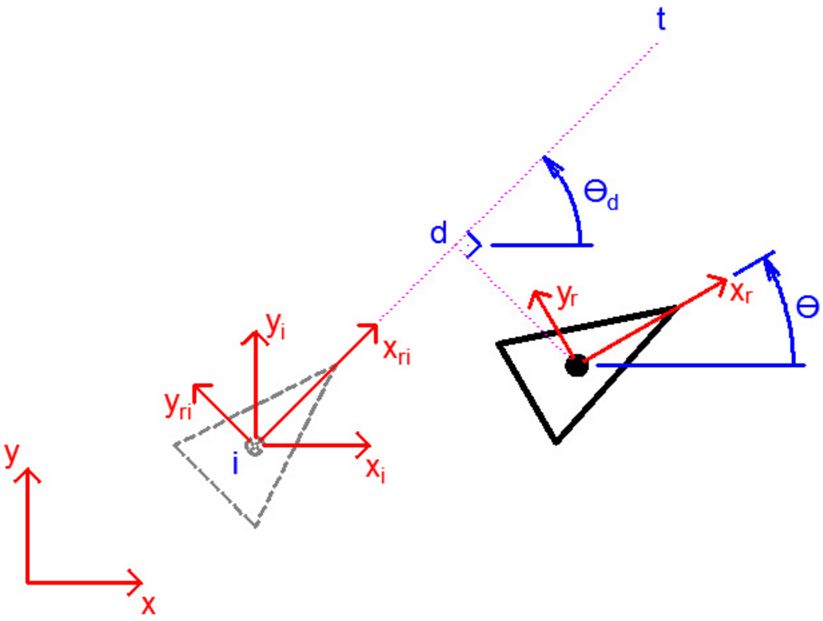


Figure 3.37: While the Line Command is running

Position Error Calculations Translating the origin of the coordinate axes $x - y$ to the point i gives a new coordinate system $x_i - y_i$. In that case, the coordinates of the rotation center of the robot becomes;

$$\begin{bmatrix} x_{i_a} \\ y_{i_a} \end{bmatrix} = \begin{bmatrix} x - x_i \\ y - y_i \end{bmatrix} \tag{3.23}$$

Rotating the coordinate axes $x_i - y_i$ through the angle θ_i about the origin gives a new coordinate system $x_{r_i} - y_{r_i}$.

In that case, actual robot position (point a) coordinates can be written as;

$$\begin{bmatrix} x_{ri_a} \\ y_{ri_a} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x_{i_a} \\ y_{i_a} \end{bmatrix} \tag{3.24}$$

Substituting x_i and y_i into Equation 3.24 gives;

$$\begin{bmatrix} x_{ri_a} \\ y_{ri_a} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} (x - x_i) \\ (y - y_i) \end{bmatrix} \tag{3.25}$$

Here, y_{ri_a} is the position error of the line command.

$$e_p = y_{ri_a} \tag{3.26}$$

Hence, the position error of the line command is;

$$e_p = -\sin(\theta_i)(x - x_i) + \cos(\theta_i)(y - y_i) \tag{3.27}$$

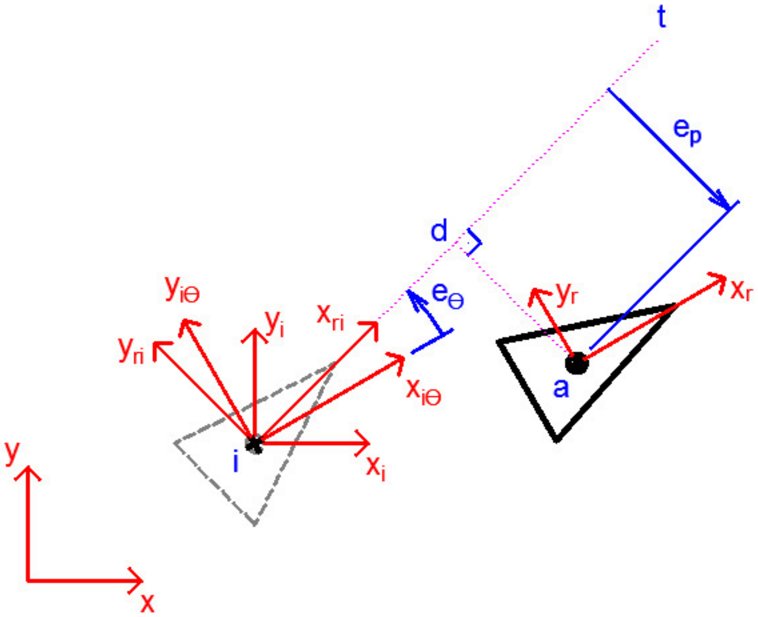


Figure 3.38: Error Representation of Line Command

Heading Error Calculations Translating the origin of the coordinate axes $x - y$ to the point i gives a new coordinate system $x_i - y_i$. In that case, the coordinates of the target point becomes;

$$\begin{bmatrix} x_{i_t} \\ \\ y_{i_t} \end{bmatrix} = \begin{bmatrix} x_t - x_i \\ \\ y_t - y_i \end{bmatrix} \tag{3.28}$$

Rotating the coordinate axes $x_i - y_i$ through the angle θ about the origin gives a new coordinate system $x_{ri} - y_{ri}$.

In that case, coordinates of target point t can be written as;

$$\begin{bmatrix} x_{ri_t} \\ \\ y_{ri_t} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x_{i_t} \\ \\ y_{i_t} \end{bmatrix} \tag{3.29}$$

Substituting x_{i_t} and y_{i_t} into Equation 3.29 gives;

$$\begin{bmatrix} x_{ri_t} \\ \\ y_{ri_t} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x_t - x_i \\ \\ y_t - y_i \end{bmatrix} \tag{3.30}$$

In this case, the angle of the virtual line between origin and t_{ri} should be the heading error. Calculating this angle in the range of $-\pi$ to π by using the $atan2$ function;

$$e_\theta = atan2(y_{ri_t}, x_{ri_t}) \tag{3.31}$$

Hence, the heading error is of the line command is;

$$e_\theta = atan2(-\sin(\theta)(x_t - x_i) + \cos(\theta)(y_t - y_i), \cos(\theta)(x_t - x_i) + \sin(\theta)(y_t - y_i)) \tag{3.32}$$

3.4.2.2 Rotate Command

For a point for the robot to follow, we have to know the initial (start) point of the command, and calculate the end (target) heading of the robot at the begining of that command. Additionally, we have to check if our robot is exactly on the desired position all along the command is running. For that purpose, we have to know the position error at any time during the command is running. At the same time, we have to set the absolute value of desired wheel speeds ratio required for the controllers.

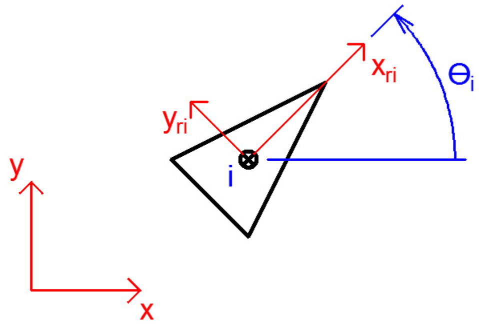


Figure 3.39: Initial state of Rotate Command

Absolute Desired Wheel Speeds Ratio Wheels should have exactly the same speeds in magnitude, but in opposite directions for rotate command. Their absolute ratio should be simply with opposite signed ratio coefficients;

$$R_{Abs_{Des}} = 1 \tag{3.33}$$

Target Heading Angle The target heading angle of the robot for rotate command is simply;

$$\theta_t = mod(\theta_i + \triangle\theta, 2\pi) \tag{3.34}$$

Target Point The target point (position) of the robot for rotate command should be exactly the same as the initial position. So, it should be simply;

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{3.35}$$

Position Error Calculations Translating the origin of the coordinate axes $x - y$ to the point a gives a new coordinate system $x_a - y_a$. In that case, the coordinates of the initial point i becomes;

$$\begin{bmatrix} x_{a_i} \\ y_{a_i} \end{bmatrix} = \begin{bmatrix} x_i - x_a \\ y_i - y_a \end{bmatrix} \tag{3.36}$$

Rotating the coordinate axes $x_a - y_a$ through the angle θ about the origin gives a new coordinate system $x_r - y_r$.

In that case, initial point i coordinates can be written as;

$$\begin{bmatrix} x_{r_i} \\ y_{r_i} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_{a_i} \\ y_{a_i} \end{bmatrix} \tag{3.37}$$

Here, x_{r_i} is the position error of rotate command.

$$e_p = x_{r_i} \tag{3.38}$$

Hence, the position error of the rotate command is;

$$e_p = \cos(\theta)(x_i - x_a) + \sin(\theta)(y_i - y_a) \tag{3.39}$$

3.4.2.3 Arc Command

For an arc-shaped trajectory for the robot to follow, we have to know the initial (start) point of that arc, know the initial heading angle of the robot, and calculate the center point of that arc and the target (end) point of that arc at the begining of that command. Additionally, we have to check if our robot is

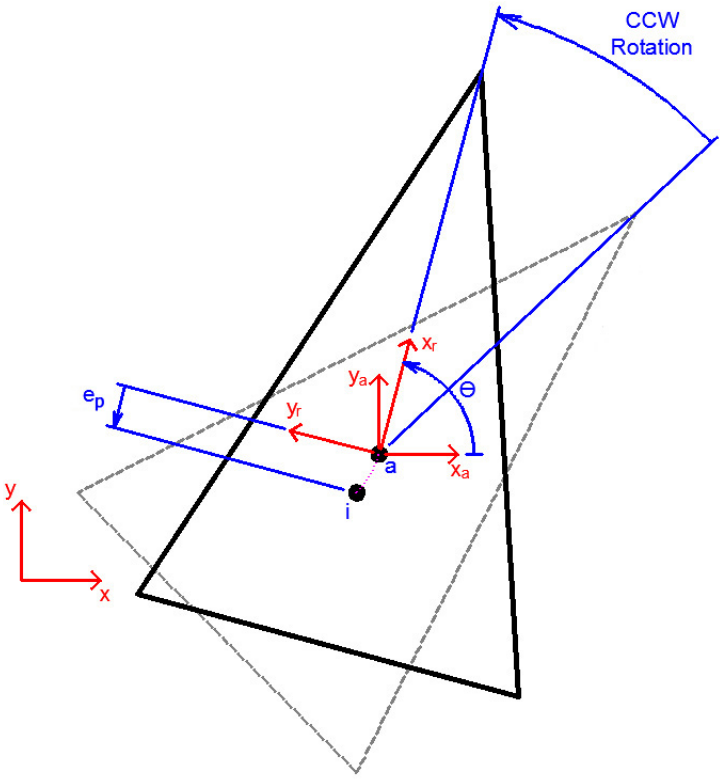


Figure 3.40: Negative Position Error Representation of Rotate Command for CCW Rotation

exactly on the desired trajectory and has the desired varying heading angle all along the command is running. For that purpose, we have to know position error and heading angle error at any time during the command is running. At the same time, we have to set the absolute value of desired wheel speeds ratio required for the controllers.

Absolute Desired Wheel Speeds Ratio Wheels should have different speeds for arc command. Their ratio should be;

$$R_{Abs_{Des}} = \frac{r_d - (signum(\triangle\alpha) * \frac{L}{2})}{r_d + (signum(\triangle\alpha) * \frac{L}{2})} \tag{3.40}$$

where;

$R_{Abs_{Des}}$: absolute value of the driving speeds ratio of two wheels

L : the distance between two traction wheels

$signum()$: the function that returns the sign of its parameter

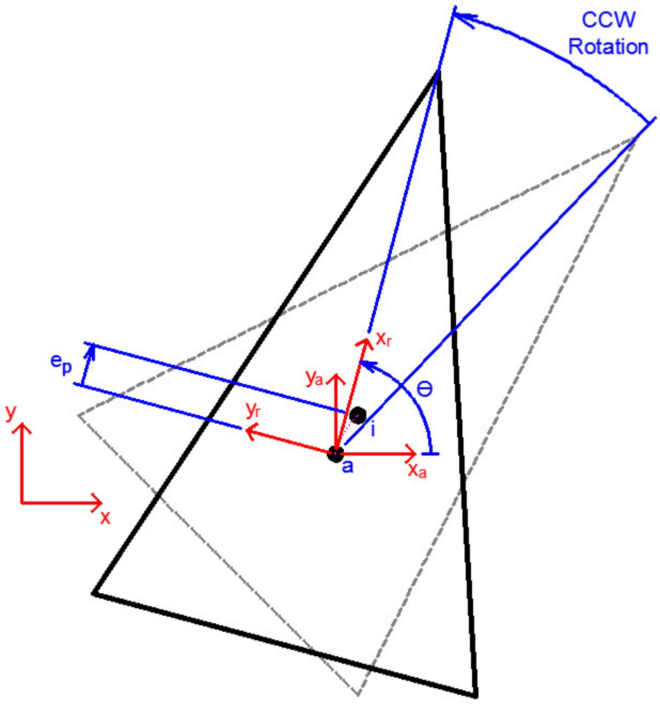


Figure 3.41: Positive Position Error Representation of Rotate Command for CCW Rotation

Center Point Calculations Translation of the origin of the coordinate axes $x - y$ to the point i gives a new coordinate system $x' - y'$.

Transformation (rotation) of the coordinate axes $x' - y'$ through the angle $-\theta_i$ about the origin gives a new coordinate system $x'' - y''$.

Now that the coordinate system $x''-y''$ is exactly on the robot body-fixed-frame $x_r - y_r$, for the coordinates of center point of the command point c, it can be said that;

$$x_{c''} = 0 \tag{3.41}$$

$$y_{c''} = \begin{cases} r_d, & \text{for } \triangle\alpha > 0 \\ -r_d, & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.42}$$

Transforming the coordinate system $x''-y''$ back to the coordinate system $x'-y'$,

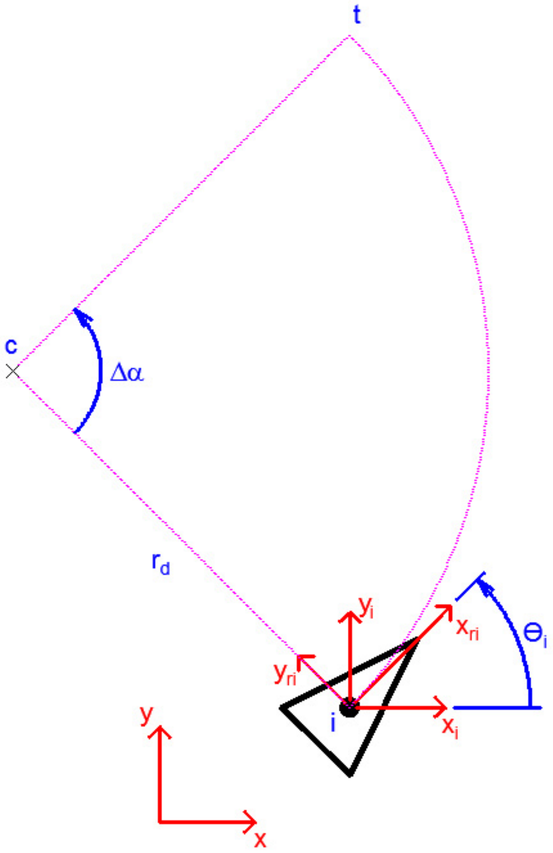


Figure 3.42: Initial State of Arc Command

coordinates of center point c becomes;

$$\begin{bmatrix} x_{c'} \\ y_{c'} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) - \sin(\theta_i) \\ \sin(\theta_i) + \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x_{c''} \\ y_{c''} \end{bmatrix} \tag{3.43}$$

Substituting $x_{t''}$ and $y_{t''}$ into Equation 3.43 gives;

$$x_{c'} = \begin{cases} -r_d \sin(\theta_i), & \text{for } \triangle\alpha > 0 \\ r_d \sin(\theta_i), & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.44}$$

$$y_{c'} = \begin{cases} r_d \cos(\theta_i), & \text{for } \triangle\alpha > 0 \\ -r_d \cos(\theta_i), & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.45}$$

Translating the coordinate system $x' - y'$ back to the global coordinate system $x - y$, coordinates of center point c becomes;

$$\begin{bmatrix} x_c \\ \\ y_c \end{bmatrix} = \begin{bmatrix} x_{c'} + x_i \\ \\ y_{c'} + y_i \end{bmatrix} \tag{3.46}$$

Substituting $x_{c'}$ and $y_{c'}$ into Equation 3.46 gives;

$$x_c = \begin{cases} -r_d \sin(\theta_i) + x_i, & \text{for } \triangle\alpha > 0 \\ r_d \sin(\theta_i) + x_i, & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.47}$$

$$y_c = \begin{cases} r_d \cos(\theta_i) + y_i, & \text{for } \triangle\alpha > 0 \\ -r_d \cos(\theta_i) + y_i, & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.48}$$

Target Point Calculations Translation of the origin of the coordinate axes $x - y$ to the point c gives a new coordinate system $x' - y'$. In this case, the coordinate of point i is;

$$\begin{bmatrix} x_{i'} \\ \\ y_{i'} \end{bmatrix} = \begin{bmatrix} x_i - x_c \\ \\ y_i - y_c \end{bmatrix} \tag{3.49}$$

Transformation of point i' through the angle $\triangle\alpha$ about point c' gives the coordinates of point t';

$$\begin{bmatrix} x_{t'} \\ \\ y_{t'} \end{bmatrix} = \begin{bmatrix} \cos(\triangle\alpha) - \sin(\triangle\alpha) \\ \\ \sin(\triangle\alpha) + \cos(\triangle\alpha) \end{bmatrix} \begin{bmatrix} x_{i'} \\ \\ y_{i'} \end{bmatrix} \tag{3.50}$$

Translating the coordinate system $x' - y'$ back to the coordinate system $x - y$, coordinates of target point t becomes;

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \cos(\triangle\alpha) - \sin(\triangle\alpha) \\ \sin(\triangle\alpha) + \cos(\triangle\alpha) \end{bmatrix} \begin{bmatrix} x_{i'} \\ y_{i'} \end{bmatrix} \tag{3.51}$$

Substituting $x_{i'}$ and $y_{i'}$ into Equation 3.51 gives;

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} \cos(\triangle\alpha) - \sin(\triangle\alpha) \\ \sin(\triangle\alpha) + \cos(\triangle\alpha) \end{bmatrix} \begin{bmatrix} x_i - x_c \\ y_i - y_c \end{bmatrix} \tag{3.52}$$

Position Error Calculations Actual intantaneous radius of the arc is;

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{3.53}$$

$$\tag{3.54}$$

Hence, the position error of the arc command is;

$$e_p = r_d - r \tag{3.55}$$

Heading Error Calculations Translating the origin of the coordinate axes $x - y$ to the point a gives a new coordinate system $x' - y'$. In that case, the coordinates of the center point becomes;

$$\begin{bmatrix} x_{c'} \\ y_{c'} \end{bmatrix} = \begin{bmatrix} x_c - x \\ y_c - y \end{bmatrix} \tag{3.56}$$

Transformation of the coordinate axes $x' - y'$ through an angle β about the origin gives a new coordinate system $x'' - y''$. Angle β depends on the direction of the sweep angle of the arc as follows;

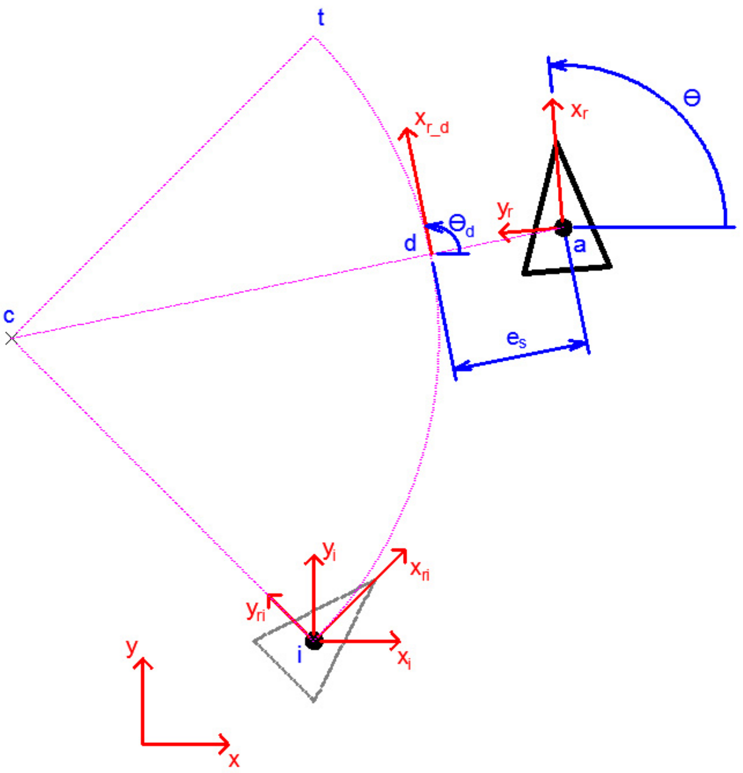


Figure 3.43: Error Representation of Arc Command

$$\beta = \begin{cases} -(\theta + \frac{\pi}{2}), & \text{for } \triangle\alpha > 0 \\ -(\theta - \frac{\pi}{2}), & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.57}$$

Therefore, coordinates $x_{c''}, y_{c''}$ can be written as;

$$\begin{bmatrix} x_{c''} \\ y_{c''} \end{bmatrix} = \begin{bmatrix} \cos(\beta) - \sin(\beta) \\ \sin(\beta) + \cos(\beta) \end{bmatrix} \begin{bmatrix} x_{c'} \\ y_{c'} \end{bmatrix} \tag{3.58}$$

Substituting $x_{c'}$ and $y_{c'}$ into Equation 3.58 gives;

$$\begin{bmatrix} x_{c''} \\ y_{c''} \end{bmatrix} = \begin{bmatrix} \cos(\beta) - \sin(\beta) \\ \sin(\beta) + \cos(\beta) \end{bmatrix} \begin{bmatrix} x_c - x \\ y_c - y \end{bmatrix} \tag{3.59}$$

In this case, the angle of the virtual line between origin and c'' should be equal to the heading error. Calculating this angle in the range of $-\pi$ to π by using the $atan2$ function;

$$e_\theta = atan2(y_{c''}, x_{c''}) \tag{3.60}$$

Hence, the heading error of the arc command is;

$$e_\theta = atan2((x_c - x)\sin(\beta) + (y_c - y)\cos(\beta), (x_c - x)\cos(\beta) - (y_c - y)\sin(\beta)) \tag{3.61}$$

Writing it explicitly for different sweep directions;

$$e_\theta = \begin{cases} atan2((x_c - x)\sin(-(\theta + \frac{\pi}{2})) + (y_c - y)\cos(-(\theta + \frac{\pi}{2})), & \\ (x_c - x)\cos(-(\theta + \frac{\pi}{2})) - (y_c - y)\sin(-(\theta + \frac{\pi}{2}))), & \text{for } \triangle\alpha > 0 \\ atan2((x_c - x)\sin(-(\theta - \frac{\pi}{2})) + (y_c - y)\cos(-(\theta - \frac{\pi}{2})), & \\ (x_c - x)\cos(-(\theta - \frac{\pi}{2})) - (y_c - y)\sin(-(\theta - \frac{\pi}{2}))), & \text{for } \triangle\alpha < 0 \end{cases} \tag{3.62}$$

3.4.2.4 Random Pattern Command

Random Pattern Command is a high-level command that uses low-level commands Line and Rotate. The robot starts its operation by a line command, and always checks boundaries in order to stay in bounds. If it came to a bound, it immediately stops the line operation and adds an urgent rotate command to its commands array's first order. The angle of this rotation is random, nevertheless it uses some algorithms to make its coverage performance improved.

When Random Pattern Command is running and a border is sensed, the robot follows this steps to calculate a random angle for rotating back to the yard and keep mowing in an efficient way:

- Takes the percentage of the position of currently stopped point on that edge. Say, the percentage is p.

- Calculates the position of the point on the left-adjacent-edge that has the same percentage of edge-length. Calls that point, point-L.

- Calculates the position of the point, this time on the right-adjacent-edge that has $(100 - p)$ percent of the edge-length. Calls that point, point-R.

- Calculates $atan2$ angles of point-L and point-R. Say, they are θ_L and θ_R respectively.

- Examine the signs of θ_L and θ_R, and compare them if θ_L is greater than θ_R or not. Choose the nearest rotation direction (CW or CCW).

In short, the robot generates and uses a random angle between θ_L and 2π or -2π and θ_R in order to use as the Rotate Command's $\delta\theta$ parameter.

3.4.2.5 Parallel Swath Command

Parallel Swath Command is a high-level command that uses low-level commands Line and Rotate in an appropriate sequence. Depending on the height and width parameters of the command, a specific area would be swept properly by many Line and Rotate commands. When the command initiates, several Line ve Rotate commands are added to the Commands-Array, which is the main part of the software architecture. Directions are automatically determined by the signs of two parameters.

Proposed Enhanced Navigation Method is applied for Parallel Swath Command in order to see the performance of that newly proposed method by several tests. Parallel swath algorithm requires an additional reference frame to be added for the enhanced navigation method succesfully applied.

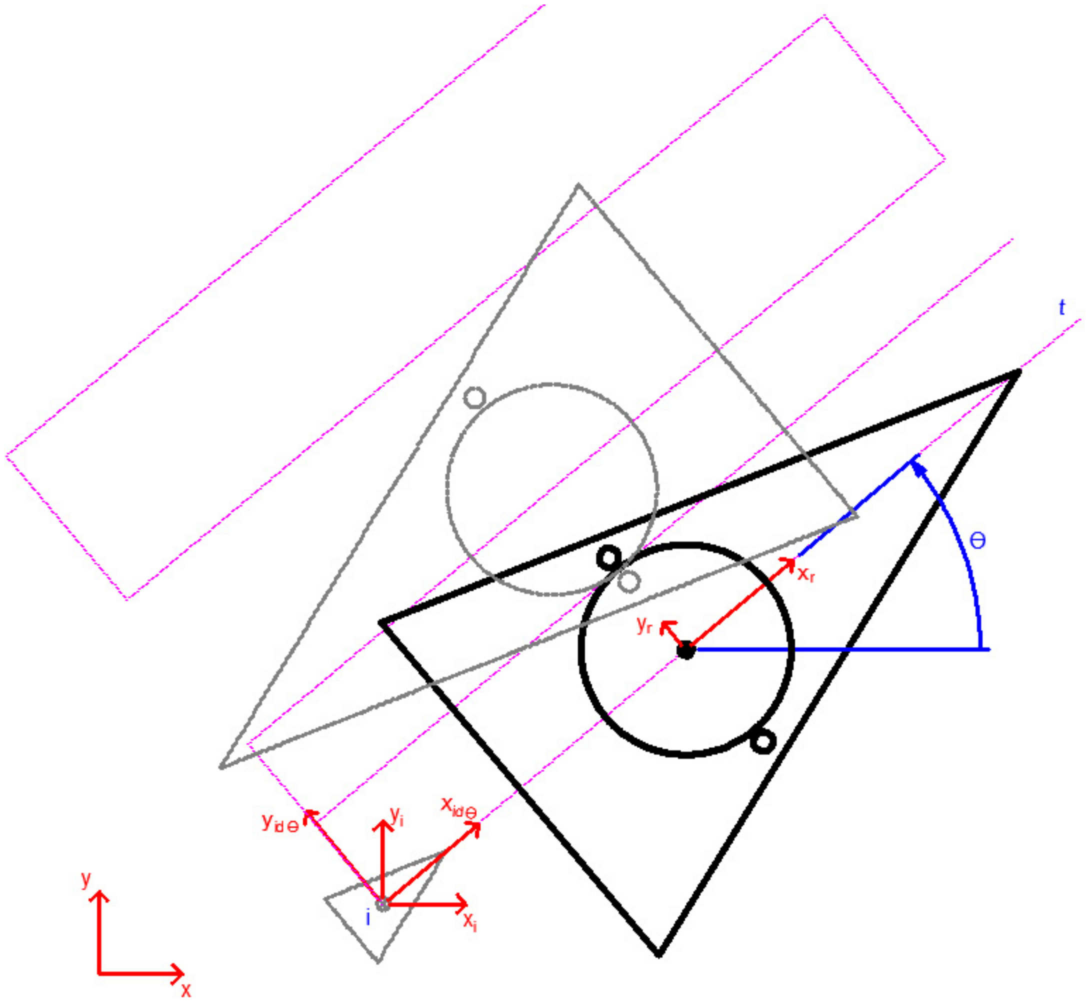


Figure 3.44: Parallel Swath Motion

3.4.2.6 Rectangular Inward Spiral Command

Rectangular Inward Spiral Command is a high-level command that uses low-level commands Line and Rotate in an appropriate sequence. Depending on the height and width parameters of the command, a specific area would be swept properly by many Line and Rotate commands. When the command initiates, several Line ve Rotate commands are added to the Commands-Array, which is the main part of the software architecture. Directions are automatically determined by the signs of two parameters.

3.4.2.7 Obstacle Avoidance

Obstacle Avoidance behaviour runs as an interrupt routine for motion planner to complete one command step sent by the trajectory planner. The methods that this behaviour uses are not derived and explained in detail.

3.4.2.8 Enhanced Navigation Method

Enhanced Navigation Method is a newly proposed method, that is developed and tested by the scope of the thesis. The method is actually based on a very basic approach: if we can sense the height differences of grass and know the accurate position information of an area that is already mowed, we can use this information as an external reference in order to use for correcting robot's position and heading angle.

When a high-level command is running, the robot always knows the exact position information of previously-completed commands. For instance, if a Parallel Swath Command is running, at the begining of the third line (second line is just for the blade-width offset), robot knows that it should be going parallel to the previously-completed first line. Similarly, at the begining of the fifth line (fourth line is again just for the blade-width offset), robot knows that it should be going parallel to the previously-completed third line. Therefore, the robot can check the height of grass under it, either left or right side depending on the previous commands.

In order to sense the height of grass, in other words, to check if the area has been already mowed or not, Infrared Switches are used. An IR switch consists of an IR transmitter and an IR receiver components. The gap between those two, used for placing the object to be sensed between the transmitter and the receiver. If the IR receiver receives the IR light transmitted from the IR transmitter, IR switch returns "false" meaning that there is no object in between the gap, else it returns "true" meaning that there is an object in between the gap.

Grass is a really hard-to-detect object relatively. Trying to sense the grass by

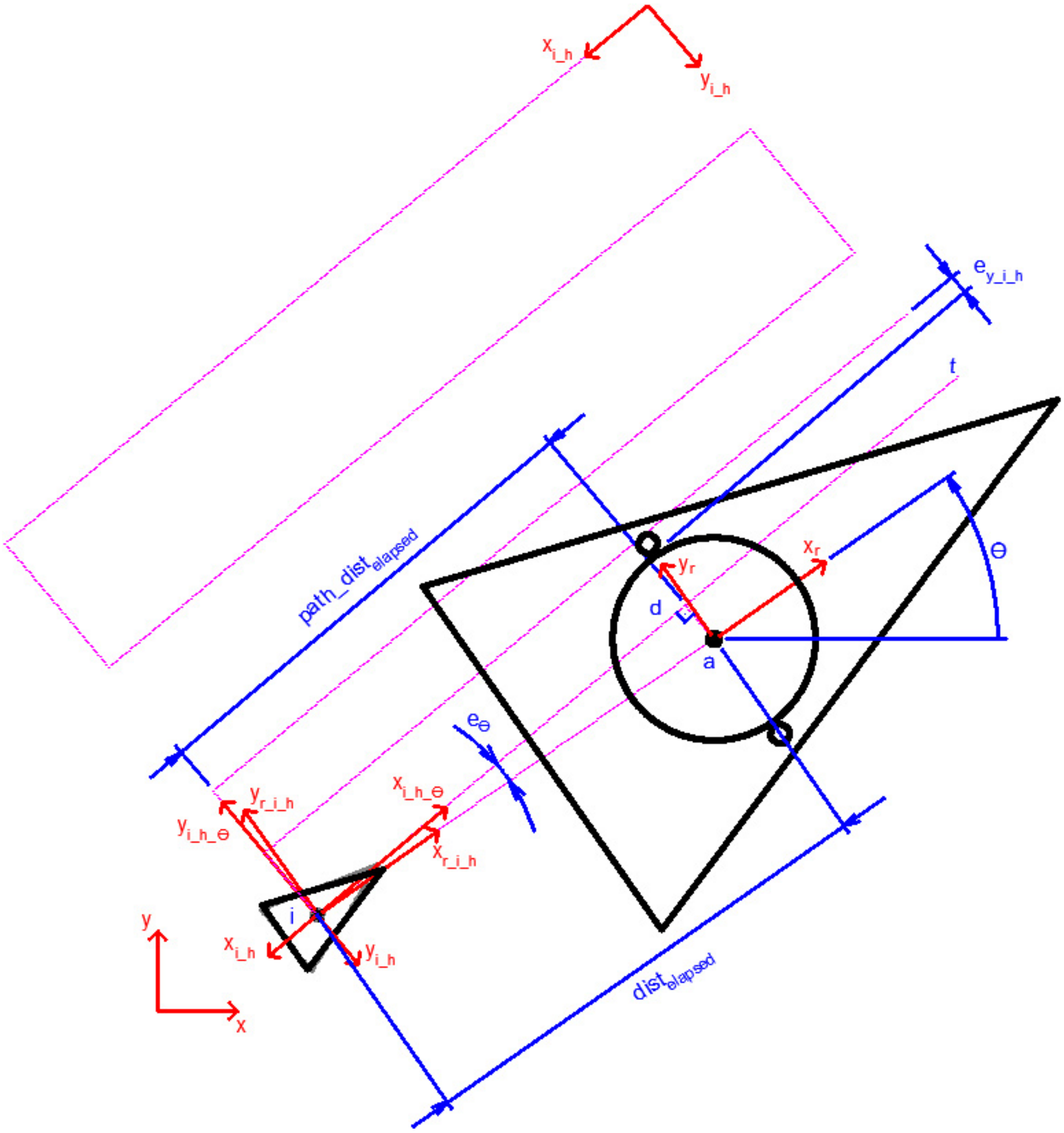


Figure 3.45: Slippage-Related Errors in Parallel Swath Motion

using an IR switch gives a really noisy-like data. In order to use the data as a reliable information, taking average approach is used in the application. Implicitly, we continuously gather the switch status information, and use the average value of last 5 seconds for "grass detection" decision. At that point, the answer of "What is the true ratio that the average should be in order to accept the area as mowed?" question is "It depends.". That is to say if your backyard has intense grass, the ratio limit can be taken a low value, however if you have sparse grass, the ratio limit should be taken higher. In our tests, appropriate values for our test areas are used. An adaptive ratio limit approach would be applied as a future work.

Mathematically, a minimum-position-error estimation approach is used for that method. Meaning, if the robot saw non-mowed grass at an area that it should not be seen, it can consider that it has made a lateral position error of at least the width of IR switch gap. In other words, the minimum distance the robot should shift laterally for the IR switch to see the mowed area is surely equal to the width of IR switch gap. After making this estimation, related heading error can be calculated by using the elapsed distance of that line command. Estimating those position and heading angle errors, the robot updates its current global position and heading angle information. [8] In other words, robot uses the previously mowed area as an external reference in order to correct its dead-reckoning based position and heading angle information. The robot checks that areas just for the main lines of Parallel Swath motions in this thesis.

Figure 3.45 shows the subjected error values. For simplicity, the reference frame of Parallel Swath command is not used in calculations here. Considering the physical dimensions, we can know what a minimum position error estimation value should be for our platform. In other words, it is basically a constant value as;

$$e_{s_y} = 1[cm] \tag{3.63}$$

Position controller directly adds that value to the y-component of the global position. Related heading error can be calculated by using the elapsed distance of the line, and can be written as;

$$e_{\theta} = \sin^{-1}(\frac{e_{s_y}}{distance_{elapsed}}) \tag{3.64}$$

in radians. Position controller directly adds that value to the current heading angle of the system. [3] Last, related x-component of the position error can be calculated again by using the elapsed distance of the line and e_{s_y} value, and can be written as;

$$e_{s_x} = \sqrt{(distance_{elapsed})^2 + (e_{s_y})^2} \tag{3.65}$$

in centimeters. Again it is directly added to the x-component of the global position. At the end, robot has corrected its global position and heading angle information by using a relatively-known external reference.

3.5 Controller Design

There are three controllers used in the system architecture. One of them is the wheel speed controller, which always controls the velocities of drive motors if they are running at the desired speed or not. Other two are the position and heading angle controllers used by the motion planner.

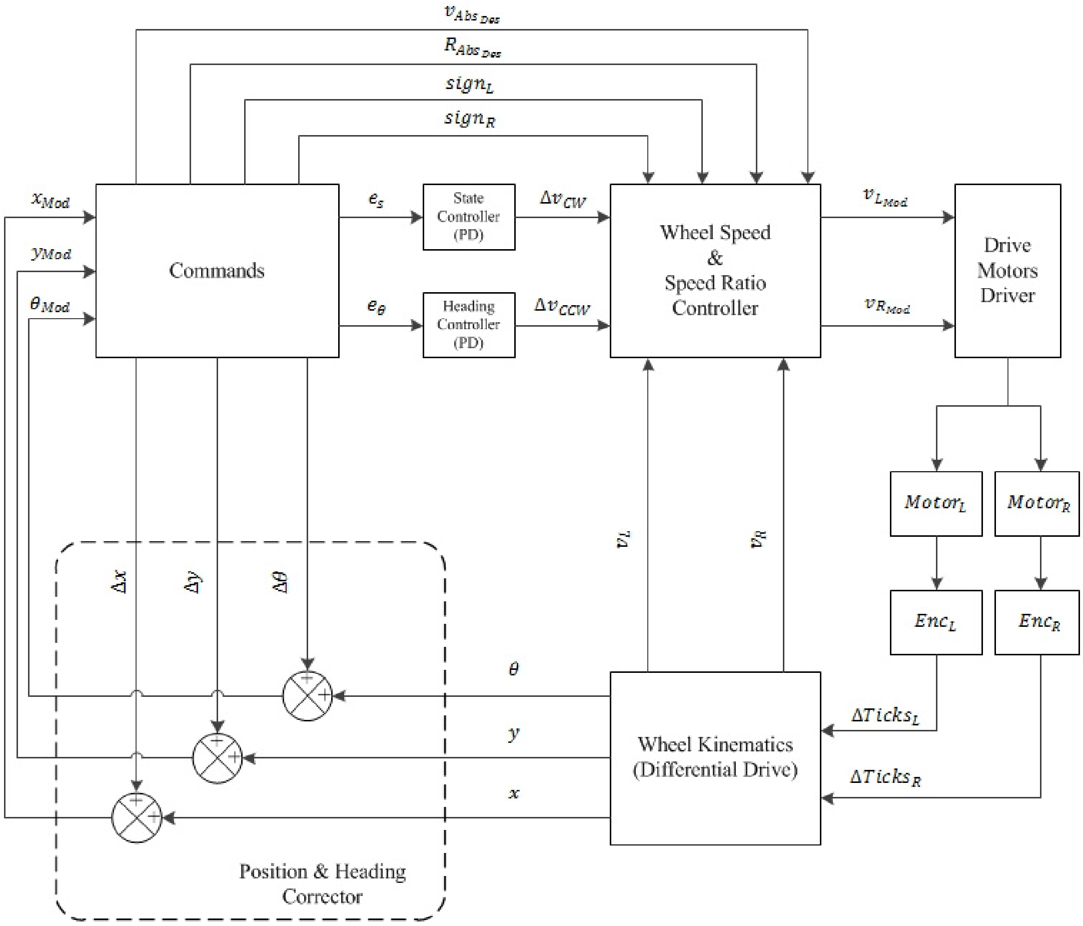


Figure 3.46: Main Block Diagram of the System

Wheel Speed Controller Wheel speed controller uses the difference between the desired and the actual speeds of a wheel as the speed error. It is actually an I-Controller allowing the robot to accelerate and decelerate quite smoothly. Proportional gain could also be added to this controller, however the main issue here is not trying to reach the desired value immediately or keeping the actual speed at desired value accurately, but just accelerating or decelerating the robot to the desired velocity smoothly. Regarding the derivative gain, since the system does not have acceleration and deceleration inputs (functions) particularly, in other words since the speed input has sudden changes, it is not convenient to use the derivative gain for that controller.

Position and Heading Angle Controllers Position and heading angle controllers use the position error and the heading angle error values in order to calculate how much speed needs to be added to a wheel speed. Here, both of them are PD-Controllers. Integral gain is not used here, because the need for the existing integral value obsoletes after the error has zeroized. Position error is zeroized when the robot is reached to the desired trajectory, and heading error is zeroized when the robot heading is reached to the desired heading angle. [16] [10]

As can be seen in the derivations in Mathematical Modelling section, a positive $\triangle v$ value means that the robot should rotate a little bit clockwise in order to steer for the desired trajectory, and a negative $\triangle v$ value means that the robot should rotate a little bit counter-clockwise in order to steer for the desired trajectory. What the position and heading angle controllers do is to calculate this $\triangle v$ values on their own way. Afterwards, we subtract the $\triangle v$ value came from the heading angle controller from the $\triangle v$ value came from the position controller. This operation gives us the resultant $\triangle v$ so that it can be added to the absolute desired wheel speed in order to steer a little bit. This modified absolute speed will then be used in the wheel speed controller to be compared with the actual wheel speed for speed error calculation.

In additon to speeding up one wheel, the wheel speeds ratio should be taken into account. Overall system architecture is based on speeding one wheel up

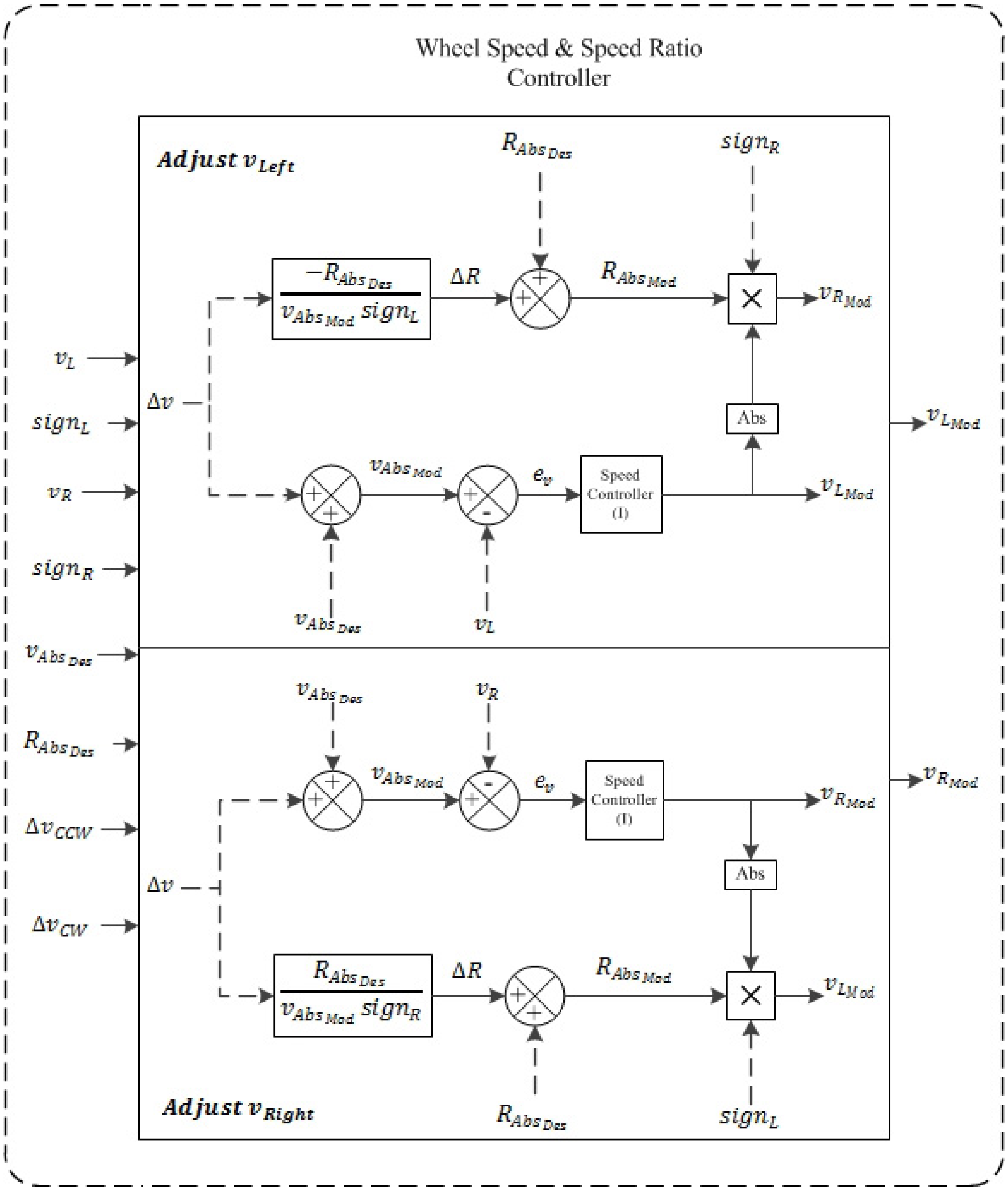


Figure 3.47: Wheel Speed and Speed Ratio Controller

while keeping the speed of the other absolutely at the desired speed. Since all the physical dimensions are definite, it can easily be calculated how much the modified ratio should be in order to keep one wheel's speed as is, while speeding the other wheel up. So, what we do here is calculating a $\triangle R$ value using the controllers' output ($\triangle v$), and adding this $\triangle R$ value to the desired absolute Ratio came from the currently running command. Depending on the system architecture, all those calculations are done by using absolute values, because the directions of the wheels are defining by commands itselves.

Naturally, the system is open for a blow-up since the position error value may go to infinity. Clearly, the robot may be say, 1 meter away from the desired trajectory. This type of a case causes the error value, and accordingly the required $\triangle v$ value goes to physically-impossible-to-do levels. In order to avoid from this type of blow-up situations, an "Error Limiter" approach is used for the wheel speed controller. The Error Limiter limits the calculated $\triangle v$ to a value that can speed the wheel up at most 50% of the absolute desired speed.

One of the most important difficulties arises with that question: "Okay, the robot is following a straight line with many errors, but at which exact target point we should say the robot to stop?". Indeed, the robot probably will never be reached to the exact target point in real life. We have to provide a solution, rather an approximation for the robot to be able to think that: "Even if I am not exactly on the desired target point, I had had better finalize the currently running command now, and initiate the next one.". So, we have two approximations there. One of them is a tolerance value. All commands are finalized when the robot is reached a specific region (in the end, a little circle) that the desired target point and the specified tolerance value creates. However, the robot may not be reached to this little region forever. The other approximation comes into play at this point. It is using the remaining distance value (or remaining angle for rotations). For the line command for instance, remaining distance value getting lower and lower during the whole command running time. However, if the robot is very close to the target point but not in the tolerance region, it passes through the point while keeping the command running. If we consider the remaining distance value, we see that it starts to increase at a very close point to the target point. At exactly that moment, robot can finalize the currently active command, and initiate the next one.

Both of the command-finalization point approaches have an keeping-error issue. Namely, the position error has been kept, and will be reflected to the next command. In order to avoid this error-transfer between commands, an "Initial Value Overwrite" approach has been used. It keeps the target point and the target heading-angle information of the lastly-completed command, and transfers these information to the newly-initiating command so that the desired and target values for this new command could be calculated by using exactly the values we want theoretically. This property allows the robot to follow a perfect rectangle created by many commands, or to complete high-level tasks like Parallel Swath and Rectangular Inward Spiral successfully. [12]

3.6 Software Architecture

Considering a lawn mower, it is impossible to use a pre-defined region even to give the construction of the software architecture a start. Naturally, all of our backyards are completely different from each other in terms of both shape and size. Additionally, they all have many static and dynamic abstacles obviously differing with all aspects. There are some illustrations in Figure 3.48 that show some relatively easy-to-imagine gardens need their lawns to mow, and virtual maps schematically drawn on them.

As it can easily be understood, it is really hard to cover a random lawn area for an ALM, and it is also so hard to handle this problem mathematically even for us. It would be better divide the problem into smaller pieces. Based on this approach, an area to be covered can be handled as if it consists of just two primitive objects: line and arc. Using a primitive *line* and a primitive *arc* command with an additional *rotate (by its own-axis)* command with the width of mower blade, any lawn area can theoretically be covered.

Although these three main commands are considered to be enough when the coverage is the issue, dealing with this type of low-level commands is definitely not reasonable. Rather, defining some high-level commands that use these three primitive commands, and always dealing just with them is surely sensible. Therefore, three main high-level coverage commands are defined in this project: *Rectangular Inward Spiral*, *Parallel Swath* and *Random Pattern*. Surely, some behaviour based commands will also be included such as an *Obstacle Avoidance*, a *Turn Back to the Charge Station* or a *Go-to-Goal* command in the future of

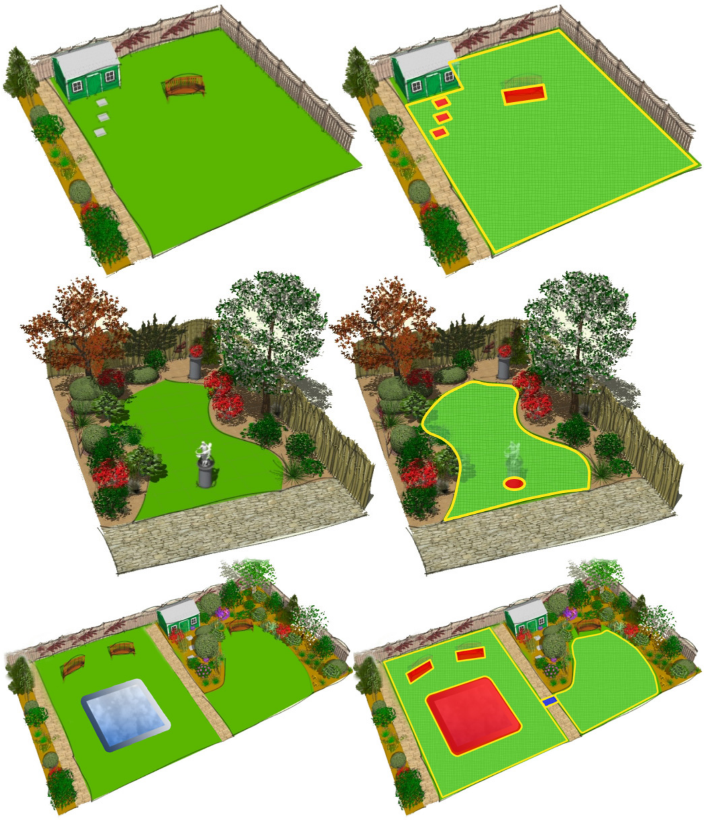
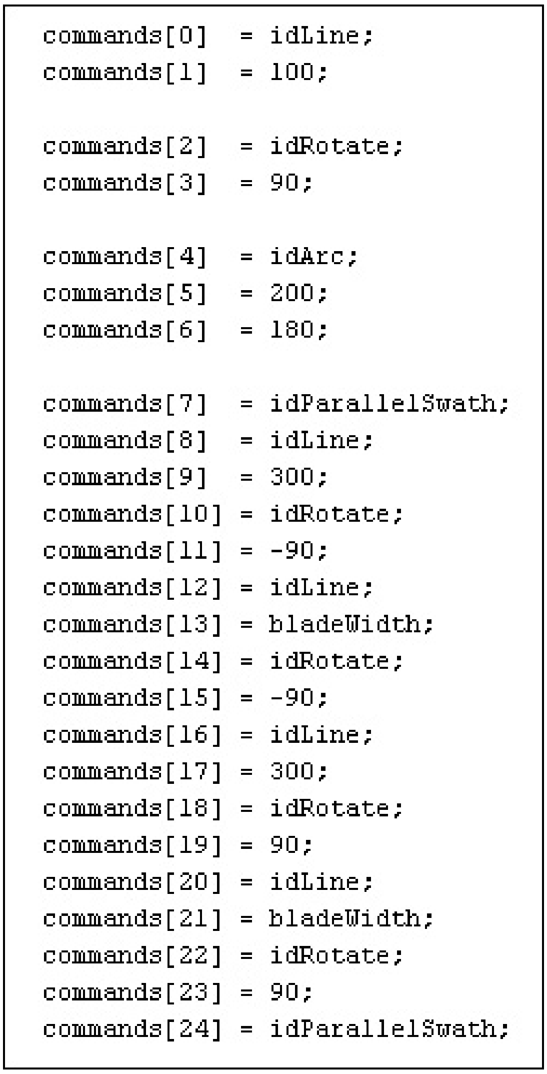


Figure 3.48: Sample Mapping Schematics

this project, however those behaviours are not necessary for the scope of this thesis right now. Both low-level and high-level available command definitions will be explained in the following sections.

As the whole software architecture of this project is considered, there is mainly a *trajectory planner* that just handles with high-level commands, and a *motion planner* that is responsible for running low-level commands successfully. *Obstacle Avoidance* will be running as an interrupt routine for motion planner to complete one command step of the trajectory planner. If there is no abstacle on the road, motion planner works as if it is just a path controller. Additionally, there is a low-level controller which always controls the velocities of both traction motors if they are running at the desired velocities or not. They will also be explained in detail in the following sections.

If all those low-level and high-level commands are considered in the sense of a programmer, it probably be thought that there should be a list that includes all the commands to be performed by the robot in sequence. As can be seen in the listing of software, there is a main 1-dimension array called *Commands-Array* used to handle this objective. The main timer in software, checks if Commands-Array has a command required to be initiated or required to be kept running in each time interval. If there is no command in the array, the robot automatically adds one (Random Pattern Command) if the autorun option is already activated by user. If the array has one or more commands in it, the first ordered command, whether it is just a low-level or a high-level command, is automatically be initiated. After completion of a command, a "Finish Command" function removes the first command from the Commands-Array, and reorder all elements of that array automatically.



```
commands[0]  = idLine;
commands[1]  = 100;

commands[2]  = idRotate;
commands[3]  = 90;

commands[4]  = idArc;
commands[5]  = 200;
commands[6]  = 180;

commands[7]  = idParallelSwath;
commands[8]  = idLine;
commands[9]  = 300;
commands[10] = idRotate;
commands[11] = -90;
commands[12] = idLine;
commands[13] = bladeWidth;
commands[14] = idRotate;
commands[15] = -90;
commands[16] = idLine;
commands[17] = 300;
commands[18] = idRotate;
commands[19] = 90;
commands[20] = idLine;
commands[21] = bladeWidth;
commands[22] = idRotate;
commands[23] = 90;
commands[24] = idParallelSwath;
```

Figure 3.49: Commands-Array Architecture

A command may reserve more than one element of the array. For instance, the

Line command needs for two elements; one is required for the command-ID, and the other is required for the only parameter of the Line command, length. Likewise, since it has just one parameter Rotation Angle, Rotate command also uses (reserves) two elements of the array. Two parametered commands like the Arc command, surely needs 3 elements of the array to be reserved. High-level commands use Commands-Array in a quite different way. They reserve one element for the command-ID at the beginning of the high-level command, and additional one element at the end of the high-level command. Elements between them of course reserved for the low-level commands to be run under that high-level command. The number of those low-level commands under high-level commands depends on the area to be covered by that high-level command.

There are three main motion functions on the system. They are line, arc and rotate functions. Kinematic equations of these three main functions derived and explained in detail before. Related low-level commands are defined in the software architecture as follows. Also the architecture of trajectory planning high-level commands such as Rectangular Inward Spiral and Parallel Swath are explained below. Additionally, proposed enhancement technique is explained below. Namely, Parallel Swath command is now has two modes of which their results will be compared after the tests. One of them is running based on the kinematic equations only, and the other one is taking the status of grass near to the blade into account if it is mowed or non-mowed.

The software architecture has a main Timer object, and all of those commands or other calculations run in each time interval. It is actually runs as a timer interrupt routine, interrupting the software in each time interval. This time interval is already the dt term used in all derivative, integral or wheel speed calculations. Second interrupt routine of the software architecture is actually a hardware interrupt, used by encoder ticks. Both encoders interrupt the slave controller, so that the ticks can be counted successfully. Slave controller transmits the delta tick values to the master controller when it is requested. It is already requesting in each time interval from the slave, so that the commands can use updated value of position, heading angle and wheel speeds information.

For all the low-level commands below, if the "Initial Value Overwrite" option is activated, then all the initialization calculations would be done by not using the actual initial values, but the previous command's target point coordinates and heading angle.

Again all commands below, check if the robot has been reached to the target point (within the conditions explained in Controller Design section) in each time interval. If it has been reached, the command calls the "Finish Command" function defined in the software. Lastly, this function stops the motors, makes all integral terms zero, and remove the completed command's information from the Commands-Array.

3.6.1 Line Command

Command Syntax: lin(l)

where;
l : desired length of the line to be tracked

Running the Line command first time; absolute desired wheel speeds ratio, target point coordinates and desired heading angle for all through the straight line are calculated depending on the actual (initial) position and heading angle values.

Keeping the Line command runnning by calling this function again and again in sequential time intervals, the position error and the heading angle error values are continued to be calculated.

3.6.2 Rotate Command

Command Syntax: rot($\triangle\theta$)

where;
$\triangle\theta$: desired rotation angle for the heading of the robot

Running the Rotate command first time; absolute desired wheel speeds ratio, target point coordinates and target heading angle values are calculated depending on the actual (initial) position and heading angle values.

Keeping the Rotate command runnning by calling this function again and again in sequential time intervals, the position error value is continued to be calculated.

3.6.3 Arc Command

Command Syntax: arc($r_d, \triangle\alpha$)

where;
r_d : desired radius of the arc to be tracked
$\triangle\alpha$: desired sweep angle of the arc to be tracked

Running the Arc command first time; absolute desired wheel speeds ratio, target point coordinates and desired heading angle for all through the arc-shaped path are calculated depending on the actual (initial) position and heading angle values.

Keeping the Arc command runnning by calling this function again and again in sequential time intervals, the position error and the heading angle error values are continued to be calculated.

3.6.4 Random Pattern Command

Command Syntax: rPa()

One of the high-level commands is Random Pattern. This trajectory planner uses low-level commands Line and Rotate in a suitable sequence.

Running the Random Pattern command first time; necessary low-level commands for the desired trajectory to be successfully passed, are calculated depending on the actual (initial) position and heading angle values. Those commands are created, and will automatically be added to the Commands-Array in order. Additionally, this function removes command's initialization information from the Commands-Array.

After all the related low-level commands have been done, this function removes command's finalization information from the Commands-Array.

3.6.5 Parallel Swath Command

Command Syntax: pSw(h, w)

where;
h : desired height of the area to be mowed
w : desired width of the area to be mowed

Parallel Swath is an other high-level command. This trajectory planner also uses low-level commands Line and Rotate in a suitable sequence.

Running the Parallel Swath command first time; necessary low-level commands for the desired trajectory to be successfully passed, are calculated depending on the actual (initial) position and heading angle values, and given height and width parameters. Those commands are created, and will automatically be added to the Commands-Array in order. Additionally, this function removes command's initialization information from the Commands-Array.

After all the related low-level commands have been done, this function removes command's finalization information from the Commands-Array.

3.6.6 Rectangular Inward Spiral Command

Command Syntax: iSp(h, w)

where;
h : desired height of the area to be mowed
w : desired width of the area to be mowed

Rectangular Inward Spiral is an other high-level command. This trajectory planner also uses low-level commands Line and Rotate in a suitable sequence.

Running the Rectangular Inward Spiral command first time; necessary low-level commands for the desired trajectory to be successfully passed, are calculated depending on the actual (initial) position and heading angle values, and given height and width parameters. Those commands are created, and will automatically be added to the Commands-Array in order. Additionally, this function removes command's initialization information from the Commands-Array.

After all the related low-level commands have been done, this function removes command's finalization information from the Commands-Array.

CHAPTER 4

MANUFACTURING AND SYSTEM INTEGRATION

4.1 Manufacturing

In this section, manufacturing of the mechanical components of ALM is briefly presented.

The external cover of the ALM is designed to be made by vacuum forming process. Vacuum forming is a specific form of thermoforming, in which a plastic sheet is heated to the forming temperature and stretched over a concave (female) or convex (male) mold. Afterwards, the air between the mold and the sheet is vacuumed onto the surface of the mold to shape the plastic precisely. In this study, a wooden mold was manufactured by conventional 3-axis CNC milling for this purpose.

For manufacturing the desired outer cover of the ALM, a wooden mold has been manufactured by conventional 3-axis CNC milling in the first place. Afterwards, this male mold is placed in the vacuum forming machine and preheated to the desired temperature in order to form the plastic sheet. Some lubricant is also applied onto male mold to make the mold-plastic separation easier.

A hard polyethylene plastic sheet is trapped in a fixed frame in the vacuum forming machine and heated up to the forming temperature afterwards. When heated sheet started to loosen, the male mold is lifted and the sheet is thereby stretched by the mold.

Finally, a pump is activated and the air between mold and sheet is vacuumed

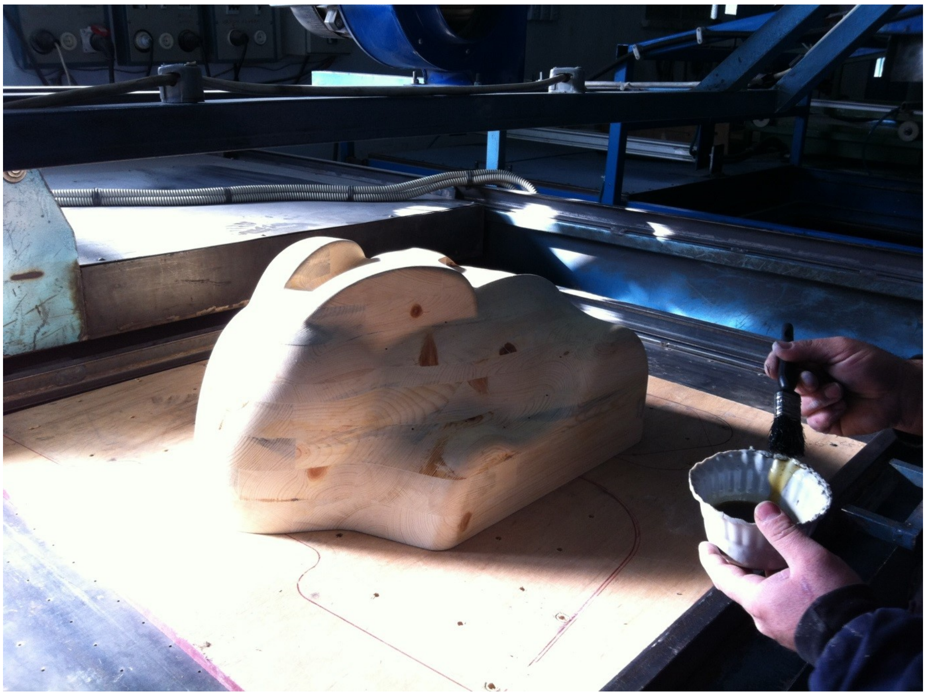


Figure 4.1: Wooden Mold of the Outer Cover

out. With the help of some manual intervention, the polyethylene sheet finally gets the exact form of the male mold.

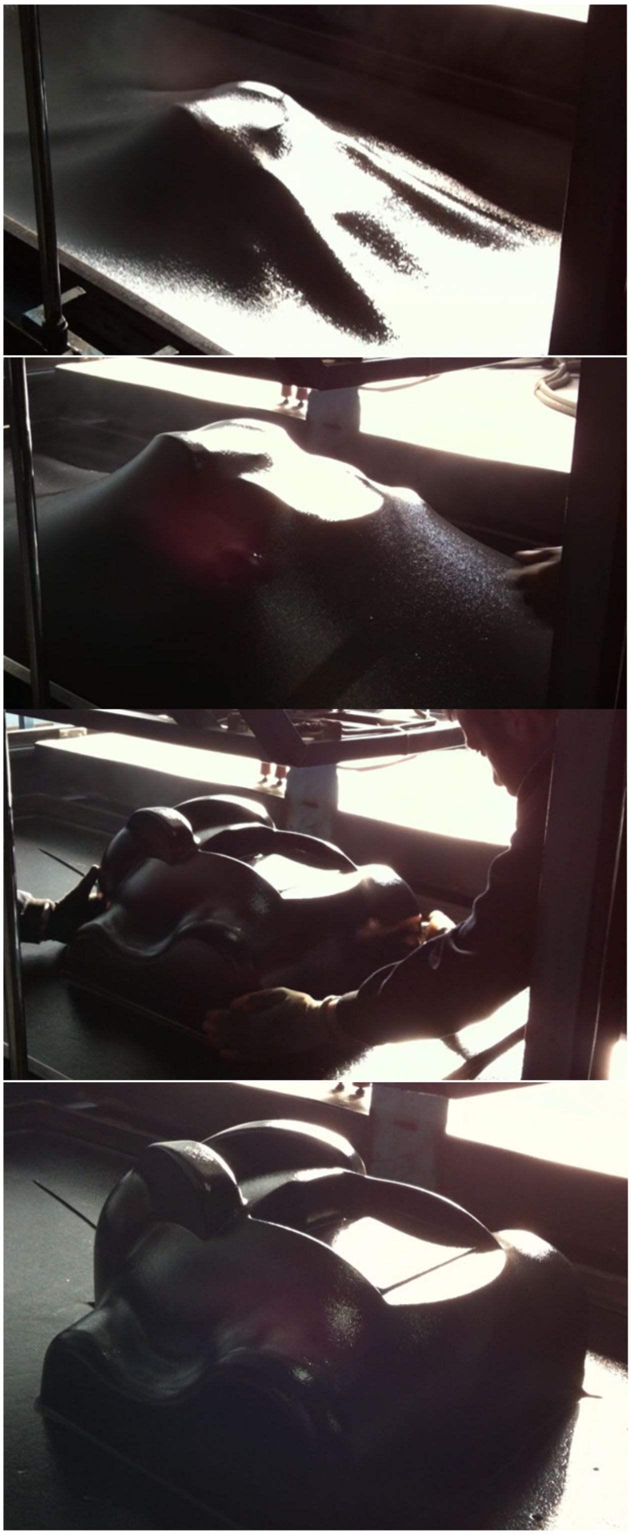
The stages of vacuum forming of outer cover are presented in Figure 4.2.

The cover is produced in various colors and textures. Final decision is made for a white colored cover, which is selected to emphasize the rabbit shape. No coating or painting is applied afterwards. The final shape of the outer cover is given in Figure 4.3.

Except for the outer cover, all of the mechanical components of the ALM are produced by conventional manufacturing methods for this first prototype. The main body, rear cap and US sensor enclosures are produced by Delrin plastic material, by 3-axis CNC milling.

All of the solid aluminum components are also produced by CNC milling and/or turning. The sheet metal aluminum parts have been laser machined, press-formed and corner welded. Besides aluminum sheet metal parts, welding is not used for assembling any of the system components. Therefore, all mechanical components are modular as desired.

After manufacturing, all aluminum parts are coated with white colored eloxal-coating. Drive wheels and cutting blades are protected from oxidation by hot dip galvanization.



Stage 1

Plastic sheet is heated up to forming temperature and mold is lifted upwards.

Stage 2

Male mold stretches the plastic sheet and gives its shape partially.

Stage 3

Air between mold and sheet is vacuumed out and some critical corners have been aligned manually.

Stage 4

The plastic sheet is cooled down and final shape is obtained.

Figure 4.2: Vacuum Forming Stages of the Outer Cover

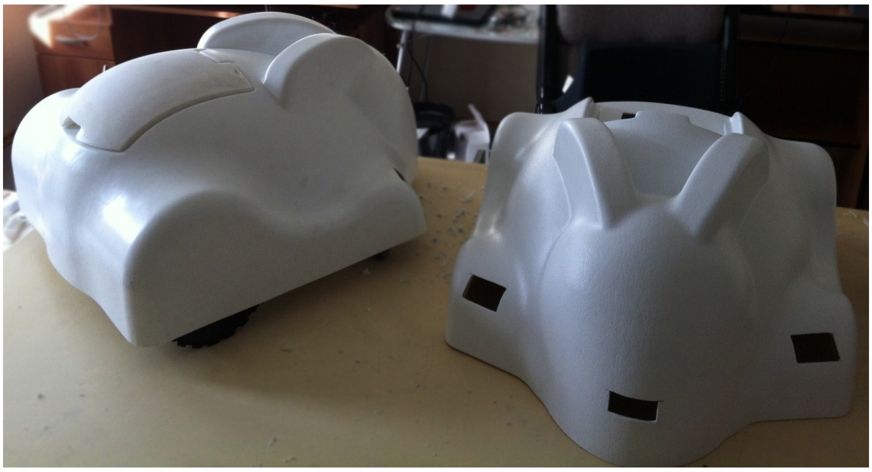


Figure 4.3: Manufactured Outer Cover

Some of the manufactured components are presented in Figure 4.4.

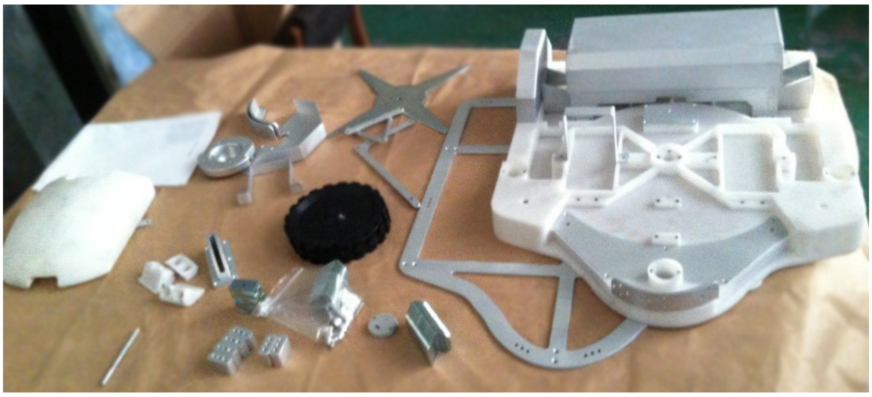


Figure 4.4: Some Manufactured Components

4.2 System Integration

After the completion of manufacturing processes, the mechanical body is assembled first. Necessary electromechanical hardware is placed on the body in advance. "Drive motors", drive wheels, idle wheels, "mower motor", cutting height adjustment mechanism, IR sensors and necessary covers are mounted onto the main body by using bolted connections.

The external components of the cutting height adjustment mechanism are mounted onto the outer cover afterwards. Then, US sensors are placed to their locations on the outer cover.

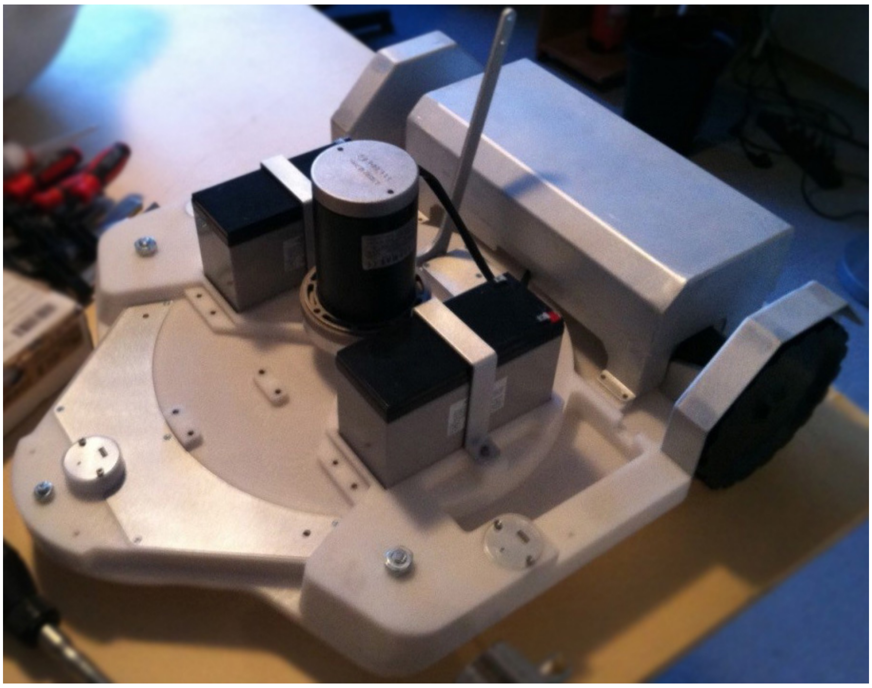


Figure 4.5: Mechanical and Electromechanical Integration on Main Body

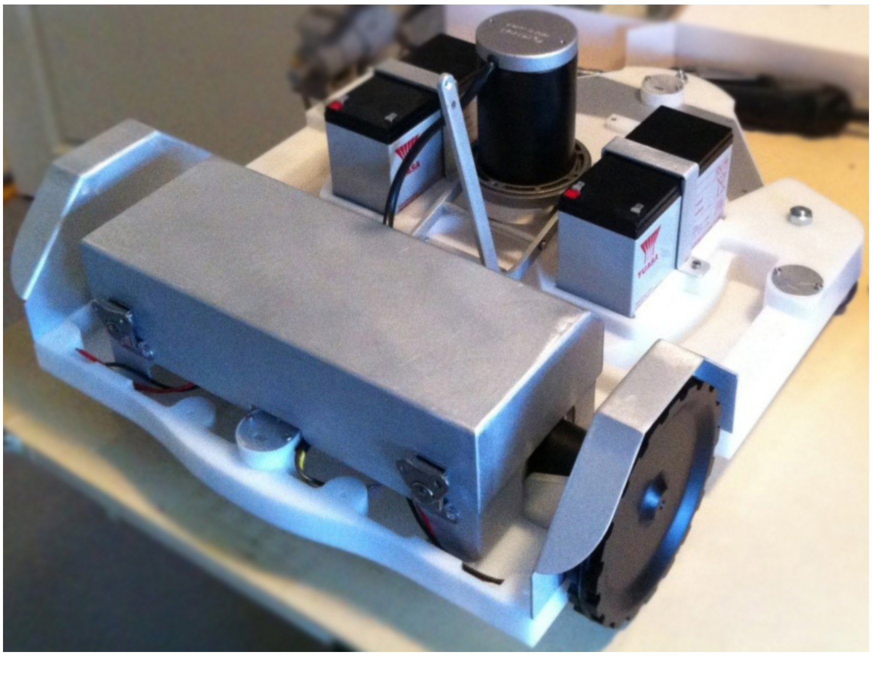


Figure 4.6: Mechanical and Electromechanical Integration on Main Body

After the completion of the integration of mechanical subsystems and sensors, the outer cover is bolted to the main body. Afterwards, the cutting blade is assembled and fixed by using its particular lock-pin.

The completed mechanical assembly of the ALM is given in Figure 4.9.

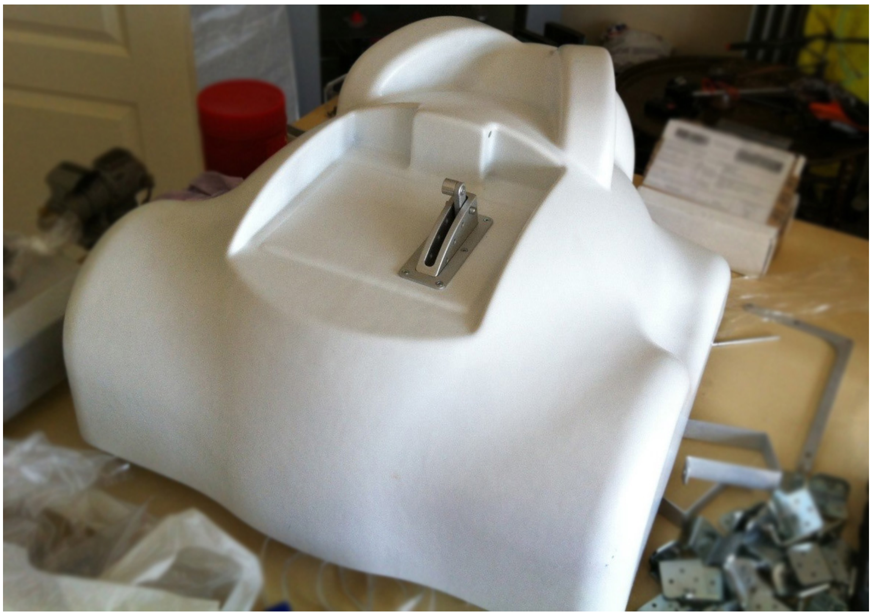


Figure 4.7: External Components of the Cutting Height Adjustment Mechanism

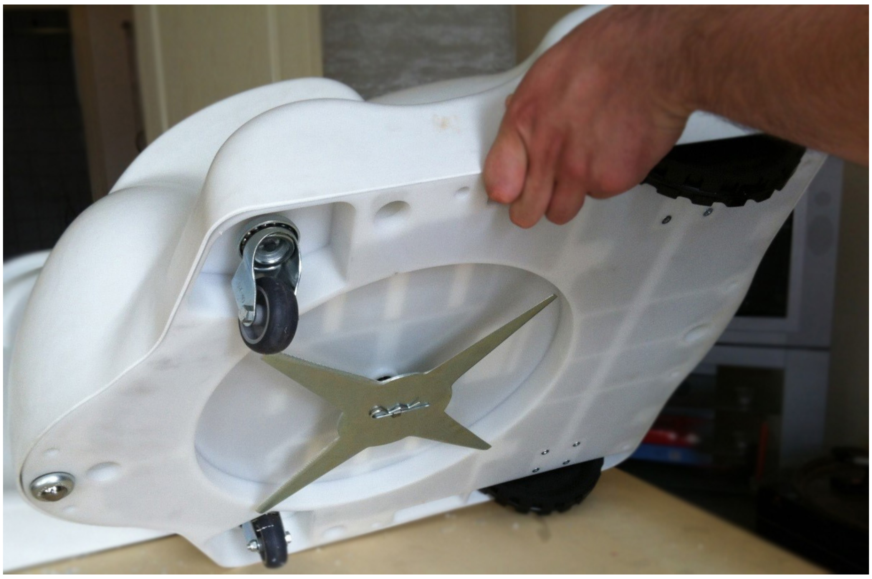


Figure 4.8: Cutting Blade Integration

Integration of all electronic components have been completed by using a reasonable harness structure for a prototype.

The final assembly of the product with all components is presented in Figure 4.11.

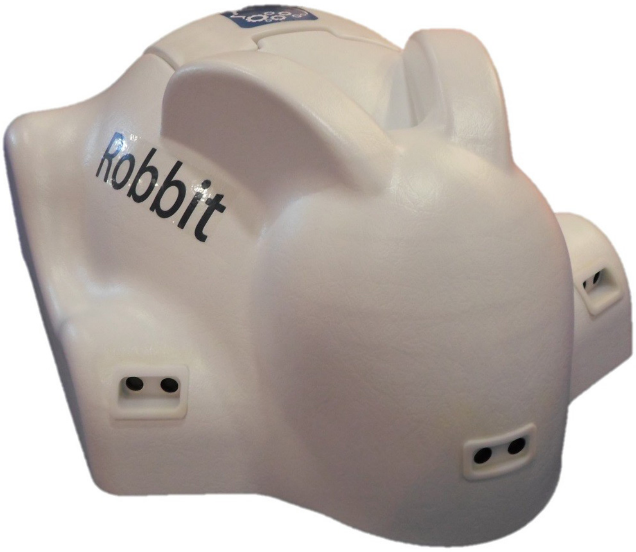


Figure 4.9: Mechanically Assembled Autonomous Lawn Mower

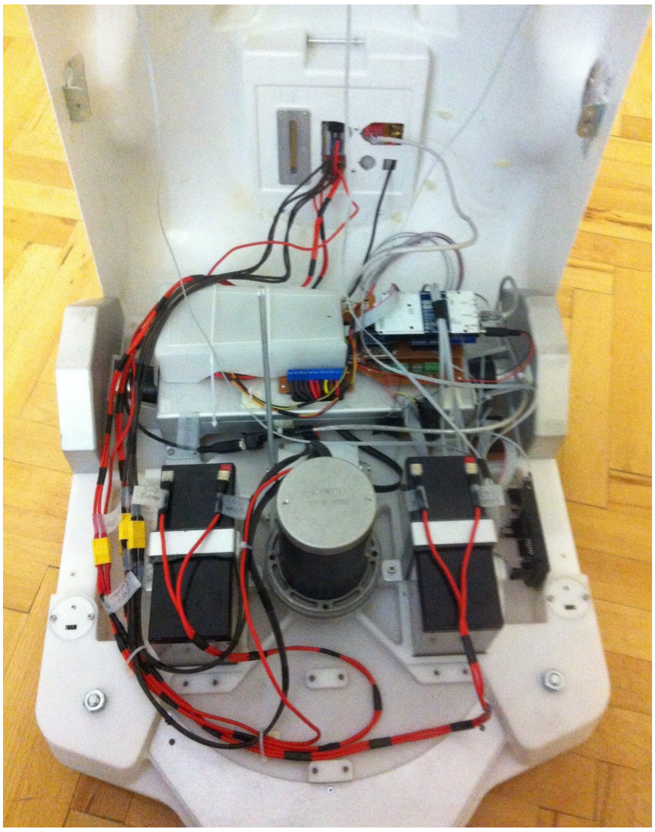


Figure 4.10: Internal View

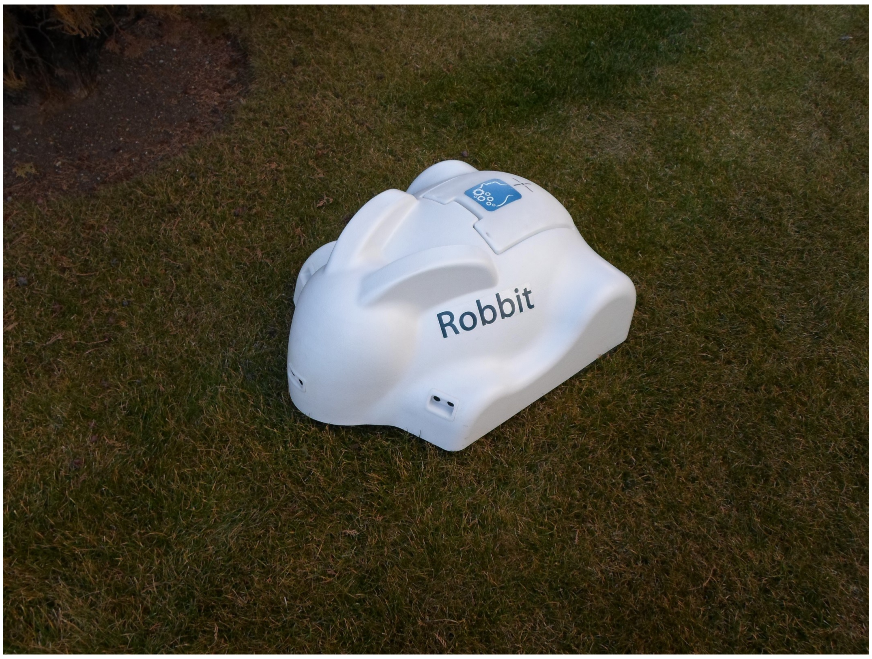


Figure 4.11: Final Assembly

CHAPTER 5

TESTING

This chapter is focused on the comprehensive physical testing of the ALM. The tests reveal the performance of the developed system and the proposed position correction technique. Test scenarios, test setups and performance comparisons for varying parameters are mentioned in detail below.

Before all performance tests, both unit and system level functionality tests have been completed. The functionality of all hardware and software are proved. The details of unit and system level functionality tests are not explained in the thesis, since the core aspect of this work is to reveal the operational success (i.e. the coverage performance) of the ALM.

5.1 Test Scenarios and Aspects

In order to evaluate navigation and coverage performances of the ALM, test scenarios are derived in logical and hierarchical manner. The main aspect of physical testing is to determine coverage performance of the ALM for different geometrical patterns and for different terrain conditions. For this purpose, some major geometrical patterns like rectangular inward spiral, parallel swath and random patterns have been executed on different terrain configurations for indoor and outdoor tests.

In this controlled set of experiments concept, physical tests are divided into two subgroups; indoor and outdoor tests.

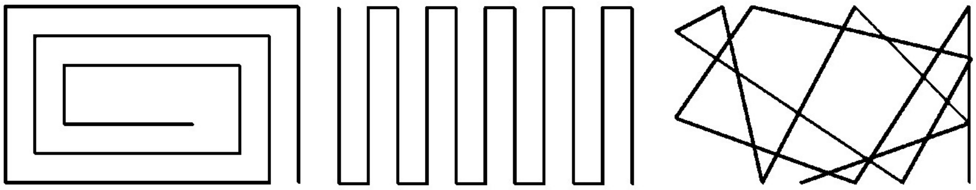


Figure 5.1: Illustrations of Rectangular Inward Spiral, Parallel Swath and Random Coverage Patterns

Heading and positioning accuracy of the ALM is intended to be determined in an ideal, non-erroneous environment. Indoor flat floor tests have been made for this purpose. With these tests, the success of the control architecture of the robot is proved and all of the controller parameters have been adjusted for the desired driving speed of the ALM.

In indoor tests, it is mainly aimed to determine the deviation from the desired trajectory, amount of slippage and corresponding coverage percentage for the ALM. Indoor tests are also divided into two subgroups. They are performed on flat parquet floor and on flat synthetic grass. For each coverage patterns, operation completion time and coverage percentages are calculated and compared.

All of the physical indoor and outdoor tests have been decided to be performed in an approximately 5 m^2 area. The reason of this selection and the dimensions of the physical area are explained in the next section.

As stated before, it is aimed to reveal trajectory-following behavior and coverage success of the ALM in indoor tests. For this purpose, the desired trajectories and the mowing area for three different coverage patterns have been presented. Random pattern on the other hand, inherently has no desired trajectory.

For each coverage pattern, operation completion time and coverage percentages are also calculated and compared. For inward spiral and parallel swath coverage patterns, a single sample is taken from set of tests and presented in this study. However, three samples have been chosen and averaged for random pattern operations, since the random pattern algorithms acts randomly all the time and there are no certainties for coverage percentage and operation completion time.

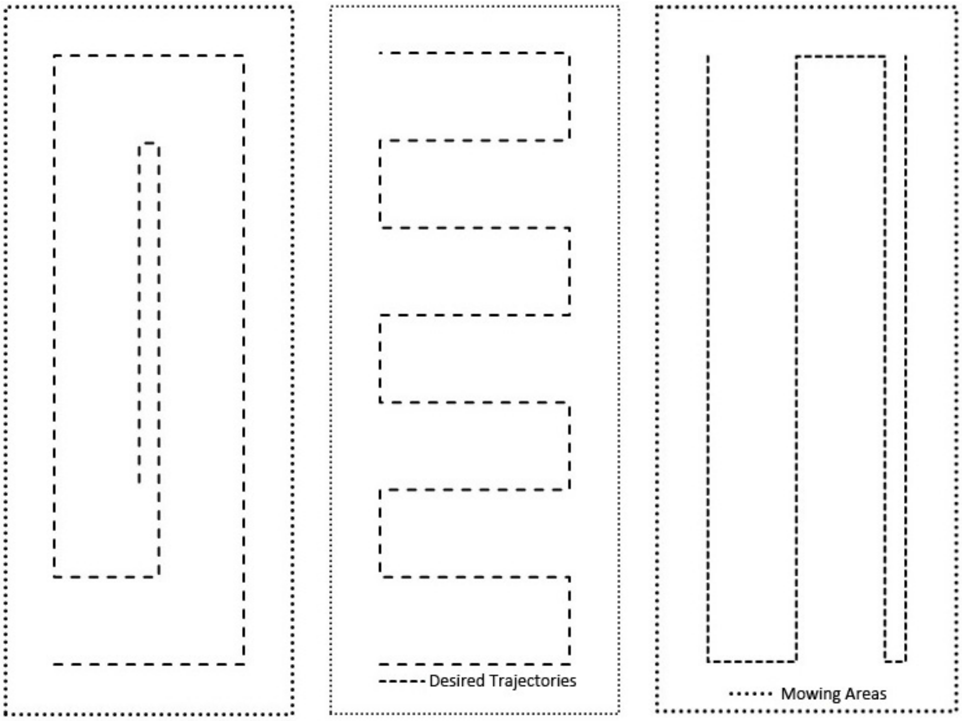


Figure 5.2: Desired Trajectories (Scaled)

With the flat parquet floor and synthetic grass indoor tests, the navigation characteristics, coverage percentages and operation completion periods of the ALM are desired to be obtained.

After determining response characteristics of the ALM for different terrains and patterns in indoor environment, the last set of tests are executed on outdoor grass terrain. Although synthetic grass terrain simulates the real grass terrain behavior closely, outdoor environment tests reveal the actual operational behavior of the robot.

The crucial part of the testing procedure is done on outdoor grass terrain, in which the sole odometric navigation and integrated position correction techniques are compared. The inner swaths start after many rotations in rectangular inward spiral pattern in which rotation errors depose cumulatively. Therefore, for this comparison, only the use of parallel swath pattern is found to be logical. These outdoor comparisons using parallel swath pattern are made for both lengthwise and widthwise of mowing areas.

5.2 Test Setup

In order to evaluate the performance of the robot, navigating with different techniques for varying environments, a basic, generic test setup has been built. Simply, a camera was placed at the top-center of the test areas for both indoor and outdoor tests. Its maximum field of view dimensions have been taken as the borders of mowing area. The dimensions of borders are 295 x 162.5 cm^2, which corresponds to an area of approximately 5 m^2. The schematic of the test area is given in Figure 5.3.

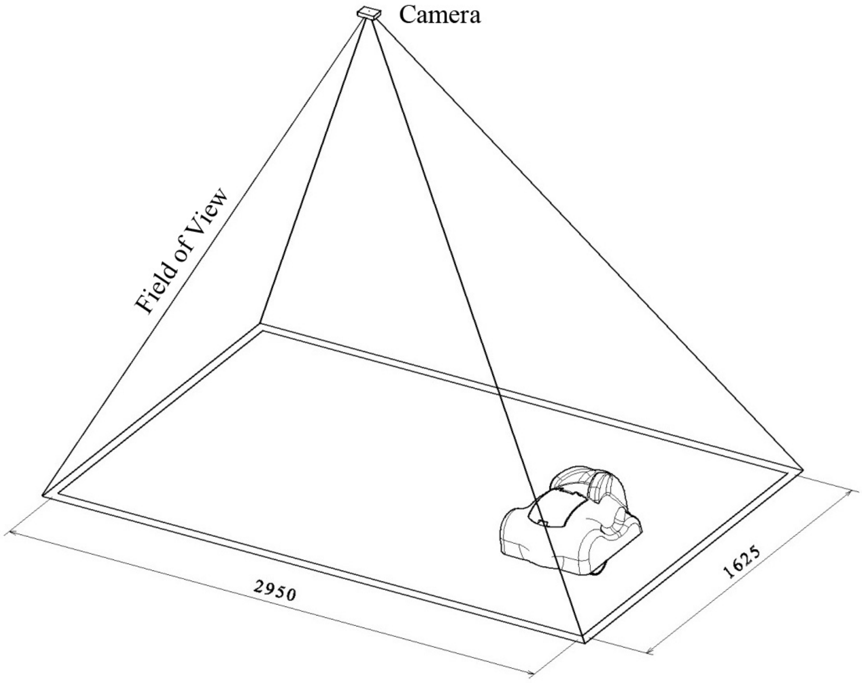


Figure 5.3: Test Setup Schematic (Scaled, Actual Proportions)

The indoor flat floor tests have been performed on a wooden parquet floor where strips are pasted on to represent borders. A sample camera view of indoor flat floor tests is given in Figure 5.4.

For indoor synthetic grass tests, a synthetic grass layer is laid onto parquet floor and trimmed to resize in desired dimensions. A camera view of the synthetic grass test setup is given in Figure 5.5.

For a fair comparison, the outdoor tests are performed within the same area, which is approximately 5 m^2 (295 x 162.5 cm^2). The camera is placed to the

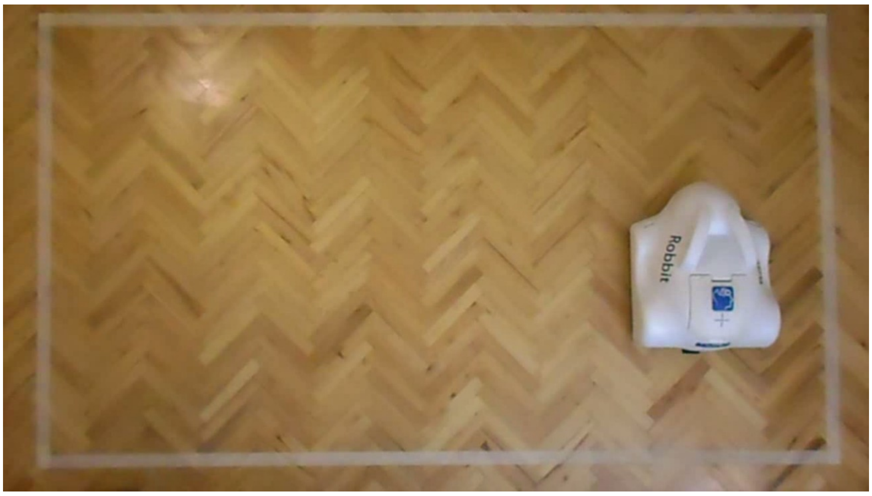


Figure 5.4: Camera View of Indoor Flat Floor Tests

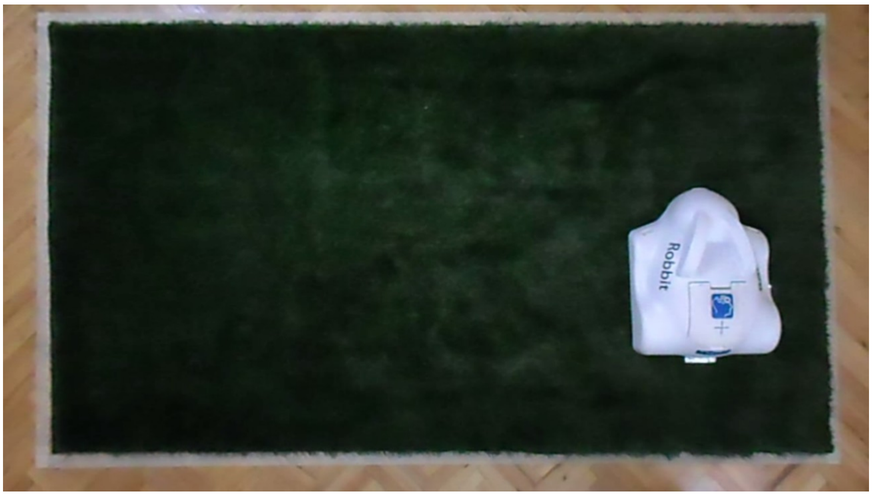


Figure 5.5: Camera View of Indoor Flat Synthetic Grass Tests

ceiling height from the ground and the borders are also marked with stripes. The outdoor test setup is given in Figure 5.6. The camera view of the outdoor test setup is also given in Figure 5.7.

5.3 Data Acquisition and Post Processing

More than 30 tests were accomplished for indoor and outdoor environments. Within all these tests, the odometric data of encoders are collected from the controller's serial port and from the robot motion recorded by a camera. Since the camera naturally has an optical distortion, post processing has also been made to make raw video data meaningful. By using an available software, recordings

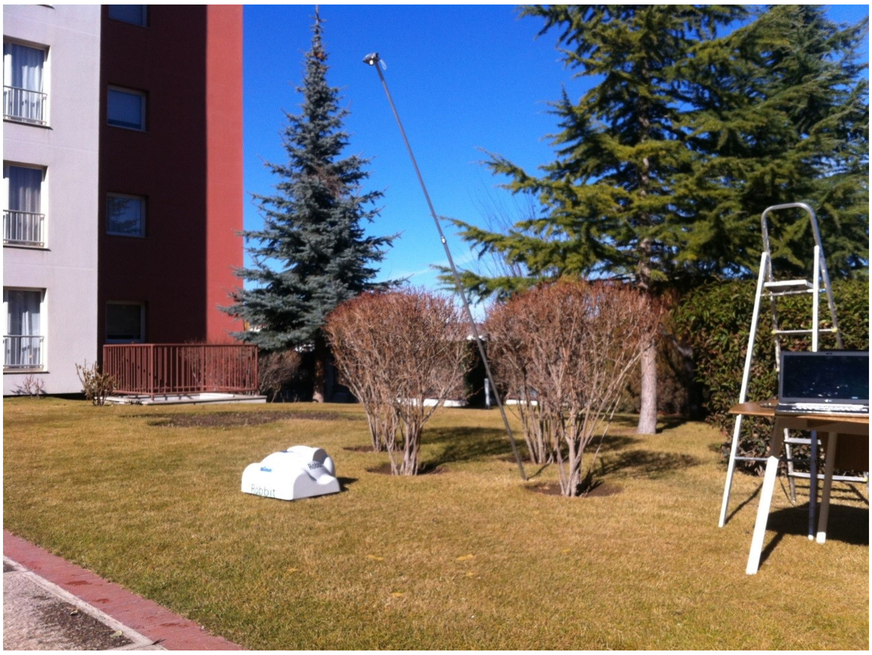


Figure 5.6: Outdoor Test Setup

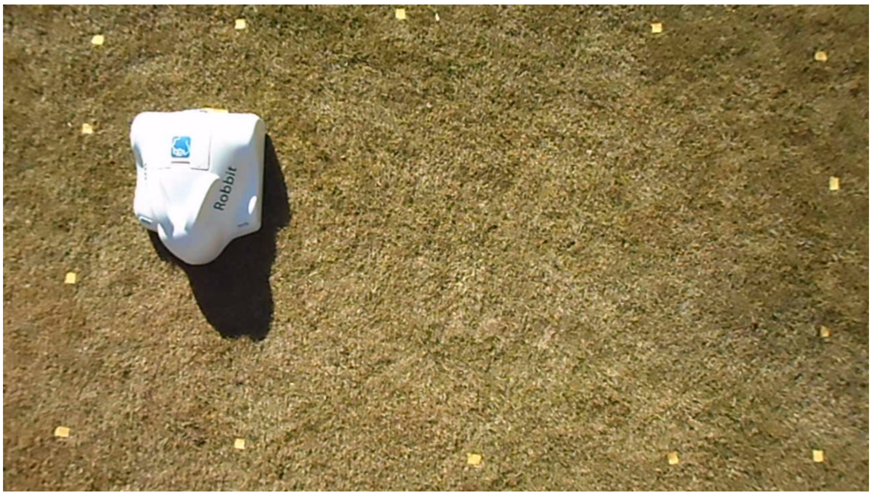


Figure 5.7: Camera View of Outdoor Tests

are optically compensated to obtain the actual trajectory. Also, the operation times are obtained for each coverage pattern-terrain combination.

Finally, the actual trajectories are stroked by the width of the cutting blade. The coverage percentage is calculated by determination of the ratio of mowed and non-mowed areas.

5.4 Indoor Tests

In indoor tests, the desired trajectory is predefined to ALM as mentioned in Section 5.1. Depending on coverage patterns, six different test results were given for flat parquet and synthetic grass terrains in this section. These are;

- Rectangular Inward Spiral Pattern
- Widthwise Parallel Swath Pattern
- Lengthwise Parallel Swath Pattern
- Random Pattern (x3)

For all indoor test samples, the data is collected from the serial port is plotted and compared with the desired trajectory (except for the random pattern). The data taken from serial port represents the own knowledge of the robot about its position. So, the amount of difference between the desired trajectory and the actual robot knowledge is revealed. In other words, the level of awareness of the robot about how much it is going off the road is revealed.

The actual trajectory, processed from recorded videos is also plotted and compared with the serial data gathered from the controller. In addition, actual trajectories are stroked (by image processing) according to the blade width and subtracted from the mowing area. Assuming that the robot is going to mow every single grass within its cutting blade width, the coverage percentage for each operation is calculated by this way.

5.4.1 Indoor Flat Floor (Parquet) Test Results

5.4.1.1 Indoor Flat Floor Rectangular Inward Spiral Pattern Results

By image processing, the coverage percentage for rectangular inward spiral pattern on flat parquet floor is calculated as 99.18%. The dashed line in Figure 5.10 represents the mowing area borders. The operation completion time for rectangular inward spiral pattern on flat parquet floor is calculated as 2 min.

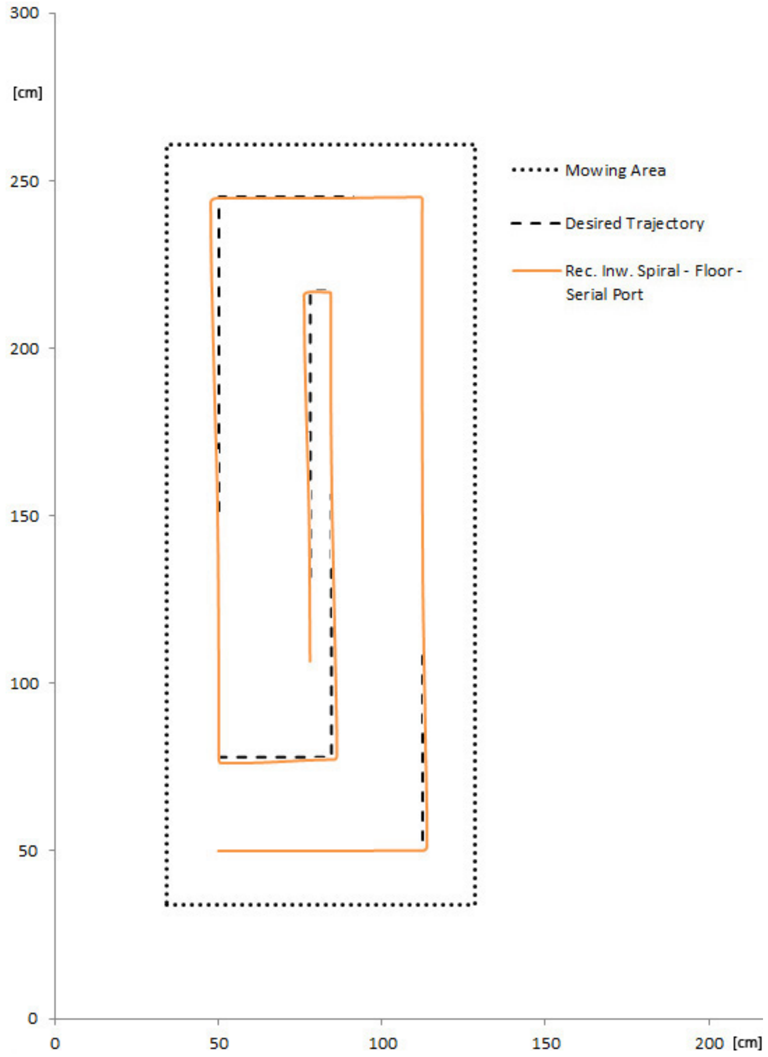


Figure 5.8: Ind. Flat Floor Rect. Inw. Spi. Serial Data Compared with Desired Trajectory (Scaled)

5.4.1.2 Indoor Flat Floor Widthwise Parallel Swath Pattern Results

In a similar way, the coverage percentage for widthwise parallel swath pattern on flat parquet floor is calculated as 98.03%. The robot had completed this operation in 2 min 30 sec.

5.4.1.3 Indoor Flat Floor Lengthwise Parallel Swath Pattern Results

The coverage percentage for lengthwise parallel swath pattern operation on flat parquet floor is calculated as 99.53% and the operation completion time is 2 min.

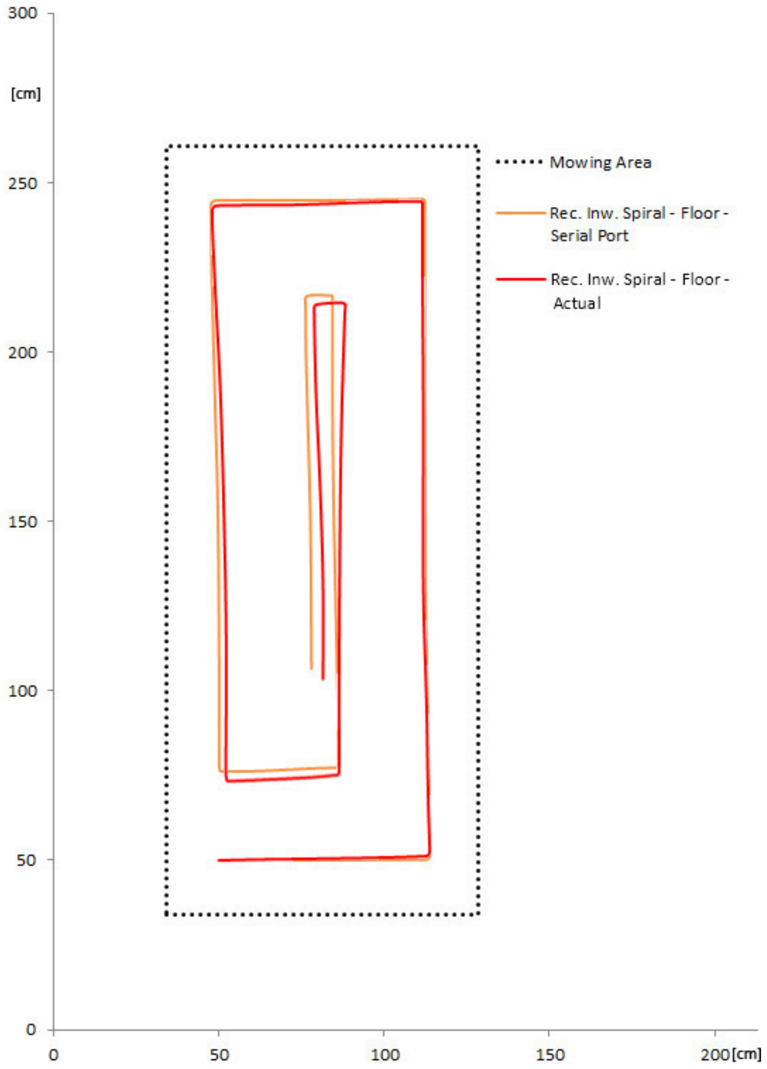


Figure 5.9: Ind. Flat Floor Rect. Inw. Spi. Actual Data Compared with Serial Data (Scaled)

5.4.1.4 Indoor Flat Floor Random Pattern Results

As mentioned earlier three random pattern coverage test results were presented in order to obtain an average coverage percentage. Since random operations do not have a prior path to follow, sole trajectories for these operations are not given in figures. Instead, actual coverages for random operations is presented. Since the maximum coverage completion time between rectangular inward spiral, widthwise and lengthwise parallel swath patterns is 2 min 30 sec (in widthwise parallel swath pattern), all random operations are terminated at exactly 2 min 30 sec for a fair comparison.

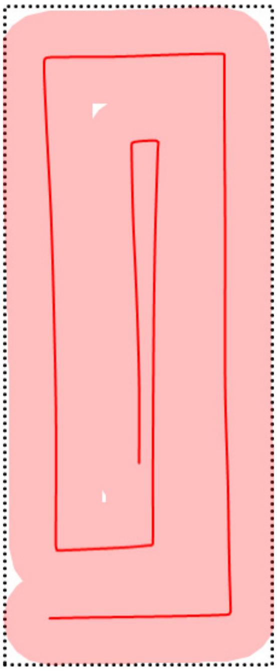


Figure 5.10: Ind. Flat Floor Rect. Inw. Spi. Actual Coverage (Scaled)

The coverage percentages for random pattern operations on flat parquet floor is calculated as 74.72%, 66.10% and 51.32%.

5.4.2 Indoor Synthetic Grass Test Results

5.4.2.1 Indoor Synthetic Grass Rectangular Inward Spiral Pattern Results

It can be seen in Figure 5.19 that the ALM exceeded the mowing area. When the serial data comparison figure is investigated, infinitesimal deviations can be seen from the desired trajectory. The reason of the relatively large deviation in actual behavior is slippage. Especially in rotations, ALM cannot satisfy a desired angle of 90°. Therefore, the robot moves out of bounds as rotation errors deposit.

For determining coverage percentages, only the mowed area inside the desired mowing area is taken into account. The coverage percentage for rectangular inward spiral pattern on synthetic grass is calculated as 89.25% by image processing. The operation completion time for rectangular inward spiral pattern on synthetic grass is calculated as 2 min 5 sec.

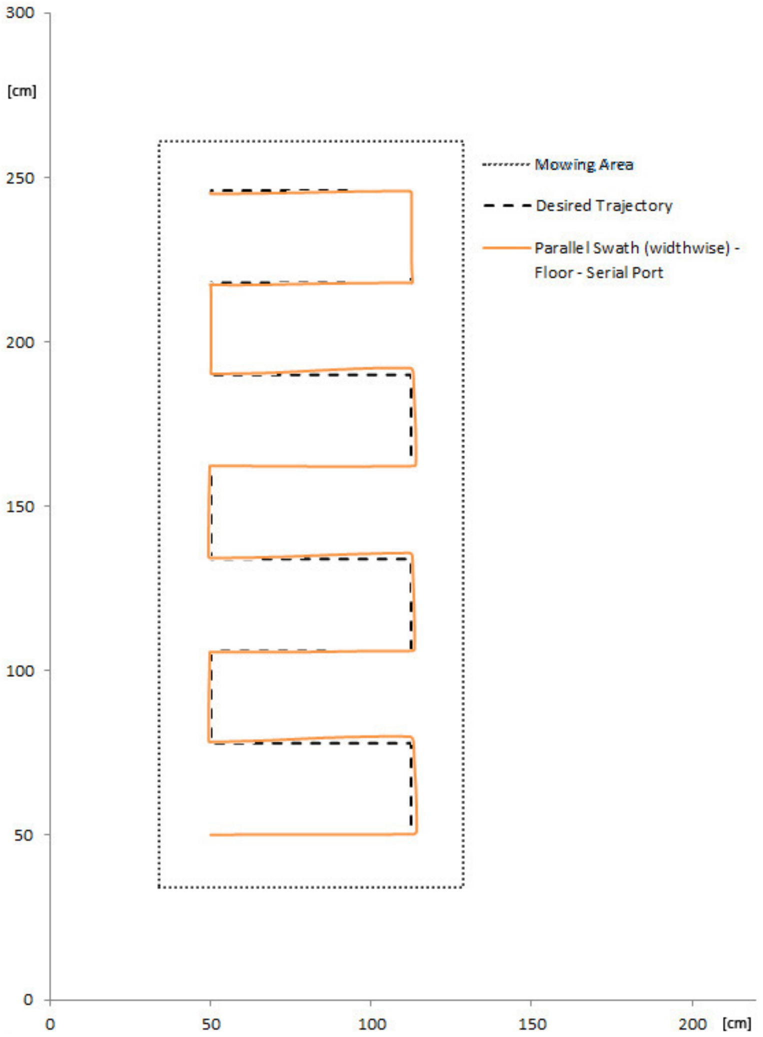


Figure 5.11: Ind. Flat Floor Widthwise Par. Swa. Serial Data Compared with Desired Trajectory (Scaled)

5.4.2.2 Indoor Synthetic Grass Widthwise Parallel Swath Pattern Results

The slippage on synthetic grass can also be seen in Figure 5.22. Coverage of the ALM on synthetic grass with widthwise parallel swath is given in Figure 5.23.

The coverage percentage for widthwise parallel swath pattern on synthetic grass terrain is calculated as 89.40%. The robot completed its operation in 2 min 37 sec.

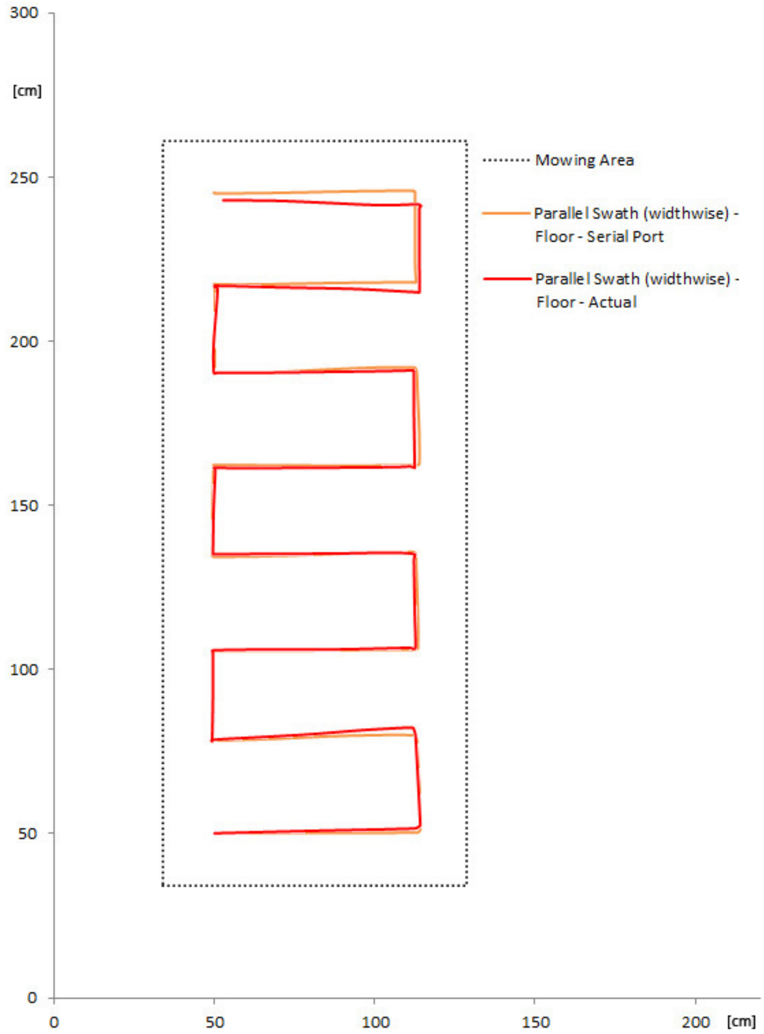


Figure 5.12: Ind. Flat Floor Widthwise Par. Swa. Actual Data Compared with Serial Data (Scaled)

5.4.2.3 Indoor Synthetic Grass Lengthwise Parallel Swath Pattern Results

The coverage percentage for lengthwise parallel swath pattern operation on synthetic grass is calculated as 69.35% and the operation completion time is 2 min.

5.4.2.4 Indoor Synthetic Grass Random Pattern Results

For making a controlled comparison, only 2 min 37 secs of operation data, which is the maximum period of indoor synthetic grass tests (for widthwise parallel swath pattern), have been used for random operation comparisons on synthetic

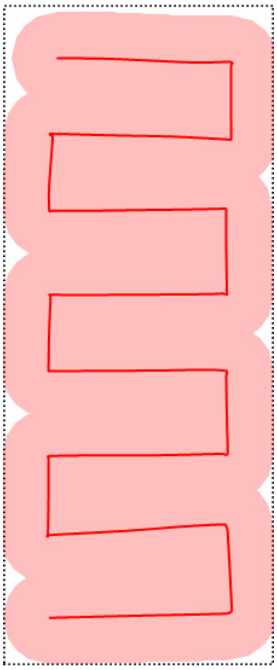


Figure 5.13: Ind. Flat Floor Widthwise Par. Swa. Actual Coverage (Scaled)

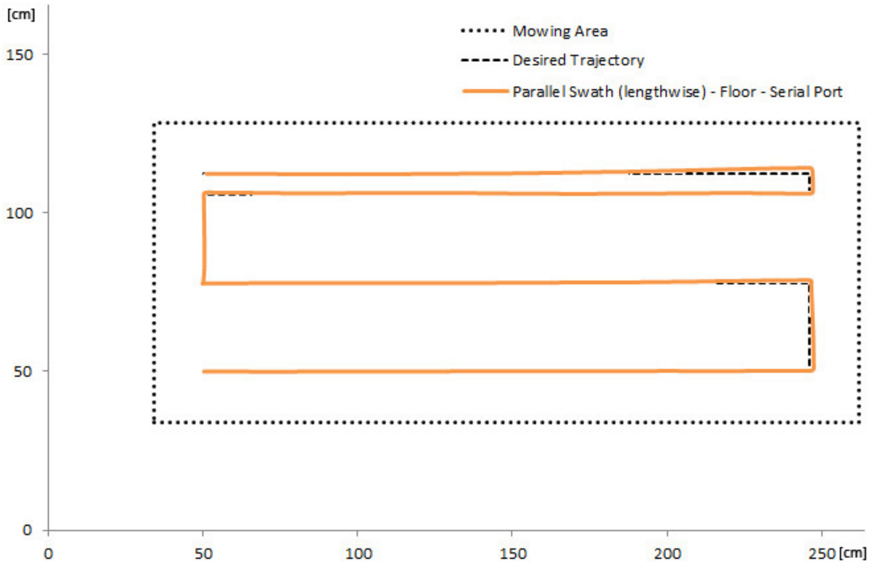


Figure 5.14: Ind. Flat Floor Lengthwise Par. Swa. Serial Data Compared with Desired Trajectory (Scaled)

grass terrains.

The coverage percentages for random pattern operations on flat synthetic grass is calculated as 58.43%, 60.63% and 55.74%.

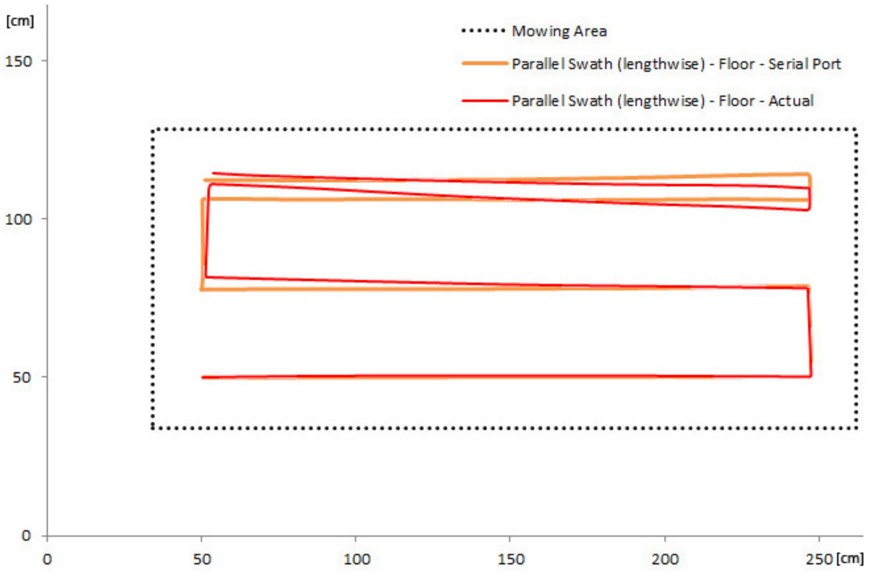


Figure 5.15: Ind. Flat Floor Lengthwise Par. Swa. Actual Data Compared with Serial Data (Scaled)

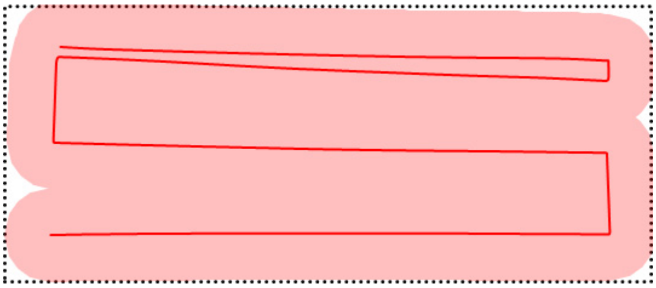


Figure 5.16: Ind. Flat Floor Lengthwise Par. Swa. Actual Coverage (Scaled)

5.4.3 Indoor Synthetic Grass Test Results

The sample test results for the flat parquet floor and synthetic grass terrain are presented in detail in Section 5.4.1 and 5.4.2. All of the indoor tests have been performed with the same velocity, the same hardware and the same controller parameters.

In this section, the results are sorted regarding the terrain and coverage pattern type. Comparisons of indoor tests regarding the coverage percentage and operation completion time are given in Table 5.1.

When the flat parquet floor coverage percentages are considered, it can be said that the amount of slippage is negligible since coverage percentages of all geometrical patterns are very close to 100%. It can also be seen that the rectangular

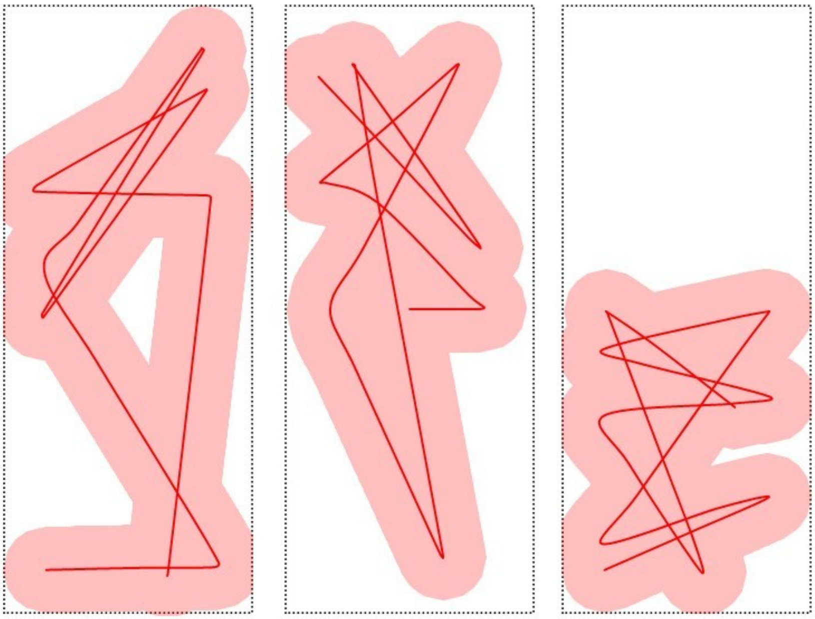


Figure 5.17: Ind. Flat Floor Actual Coverage for Three Different Random Pattern Operations (Scaled)

inward spiral and lengthwise parallel swath patterns achieved a slightly better performance than widthwise parallel swath, when operation completion periods and coverage percentages are considered. This is inevitable since the number of rotations is very close in between rectangular inward spiral and lengthwise parallel swath (7 and 6 respectively), where widthwise parallel swath operation includes two times more rotations (14). Therefore, they eventually slows down the operation.

When flat parquet floor coverage percentages are compared with the synthetic grass results, it can be seen that the slippage is increased in a significant manner. At least 10% drop in coverage percentages is observed when synthetic grass test results are compared with the parquet floor results. This difference can also be seen on actual-serial data comparison plots for synthetic grass tests. Due to this slippage deposition, it can be seen that the ALM moves out of the desired mowing area, in almost every operation performed on synthetic grass. Also a small lag is observed when operation completion periods are compared. This lag mainly occurs from synthetic grass resistance, which is especially dominant in rotations.

Moreover, it is also observed that the coverage percentage success hierarchy

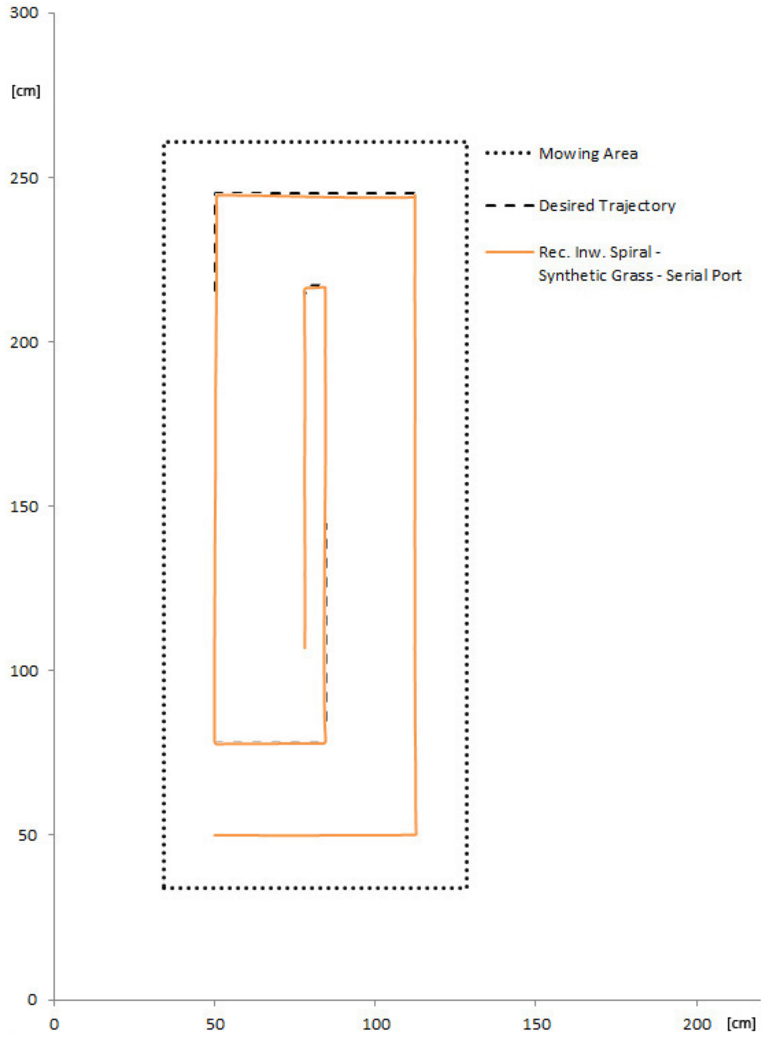


Figure 5.18: Ind. Synt. Grass Rect. Inw. Spi. Serial Data Compared with Desired Trajectory (Scaled)

changes for synthetic grass terrain. In parquet floor tests, the lengthwise parallel swath algorithm yields the best coverage success, whereas it took the worst score except for the random pattern on synthetic grass.

The reason for that may be seen confusing at the first glance. It is commonly known that most of the position and heading errors deposit in rotations, therefore it can be thought that the coverage percentage increases as the number of rotations decrease. This idea is not wrong conceptually, and it has also been proven in parquet floor tests. However, if the slippage is inevitable, an increase in number of rotations in a geometrical pattern affects coverage percentage in a positive manner. Because, two consecutive rotations with short distances be-

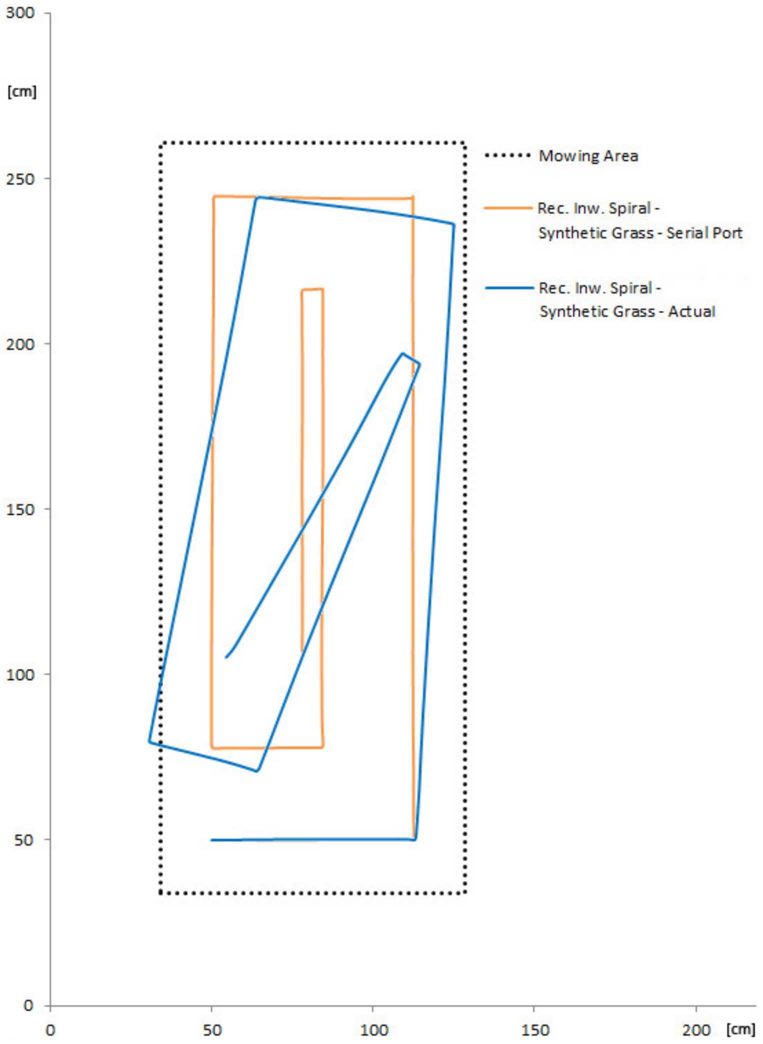


Figure 5.19: Ind. Synt. Grass Rect. Inw. Spi. Actual Data Compared with Serial Data (Scaled)

tween, compensate their heading errors a little, where a relatively long line after a rotation increases positional errors since the heading of the robot is less than desired. Therefore, the usage of geometrical patterns with more rotations is found to be beneficial for grass terrains.

Additionally, it is seen that the geometrical patterns cover the desired area faster than the random pattern since random behavior is inherently inefficient.

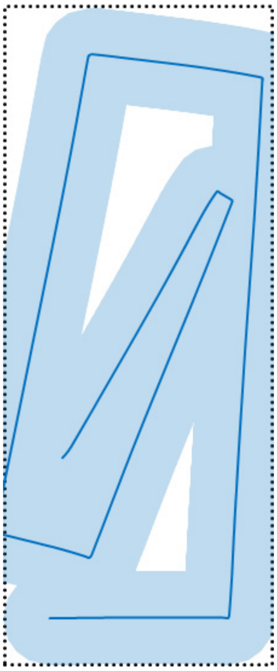


Figure 5.20: Ind. Synt. Grass Rect. Inw. Spi. Actual Coverage (Scaled)

5.5 Outdoor Tests

As mentioned before, navigation characteristics of the ALM has been obtained from indoor tests. The differences between coverage percentage, operation completion time, deviation from trajectory and amount of the slippage have already been determined for varying coverage patterns, both for flat parquet floor and synthetic grass in indoor tests.

In outdoor tests, it is only aimed to reveal the differences in autonomous navigation behavior of sole odometry and proposed enhancement technique. In order to determine the odometric navigation enhancement, two sets of tests are performed. On a decent grass terrain, Widthwise and Lengthwise Parallel Swath Pattern coverage operations are performed several times with and without applying the IR position correction method. It is mentioned before that the rectangular inward spiral coverage algorithm is found unreliable for this comparison since the inner swaths start after four rotations, where a great amount rotation error deposition has been occurred. Also, random pattern behaviors are not tested in outdoor environment since the proposed enhancement for odometric navigation is only valid for geometrical patterns.

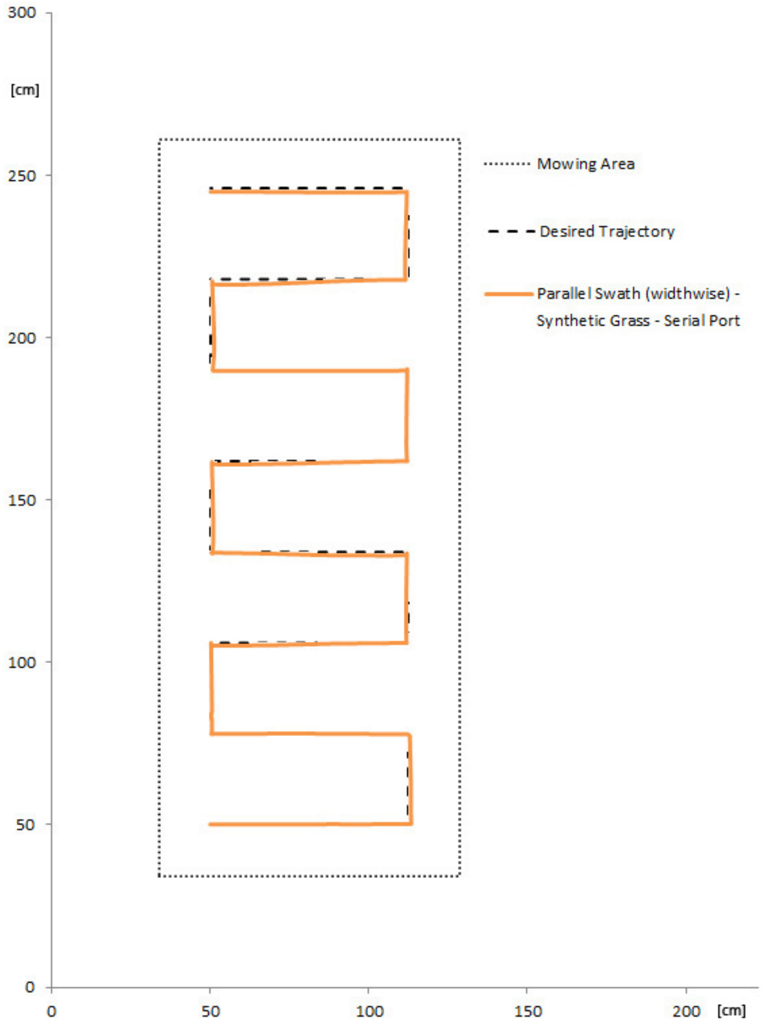


Figure 5.21: Ind. Synt. Grass Widthwise Par. Swa. Serial Data Compared with Desired Trajectory (Scaled)

In this context, the outdoor tests without lawn mowing are performed first. In this set, several widthwise and lengthwise parallel swath coverage algorithms are executed. Afterwards, mower motor is turned on and same parallel swath patterns are performed on lawn. As the ALM mows whole stripe of grass according to its blade width, the success of the proposed correction method is intended to be shown.

Like indoor tests, data obtained from serial port is plotted and compared with the desired trajectory for enhanced navigation technique. The actual trajectory, processed from recorded videos is also cross plotted and compared with desired trajectory. Then, the coverage percentages and the operation completion times are calculated.

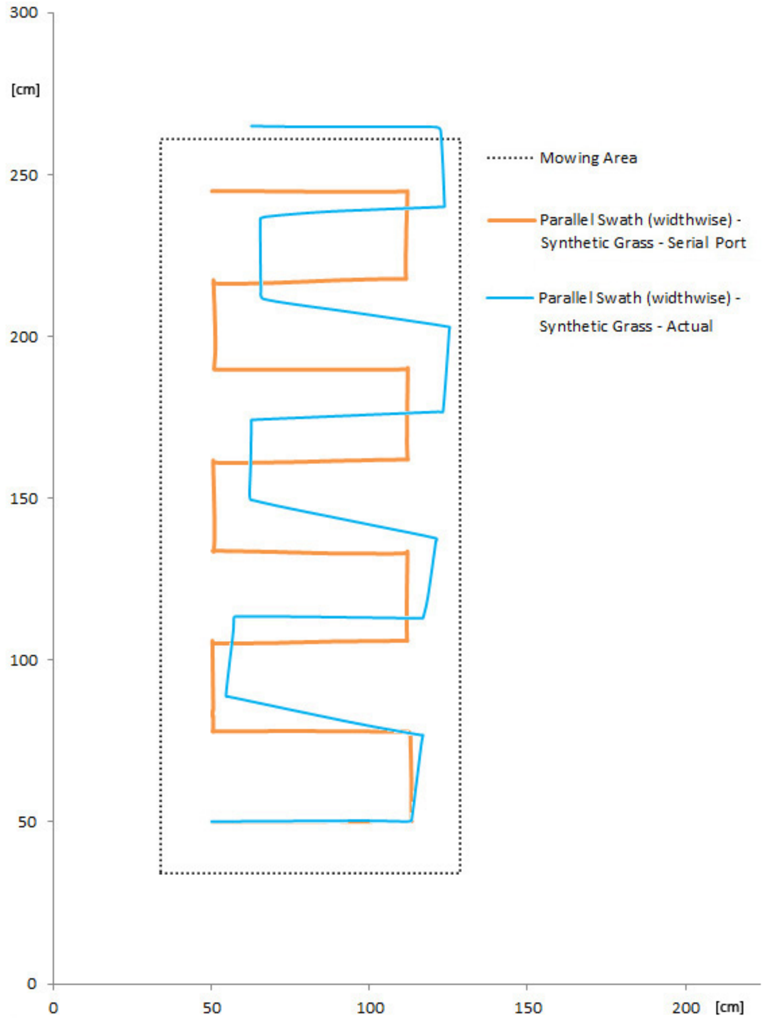


Figure 5.22: Ind. Synt. Grass Widthwise Par. Swa. Actual Data Compared with Serial Data (Scaled)

Sample test results are presented in following sections.

5.5.1 Outdoor Widthwise Parallel Swath Pattern Results

It can be seen in Figure 5.28 that the ALM corrects its position when it senses the presence of lawn on the area that it had swathed. This behavior can also be seen from robot's eye in Figure 5.29. As the position correction intterupts occured by IR switches, small bumps on the robot's trajectory are occured.

The slippage errors can be seen from the comparison of serial and actual data. This comparison is given Figure 5.30.

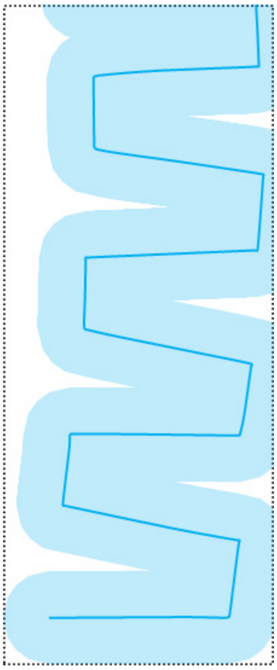


Figure 5.23: Ind. Synt. Grass Widthwise Par. Swa. Actual Coverage (Scaled)

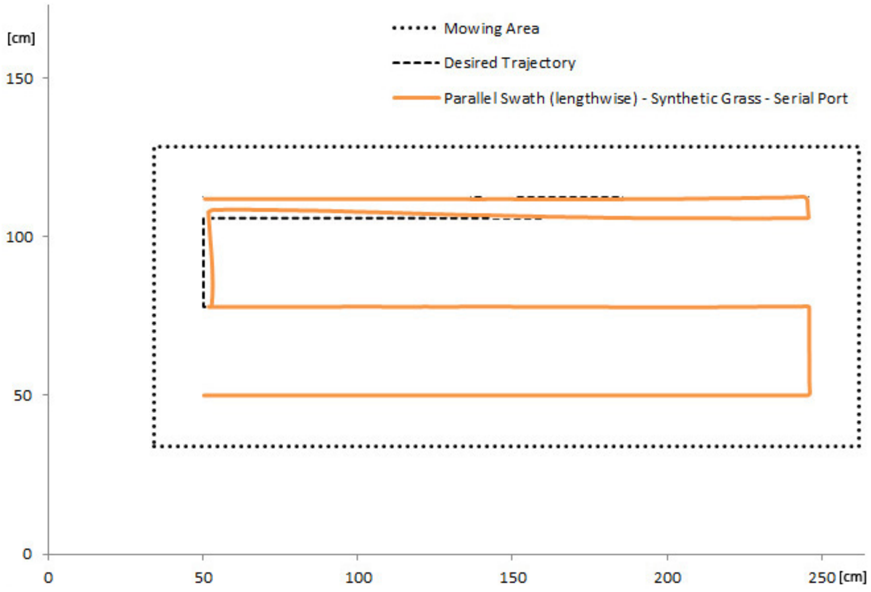


Figure 5.24: Ind. Synt. Grass Lengthwise Par. Swa. Serial Data Compared with Desired Trajectory (Scaled)

Similar to indoor tests, the outdoor trajectories are stroked along the blade width for computing coverage percentages. Coverages for sole and enhanced odometric navigation techniques are given in Figure 5.31.

The coverage percentage for odometric navigation, using IR data for position correction is calculated as 92.43%, where 90.26% can be achieved for the width-

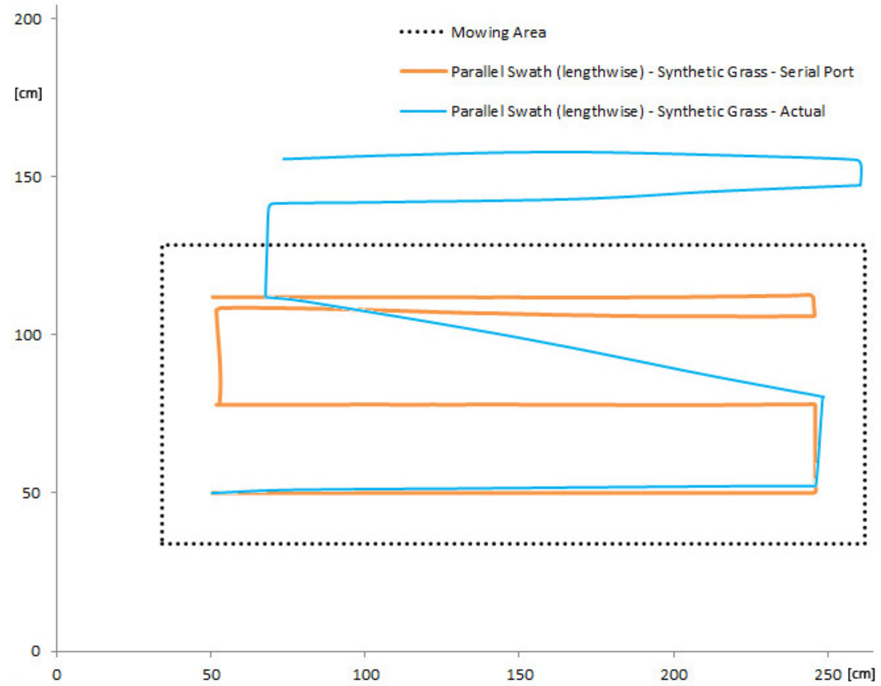


Figure 5.25: Ind. Synt. Grass Lengthwise Par. Swa. Actual Data Compared with Serial Data (Scaled)

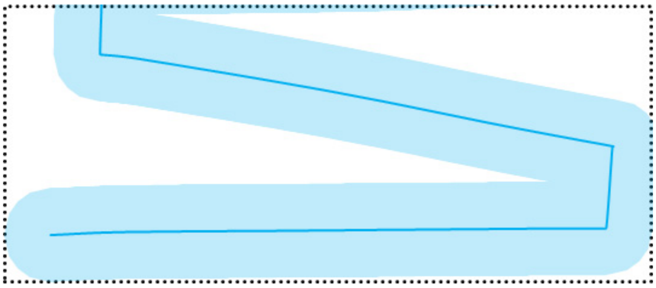


Figure 5.26: Ind. Synt. Grass Lengthwise Par. Swa. Actual Coverage (Scaled)

wise parallel swath algorithm which does not utilize position correction method.

5.5.2 Outdoor Lengthwise Parallel Swath Pattern Results

In a similar manner, deviation from desired trajectory for sole and enhanced odometric navigation is given in Figure 5.32.

Serial port data, compared with the desired trajectory for enhanced odometric navigation technique is given in Figure 5.33.

Comparison of serial and actual data which reveals the slippage deposition is

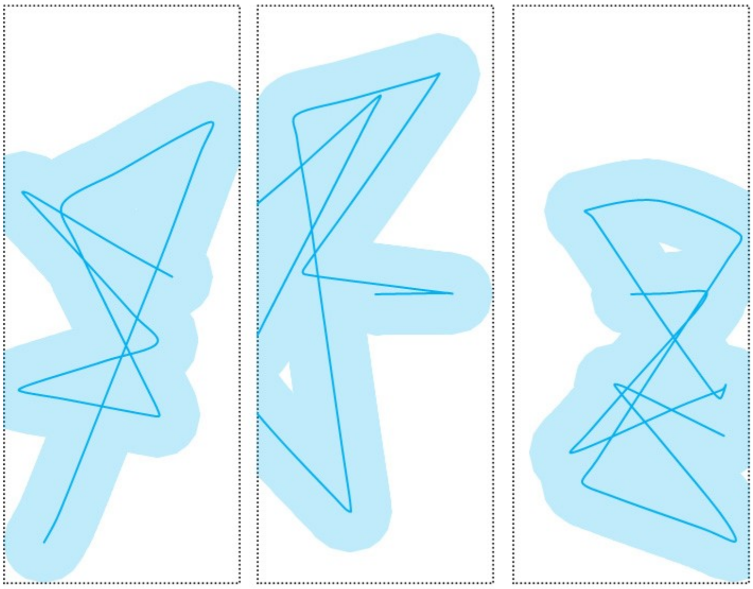


Figure 5.27: Ind. Synt. Grass Actual Coverage for Three Different Random Pattern Operations (Scaled)

given in Figure 5.34.

Finally, actual coverages for sole and enhanced odometric navigation techniques for lengthwise parallel swath pattern are given in Figure 5.35.

The coverage percentage for odometric navigation, using IR data for position correction is calculated as 82.03% for the widthwise parallel swath algorithm. Only 70.75% of the mowing are can be covered by sole odometric navigation by the same pattern.

5.5.3 Comparisons and Conclusions of Outdoor Tests Results

The general odometric coverage behavior for varying geometrical patterns is investigated in Section 5.4.3 by indoor tests. In outdoor tests, it is intended to see the effects of the enhancement made over odometric navigation for position correction. The results comparing this effect are given in Section 5.5.1 and 5.5.2.

The overall results are compared in Table 5.2, where the standard and enhanced navigation techniques are compared for parallel swath coverage algorithms.

It is seen that the coverage performance is increased from 90.26% to 92.43%

Table 5.1: Comparisons of Indoor Test Results

Indoor Tests		**Operation Completion Time**	**Coverage Percentage**
Parquet Flat Floor Tests	Rect. Inw. Spi.	2 min	99.18%
	Wid. Par. Swa.	2 min 30 sec	98.03%
	Len. Par. Swa.	2 min	99.53%
	Random (Ave.)	2 min 30 sec (Terminated)	64.04%
Synthetic Grass Tests	Rect. Inw. Spi.	2 min 5 sec	89.25%
	Wid. Par. Swa.	2 min 37 sec	89.40%
	Len. Par. Swa.	2 min	69.35%
	Random (Ave.)	2 min 37 sec (Terminated)	58.26%

Table 5.2: Comparisons of Outdoor Test Results

Outdoor Tests		**Operation Completion Time**	**Coverage Percentage**
Sole Odometry	Wid. Par. Swa.	2 min 34 sec	90.26%
	Len. Par. Swa.	2 min	70.75%
Enhanced Technique	Wid. Par. Swa.	2 min 36 sec	92.43%
	Len. Par. Swa.	2 min 1 sec	82.03%

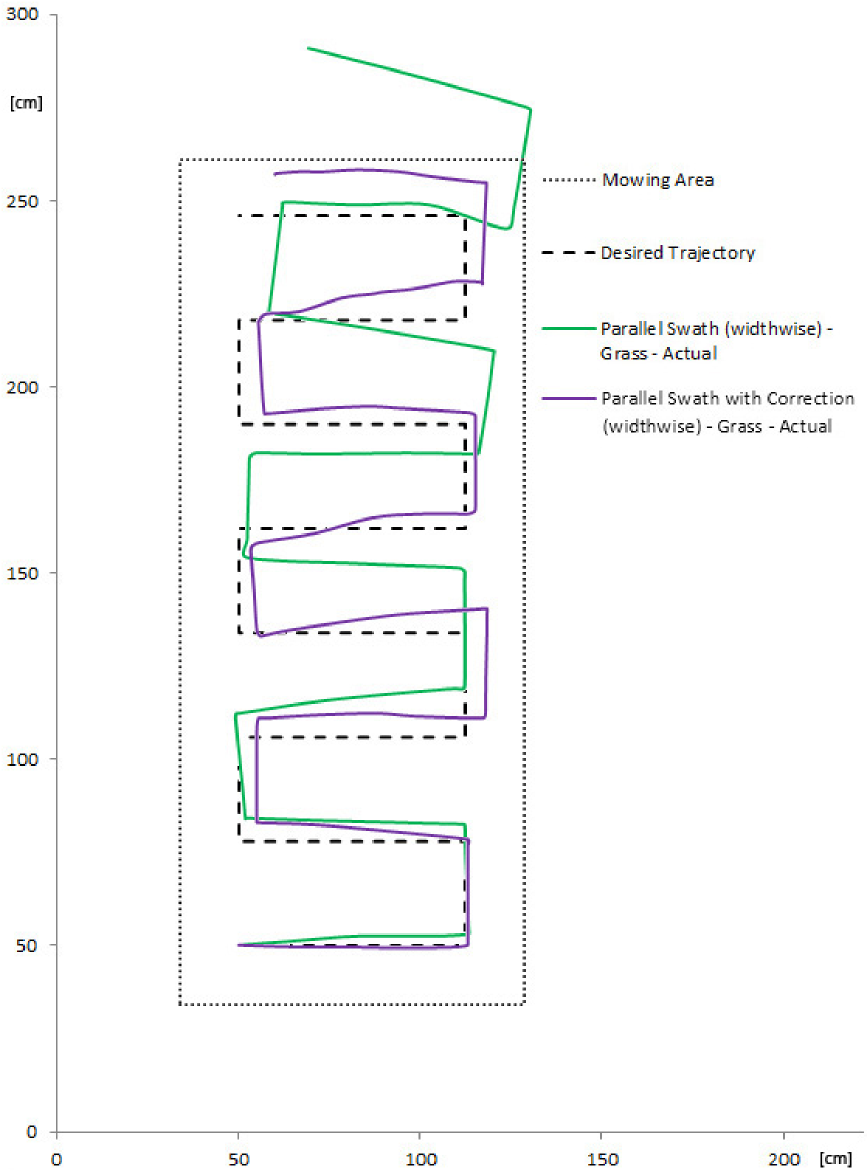


Figure 5.28: Out. Wid. Par. Swa. Sole and Enhanced Technique Actual Data Compared with Desired Trajectory (Scaled)

by accounting IR data for positional corrections for widthwise parallel swath, which corresponds to a 2.4% overall increase. This amount is more significant for lengthwise parallel swath operation, in which coverage percentage is enhanced from 70.75% to 82.03% which means 15.9% overall increase.

Therefore, it is experienced that the positional correction mechanism has more impact on lengthwise parallel swath, compared to the widthwise parallel swath pattern. The reason is consistent with the explanation given in Section 5.4.3. A decrease in number of rotations also affects coverage percentage in a negative

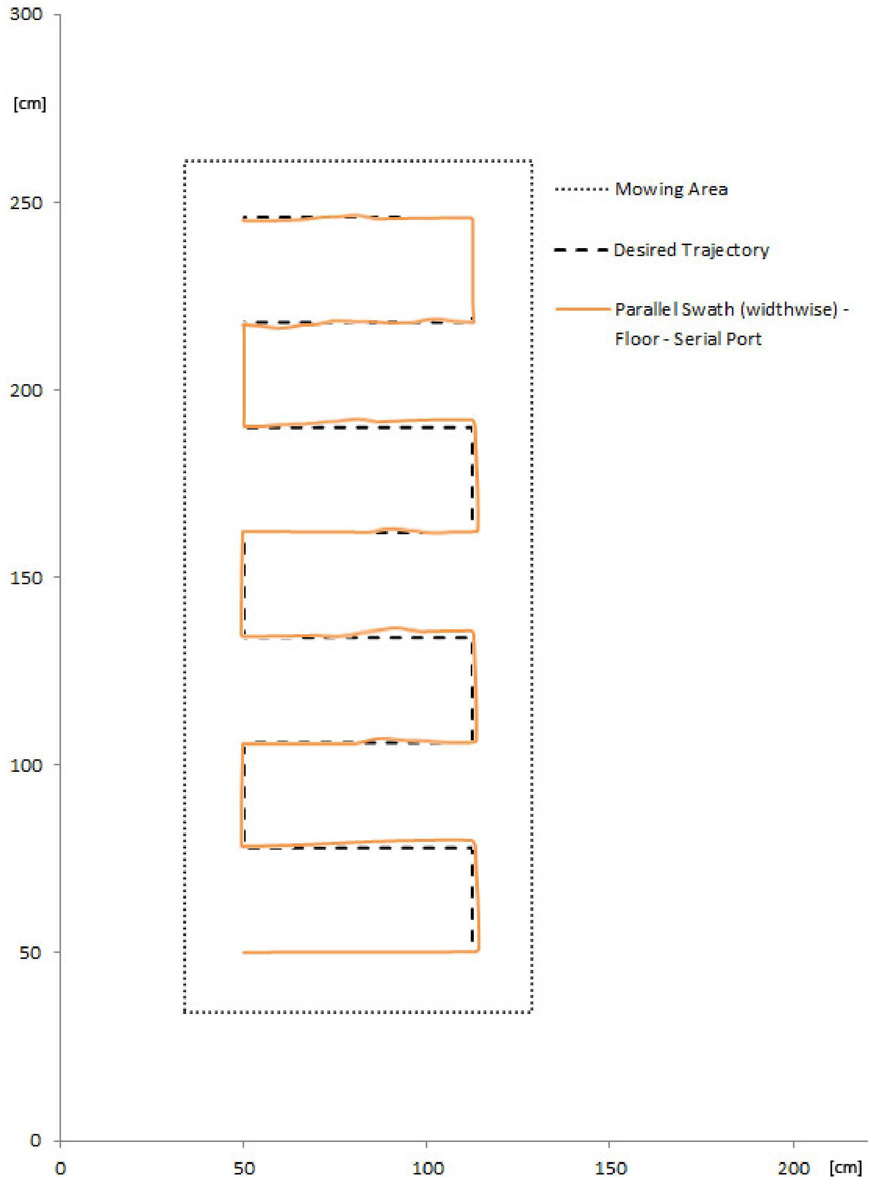


Figure 5.29: Out. Wid. Par. Swa. Serial Data Compared with Desired Trajectory (Scaled)

manner, and by improving error correction, coverage performance is improved in a great manner. This effect is less significant in widthwise parallel swath pattern since the consecutive rotations provide so called "auto improvements" for heading errors.

Finally, it is concluded that the proposed enhancement method (the position correction mechanism utilizing mowed and non-mowed area data between swaths) is working properly and increases the overall coverage performance of the ALM.

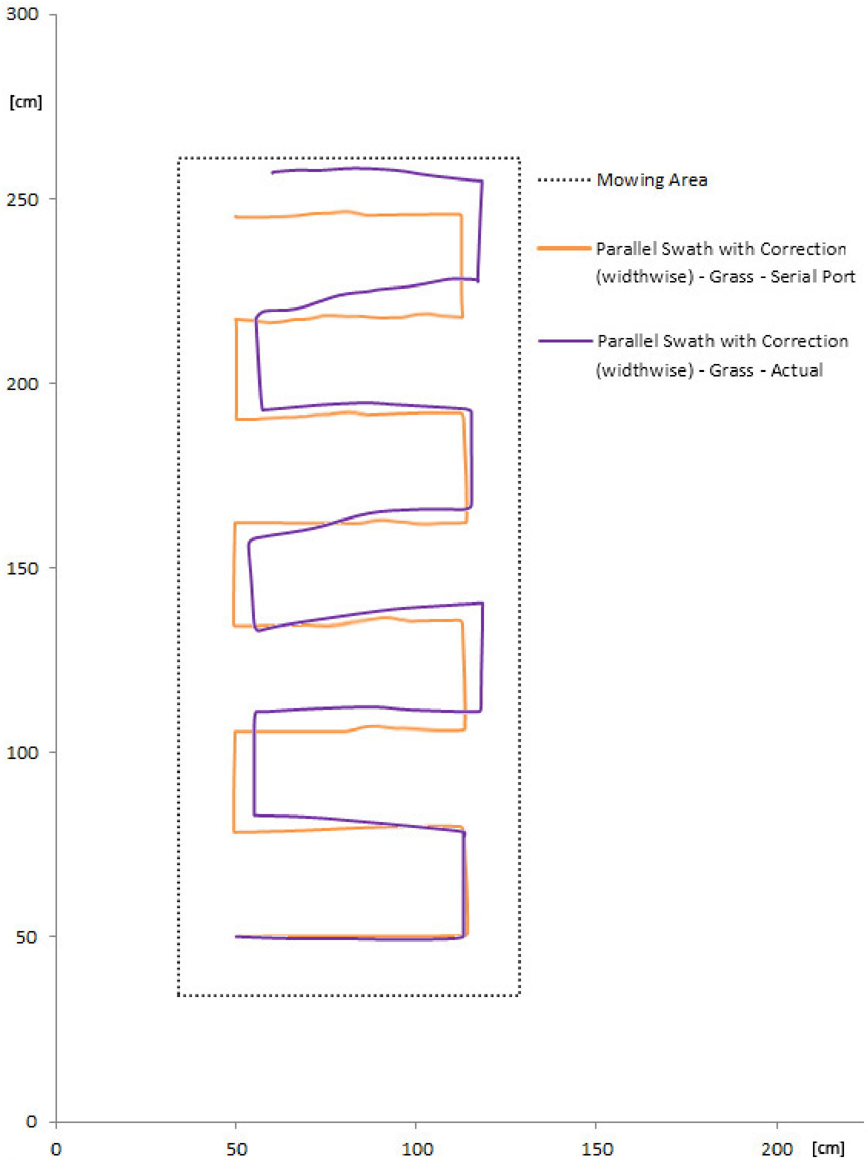


Figure 5.30: Out. Wid. Par. Swa. Serial Data Compared with Actual Data (Scaled)

5.5.4 Conclusion

In the physical testing phase of the development procedure; actual response of the robot, its possible improvements and overall design success are investigated by indoor and outdoor tests.

First of all, the success of drive wheel design is proven. With the aid of the comparably heavy, synthetic rubber coated wheels, the ALM followed the desired trajectory very accurately, even though the varnished flat parquet terrain cannot

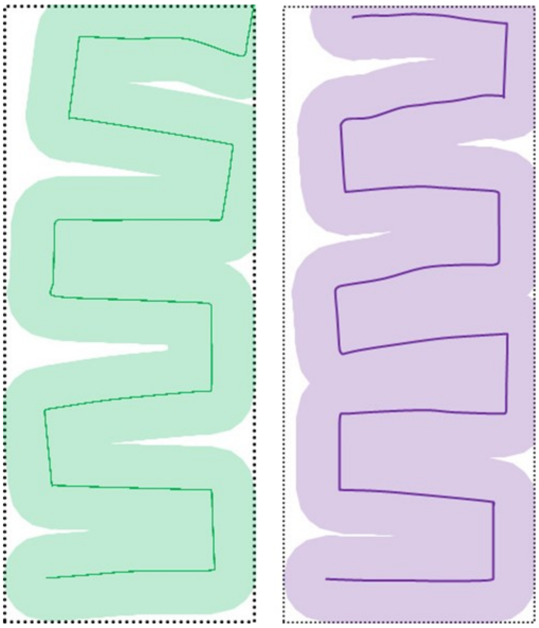


Figure 5.31: Actual Coverage of Out. Wid. Par. Swa. Sole and Enhanced Techniques (Scaled)

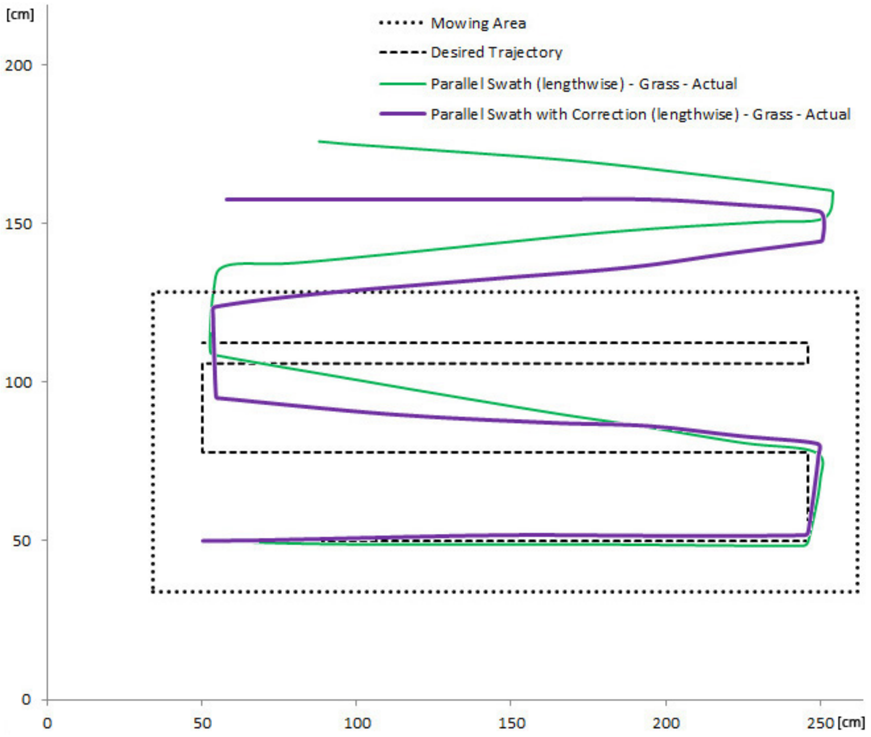


Figure 5.32: Out. Len. Par. Swa. Sole and Enhanced Technique Actual Data Compared with Desired Trajectory (Scaled)

be considered as a surface with high friction coefficient. The coverage ratio of the ALM on that terrain is almost unity for all tested coverage patterns. Besides the wheel performance, the success of the software architecture and controller

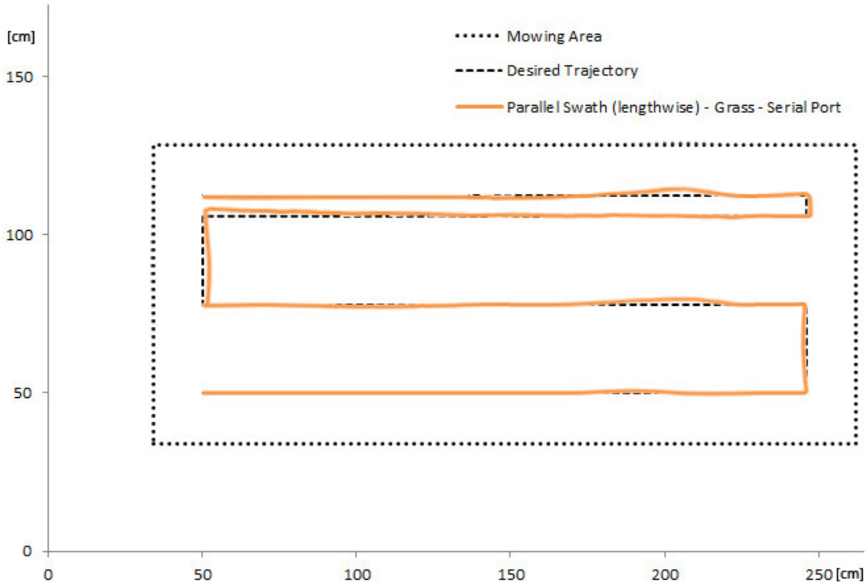


Figure 5.33: Out. Len. Par. Swa. Serial Data Compared with Desired Trajectory (Scaled)

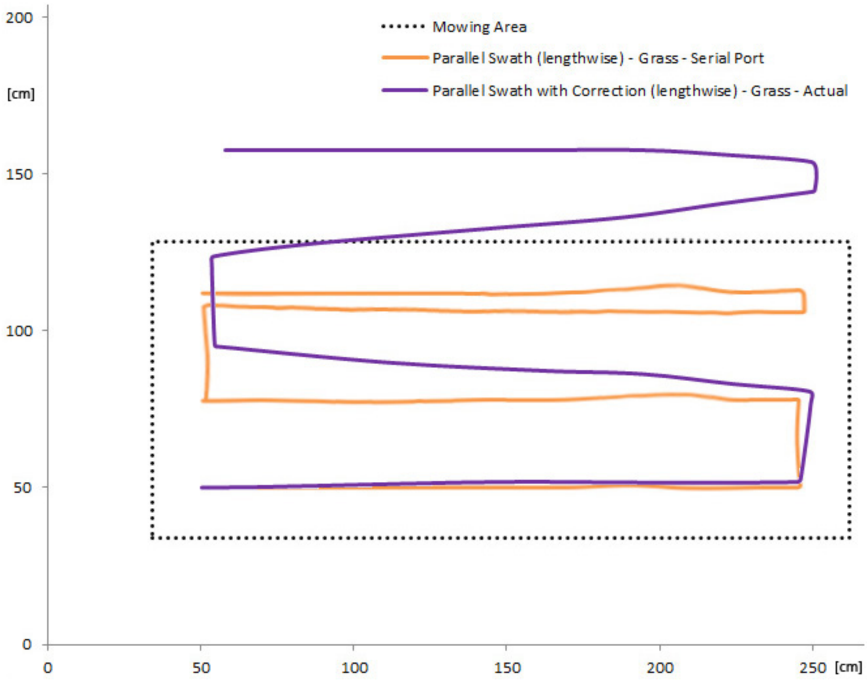


Figure 5.34: Out. Len. Par. Swa. Serial Data Compared with Actual Data (Scaled)

design cannot be underestimated for this achievement.

After successful operations performed on the flat parquet floor, indoor tests are continued with synthetic grass terrain. With synthetic grass tests, the outdoor response of the ALM is tried to be simulated. As expected, slippage played an

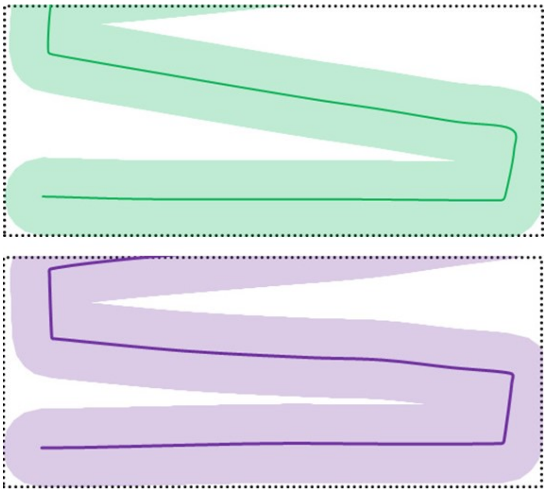


Figure 5.35: Actual Coverage of Len. Wid. Par. Swa. Sole and Enhanced Techniques (Scaled)

important role and negatively affected the trajectory-following behavior of the robot. As the amount of slippage is increased by time and number of rotations, it is observed that the ALM moves out from the desired area in time, for every coverage pattern. On a relatively slippery terrain, the coverage performance is inevitably decreased.

Also the differences between coverage patterns are revealed among these indoor tests. It has been determined that the geometrical coverage patterns (rectangular inward spiral, parallel swaths) are far more efficient and faster than the random algorithms. Random coverage patterns are actually quite practical in environments, where achievement of an accurate positioning can be hard and costly, but unfortunately these operations are inefficient by its nature.

Contrary to conventional opinion, it has been seen during indoor synthetic grass tests that better coverage performance can be obtained by shortening straight paths and raising the number of rotations, instead of navigating same area with travelling long distances with fewer rotations. Even the operation time is a little longer than the others; widthwise parallel swath coverage algorithm provides the best coverage performance on this relatively slippery terrain.

In the outdoor tests, the cutting blade design is intended to be proven first. It has been shown that the cutting blade with four cutting edges worked fine for lawn mowing. Also it is concluded that multiple cutting edges cut grass into smaller clippings, therefore they provide better mulching performance.

Afterwards, the exact same odometric navigation technique with same parameters, proved in indoor environment has been tested in outdoors. The results showed us that the resistance and slipperiness of the synthetic grass is nearly the same with real grass. The widthwise and lengthwise parallel swath patterns resulted at nearly same coverage performance. This argument confirmed our idea of simulating outdoor environment by synthetic grass and validated previous test results.

Finally one of the core aspects of the thesis is tested on outdoor environment. The proposed minimalist position correction technique, which utilizes the IR switch data in order to identify mowed and non-mowed areas after an inner swath, is tested. Test results showed that this enhanced technique can improve coverage performance up to 16%, compared to the sole odometric navigation.

Although enhanced odometric navigation technique proposed in this thesis, raises the coverage performance in a significant manner (especially for lengthwise parallel swath), the physical conditions of the outdoor terrain must not be ignored. The outdoor tests have been performed on a nearly "ideal" garden, in which grasses are dense and the terrain is nearly flat and dry. On more rugged or slippery lawns, this performance improvement will be decreased inevitably.

Despite all, the enhanced odometric navigation technique, proposed in this thesis have accomplished its mission on outdoor environment and it is shown that this technique can be a serious minimalist alternative for relatively expensive solutions in order to overcome inherent odometric positioning problems.

CHAPTER 6

CONCLUSION

6.1 Conclusion

In this study, it is mainly aimed to build an affordable ALM, which can assure the necessities expected from a commercial product.

An ALM must provide satisfactory performance in order to encounter consumer expectations, whereas it must be as cheap as possible to become a player in the domestic robotics market. Starting from this point, major objectives of an ALM can be stated as leanness and simplicity, while assuring the quality and performance. With these requirements, an ALM was sketched, designed, optimized, manufactured and assembled. All mechanical design, software development and system integration tasks are completed from scratch.

Functionality, reliability and visual quality are determined as primary aspects for the mechanical design of the ALM. An unconventional mowing method, mulching is selected in order to cohere with the concept of autonomy and for energy efficiency. In addition, differential drive steering system for the robot is found to be advantageous for mowing outdoors.

In order to reduce the overall cost of the system, navigation techniques which require expensive hardware and high computational power are avoided. Instead, it is shown that a minimalist approach with limited hardware can also provide adequate results for outdoor navigation. Although it is vulnerable to the inevitable slippage-based error deposition, odometric dead-reckoning method is selected since it is cheap and simple.

Precautions for slippage problem have been started from mechanical design. In order to improve traction, drive wheels are designed as frictional as possible. Also, the center of gravity of the robot is drawn closer as much as possible to the wheel axis.

The major enhancement made for increasing the outdoor navigation performance is to utilize IR switches, which are mounted at the bottom of the robot. The slippage-based error deposition is aimed to be compensated by the identification of mowed and non-mowed lawn areas.

After manufacturing and system integration, many indoor and outdoor tests are performed in order to determine navigation characteristics of the robot. The functionality and the success of the control architecture of the robot are proved by indoor tests. Moreover, various coverage patterns are tested and their superiorities are revealed.

The proposed enhancement is tested in outdoor grass terrain. It is proved that the proposed position correction technique improves the outdoor coverage performance up to 16%. The overall coverage with this technique is calculated above 80% for lengthwise parallel swath pattern. Coverage percentages above 80% are generally considered as a successful operation for ALMs. It can be concluded that the developed ALM accomplished its mission.

Additionally, it has been shown that the major performance criterion for an ALM -coverage performance- can be satisfied with a low cost hardware. Two IR switches cost below 10 USD, which is negligible when compared to sophisticated sensors.

As a future work, mechanic and electronic hardware can be further developed. The body and the mowing system designs can be improved to sustain coverage performance over more rugged and slippery terrains. Also, the position correction method which uses IR switches may be improved by considering different types of sensors as a future work.

In conclusion, the mechanical, electrical and software designs of the ALM is succeeded for both indoor and outdoor terrains.

REFERENCES

[1] World robotics 2012 service robots. Website, 2013. `http://www.ifr.org/service-robots/statistics/` [Accessed 19-August-2013].

[2] R.C. Arkin. *Behaviour Based Robotics.* MIT Press, 1998.

[3] J. Borenstein. Internal correction of dead-reckoning errors with the compliant linkage vehicle. *Journal of Robotic Systems, Vol. 12, No. 4 April 1995, pp. 257-273*, 1995.

[4] Bernard Etkin. *Dynamics of Atmospheric Flight, Stability and Control.* Wiley, 1971.

[5] H.R. Everett. Sensors for mobile robots. *A K Peters, Ltd., Wellesley, MA*, 1995.

[6] Owen Holland. The grey walter online archive - background information. Website, 2013. `http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html` [Accessed 19-August-2013].

[7] Ilshim Global Co, Ltd. *Windoro User Manual.* `http://roboshop.lt/wp-content/uploads/2013/03/windoro-user-manual-en.pdf` [Accessed 21-February-2014].

[8] L. Feng J. Borenstein. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions Robotics and Automation, vol. 12, pp. 869–880.*, 1996.

[9] L. Feng D. Wehe J. Borenstein, H. Everett. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems 14(4): 231–249*, 1997.

[10] Y. Koren J. Borenstein. Motion control analysis of a mobile robot. *Transactions of ASME, Journal of Dynamics, Measurement and Control, Vol. 109, No. 2, pp. 73-79*, 1987.

[11] K. D. Kuhnert and W. Seemann. Design and realization of the highly modular and robust autonomous mobile outdoor robot amor. Technical report, Robotics and Applications and Telematics, 2007.

[12] A. Martinelli. The odometry error of a mobile robot with a synchronous drive system. *IEEE Transactions on Robotics and Automation, pp. 399–405.*, 2002.

[13] Mitka Eleftheria Mouroutsos G. Spyridon. Classification of domestic robots, 2012.

[14] Nils J. Nilsson. Shakey the robot. *Technical Note 323. AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025*, 1984.

[15] The Ninth Annual Robotic Lawnmower Competition. *Competition Rulebook*, revision 2012.1.1 edition, May 31 - June 3 2012.

[16] Katsuhiko Ogata. *Modern Control Engineering.* Prentice Hall, 2009.

[17] Illah R. Nourbakhsh Rolan Siegwart. *Introduction To Autonomous Mobile Robots.* The MIT Press, 2004.

[18] Wright State University. Competition report. Technical report, Lawn Rider Competition, 2012.

[19] Glen D. Weinstein. Notice of annual meeting of stockholders. Technical report, iRobot Corporation, April 10 2013.