REDUNDANT MOBILE ROBOT CONTROL

A Thesis Submitted to
The Graduate School of Engineering and Science of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Mechanical Engineering

by
Onur ÇELİK

December 2016

İZMİR

We approve the thesis of **Onur CELIK**

Examining Committee Members:

Assist. Prof. Dr. M. İ. Can DEDE
Department of Mechanical Engineering, İzmir Institute of Technology

Prof. Dr. Serhan ÖZDEMİR
Department of Mechanical Engineering, İzmir Institute of Technology

Assist. Prof. Dr. Levent ÇETİN
Department of Mechatronics Engineering, İzmir Katip Celebi University

30 December 2016

Assist. Prof. Dr. M. İ. Can DEDE
Supervisor, Department of Mechanical
Engineering, İzmir Institute of Technology

Prof. Dr. Metin TANOĞLU
Head of the Department of Mechanical
Engineering

Prof. Dr. Bilge KARAÇALI
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor Assist. Prof. Dr. M. İ. Can DEDE for his understanding, wisdom, patience, guidance and suggestions.

I would also like to thank my dear friends and colleagues in IZTECH Robotic Lab. for their friendship and support.

ABSTRACT

REDUNDANT MOBILE ROBOT CONTROL

Indoor mobile robots are one of the widely researched and developing technologies in robotic field since they can be used in the service robotics and industrial application domains. Moreover, a sub-category of mobile robots, omnidirectional mobile robots, allow performing tasks in narrow indoor spaces by providing better motion capabilities. Additionally, redundancy in mobile robots is started to be used for various advantages including fault tolerance and increased payloads.

The objective of this thesis is to improve the design of the omnidirectional mobile robot that was previously constructed in IRL (IzTech Robotics Lab) and develop a redundancy resolution algorithm in order to control the redundant omnidirectional mobile robot to tolerate faults in the actuation system. Initially, the mechanical structure of the mobile robot is improved by the addition of a suspension system for each wheel assembly. A new onboard controller hardware is used and a new top-level controller is employed to be used along the redundancy resolution algorithm. Additionally, previously developed obstacle avoidance algorithm is improved by employing a new configuration of sensors and including a virtual damper to compensate for variable velocity level while approaching to an obstacle.

The fault tolerance algorithm is developed in this thesis by integrating a pseudo inverse of the Jacobian matrix that is subjected to a virtual weighted matrix so that the motion of the mobile robot will sustain its motion even though there is an efficiency drop in one of the actuators. Top-level control algorithm along with the fault tolerance and the obstacle avoidance algorithms is experimentally tested and test results indicate that the mobile robot can achieve the primary task in the case of one of the actuator's efficiency drops down to 70% or in the case of multiple obstacles on the path of the robot.

ÖZET

ARTIKSIL MOBİL ROBOT KONTROLÜ

İç mekan mobil robotları, servis robotları ve endüstriyel alanlar gibi bir cok alanda kullanıldığı icin en çok araştırılan ve geliştirilmekte olan teknolojilerden biridir. Dahası, mobil robotların bir alt kategorisi olan çok yönlü mobil robotlar, dar mekan alanlarında daha iyi hareket kabiliyeti sağlar. Ek olarak artıksılllık mobil robotlarda farklı yardımcı görevleri gerekleştirmekte kullanılır.

Bu tezin amacı IRL (İyte Robotik Laboratuvarı) üretilmiş olan çok yönlü mobil robotun tasarımını iyileştirmek ve hareket sistemindeki arızaları tolere edecek şekilde artıksıl çok yönlü robotu çözüm algoritması geliştirmektir. İlk olarak, mobil robotun mekanik yapısı, bütün tekerleklere süspansiyon sistemi ilave edilerek geliştirildi. Yeni bir yerleşik denetleyici donanımı kullanıldı ve yeni üst düzey kontrolcü yaratıldı. Buna ek olarak, daha önce geliştirilmiş engelden kaçış algoritması, engellere yaklaşırken değişken hız seviyesini telafi etmek için yeni sensör konfigürasyonu ve sanal sönümleyici kullanılarak geliştirildi.

Bu tezde, hata tolare etme algoritması, hareketli bir robotun hareketi, hareket sistemlerinden birinde verim düşüşü olsa bile hareketini sürdürecek şekilde sanal bir ağırlık matrisine tabi tutulan Jacobian matrisinin sahte bir tersinin entegre edilmesiyle geliştirilmiştir. Engelden kaçınma algoritması ve hata toleransı ile birlikte üst düzey kontrol algoritması deneysel olarak test edilmiştir. Test sonuçları, mobil robotun hareket sistemlerinden birinin verimlilik düşüşünü %70 oranında telafi edebildiğini veya robot yolundaki birden çok engelle karşılaşması durumunda görevini yerine getirebildiğini göstermektedir.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF SYMBOLS

V_x, V_y	Mobile Rbot Linear Velocity
ω_v	Mobile Robot Rotational Velocity
ϑ	Velocity vector's angle
V_i	Wheel's linear velocity
ω_i	Wheel's angular velocity
L	Link length
$B_{vx,}$ $B_{vy,}$ B_w	coefficient of viscous friction
$C_{vx,}$ $C_{vy,}$ C_w	coefficient of coulomb friction
F	vehicle force
Γ	torque around z direction
T	Motor torque
K_T	Torque constant
i_a	actuator current
f_i	wheel's traction forces
k	coefficient of spring
c	coefficient of damper
J	Jacobian Matrix
J^+	Pseudo inverse of the Jacobian Matrix
J_w^+	weighted Pseudo inverse of the Jacobian Matrix
W	diagonal weight matrix
W_i	performance coefficient of actuator

CHAPTER 1

INTRODUCTION

A mobile robot is an automatic machine which is able to move in the unlimited surrounding environment. The mobile robot technology has been developing since the 1940s. The first examples of mobile robots were a tracked robot and a remotely controlled main battle tank which were used for military tasks. Since then, mobile robots are widely investigated. They are mostly employed because of their ability to be used in different terrains for a variety of applications. The field of mobile robotics consists of a complete and concise study of modeling, control, navigation methods, and trajectory planning for each different type of mobility. Hence, it is a field that colligates many different disciplines such as mechanical, system, electronic, mechatronics, and computer engineering.

The interest in mobile robots is continuously increasing since due to the freedom of movement, mobile robots are more flexible and can perform tasks that cover larger spaces than their conventional fixed counterparts. Mobile robots are also classified depending on their working environments as illustrated in Figure 1.1. Since this thesis focuses on ground mobile robots, the classification of ground mobile robots is also given considering their mobility as shown in Figure 1.1.

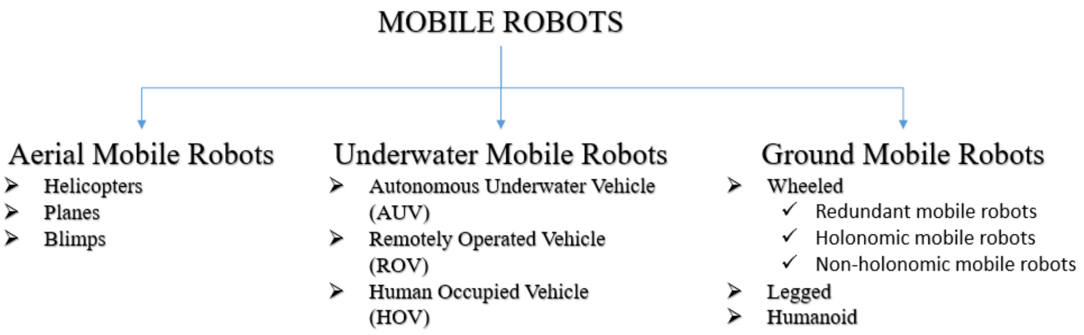

MOBILE ROBOTS
Aerial Mobile Robots
➢ Helicopters
➢ Planes
➢ Blimps
Underwater Mobile Robots
➢ Autonomous Underwater Vehicle (AUV)
➢ Remotely Operated Vehicle (ROV)
➢ Human Occupied Vehicle (HOV)
Ground Mobile Robots
➢ Wheeled
✓ Redundant mobile robots
✓ Holonomic mobile robots
✓ Non-holonomic mobile robots
➢ Legged
➢ Humanoid

Figure 1.1 Classification of mobile robots

(source: Jizhong Xiao- Mobile Robot Locomotion)

1.1 Aerial Robots

Aerial robots, which are also called drones, are commonly used in military and special purpose operations. Beyond the military applications, aerial robots have become more popular for civilian usage having a wide variety of applications such as carrying drones, search and rescue robots, film-making, delivering medical supplies to remote and inaccessible regions. Amazon carrier drone, DJI phantom drone and SpaceX rocket can be given as recent examples of aerial robots as shown in Figure 1.2.

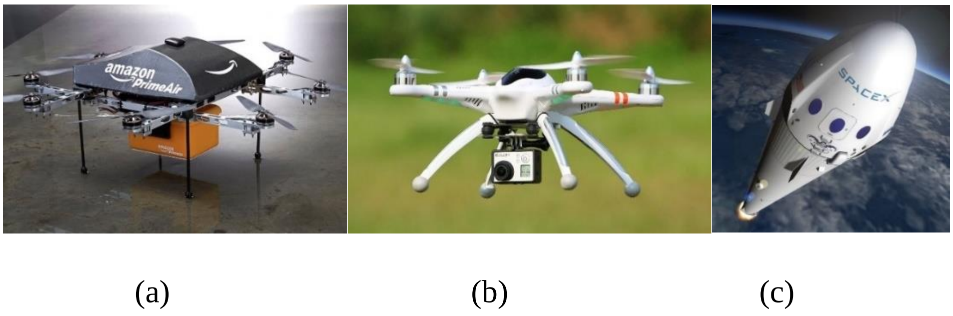


Figure 1.2 Aerial Robots

(a)- Amazon carrier drone (source: Amazon, 2016)

(b)- DJI phantom drone (source: Oddgifts, 2016)

(c)- SpaceX aerial robots (source: SpaceX, 2016)

1.2 Underwater Robots

Underwater Robots are designed to perform operations in deep sea and shore environments. Underwater mobile robots, which are generally known as ROV (remotely operated vehicle), can also travel underwater autonomously without any input from the user. These type of underwater vehicles is classified as AUVs (autonomous underwater vehicle). Underwater robots are generally used in pipeline investigations, mine detecting, at aquafarms. Additionally, underwater robots are used in scientific researches to study lakes, the ocean, and the ocean floor. A variety of sensors were adapted to underwater robots to measure the concentration of various elements or compounds, the absorption or reflection of light, and the presence of microscopic life in these research applications. In Figure 1.3 different underwater vehicles are presented.

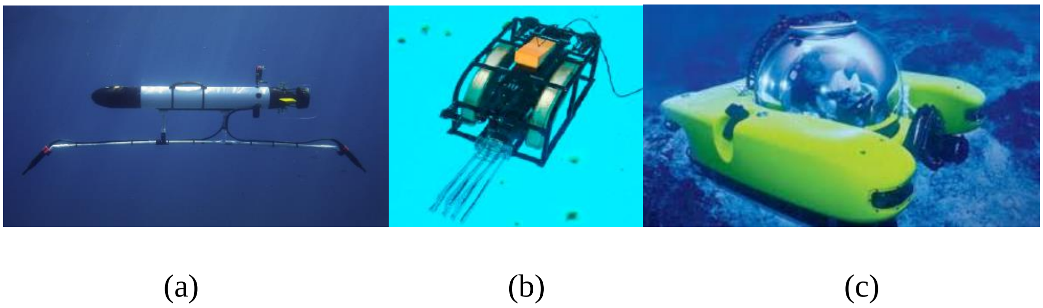


Figure 1.3 Underwater robots

(a)-Autonomous Underwater Vehicle (AUV) (White et al, 2016)

(b)-Remotely Operated Vehicle (ROV) (Sadao, 2015)

(c)-Human Occupied Vehicle (HOV) (Mnn, 2016)

1.3 Ground Mobile Robots

Ground mobile robots are remote, autonomously or semi-autonomously controlled vehicles that travel on any surface. Ground Mobile robots could be categorized with respect to the environments they are used in as indoor and outdoor mobile robots. The capability of these mobile robots varies depending on their environments.

1.3.1 Outdoor Mobile Robots

Outdoor mobile robots are designed to perform operations in outdoor environments which mostly consist of challenging terrains. Outdoor mobile robots are used in a wide variety of applications including military operations, spatial and underwater explorations, automatic agricultural operations, civil protection and fire surveillance, mining operations, hazardous-waste disposal, and entertainment. Mobile robot’s operating environments are often unknown and/or unstructured. Since they can be used in unknown and rough environments, they are constructed to be more durable and robust to disturbances existing in the environment. In Figure 1.4, a concept mobile robot which is designed for space exploration application is shown.

Figure 1.4 Outdoor mobile robots

(source: Audi, 2016)

1.3.2 Indoor Mobile Robots

Indoor mobile robots are developed to perform a variety of tasks inside commercial and civil buildings. Nowadays indoor mobile robots are a popular topic in academic studies since they are relatively easier to build and study on. Indoor mobile robots have delicate mechanism/structure when compared to outdoor mobile robots since they are generally designed to work on smooth surfaces. Typical indoor applications of mobile robots are cleaning, service robotics, surveillance, and stocking in warehouses. In Figure 1.5, different examples of indoor mobile robots are presented.

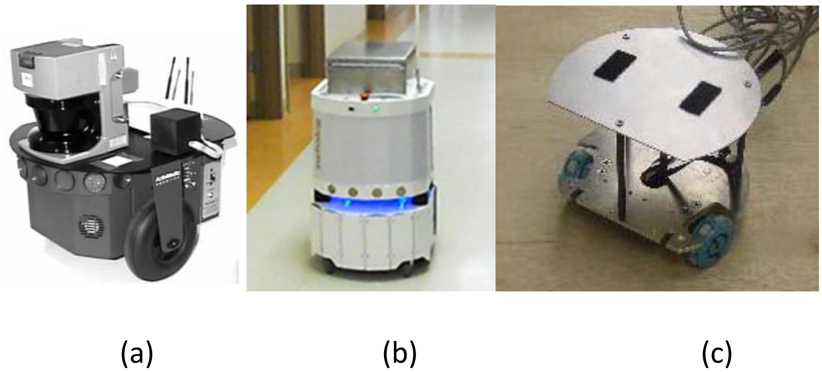


Figure 1.5 Indoor mobile robots

(a)- Pioneer 2-DX mobile robot (source: Gerkey et al. 2003)

(b)- Automated Guided Vehicle in healthcare facilities (source:Swisslog, 2016)

(c)-omnidirectional mobile robot (source: Williams et al. 2002)

Indoor mobile robots can be categorized according to their motion types as nonholonomic and holonomic motions. The details of these categories are described in following subsections briefly.

1.3.2.1 Nonholonomic Mobile Robots

The nonholonomic term in mobile robots means that the state of the motion depends on the path taken to achieve it. In robotics, if the controllable degrees of freedom (DoF) are less than the total DoF required by the task, the robot is called to be nonholonomic. This kind of mobile robots cannot move in the direction of wheel's rotation axes. An example of the nonholonomic mobile robot is given in Figure 1.6. This robot has 2 DoF to the motion.

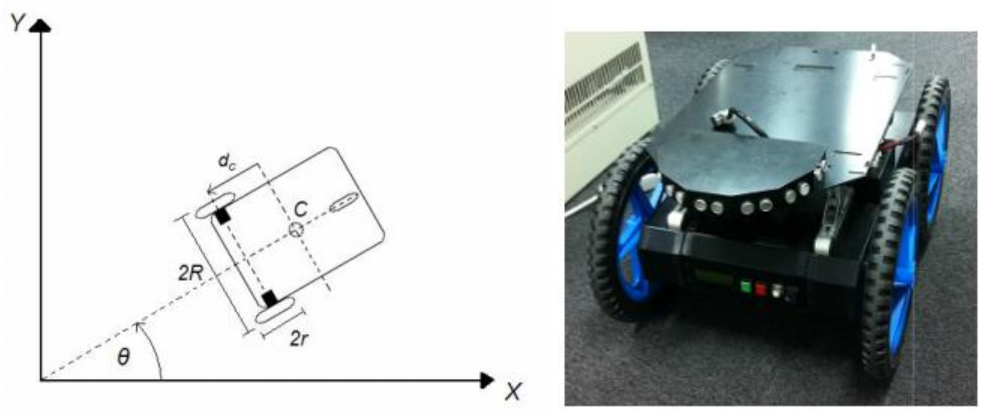


Figure 1.6 Nonholonomic Mobile Robot (source: Normalesup, 2016)

1.3.2.2 Holonomic Mobile Robots

A subtype of indoor mobile robots is holonomic mobile robots. Holonomic refers to the relationship between controllable and total DoF of a robot. If the controllable DoF is equal to total DoF, the robot is holonomic. A robot which has omnidirectional wheels is a good example of holonomic mobile robots. Holonomic mobile robots are capable of translational and rotational motions independently and simultaneously on the surface which they move on. As a result of this advantage, holonomic mobile robots are widely used in many applications in which the tasks are carried out in narrow spaces such as transporting of goods in hospitals, public cleaning, factories and sheltered workshops for disabled people. An example of the 3 DoF holonomic mobile robot is given in Figure 1.7.

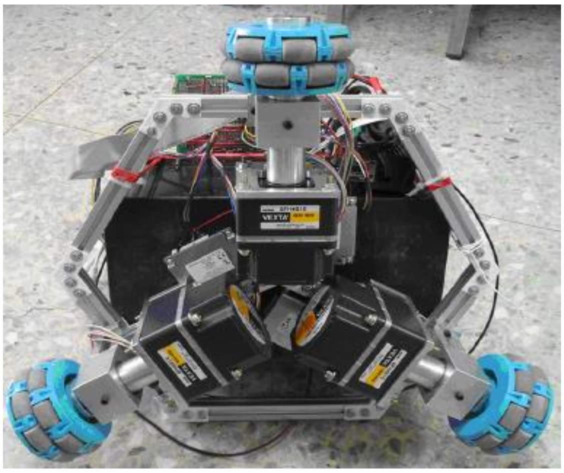


Figure 1.7 Omnidirectional mobile robot (Huang and Tsai, 2008)

The differences between nonholonomic and holonomic motion are illustrated in Figure 1.8. Nonholonomic motion is illustrated on the left side of Figure 1.8. The linear motion of the mobile robot is depended on the robot's orientation. On the other hand, in holonomic motion which is presented on the right side of the figure, the linear motion is independent of the robot's orientation.

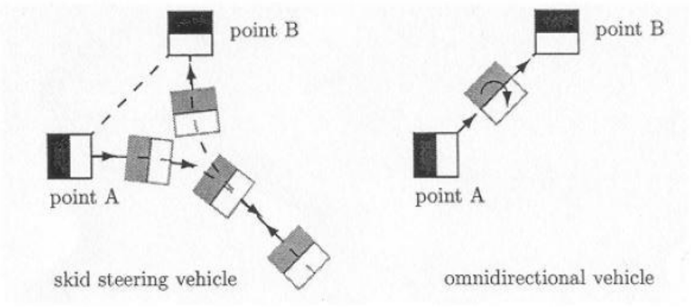


Figure 1.8 Moving from A to B: skid-steering and omnidirectional movement (Baede, 2006)

1.3.2.3 Redundant Mobile Robots

Basically, kinematic redundancy is related to the relationship of controllable DoF and task's DoF. If controllable DoF is more than the DoF required by the task, then the robot is called kinematically redundant. A total of 3 DoF is required for ground mobile robots to achieve their task on a surface. More than 3 actuators form redundancy for ground type robots.

One of the main purposes of kinematical redundancy is to increase the sensitivity (or quantity) of the solution. More than necessary actuators result in a redundant system so that the robot could accomplish its tasks with different combinations of motor speeds which provide the same Euclidean movement.

Another usage area of the redundant mobile robot is the transportation of goods for internal logistics. The advantage of redundant mobile robots is that they could be used for heavy duty tasks such as in application where they carry robotic arms and heavy payloads. A specific example of the redundant mobile robot for heavy duty task is given in Figure 1.9.

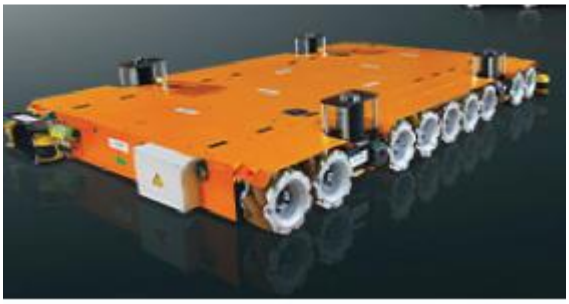


Figure 1.9 Redundant Mobile Robot for internal logistic (KUKA-robotics)

1.4 Objective of the Thesis

The aims of this thesis were redesigning the previously developed omnidirectional mobile robot by using new hardware and by implementing top-level controller along with a redundancy resolving algorithm to employ fault tolerance in the case a failure in actuation. Additionally, the previously developed obstacle avoidance algorithm is replaced with a new one that includes a virtual damping element.

In order to carry out the previously mentioned implementation of control algorithms, the following modifications and additions in hardware are made.

- A suspension is designed and assembled to the actuation system of the mobile robot.
- Previous controller, Prometheus by Diamond Systems, is replaced with UEISIM 600-1G data acquisition system produced by United Electronic Industries.
- The quantities of infrared sensors are increased and they are replaced at all eight corners of the octagonal design of the mobile platform.
- Magnetic encoders are assembled to the DC motors.

The work carried out in the controller development is summarized as follows:

- The PID controller in the motor driver unit is tuned and a top-level controller is designed and optimized to compensate for the shifts in the direction of the mobile robot.
- Virtual spring-damper model is used in the obstacle avoidance algorithm.
- Weighted pseudo-inverse of the Jacobian matrix is used in the case of a malfunctioning wheel in order to tolerate the failure.

1.5 Outline

In the following chapter, literature survey on control of mobile robots, fault toleration concept in mobile robots and previous studies carried out for obstacle avoidance is presented. The description of hardware is covered in the third chapter. This chapter consists of general information of the developed mobile robot and specifically its hardware and software. The fourth chapter is about the system modeling which consists of the description of the kinematic and the dynamic model of the developed mobile robot, and actuation system modeling. The fifth chapter presents experimental test procedures and test results for fault tolerance and obstacle avoidance algorithms. In the last chapter, the conclusion is given and future work is addressed.

CHAPTER 2

LITERATURE SURVEY

The popularity of mobile robots is increasing due to their enhanced mobility with respect to traditional robots. Holonomic robots are highly preferred systems for their advantages in maneuverability and effectiveness. The aim of this chapter is to present a survey on control algorithms of mobile robots, obstacle avoidance algorithms, fault toleration, and mobile robot components.

2.1 Survey on Control of Holonomic Mobile Robots

In (Watanabe, 1998), a review of omnidirectional mobile robots and their dynamic models are provided in which three lateral orthogonal-wheel assemblies are presented. Later, four different control strategies such as PID control method, resolved acceleration control method, fuzzy model control method, and stochastic fuzzy servo control method are discussed to provide a path for developing a feedback control system for a three-wheeled omnidirectional robot.

In Spoelder's study (2003) a neuro-fuzzy controller for sensor-based mobile robot navigation in indoor environments is presented. In this study, the control system is divided into high-level behaviors and low-level behaviors with respect to the hierarchy of robot behaviors. They also investigate their sub-behaviors in which each basic behavior reacts to data received from specific sensors.

Simpson, Jacobsen, and Jadud (2006) presented control approaches for mobile robots which are named by themselves as the subsumption architecture and Occam-pi. Occam-pi is a programming language that uses process-oriented programming model. Subsumption architecture means that a basic control system can be established for the lowest hardware level functionality of the robot and additional levels of competence can be built on top, with the little change required to the existing modules and layers. This style of control is adapted from the biological idea of reflex and cognitive actions.

In Rojas and Förster's study (2006), optimal control of an omnidirectional mobile robot is presented for a four-wheel robot and generalized for controlling mobile robots with larger number of wheels. Computing a set of consistent and optimal motor forces and speed for over-determined systems is explained. Also, the controller allows to determine if wheel is slipping on the ground or not. As a conclusion, presented method is useful for both avoiding and handling wheel slippage.

Omrcen, Zlajpah and Nemec's (2006) study presented a new approach for control of mobile robots. They called this approach as "combined control", which combined two different types of robot control, i.e. the torque and velocity control in a single robot system. Their first controller, the velocity controller, controlled each joints velocity by a PID controller with optimized parameters. The second one is torque controller. Resolved-acceleration control was used for the torque control.

In the study carried out by Liu, Zhu, Williams and Wu (2007), a nonlinear controller design for an omnidirectional mobile robot is developed. The robot controller comprised an outer-loop controller and an inner-loop controller. These are both designed using the Trajectory Linearization Control method based on a nonlinear robot dynamic model. An open-loop controller computes both the nominal control and the nominal trajectory by using a pseudo-dynamic inverse of the plant model. The second part of the control is a feedback controller which corrected the system's tracking error.

An adaptive backstepping control method for trajectory tracking and stabilization of an omnidirectional mobile robot with parameter variations and uncertainties caused by friction and slip is presented in Huang and Tsai's study (2008). They derived a second-order nonlinear dynamic model with the simplified characteristics of the used DC servomotors for three-wheeled mobile robots. This model with two unknown parameters and bounded uncertainties is constructed via the Newton's second motion law. Experimental results have shown that the advised controller is capable of dealing with the basic navigation problems which are stabilization about a desired posture and tracking a reference trajectory.

Another study about mobile robot control is done by Wang, Tsai, and Wang (2010). A dynamic motion control is designed for position control and trajectory tracking of the holonomic mobile platform which has four omnidirectional wheels. The suggested dynamic motion controller is synthesized using backstepping method to achieve both position control and time-varying trajectory tracking. The controller can be extended to carry out path tracking missions. According to simulation result, the proposed dynamic motion controller is capable of accomplishing the desired motions.

In the study presented in (Hashemi et al., 2011), the authors suggested and examined a PI-fuzzy path planner and associated low-level control system for a linear discrete dynamic model of omnidirectional mobile robots to obtain optimal inputs for drivers. Filtering of the velocity and acceleration is also applied in the path planner to satisfy planning prerequisites and prevent slippage.

2.2 Survey on Obstacle Avoidance Algorithms for Mobile Robots

Obstacle avoidance is utilized in robotics where there is a risk of collision with the obstacles existing in the operation environment. To overcome this problem, obstacle avoidance is realized in control algorithm with a variety ways. The obstacle avoidance concept is especially needed in mobile robot applications in which there might exist various obstacles to performing a task especially in the urban areas. The obstacle avoidance can be integrated into the system either as pre-computed obstacle free path planning or as a real-time algorithm with relevant feedback which is used mostly in the autonomous and semi-autonomous systems.

In (Fujimori et al., 1997), obstacle avoidance of a mobile robot was handled with a local navigation technique in which the dynamics of the robot are taken into consideration. This method requires only the distance information between the robot and the obstacle in three specified directions. The distance information in these three specified directions are the center *"d_c"*, right *"d_r"* and left *"d_l"* distances. According to these information, there are three assumptions which are (i) move only in the forward direction, (ii) turn left or (iii) turn right with the minimum rotation radius and its velocity is constant until the robot is near the goal. The navigation law is a first order differential equation. Navigation to the target and obstacle avoidance are accomplished by switching the motion direction angle of the robot.

If all sensors have detected the obstacle at the same time, which is illustrated in Figure 2.1, the desirable direction angle θ_a^* for avoiding obstacle is calculated as presented below.

$$\theta_a^*(t) = \theta(t) + sgn(d_l - d_r)(\frac{\pi}{2} - \varepsilon) \tag{2.1}$$

$$\varepsilon = tan^{-1}\left(\frac{d.cos\alpha - d_c}{d.sin\alpha}\right), \left(|\varepsilon| < \frac{\pi}{2}\right) \tag{2.2}$$

Where

$$d \triangleq \max(d_r, d_l) \tag{2.3}$$

$$sgn(x) = \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases} \tag{2.4}$$

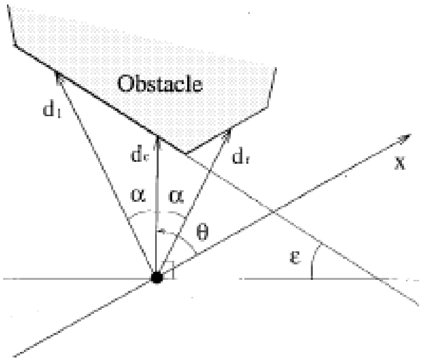


Figure 2.1 The case of three sensors detecting the obstacle

(source: Nikiforuk and Gupta, 1997)

According to Figure 2.2, if two sensors have detected the obstacle, the desirable direction angle for avoiding the obstacle is calculated as:

$$\theta_a^*(t) = \theta(t) + sgn(d_l - d_r)\left(\frac{\pi}{2} - \varepsilon\right) + sgn(d_l - d_r)\pi \tag{2.5}$$

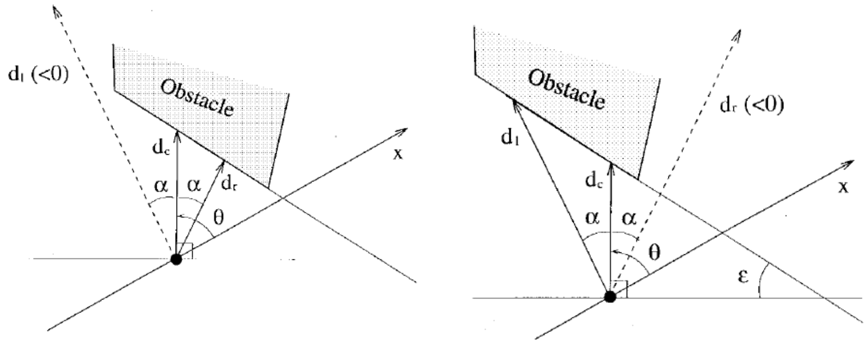


Figure 2.2 The case of two sensors detecting the obstacle

(source: Nikiforuk and Gupta, 1997)

According to Figure 2.3, when there is no obstacle in the center direction, the direction angle is calculated as:

$$\theta_a^*(t) = \theta(t) \tag{2.6}$$

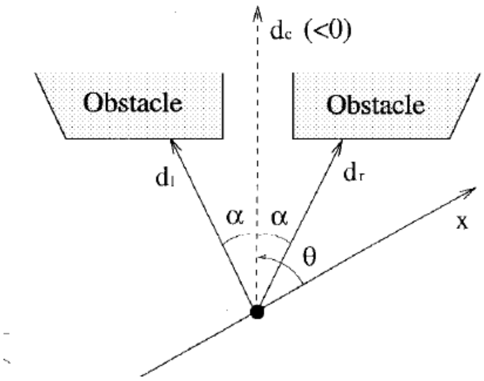


Figure 2.3 The case when the sensor for detecting the center distance does not detect an obstacle and the other two sensors detecting the obstacle

(source: Nikiforuk and Gupta, 1997)

According to Figure 2.4 below, there are two cases to be taken into consideration when only one sensor has detected the obstacle. Therefore, the desirable direction angle is calculated as below.

$$\theta_a^*(t) = \theta(t) + \frac{\pi}{2} \tag{2.7}$$

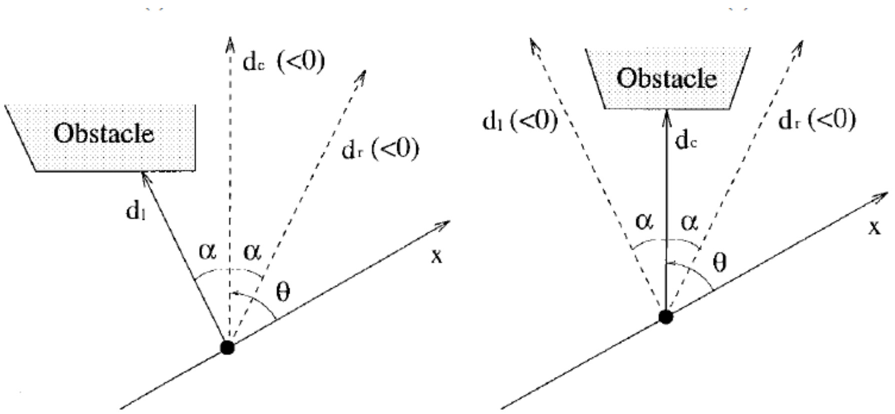


Figure 2.4 The case of only one sensor detected the obstacle

(source: Nikiforuk and Gupta, 1997)

The other approach is to use artificial potential energy for obstacle avoidance. Ge and Cui’s study (2000) presents new repulsive potential functions which take the relative distance between the robot and the goal into consideration. The global minimum of the total potential ensured the goal destination. The basic concept of the potential field method is to fill the robot’s workspace with an artificial potential field in which the robot is attracted to its target position while staying away from the obstacles.

In the case of the mobile robot moving on one-dimensional path, the attractive potential $U_{att}(q)$ and the repulsive potential $U_{rep}(q)$ are given by:

$$U_{att}(q) = \frac{1}{2}\xi x^2 \tag{2.8}$$

$$U_{rep}(q) = \frac{1}{2}\eta\left(\frac{1}{x_{obs} - x} - \frac{1}{\rho_0}\right)^2 \tag{2.9}$$

Here, q is the position of the mobile robot in the workspace, ξ and η are positive scaling factors, ρ_0 is a positive constant denoting the influence of the obstacle’s distance.

An example of the mobile robot moving in a one-dimensional path is illustrated in Figure 2.5, which consists of robot’s actual position, target position and obstacle position.

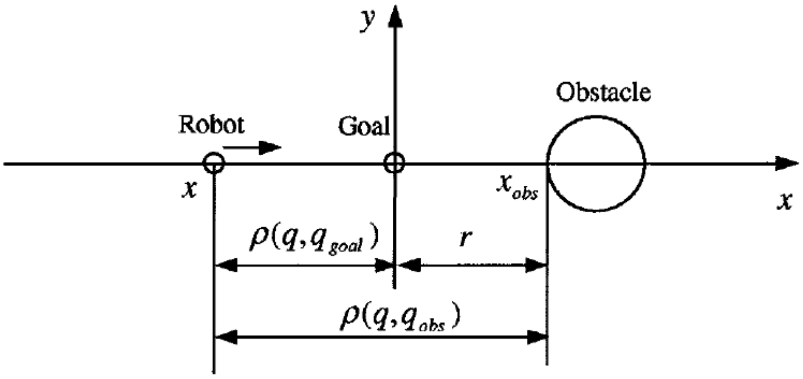


Figure 2.5 Locations of the robot, goal, and obstacle in a 1-D

(source: Ge and Cui, 2000)

Figure 2.6 shows the relation between robot position and virtual potential energy changes with respect to x direction according to specific assumption, that is $x_{obs} = 0.5$, $\rho_0 = 2$, and $\xi = \eta = 1$,

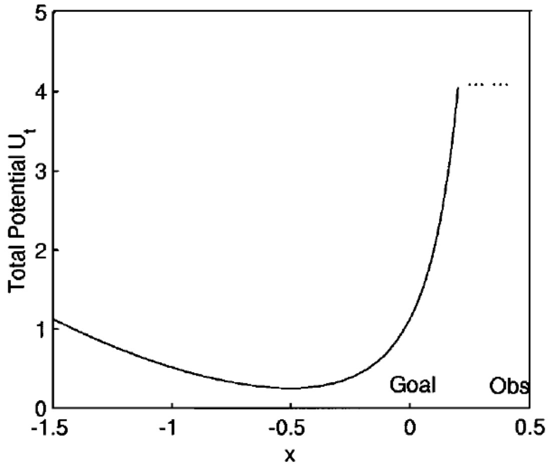


Figure 2.6 Total potential function in a 1-D case

(source: Ge and Cui, 2000)

In (Pai et al., 2012), the obstacle avoidance algorithm is established with both fuzzy and extension theory. Fuzzy theory is introduced for tuning PWM signal for motor velocity and corrects path deviations during robot motion. Extension theory is used to describe the inference process of obstacle avoidance, which allows to convert a complex problem in the real world into one expressed through extension element model.

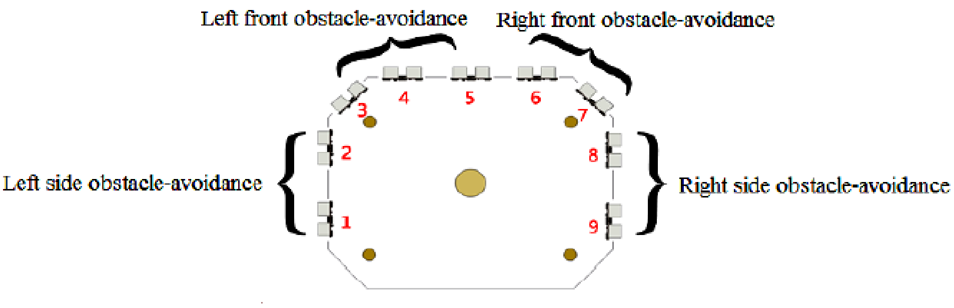
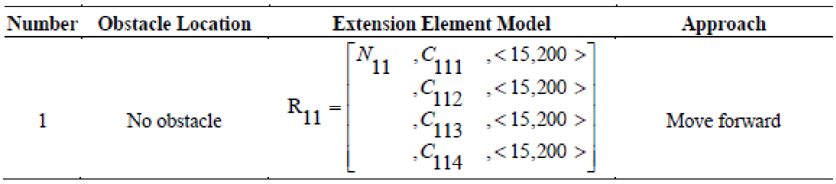

Left front obstacle-avoidance
Right front obstacle-avoidance
Left side obstacle-avoidance
Right side obstacle-avoidance
1
2
3
4
5
6
7
8
9

Figure 2.7 Ultrasonic sensor groups on the robot's frame

(source: Pai et al., 2012)

The robot plans its motion according to extension set characteristics which are name (N), characteristics (C) and characteristics value (V). The extension set characteristics are calculated from the set of data which are collected from sensor groups, which is illustrated in Figure 2.7, and extension element model is created with these characteristics.



Number	Obstacle Location	Extension Element Model	Approach
1	No obstacle	$R_{11} = \begin{bmatrix} N_{11} & ,C_{111} & ,<15,200> \\ & ,C_{112} & ,<15,200> \\ & ,C_{113} & ,<15,200> \\ & ,C_{114} & ,<15,200> \end{bmatrix}$	Move forward

Figure 2.8 the extension element model for no obstacle case

(source: Pai et al., 2012)

In the model presented in Figure 2.8, where no obstacle case is represented, the possible mobile robot motions are assigned to each extension element model as moving forward, move right, move left and so forth for the cases of obstacle detection.

2.4 Survey on Fault Tolerance for Mobile Robots

Fault tolerance is a comprehensive study area in robotics. Since robots are susceptible to failures in many applications, it is critical to take precautions and tolerate internal failures to successfully accomplish the task without the need for immediate human intervention. Faults in a robotic system can be classified as sensor-based faults, actuator-based faults, algorithm-based faults. However, this thesis focuses on only actuator-based faults.

Sensor-based fault tolerance is used to minimize faulty sensor's effect on control. Sensor based faults could be classified as bias, gain, drifting, precision, degradation, complete failure, or noise.

Algorithm-based fault tolerance is used to detect errors resulting from transient as well as permanent faults. Typical algorithm-based faults are encountered during encoding of data to be processed, redesign of target algorithm to operate on encoded data and distribution of computation of multiple execution units.

Generally, actuator-based faults often occur due to external effects during the system's operation. The actuator's situation is usually described as fault-free case, loss of effectiveness or out of order.

In the work of (Rojas and Förster, 2006), the case of the loss of one actuator for four-wheeled mobile robot is studied and the mobile robot can continue to drive precisely and still maintain its holonomic motion. The studied approach utilized the motor traction force calculations. The motor traction forces are represented as:

$$f = (f_1, f_2, f_3, f_4)^T \tag{2.10}$$

In the case of the malfunction in the first motor, f_1 is always zero. However, remaining motors can provide holonomic motion. Furthermore, the necessary motor torques are created with given Equations 2.11, 2.12, 2.13. Equation 2.11, 2.12, 2.13 present the motor forces while moving on forward without letting it rotate, moving sideways to the right without letting it rotate and rotates the robot counterclockwise, respectively.

$$\hat{f}_f = (0, 0, -1, 1)^T \tag{2.11}$$

$$\hat{f}_s = (0, -1, 1, 0)^T \tag{2.12}$$

$$\hat{f}_r = (0, 1, 0, 1)^T \tag{2.13}$$

Next, the robot is driven by generating demands using Equation 2.14 with x- and y- direction velocity demands and traction forces.

$$\frac{v_x}{\sin(\varphi)}\hat{f}_s + \frac{v_y}{\cos(\varphi)}\hat{f}_s \tag{2.14}$$

Consequently, the robot can move in the desired direction, and the two inactive motors are rolled passively, without causing any major problems, but the robot can move only half as fast as the non-faulty situation.

In another work by Yanzhe et al. (2014), they demonstrated a self-adaptation framework, which is named as ReFrESH, that contains a mechanism for fault detection and fault mitigation. The aim of this mechanism is providing a diagnosable and maintainable infrastructure support to manage the task performance in the presence of unexpected uncertainties in a real-time operating system. Additionally, ReFrESH provides a self-adaptation support for both software and hardware functionality. By augmented port-based object through the attachment of a components "Evaluator" and "Estimator", ReFrESH enables the mobile robot to autonomously detect and locate faults.

Aside the sensor-based and algorithm-based faults, faults that might occur in the mobile robots can be classified into two main groups namely manufacturing related and actuator related faults. Actuator related faults can be solved by using kinematic redundancy of the mobile robot.

Kinematical redundancy can be described based on the DoF required by the task. Considering that "*n*" represents the robot's controllable DoFs and "*m*" represents the task's DoFs.

✓ Case 1: $n<m$

The mobile robot is called deficient. This refers to the non-holonomic robots.

✓ Case 2: $n=m$

The mobile robot is called normal (regular). This refers to the standard holonomic robots.

✓ Case 3: $n>m$

The mobile robot is called redundant.

Kinematic redundancy could be used for additional tasks such as;

- Avoiding obstacles
- Avoiding kinematic singularities
- Minimizing energy consumption
- Increasing safety
- Fault tolerance

Kinematic redundancy problems are generally resolved at the velocity level. The velocity-level kinematic equation for a general robotic system is given below,

$$\bar{V} = \hat{J}_p . \dot{\bar{q}} \tag{2.10}$$

Where $\hat{J}_p$ is the Jacobian matrix, which is formulated as:

$$\hat{J}_p = \begin{bmatrix} \bar{J}_{p_1} & \cdots & \bar{J}_{p_n} \\ \bar{J}_{a_1} & & \bar{J}_{a_n} \end{bmatrix} \tag{2.11}$$

- $\bar{J}_{p_k}$ is the tip point velocity influence coefficient
- $\bar{J}_{a_k}$ is the angular velocity influence coefficient

The inverse kinematic solution is used to find the actuator velocities for a given end effector velocity. For the redundant case, there are infinite number of solution, which provides the controller designer to issue further optimization. An optimized solution is given in Equation 2.12.

$$\dot{\bar{q}} = \hat{J}^{\#} \bar{V} \tag{2.12}$$

Where,

$$\hat{J}^{\#} = \hat{J}^T (\hat{J}\hat{J}^T)^{-1} \tag{2.13}$$

$\hat{J}^{\#}$ is called Moore-Penrose inverse or pseudo-inverse of the Jacobian matrix, since $\hat{J}\hat{J}^{\#} = \hat{I}$. The cost function used to formulate this pseudo-inverse is derived as:

$$g(\dot{\bar{q}}) = \frac{1}{2} \dot{\bar{q}}^T \widehat{W} \dot{\bar{q}} \tag{2.14}$$

Where $\widehat{W}$ is a *n* x *n* symmetric positive definite matrix. The inverse kinematics solution in velocity level is then derived as:

$$\dot{\bar{q}} = \hat{J}^{\#}_W \bar{V} \tag{2.15}$$

$$\hat{J}_W^{\#} = W^{-1}\hat{J}^T(\hat{J}W^{-1}\hat{J}^T)^{-1} \tag{2.16}$$

where $\hat{J}_W^{\#}$ is called weighted pseudo-inverse of the Jacobian matrix.

2.5 Holonomic Mobile Robot System Components

In this section, the components of the holonomic mobile robots are investigated. The main parts of holonomic mobile robots are omnidirectional wheels, motors, motor controllers, sensors, batteries and data acquisition (DAQ) system.

2.5.1 Omnidirectional wheels

An omnidirectional wheel comprises a mechanical assembly of a rotating hub connected to the frame of the wheels and a series of peripheral free rotating rollers attached to it. The omnidirectional wheel can be actuated to roll forward like a normal wheel or passively roll laterally. The omnidirectional wheels are the key components to developing a holonomic robot. Making use of the passive lateral movement of omnidirectional wheels, the holonomic mobile robots can move in arbitrary direction without the need changing the direction of the wheels or the platform. In the past years, omnidirectional wheels have also been used in wheelchairs and in service vehicles in airports.

One of the first modern omnidirectional wheels is designed by the Swedish inventor Bengt Ilon around 1973 which is given in Figure 2.9.

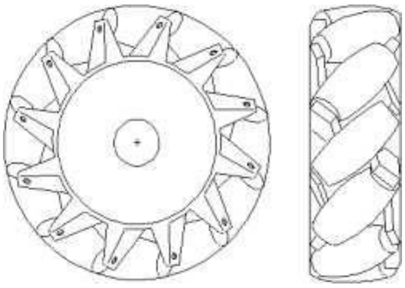
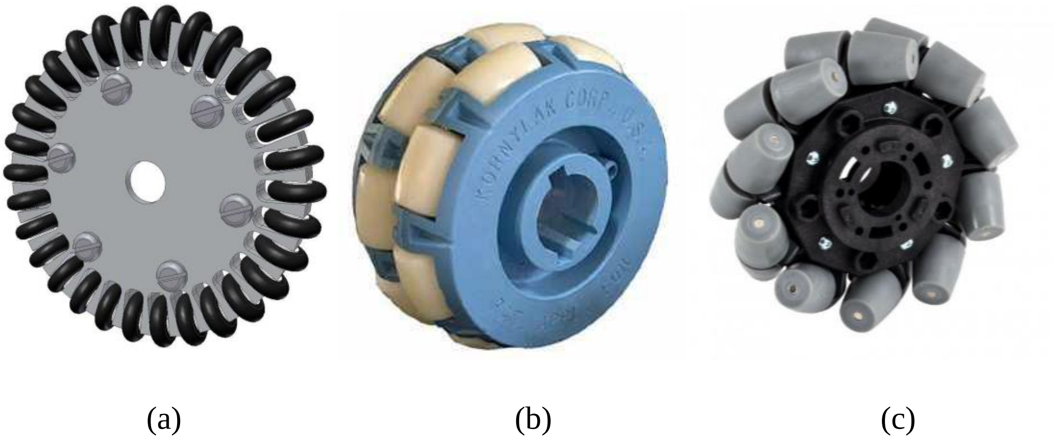


Figure 2.9 Omnidirectional wheel designed by Bengt Ilon

Ilon’s wheel is based on the principle of a central wheel with rollers assembled at an angle around the periphery of the wheel. Some of the traction force is along the mobile platform’s movement direction and the rest of it is acting towards a perpendicular direction which needs to be compensated by the traction force components of the other wheels. The profile of the wheels is nearly circular, which provides a smoother and continuous motion.

There are several designs of omnidirectional wheels as show in Figure 2.10. The term omnidirectional wheel is widely used as the synonym of the Mecanum wheel, however, the Mecanum wheel is a particular design of omnidirectional wheel.



(a) (b) (c)

Figure 2.10 (a)- Segmented omnidirectional wheel (b)-A commercial double omnidirectional wheel (c)- A mecanum wheel

The segmented omnidirectional and commercial omnidirectional wheels are based on the principle of a central wheel with passive rollers assembled periphery of the wheel. A mecanum wheel is based on the design of Ilon and sometimes named as the Swedish wheel.

The features of the omnidirectional wheels are given in Table 2.1.

Table 2.1 Omnidirectional Wheel Features

Advantages	Weak points
Capability to be mounted on a car-like platform	Limited mobility on uneven terrain particularly in lateral motion
Requires only one actuator with individual control per wheel to achieve omnidirectional motion	More complex production than spherical wheels
High payload capability	Its applications are constrained by a limited set of allowable wheel position configurations

2.5.2 Actuators for Mobile Robots

Typical actuators used in mobile robot applications are powered by hydraulics, pneumatics and electro-magnetic. Some characteristic features of actuator types are listed in the following paragraphs.

Hydraulic actuators have the greatest power to size ratio among all. This makes them advantageous when there is a need for high input power in small areas. In the work of (Tanaka et al., 2005), hydraulic actuators are utilized for their high-power output for accomplishing difficult rescue tasks. Moreover, they are simple actuators. However, control of these type of actuators is cumbersome and expensive compared to the electric actuators. Pneumatic actuators have common features with the hydraulic ones with only one difference that is the power source is air. In addition, pneumatic actuators provide some elasticity in the output and it is faster than the hydraulic actuators due to their reduced inertia. In (Granosik and Borenstein, 2004), an example of pneumatic driven actuator control for a mobile robot is explained.

Another type of the actuator is the electric driven actuators which convert electrical energy to mechanical energy. By using commutator, electric actuators or electric motors (EM) generate a rotating magnetic field to run the rotor. EMs can be grouped using the supply voltages which are DC and AC. In (Niku, 2001) actuators that are mainly used in robotics are discussed which are categorized as; DC Motors, AC Motors, Brushless DC motors, Direct Drive Electric Motors, Servomotors, and Stepper Motors.

An AC motor is an electric motor driven by an alternating current. AC motors are also divided as synchrony and asynchrony motors. The synchrony motors are used in heavy-duty jobs. On the other hand, the standard types of AC motors are asynchrony. The basic working principle is the same, which is driving the mass made by magnetic sheets via rotating electromagnetic fields. Although synchronous motor works in the same frequency with supplied current's frequency, asynchronous motors cannot reach the supplied current's frequency because of induction current. Generally, AC motors are working with three phase system.

DC motors work with direct current. DC motors are categorized into brushed and brushless DC motors. The brushed DC motor's speed is varied by changing the operating voltage or the strength of the magnetic field. The brushed DC motors are still being used in electrical propulsion, cranes and steel rolling mills. In (Yime et al., 2014), Design of a brushed DC motors PID controller for development of low-cost robotic applications is presented. The brushless motors do not have brushes and collectors. That is why this kind of motors is named brushless motors (BLDC). In the recent years, brushless motor started to be used in the robotics area in an increasing trend. The brushless motors are more efficient than similar sized brushed DC motors. In (Romero and Cocha, 2006), control of position/velocity is executed with DC brushless motors for mobile robots. Furthermore, brushless motors are more reliable, silent and durable. Brushless motors are more suitable for very various industrial applications and robotic fields.

2.5.3 Motor Controller

A motor driver is an amplifier with some embedded controller options. Generally, the function of motor drivers is to take a low-current control signal and then amplify a higher current signal that can drive a motor. The motor drivers could be categorized with respect to their drive types such as voltage (speed) drive and current (torque) drive. Motor drivers vary in their maximum supply voltage, maximum output current, rated power dissipation, load voltage and number of outputs. In the work of

(Salih et al., 2006), speed controller type motor drivers are used in mobile robot which can be given as an example for speed drive motor controllers. On the other hand, in the study of (Fierro and Lewis, 1996) control of the mobile robot is realized in the torque drive mode where computed torque method is used for driving the mobile robot in which wheel slippage and frictional effects are considered.

2.5.4 Sensors

Sensors are used to measure physical changes in its environment and convert this physical quantity into electrical signals. In the robotics area, sensors have crucial importance to acquire information from and understand robot's environment. In fact, robot's capabilities are related with sensors limits and precision.

Table 2.2 Classification of sensors used in mobile robotics applications (source:Siegwart and Norbakhsh, 2004)

General classification (typical use)	Sensor Sensor System	PC or EC	A or P
Tactile sensors (detection of physical contact or closeness; security switches)	Contact switches, bumpers Optical barriers Noncontact proximity sensors	EC EC EC	P A A
Wheel/motor sensors (wheel/motor speed and position)	Brush encoders Potentiometers Synchros, resolvers Optical encoders Magnetic encoders Inductive encoders Capacitive encoders	PC PC PC PC PC PC PC	P P A A A A A
Heading sensors (orientation of the robot in relation to a fixed reference frame)	Compass Gyroscopes Inclinometers	EC PC EC	P P A/P
Ground-based beacons (localization in a fixed reference frame)	GPS Active optical or RF beacons Active ultrasonic beacons Reflective beacons	EC EC EC EC	A A A A
Active ranging (reflectivity, time-of-flight, and geometric triangulation)	Reflectivity sensors Ultrasonic sensor Laser rangefinder Optical triangulation (1D) Structured light (2D)	EC EC EC EC EC	A A A A A
Motion/speed sensors (speed relative to fixed or moving objects)	Doppler radar Doppler sound	EC EC	A A
Vision-based sensors (visual ranging, whole-image analysis, segmentation, object recognition)	CCD/CMOS camera(s) Visual ranging packages Object tracking packages	EC	P

A, active; P, passive; P/A, passive/active; PC, proprioceptive; EC, exteroceptive.

The various types of sensors that are used in building mobile robots are tabulated in Table 2.2. Among these sensor, the working principles of the sensor that re used in this thesis study are presented in the next paragraphs.

- Gyroscope sensor – is a device that can measure angular velocity. The new generation gyroscope sensors are classified by output types as analog and digital. Inner and substrate representation of the gyroscope sensor is given in Figure 2.11.

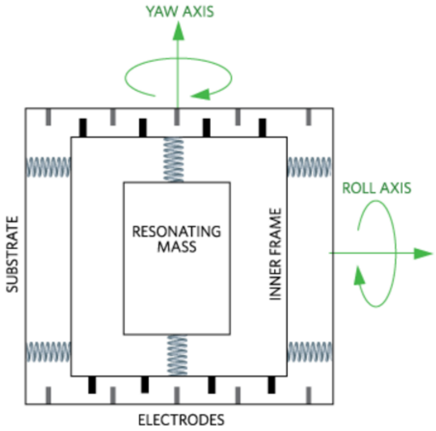


Figure 2.11 Inner and substrate representation of gyroscope sensor

(source: Maximintegrated, 2016)

A gyroscope uses the capacitive sensing method to measure the Coriolis acceleration induced by angular velocity. To do that inner frame of the sensor is connected to the outer frame that is substrate by springs, which are placed 90 degrees relative to the resonating motion.

- Infrared sensor – is a device that is used to measure distance. It works by using a specific light sensor to detect a selected light wavelength in the infrared spectrum. The operation of the IR sensor is illustrated in Figure 2.12.

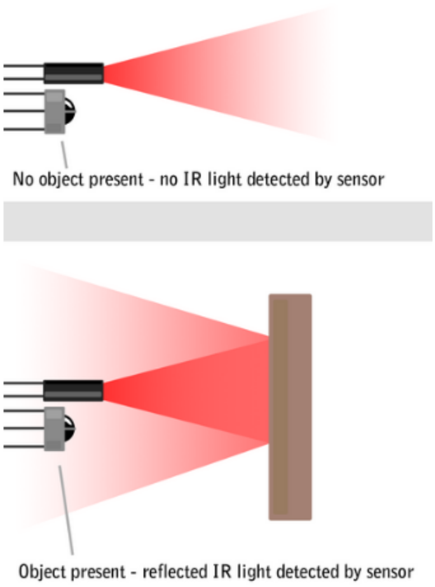

No object present - no IR light detected by sensor
Object present - reflected IR light detected by sensor

Figure 2.12 Illustration of the operation of IR sensors

(source: Rec, 2016)

IR sensors detect the light at a specific wavelength which is the infrared spectrum. These sensors include an IR light source and an IR sensor. The light emitted by the light source is reflected from the measured object and received by the IR sensor. The intensity of the reflected light determines the voltage generated by the sensor and the distance is measured using this information.

- Encoders – whether rotary or linear, absolute or incremental, there are two major encoder types which are the optical and magnetic encoder. An optical encoder uses light to identify the position for the encoder. On the other hand, a magnetic encoder works similar to determine a position of an optical encoder, but it uses magnetic fields instead of light. Incremental encoder mechanism is presented in Figure 2.13.

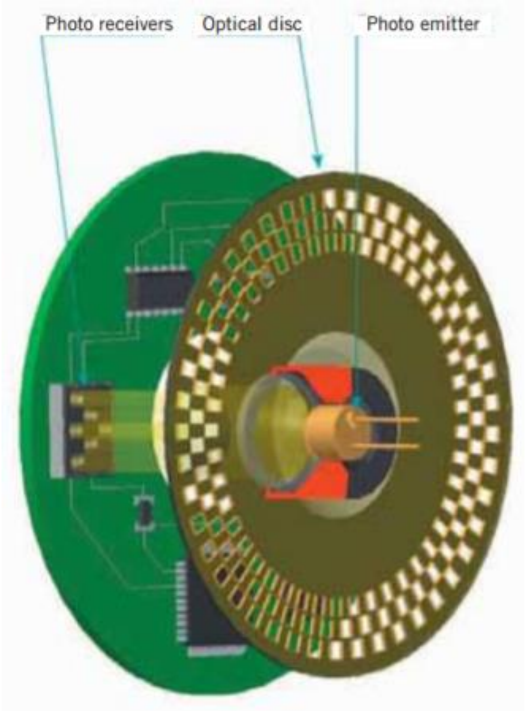


Figure 2.13 Incremental encoder mechanism (source: Evrimteknik, 2016)

The incremental encoder usually generates two different square waves, which are called channel A and B, which are out of phase by 90 degrees. These waves are illustrated in Figure 2.14. The first channel gives information about rotation speed while the second channel required the sense the direction of rotation. Another signal, which called Z or zero channel, provides the information about the absolute zero position of the encoder shaft.

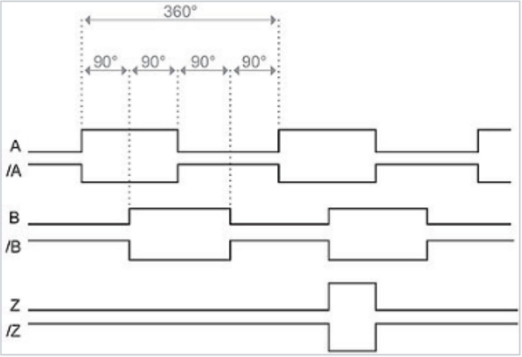


Figure 2.14 Incremental encoder signals (source: Evrimteknik, 2016)

2.5.5 Batteries

Mobile robots need an independent power source to work. Choosing the batteries is important for mobile systems since it directly effects to working life and the weight of the mobile platform.

Batteries are classified with respect to their rechargeability. Generally, rechargeable batteries are preferred for mobile systems. The most common rechargeable batteries are, NiCd, NiMH, lead acid, Li-ion, and Li-polymer.

Nickel Cadmium (NiCd) is well understood the chemical material but relatively low in energy density. NiCd is used fin applications that require longer service life and high discharge current, which can work in relatively lower temperatures (-40˚ C). NiCd is one of the most strong and durable batteries. It is the only chemistry that allows ultra-fast charging with minimal stress. Main applications are two-way radios, biomedical equipment, professional video cameras power tools, aviation, and UPS. Because of environmental concerns, NiCd is being replaced with other chemistries, but it retains its status in aircraft owing to its good safety record. The NiCd contains toxic metals and is environmentally unfriendly.

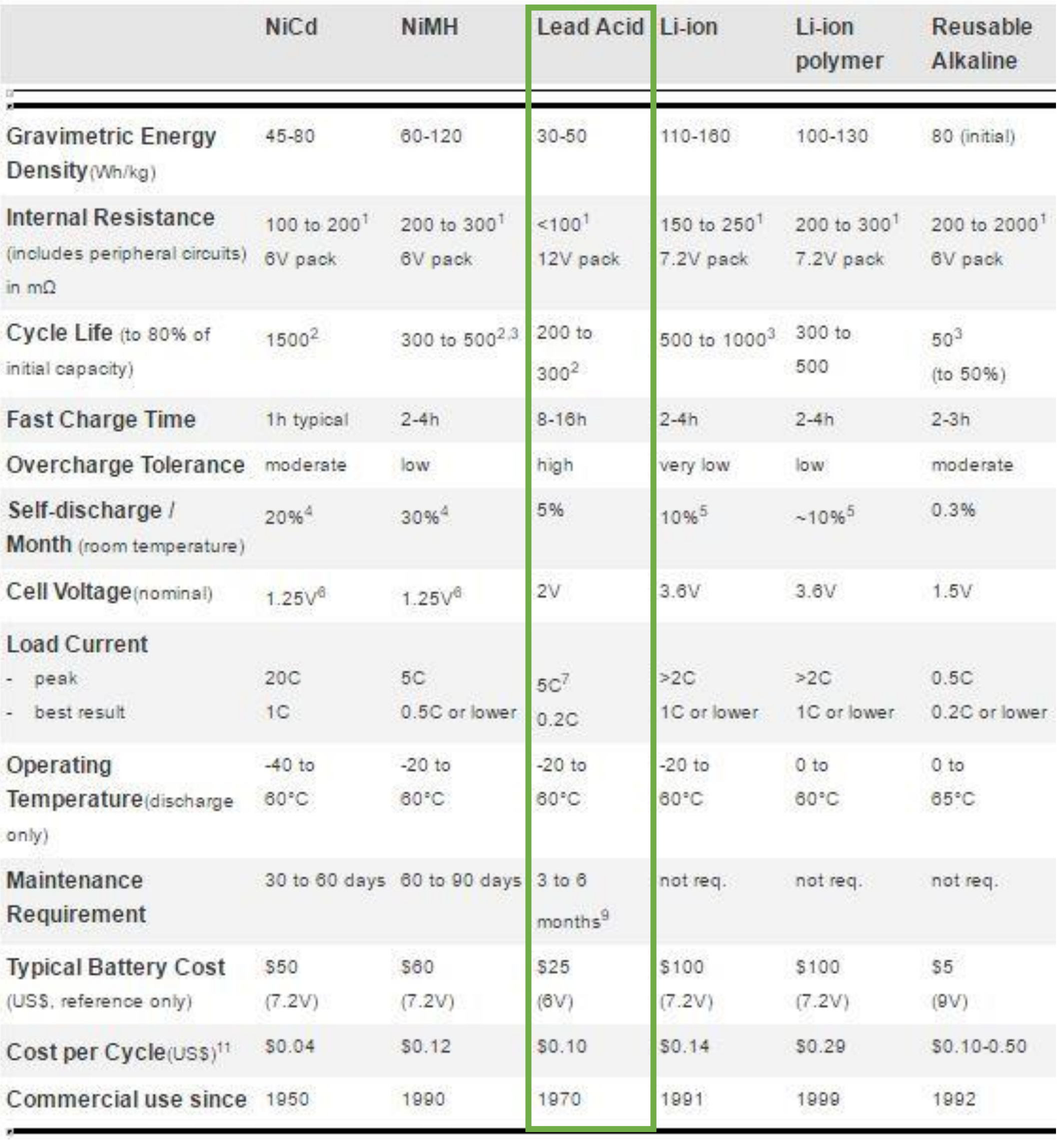
Nickel metal hydride (NiMH) serves as a replacement for NiCd as it has no toxic metals and provides higher energy density. NiMH is used in mobile phones, laptop computers, medical instruments, hybrid cars and industrial applications. NiMH is also available in AA and AAA cells for consumer use.

Lead Acid is the oldest and most economical rechargeable battery system where weight is of little concern. Lead acid is craggy, forgiving if abused and is economically priced, but it has a low specific energy and limited work life. Lead acid is popular around wheelchairs, golf cars, personal carriers, emergency lighting and uninterruptible power supply (UPS). Lead is toxic and cannot be disposed of in landfills.

Lithium-ion (Li-ion) is replacing many applications that were previously served by lead and nickel-based batteries. Li-ion is used where high-energy density and lightweight is of prime importance. Because of safety concerns, Li-ion needs a protection circuit. It is more expensive than most other batteries, but high cycle count and low maintenance reduce the cost per cycle over much other chemical substances. Li-ion batteries are popular around notebook computers and cellular phones.

Lithium Ion Polymer (Li-ion polymer) offers the properties of the Li-ion in ultra-slim geometry and simplified packaging. Main applications are mobile phones.

Table 2.3 Characteristics of commonly used rechargeable batteries (source:Batteryuniversity, 2016)



	NiCd	NiMH	Lead Acid	Li-ion	Li-ion polymer	Reusable Alkaline
Gravimetric Energy Density(Wh/kg)	45-80	60-120	30-50	110-160	100-130	80 (initial)
Internal Resistance (includes peripheral circuits) in mΩ	100 to 200[1] 6V pack	200 to 300[1] 6V pack	<100[1] 12V pack	150 to 250[1] 7.2V pack	200 to 300[1] 7.2V pack	200 to 2000[1] 6V pack
Cycle Life (to 80% of initial capacity)	1500[2]	300 to 500[2,3]	200 to 300[2]	500 to 1000[3]	300 to 500	50[3] (to 50%)
Fast Charge Time	1h typical	2-4h	8-16h	2-4h	2-4h	2-3h
Overcharge Tolerance	moderate	low	high	very low	low	moderate
Self-discharge / Month (room temperature)	20%[4]	30%[4]	5%	10%[5]	~10%[5]	0.3%
Cell Voltage(nominal)	1.25V[6]	1.25V[6]	2V	3.6V	3.6V	1.5V
Load Current - peak - best result	20C 1C	5C 0.5C or lower	5C[7] 0.2C	>2C 1C or lower	>2C 1C or lower	0.5C 0.2C or lower
Operating Temperature(discharge only)	-40 to 60°C	-20 to 60°C	-20 to 60°C	-20 to 60°C	0 to 60°C	0 to 65°C
Maintenance Requirement	30 to 60 days	60 to 90 days	3 to 6 months[9]	not req.	not req.	not req.
Typical Battery Cost (US$, reference only)	$50 (7.2V)	$60 (7.2V)	$25 (6V)	$100 (7.2V)	$100 (7.2V)	$5 (9V)
Cost per Cycle(US$)[11]	$0.04	$0.12	$0.10	$0.14	$0.29	$0.10-0.50
Commercial use since	1950	1990	1970	1991	1999	1992

The characteristics of the six most commonly used rechargeable battery systems are compared in Table 2.3 according to energy density, cycle life, exercise requirements and cost.

In this thesis, selection criteria are cost, weight, lifetime, the sensitivity of charging and cycle life. According to these criteria, lead-acid batteries are chosen to be used in this thesis study. Common advantages of lead acid batteries are given Table 2.4.

Table 2.4 Advantages of Lead acid batteries.

Inexpensive and simple to manufacture
Mature, reliable and well-understood technology
Low self-discharge
Low maintenance requirements

2.5.6 Data Acquisitions Systems

DAQ systems are the main instruments used in the laboratory research of scientists and engineers; especially, for test and measurement, automation, and so forth. The main purpose of the DAQ system is the sampling the signals that are measures of real world physical conditions (voltage, current) and converting the resulting samples into digital values that can be handled by a computer.

DAQ system generally contains digital I/O (input/output), ADC (analog to digital converter), DAC (digital to analog converter), counter inputs and necessary communication buses (Ethernet, Serial, Can bus, Profibus) to communicate with a computer or industrial controllers.

Three different types of DAQ systems that are widely used are PCI-based systems for Desktop PC's, rugged PC's, or standalone system technologies. PCI (Peripheral Component Interconnect), PCI Express, USB, wireless and Ethernet data acquisition cards can be given as examples of DAQ systems that are used alongside with desktop PCs. In the work of (Sidenbladh et al., 1999) a PCI-based control of the mobile robot is done where visual algorithms are also integrated into the system. PC 104, which is an embedded computer standard, is one of the rugged PC DAQ systems. In these systems, DAQ modules are integrated with processing and graphical unit modules generally for industrial usages which are more compact and durable. In the work of (Wang, 2009) a mobile robot is controlled by a PC 104 unit and wireless communication option. Standalone DAQ systems, such as FPGA (A field-programmable gate array), generally have an embedded processing unit on a single circuitry board that is designed to perform specific tasks. (Kale and Shriamwar, 2009) introduced a standalone controller system for controlling a mobile robot. This system has considerable costs lower than other two systems, more compact, and task-oriented.

2.6 Conclusion

In this section, a literature survey is presented for mobile robot control along with mobile robot’s components.

Firstly, a survey on control of omnidirectional mobile robots is introduced in which PID, dynamic model based, fuzzy and stochastic control methods are reviewed. PID controller is chosen because of its reliable structure for robotics. In reviewed studies, it has been seen that modeling the controller with regarding slippage and friction on wheels is important in the case of torque control of actuators, so called motors. However, in this study speed control is implemented in the system in which drivers’ embedded closed-loop controllers ensures speed tracking of motors which only leaves the problem of mismatching of the actuator (motors and wheels) characteristics caused by usage of different wheels and disturbance effects (such as nonuniformities on surface of the robot is moving on). To overcome this, the top-level controller is utilized by using additional hardware, in this case using gyroscope in the loop. Hence to improve the performance of the controller, a top-level PI controller is designed.

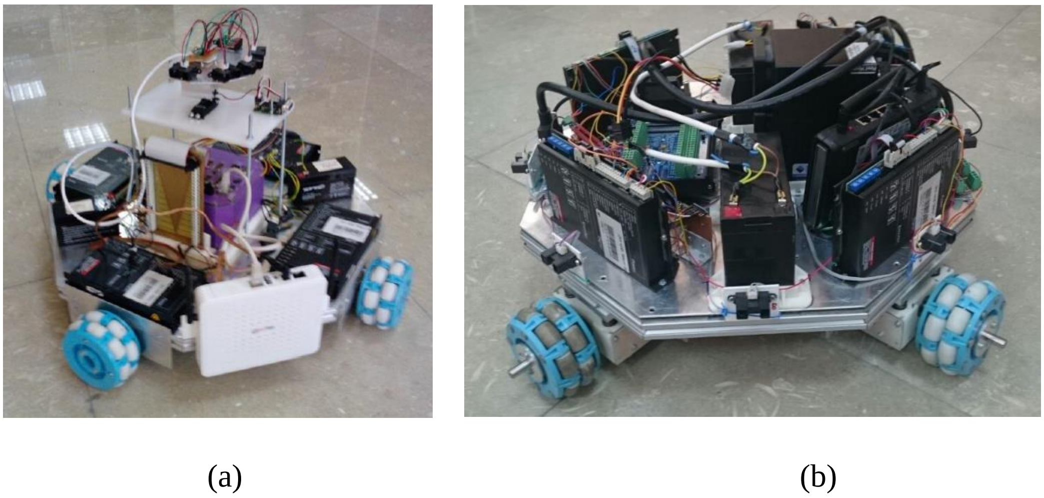
Secondly, the studies in the subject of obstacle avoidance in wheeled robots are explained to avoid possible collisions in task environment. According to our hardware, the most appropriate solution is decided to be a virtual spring-damper system. In order to improve our previous algorithm, the virtual damper is modeled and added.

Many different methods of employing fault tolerance and detecting faults in different systems are given. In this thesis study, the fault tolerance is enforced on actuator level. As reviewed in the literature, velocity level distribution of the task space motion is chosen since the actuators are driven by speed controllers and pseudo-inverse of the Jacobian approach is highly anticipated in redundant control of robots. To decrease the effect of the faulty actuator on the mobile robot, the weighted pseudo-inverse of the Jacobian matrix is developed and applied for actuator based fault tolerance. Lastly, essential components of omnidirectional robotic system components are investigated.

CHAPTER 3

DESCRIPTION OF THE HARDWARE

Mobile robots find use in many applications such as land explorations, mining, cleaning, and other indoor tasks that take place in hospitals or factories. An indoor mobile robot is investigated in this thesis study. A four-wheeled omnidirectional mobile robot is used for investigation which was designed and constructed in a previous thesis study in IzTech Robotic Laboratory (Şahin, 2012). In this study, this mobile robot is improved by integrating a new hardware and suspension system to enhance accuracy, efficiency and ease of control. The previous and new version of the mobile robot are shown in Figure 3.1.



(a) (b)

Figure 3.1 Redundant Mobile Robot (a)- Previous design of the mobile robot, (b)- New design of the mobile robot

In this thesis, relative to the original hardware, the structure and main parts are maintained the same and the new components added to the system are a new DAQ system, infrared sensors at each side of the octagonal mobile platform frame and magnetic encoders at the rear ends of the wheel actuators.

3.1 General Information and Specification of the Mobile Robot

The number of actuated wheels of the mobile robot makes it kinematically redundant. Four omnidirectional wheels are actuated with geared DC motors. The hardware used in the mobile robot and the mobile robot’s specifications are listed in Table 3.1.

Table 3.1 Hardware and Specifications of the Mobile Robot.

Motor Type	Geared DC motor
Motor Controller	Maxon ADS 50/10
Motor Controller Mode	Velocity
Power Supply	Lead Acid
Controller Device (DAQ system)	UEISIM 600-1G
Wheel Type	Omnidirectional wheels
DoF of Motion	3
DoF of Mobile Robot	4
Encoder Type	Magnetic Encoder
Sensors	Infrared range sensor Analog Gyroscope
Workspace	Indoor - ground
Weight	~15kg

The hardware used on mobile robot are shown in Figure 3.2 which shows the top view photo of mobile robot. The yellow box indicates the Ueisim daq, blue box indicates the wireless router, red boxes indicate motor drivers, orange box indicates batteries, orange circles indicate infrared sensors, green box indicates power inverter and dashed yellow box indicates I/O cards.

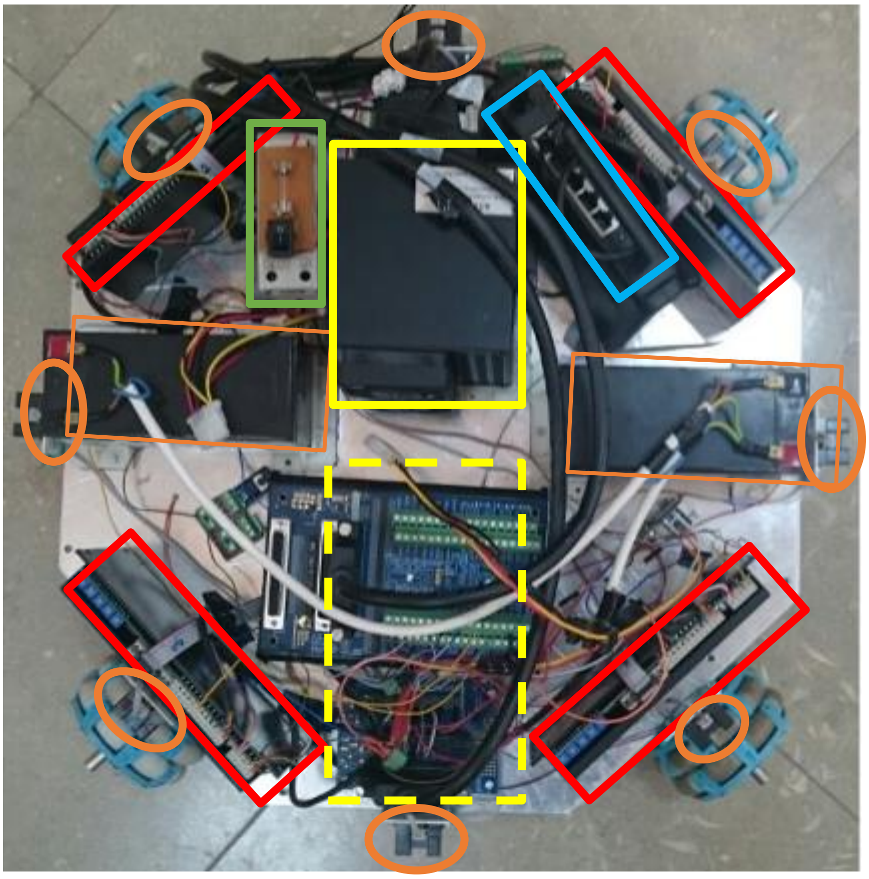


Figure 3.2 Top view of the Redundant Mobile Robot

The actuation system including the motor, the encoder and the omnidirectional wheel along with the improved shaft bedding with the new suspension are shown in Figure 3.3. In this figure, the green box indicates the DC motor, the gearbox and the encoder, the yellow box indicates the motor bedding and the new suspension system, and the blue box indicates omnidirectional wheel.

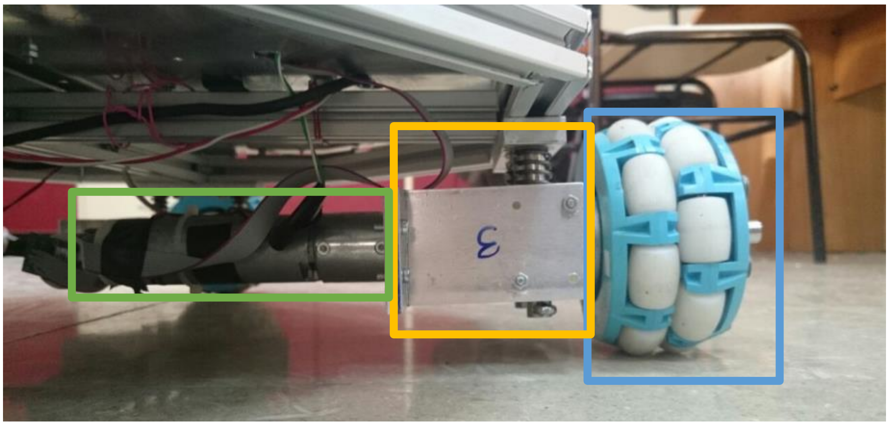


Figure 3.3 Actuation System of the Redundant Mobile Robot

3.2 Hardware Used in the Mobile Robot

The redundant mobile robot is composed of mechanical and electronic equipment. These are the controller DAQ system, the actuation systems composed of DC motor and gearbox, the motor drivers, the new suspension system, the omnidirectional wheels, the power unit, the battery, the wireless router, the infrared sensors, the magnetic encoders, and the gyroscope. Each of these components are explained in the next subsections.

3.2.1 Data Acquisition System (DAQ)

In order to run the control algorithm of the mobile robot, acquire and process data a DAQ system is required. UEISIM 600-1G DAQ is chosen as the DAQ system because of its compact and rugged design, adaptability to many different I/O (Input/Output) types, and Matlab (Simulink package) support. UEISIM 600-1G is the DAQ system in which DAQ cards with different DAQ features such as digital I/O, analog I/O, counter/timer, quadrature encoder, Serial, CAN and ARINC 429 I/O. This DAQ system supports up to six I/O slots. UEISIM DAQ has a built-in Linux type of operating system. On the mobile robot, DNA-AI-217 and DNA-AO-308 cards are used for I/O. Figure 3.4 shows the UEISIM 600-1G and its interfaces.

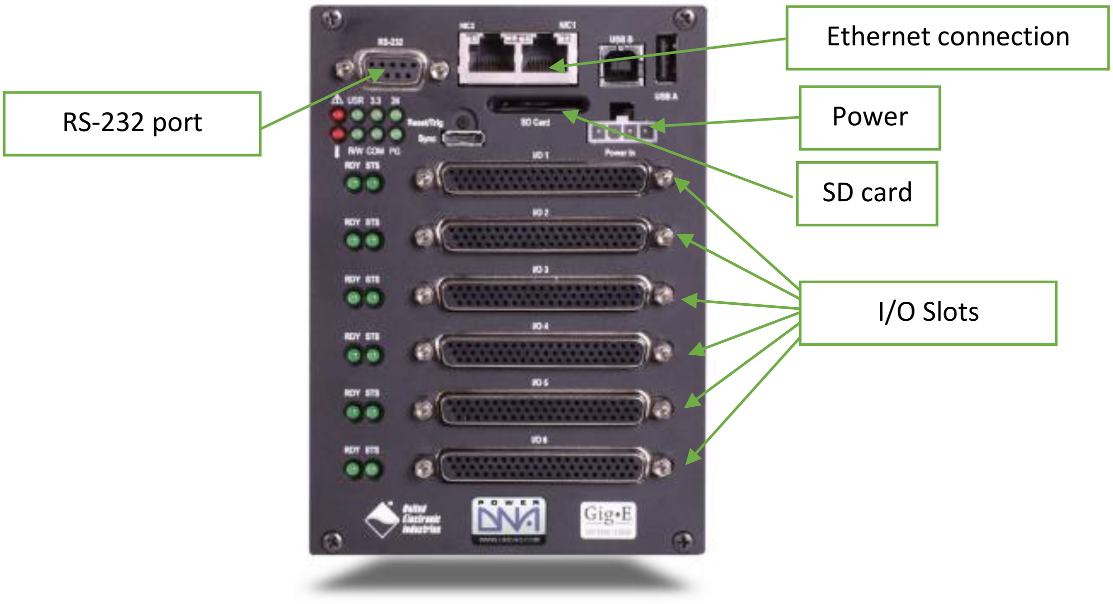
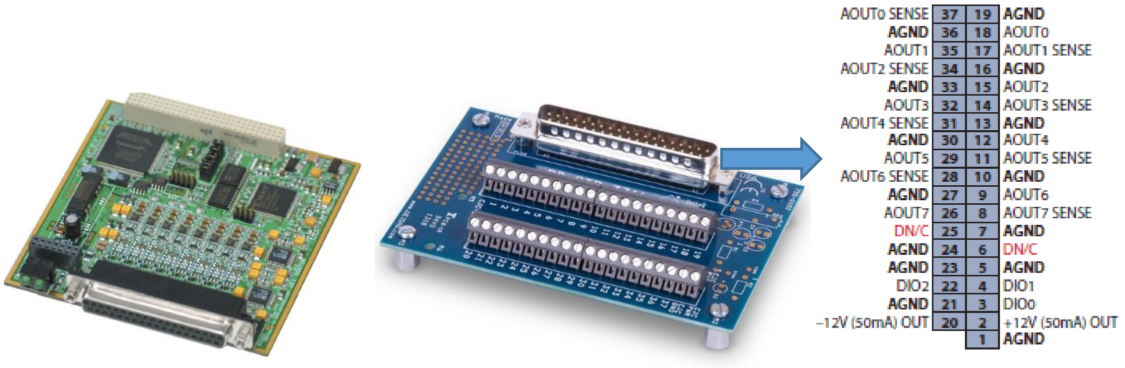


Figure 3.4 UEISIM 600-1G

DNA-AO-308 is an analog output card, which provides 8 independent analog outputs with 16-bit resolution. The range of output is ±10V and ±50mA per channel. The maximum update rate is 100 kHz for each channel. The output module and the arrangements of the pins are given in Figure 3.5.

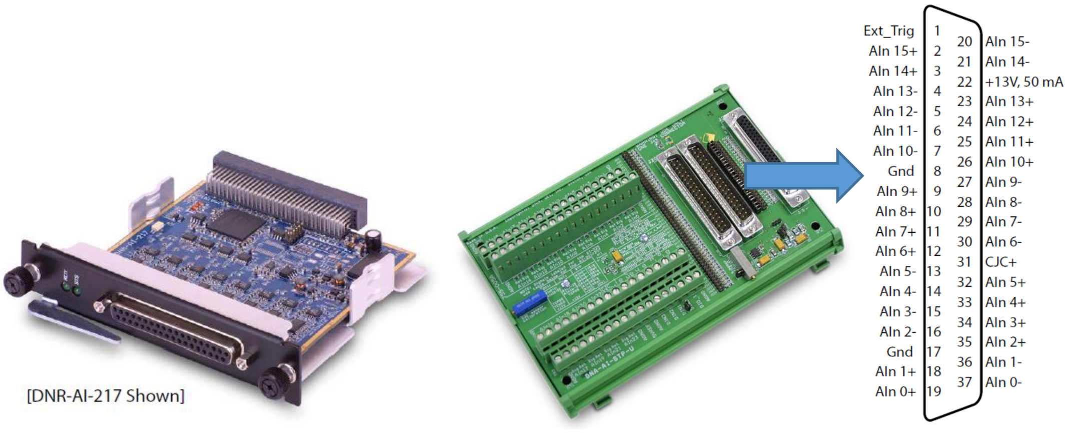


AOUT0 SENSE	37	19	AGND
AGND	36	18	AOUT0
AOUT1	35	17	AOUT1 SENSE
AOUT2 SENSE	34	16	AGND
AGND	33	15	AOUT2
AOUT3	32	14	AOUT3 SENSE
AOUT4 SENSE	31	13	AGND
AGND	30	12	AOUT4
AOUT5	29	11	AOUT5 SENSE
AOUT6 SENSE	28	10	AGND
AGND	27	9	AOUT6
AOUT7	26	8	AOUT7 SENSE
DN/C	25	7	AGND
AGND	24	6	DN/C
AGND	23	5	AGND
DIO2	22	4	DIO1
AGND	21	3	DIO0
−12V (50mA) OUT	20	2	+12V (50mA) OUT
		1	AGND

(a) (b) (c)

Figure 3.5 Analog Output module of UEISIM, (a)-Analog output card, (b)- Analog output board, (c)- Analog output pins

DNA-AI-217 is analog input card, which provides 16 input channels with 24-bit resolution. The sample rate of this card is 120kHz for each channel. Input range is ±10V. The analog input module and its pins are given in Figure 3.6.



Ext_Trig	1	20	AIn 15-
AIn 15+	2	21	AIn 14-
AIn 14+	3	22	+13V, 50 mA
AIn 13-	4	23	AIn 13+
AIn 12-	5	24	AIn 12+
AIn 11-	6	25	AIn 11+
AIn 10-	7	26	AIn 10+
Gnd	8	27	AIn 9-
AIn 9+	9	28	AIn 8-
AIn 8+	10	29	AIn 7-
AIn 7+	11	30	AIn 6-
AIn 6+	12	31	CJC+
AIn 5-	13	32	AIn 5+
AIn 4-	14	33	AIn 4+
AIn 3-	15	34	AIn 3+
AIn 2-	16	35	AIn 2+
Gnd	17	36	AIn 1-
AIn 1+	18	37	AIn 0-
AIn 0+	19		

(a) (b) (c)

Figure 3.6 Analog Input module of UEISIM, (a)- Analog input card, (b)- Analog input board, (c)- Analog input pins

3.2.2 Actuation System

The actuation system of the mobile robot consists of a DC motor, a gearbox, a motor driver, an omnidirectional wheel, a magnetic encoder and the new suspension module.

DC Motor and Gearbox

The main part of the mobile robot's actuation system is the electric motor which is used to generate the rotational motion of the wheel. Dunkermotoren 24V G30.0 model brushed DC motor with planetary gearboxes (PLG32 model) is used as the DC motor. The reduction ratio of the gearbox is 20.25:1. The specification of the motor and the gearbox are given in Appendix B.

Magnetic Encoder

The encoder is used for measuring DC motor's speed. In this study, a magnetic encoder is chosen because of its assembly advantages. The chosen magnetic encoder is the AS5040 rotary sensor which has10-bit resolution. Further specifications of the magnetic encoder are given in Appendix C. The magnetic encoder is mounted at the rear end of the motors. Necessary parts for assembly of the sensors are produced with a 3D printer. The mounting details are given in Figure 3.7.

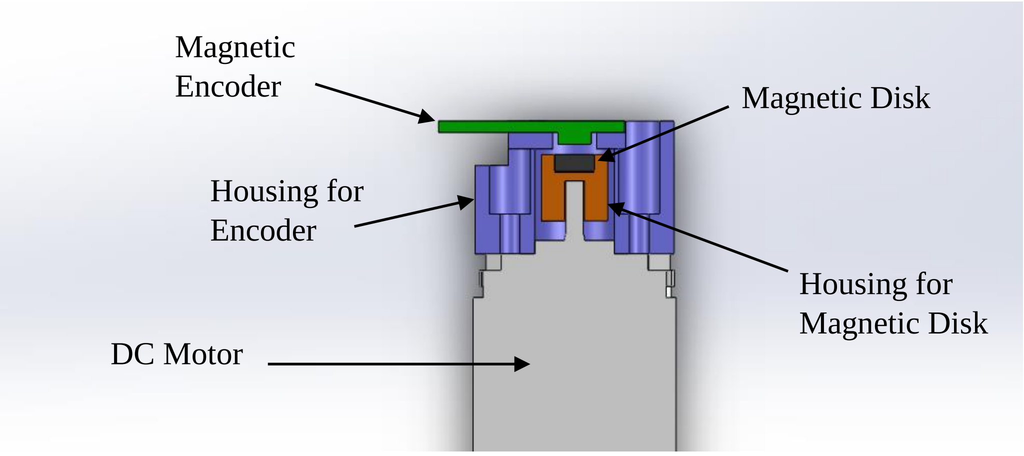


Figure 3.7 Assembly details of the magnetic encoder

Motor Driver

The motor driver is a necessary for actuating the wheels and thus, controlling the mobile robot. Actuators used in this study are driven by Maxon ADS 50/10 DC motor drivers. The specifications of the motor driver are given in Appendix D. There are five sections on the motor driver which are the power, the signal, the encoder, the modes and the settings sections.

In the power section, pin 1 and pin 2 are used to supply desired amount of voltage to DC motor and pin 4 and pin 5 are connection ports to the main power supply.

In the signal section, pin 1 and pin 2 are used to acquire desired motor driving signals (i.e. for speed operation mode, it is the motor speed) and pin 3 is used to enable the motor operation. Furthermore, pin 7 and pin 8 are used for monitoring the motor speed and the motor current.

The encoder section consists of quadrature encoder connection ports. Motor driver calculates motor speed using the signals acquired from the encoder.

In the mode section, six jumper switches are used to change the controller mode. The combinations of these pins change the mode. This motor driver has four control modes which are, (i) Speed control using tachometer, (ii) Speed control using encoder, (iii) IxR compensated speed control (iv) Current control.

In setting section, five potentiometers are used to tune controller parameters. These sections are illustrated in Figure 3.9. Inside of the motor driver, there is a built-in PID controller, whose parameters are tuned by three hidden potentiometers under the casing which is shown in Figure 3.10.

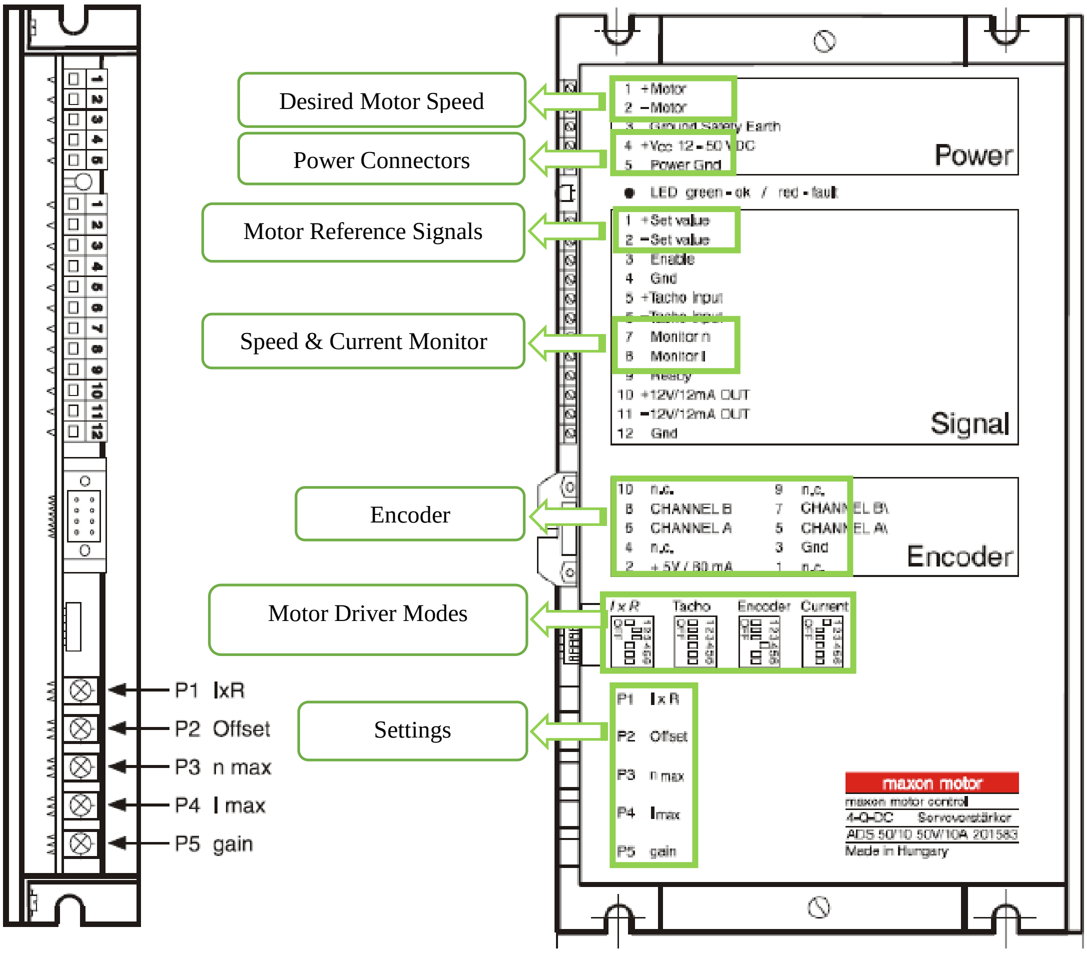


Figure 3.9 Connectors of the motor driver

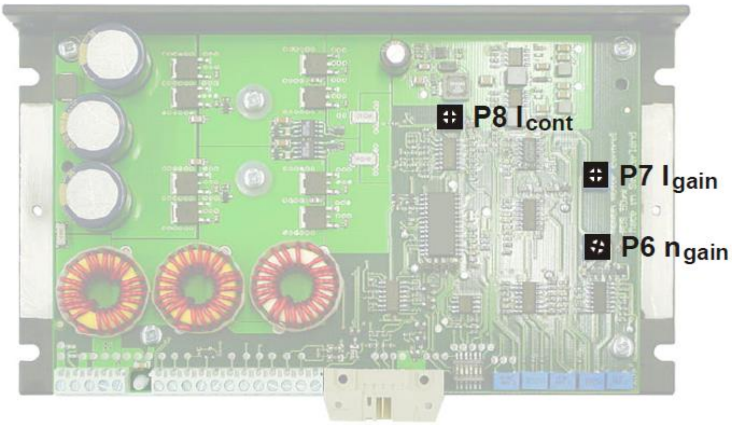


Figure 3.10 Hidden potentiometers of the Maxon motor amplifier

In this study, generally, speed control with encoder mode is used for running and testing the control algorithms. However, in order to identify the DC motor characteristics (transfer function), current mode is used.

In the speed mode, firstly, magnetic encoder’s data is processed in the motor driver to be used in the in-drive controller for speed mode operation. Maxon motor driver’s speed monitor’s output gives a voltage which is proportional the speed. Output voltage range is -10V to +10V. 10V corresponds to the maximum motor’s speed.

To calibrate of encoders, each motor is run independently and motors’ speed is measured with a laser tachometer. After, real motors’ speeds are paired with output voltages. Motors’ speeds are calibrated using this methodology.

Suspension System

The suspension system is designed to reduce the vibration on mobile robot’s platform which caused by omnidirectional wheels and to improve road-holding. Additionally, with the suspension system, the mobile robot can work on relatively rougher surfaces.

Another advantage of the suspension system is reducing the vibration on mobile robot’s platform. This is necessary since there are many electronic devices that should not be subjected to excessive vibrations and mechanical connections on the mobile robot platform should not loosen up during operation.

The suspension system consists of a spring, linear and rotational bearings, a DC motor housing, a coupling and a sliding body. In Figure 3.11, exploded view of the suspension system is illustrated and the assembly of the suspension systems on the mobile platform frame is given in Figure 3.12.

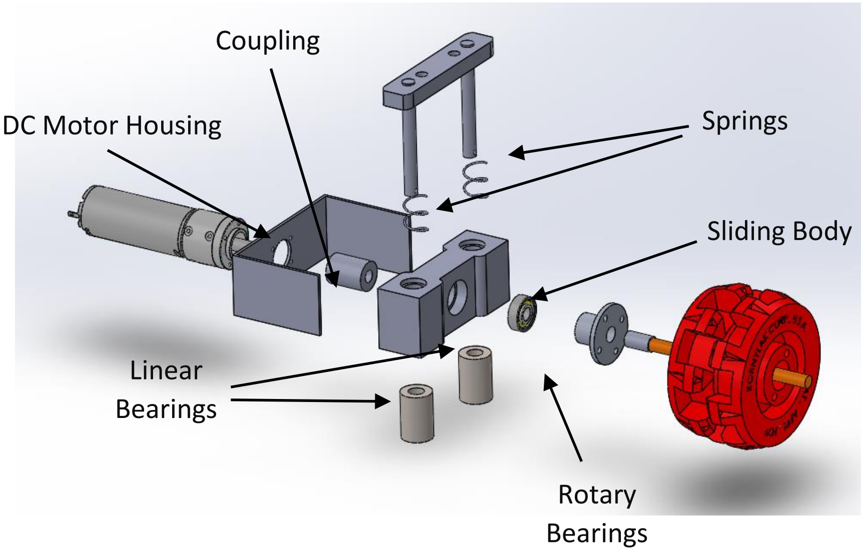


Figure 3.11 Suspension System exploded view of Mobile Robot

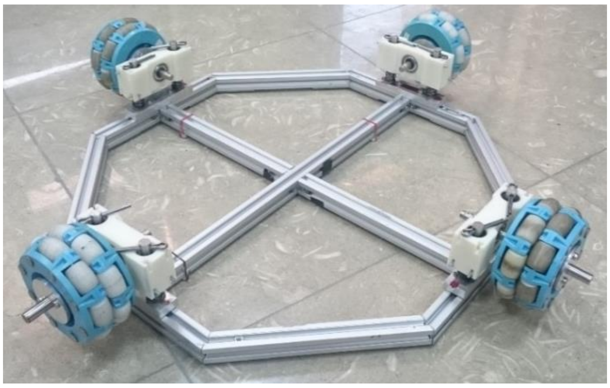


Figure 3.12 Assembly of Suspension System to Mobile Robot body

Omnidirectional Wheels

Omnidirectional wheels are chosen because they allow holonomic motion. The omnidirectional wheels produced by Kornylak Corp, which is shown in Figure 3.13, are used in the mobile robot. The wheels have a special design with eight passive rollers which are assembled perpendicular to the axis of rotation of the wheel around the outer diameter of the wheel. However, the omnidirectional wheels have a disadvantage since it causes slip depending on the material used for the passive rollers.

Specifications of omnidirectional wheels are given below:

- Lightweight wheels
- Self-lubricating
- High impact plastic
- Low friction coating
- Stainless steel axles
- Washable even steam cleaning
- Corrosion resistant

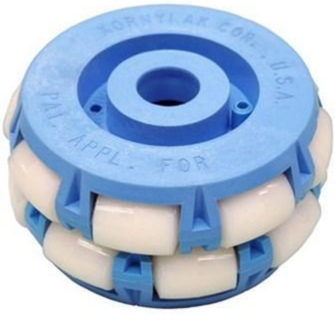


Figure 3.13 Omnidirectional wheel by Kornylak Corp.

There are three types of rollers used by Kornylak Corp, which are listed below;

- Nylon
- Synthetic rubber coated polypropylene
- Solid urethane roller with molded axle

The material used on wheels’ roller effects its friction properties and slipping characteristics which directly influences the overall mobile robot performance. There are two types of wheels used on the mobile robot as two of each type. Among these, one type has synthetic rubber coated polypropylene and the other one has nylon rollers. Wheels with nylon rollers are attached to the first and the third actuator while the second and the fourth have the other type of rollers. Wheels with nylon rollers have low friction coefficient with respect to the ones that have rollers produced with rubber coated polypropylene.

3.2.3 Power Management System

The power management system consists of a power unit that distributes energy to the hardware and the batteries as the energy source.

Power Unit

The power unit consists of voltage regulator cards to supply various voltage levels to a variety of electronic components of the mobile robot. There are two kinds of voltage regulator cards used in the power unit, which are step-up and step-down. The step-up regulation cards are used to supply 24 VDC to DAQ system and the actuators from the 12 VDC batteries. The step-down regulator cards are used to supply 5 VDC to sensors and 9 VDC to the wireless router from the same battery source.

Battery

The most suitable battery for the mobile robot application according to the selection criteria given in Chapter 2 is lead acid type battery. The capacity of the battery is 7Ah and the supplied voltage is 12 V. With this features, this battery can support 1 hour of operation considering the average energy usage of the mobile robot. The selected battery is given in Figure 3.14.

Figure 3.14 Ttec 12V/7Ah Lead acid battery

3.2.4 Wireless Communication System

The mobile robot is operated with wireless signals. A wireless router on the mobile robot operates in access point mode where master PC communicates directly with the UEISIM DAQ through the wireless router. The communication is illustrated in Figure 3.15. As seen in the figure, instead of the Ethernet connection between PC and DAQ, a router converts this connection into wireless. In this thesis study, the wireless communication is used to embed the control algorithms into the robot's controller and to monitor and log the data from DAQ.

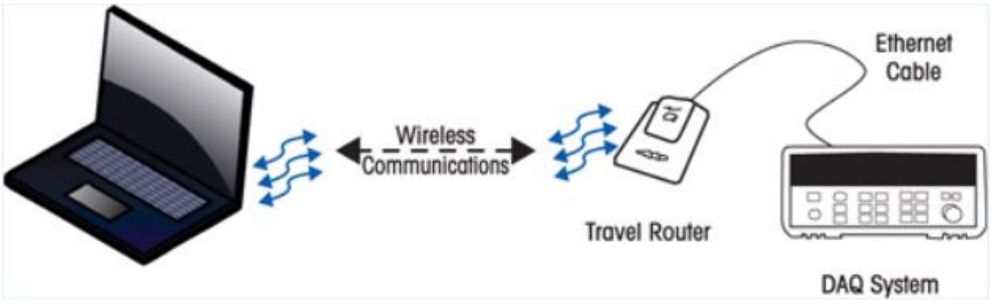


Figure 3.15 Wireless communication of the DAQ system with the host PC (source: Evaluationengineering, 2016)

3.2.5 Infrared Sensor

In this study, SHARP GP2Y0A02YK infrared range sensors are used to detect the obstacles on the robot’s path. This sensor’s working range is from 20 cm to 150 cm. In order to calculate the distance of an obstacle, the analog output voltage from the sensor should be converted distance with a transforming function. In our case, the critical distance is selected to be in between 25cm to 60cm since the sensors output signal shows more linearity in this range.

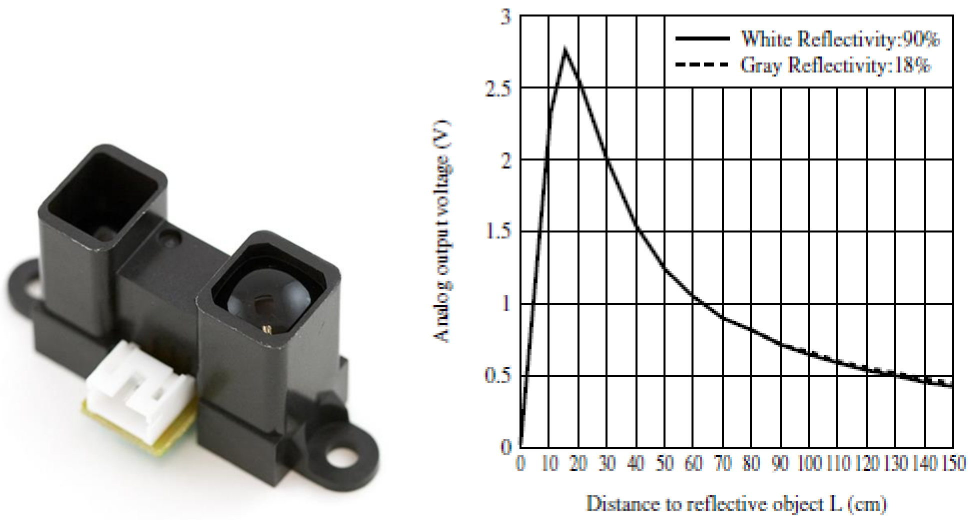


Figure 3.16 Relation of analog output voltage and distance of infrared sensor

To find transforming function to be used to calculate the distance in the range between 25cm and 60cm, the plot in Figure 3.16 is used and experimental data are collected for verification of this function. Equation 3.1 represents only this interval's transforming function.

$$L = -1{,}3599V^3 + 12{,}463V^2 - 52{,}769V + 95{,}05 \tag{3.1}$$

3.3 Software Used for Control of Mobile Robot

As the remote (slave) controller on the mobile robot, the UEISIM system is chosen and a notebook PC is used as a master controller and deploying embedded codes into the remote controller. Matlab Simulink software is used on the notebook PC to design and test the control algorithms by controlling the mobile robot via exchanging information through the DAQ system. Matlab Simulink software is chosen for its simulation capability, real-time operation capability and suitability for the UEISIM DAQ system. Matlab Simulink software and Simmechanics tools are used to simulate mobile robot and test the control algorithms before implementing them on the real system. The system and motor models are identified by using the System Identification Toolbox in Matlab software.

Embedded codes for controlling mobile robot are generated by using UEISIM support for Simulink which are later embedded and run on the remote UEISIM DAQ system via Ethernet SSH (secure shell) protocol. Putty, is serial communication terminal connection software which is used for accessing remote hardware using Serial or Ethernet connection. To connect to the mobile robot's OS (operating system) remotely and embed the devised controller algorithms, putty software is utilized. UEISIM support for Simulink allows to either embedding the codes inside the controller for standalone operation or for remotely controlling the device (teleoperation mode). In teleoperation mode, the remote controller works as a standalone controller and Simulink Desktop Real Time is used in the notebook PC to send and receive master commands via UDP (User Datagram Protocol).

3.4. Conclusion

In this chapter, general information and specifications of the omnidirectional mobile robot are presented. Then, the hardware and the software used in the system are introduced in the order of the DAQ system, the actuators, the power management system, the wireless communication system, the sensory system and the software. In the next Chapter, the modelling work carried out for the actuation system introduced in this Chapter is presented along with the total system modelling including the kinematic and dynamic models.

CHAPTER 4

THE SYSTEM MODELING

One of the main purposes of this thesis study is to develop a suitable controller for the redundant mobile robot constructed in this thesis work. In the previous study conducted by (Şahin, 2012), the mobile robot was controlled and run without using the capabilities brought up as a result of having redundancy. In this Chapter, initially the overview of the controller architecture is provided as low-level controller to control the rotational speed of the wheels and the top-level controller which is making use of the gyroscope signals as feedback signals. Later, the kinematic model of the redundant mobile robot is described. Experimental work carried out to estimate the motor model, actuation system model and the system's dynamic model are provided to complete the modeling work. The necessary algorithms to be used in this thesis work are introduced for obstacle avoidance and fault tolerance.

4.1 Overview of the Control Architecture of the Redundant Mobile Robot

In order to control the mobile robot, a top-level controller is developed and employed. A low-level control loop runs in the motor driver as illustrated in Figure 4.1. The motor driver has its own built-in PID controller, which receives encoder data so that each wheel's velocities are regulated.

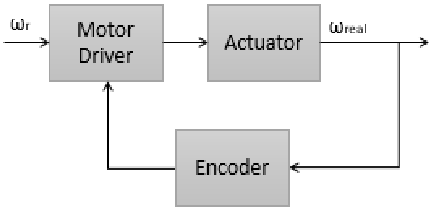


Figure 4.1 Low-level control scheme

The top-level control is used for minimizing the disturbance effects caused by environment effects and manufacturing errors. In order to achieve this, a gyroscope sensor is used to supply feedback signals to be used in the controller. The top-level control is illustrated in Figure 4.2.

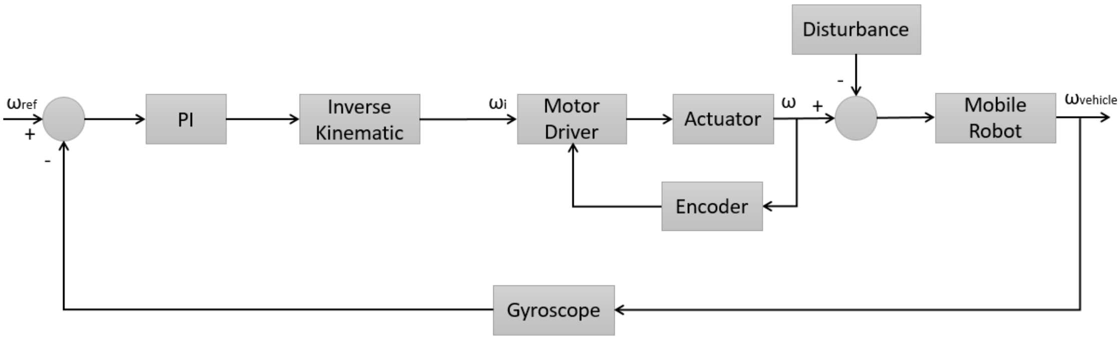


Figure 4.2 Top-level control of omnidirectional mobile robot

4.2 Kinematics of Mobile Robot

Kinematics is the study of the mathematics of motion without considering the forces that affect the motion. Kinematics as a field of study is often referred to as the "geometry of motion" and as such may be seen as branch of mathematics. Kinematic model of mobile robots is investigated in two parts which are forward and inverse kinematic models of mobile robots.

To carry out kinematics work, the reference coordinate system for motion is given in Figure 4.3. Where V_i and ω_i (i =1,2,3,4) refer to wheel's linear and angular velocities.

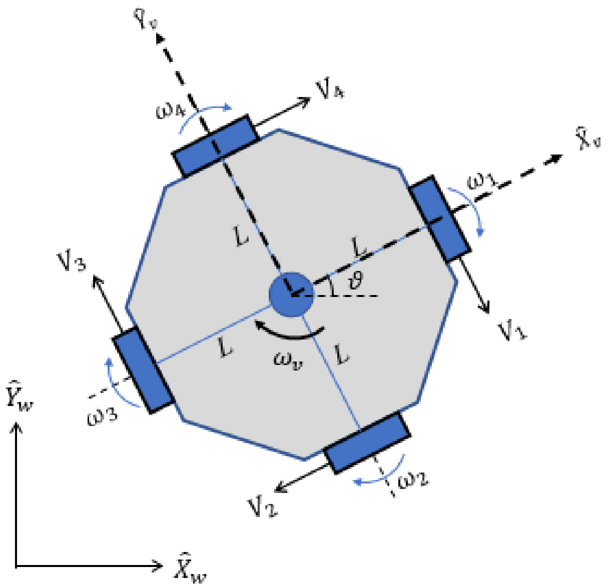


Figure 4.3 Top view of four wheeled mobile robot

In this figure, L is the distance between the mass center and the mid-point of the wheels in the lateral direction. ϑ is the angle between robot's frame, $\hat{X}_v$ and $\hat{Y}_v$, and world coordinate frame which is determined by the unit vectors $\hat{X}_w$ and $\hat{Y}_w$.

4.2.1 Forward Kinematics Model

In the forward kinematic model, task space's velocities are defined as a function of joint velocities. Equations relating the wheels' linear velocities to the mobile robot's task space velocities are given in the following equations:

$$V_x = V_1.\sin\vartheta - V_2.\cos\vartheta - V_3.\sin\vartheta + V_4.\cos\vartheta \tag{4.1}$$

$$V_y = -V_1.\cos\vartheta - V_2.\sin\vartheta + V_3.\cos\vartheta + V_4.\sin\vartheta \tag{4.2}$$

$$\omega_v = (V_1 + V_2 + V_3 + V_4)/L \tag{4.3}$$

In Equations 4.1, 4.2, 4.3, the velocity of the robot on local x and y directions are signified with V_x and V_y, respectively. The angular velocity of center of mass is signified with ω_v. Jacobian matrix, $\hat{J}$, developed as a result of velocity-level forward kinematics analysis is given below:

$$\begin{bmatrix} V_x \\ V_y \\ \omega L \end{bmatrix} = \hat{J}. \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \tag{4.4}$$

$$\hat{J} = \begin{bmatrix} -\sin\vartheta & -\cos\vartheta & \sin\vartheta & \cos\vartheta \\ \cos\vartheta & -\sin\vartheta & -\cos\vartheta & \sin\vartheta \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.5}$$

4.2.2 Inverse Kinematics Model

Because of the redundancy, Jacobian matrix, given in Equation 4.5, is not a square matrix. Hence, pseudo inverse method used to find minimum norm of wheel velocities. Pseudo inverse of the Jacobian matrix $,J^{+}$, for robot is derived as below

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \hat{J}^{+} \begin{bmatrix} V_X \\ V_Y \\ -\omega_V . L \end{bmatrix} \tag{4.6}$$

$$\hat{J}^{+} = \hat{J}^{T}\left(\hat{J}.\hat{J}^{T}\right)^{-1} = \begin{bmatrix} 0{,}5\sin\vartheta & -0{,}5\cos\vartheta & 0{,}25 \\ -0{,}5\cos\vartheta & -0{,}5\sin\vartheta & 0{,}25 \\ -\,0{,}5\sin\vartheta & 0{,}5\cos\vartheta & 0{,}25 \\ 0{,}5\cos\vartheta & 0{,}5\sin\vartheta & 0{,}25 \end{bmatrix} \tag{4.7}$$

Then linear velocities of wheels are:

$$V_1 = 0{,}5.V_x.\sin\vartheta - 0{,}5.V_y.\cos\vartheta - 0{,}25.\omega_v.L \tag{4.8}$$

$$V_2 = -0{,}5.V_x.\cos\vartheta - 0{,}5.V_y.\sin\vartheta - 0{,}25.\omega_v.L \tag{4.9}$$

$$V_3 = -0{,}5.V_x.\sin\vartheta + 0{,}5.V_y.\cos\vartheta - 0{,}25.\omega_v.L \tag{4.10}$$

$$V_4 = 0{,}5.V_x.\cos\vartheta + 0{,}5.V_y.\sin\vartheta - 0{,}25.\omega_v.L \tag{4.11}$$

These actuator velocities calculations are for ideal conditions (without slipping and homogeneous mass distribution).

4.3 Experimental Estimation of Dynamic Equations of the Mobile Robot

Dynamic modeling of the mobile robot is necessary to design its controller. Using Newtonian mechanics, dynamic equations of the mobile robot relating the forces/torque (F_x, F_y, Γ) supplied by the actuations system to the motion of the mobile robot is given below;

$$F_x(t) = M.\frac{dV_x(t)}{dt} + B_{vx}V_x(t) + C_{vx}sgn\big(V_x(t)\big) \tag{4.12}$$

$$F_y(t) = M.\frac{dV_y(t)}{dt} + B_{vy}V_y(t) + C_{vy}sgn\left(V_y(t)\right) \tag{4.13}$$

$$\Gamma(\mathrm{t}) = J.\frac{d\omega_v(t)}{dt} + B_W\omega_v(t) + C_W sgn\big(\omega_v(t)\big) \tag{4.14}$$

In the above equations, robot's mass and moment of inertia are denoted by M and J. Viscous friction and Coulomb friction is denoted with B and C, respectively. The coefficients of these friction terms that are related to V_x, V_y and ω_v velocities are estimated by carrying out the necessary experimentation.

The relationships between the total traction forces/torque acting on the robot (F_x, F_y, Γ) and the wheel's traction forces denoted with f_i for the i$^{\text{th}}$ wheel are given in Equations 4.15, 4.16 and 4.17.

$$F_x(t) = f_4(t) - f_2(t) \tag{4.15}$$

$$F_y(t) = f_1(t) - f_3(t) \tag{4.16}$$

$$\Gamma(\mathrm{t}) = \big(f_1(t) + f_2(t) + f_3(t) + f_4(t)\big).L \tag{4.17}$$

For the estimation of dynamic parameters, the total forces and torque subjected to the robot as a result of wheel traction forces are calculated by using the set of Equations from 4.15 to 4.19 and measured motor currents. The relation between a wheel's traction force (f) and the torque generated by the DC motor (T) and also, the relation between the current supplied to the DC motor (i_a) and the torque generated by the DC motor (T) are given in Equations 4.18 and 4.19, respectively. In these equations, the radius of the wheels and the torque constant of the DC motors are denoted with r and K_T, respectively.

$$f(t) = \frac{T(t)}{r} \tag{4.18}$$

$$T(t) = K_T . i_a(t) \tag{4.19}$$

The identification process is carried out while the robot is moving at a constant velocity. Because of constant velocity, the dynamic equations are simplified as follows:

$$F_x(t) = B_{vx}V_x(t) + C_{vx}sgn\big(V_x(t)\big) \tag{4.20}$$

$$F_y(t) = B_{vy}V_y(t) + C_{vy}sgn\left(V_y(t)\right) \tag{4.21}$$

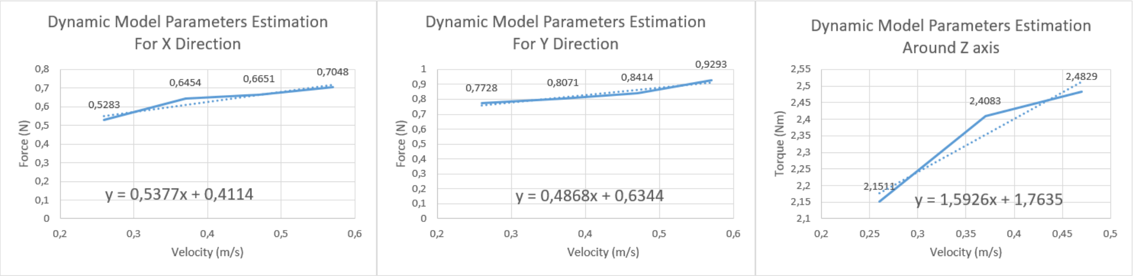
$$\Gamma(\mathrm{t}) = B_W\omega_v(t) + C_W sgn\big(\omega_v(t)\big) \tag{4.22}$$

Experiments are conducted for V_x, V_y and ω_v velocities separately. For each direction, four different velocity levels are chosen as demands and supplied currents to the motors are recorded and traction forces of the wheels are calculated based on these data. Making use of the wheels' traction forces, the total traction force/torque of the mobile platform along direction of the demanded velocity are calculated. These values are shown in Table 4.1 for each experiment. It should be noted that the measurements are done after the robot reaches the constant velocities.

Table 4.1 Demanded velocities and resulting total traction forces and torque

$\boldsymbol{V_X}$ ***(m/s)***	$\boldsymbol{F_X}$ ***(N)***		$\boldsymbol{V_Y}$ ***(m/s)***	$\boldsymbol{F_Y}$ ***(N)***		$\boldsymbol{\omega_v}$ ***(rad/s)***	**Γ** **(N.m)**
0,26	0,5283		0,26	0,7728		0,26	2,1511
0,37	0,6454		0,37	0,8071		0,37	2,4083
0,47	0,6651		0,47	0,8414		0,47	2,4829
0,57	0,7046		0,57	0,9293		-	-

Collected data are used to fit linear equations using least squares method for each motion direction. Using these equations, dynamic model's parameters are estimated for three axes separately. Estimated parameters are given in Table 4.2.



(a) (b) (c)

Figure 4.4 Dynamic model parameters estimation (a)- for X direction, (b)- for Y direction, (c)- around Z axis

In Figure 4.4, the line fit for dynamic model parameters from the experimental data are shown. In this figure, dashed lines give the result of line fit and the experimental data is shown with solid line. The line equations are given Table 4.2.

Table 4.2 Resulting equations and estimated friction values

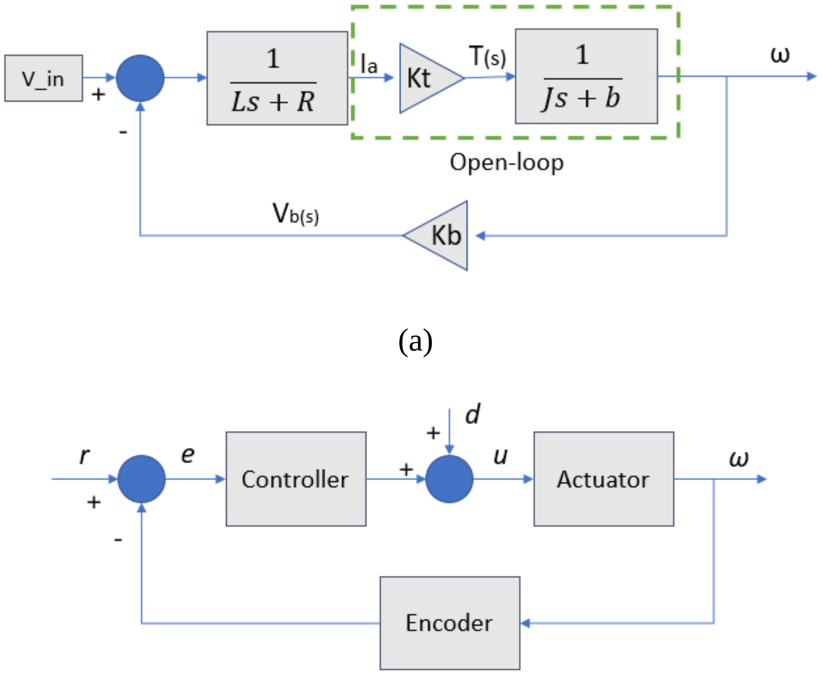
	Equations	B_x, B_y, B_W	C_x, C_y, C_W
V_x	F_x = 0,5377 V_x + 0,4114	0,5377	0,4114
V_y	F_y = 0,4868 V_y + 0,6344	0,4868	0,6344
ω_v	Γ =1,5926 ω_v + 1,7635	1,5926	1,7635

The differences between coefficients according to motion direction caused by the different level abrasion of the gearbox of the actuators.

4.4 Modeling of the Mobile Robot Components

In order to identify the characteristic behavior of the actuation system, experiments are carried out in which the system is excited with a known signal and corresponding response is observed. Firstly, the measured data is to be imported and examined. Then, the model estimation algorithms built in Matlab are used for the measured data.

The actuation system is divided into two parts which are the open-loop and the close-loop which are illustrated in Figure 4.5. The open-loop transfer function gives a relation between torque input and velocity output of the motor coupled to the omnidirectional wheel used in this study. The motor-wheel assembly in these tests is positioned so that the wheel can freely rotate. On the other hand, the close-loop transfer function gives a relation between set voltage from the DAQ to send a velocity demand and velocity output of the actuator-wheel assembly. In this next parts of this subsection, the work carried out to identify the open-loop and the close-loop transfer functions are explained and the results are presented.



(a)

(b)

Figure 4.5 Actuation system model (a)-open loop model, (b)-close loop model

4.4.1 DC Motor Model

The mechanism of the DC motors is characterized with following two equations below.

$$V = K_v * \omega + i * R + L * \frac{di}{dt} \tag{4.23}$$

$$\tau = K_t * i - b * \omega - J * \frac{d\omega}{dt} \tag{4.24}$$

Here;

- V is the input voltage
- τ is the torque
- i is the current
- Back-emf coefficient, K_v, torque constant, K_t, armature resistance, R,field inductance, L, shaft friction, b, and inertia of the rotor, J, are mechanical and electrical parameters of the motor.

This DC motor model is developed as a Simulink model as show in Figure 4.6. The parameters of the motor are selected from corresponding DC motor catalogue.

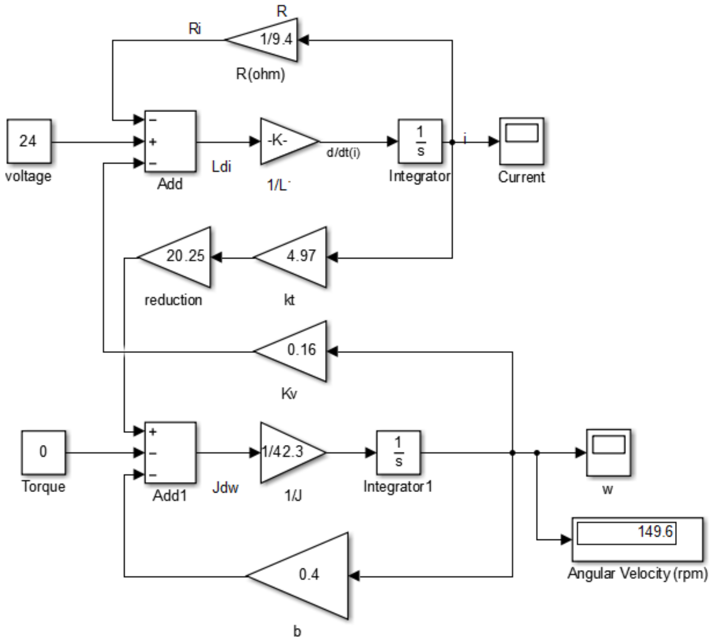


Figure 4.6 Theoretical DC Motor Model for Dunkermotoren G30.0

After applying 24VDC of set value (maximum) to the DC motor model, the response is acquired as in Figure 4.12 with theoretical model of the DC motor.

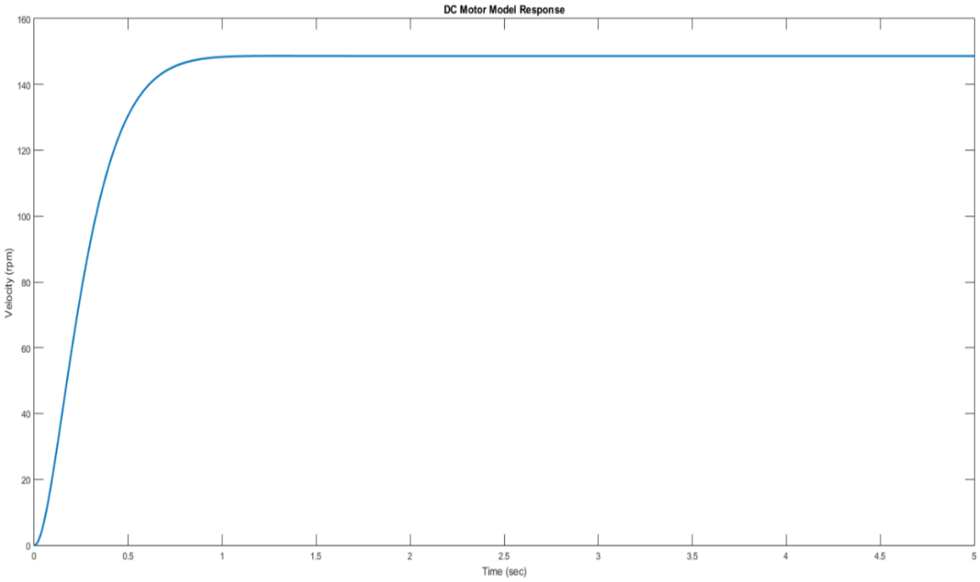


Figure 4.7 Theoretical DC Motor Model Response for an input of 24VDC

4.4.2 Open-loop Transfer Function Identification

Open-loop transfer function gives the relation between current (torque) input and motor’s angular velocity which is indicated in Figure 4.5-(a). In order to identify open-loop transfer function, input current and output velocity data are acquired from experimental setup and processed in Matlab using System Identification Toolbox. For a given reference step current input, response of the actuators are acquired from the magnetic encoder and best-fit transfer functions are obtained for each actuator by processing these data in Matlab. The identification process is executed for each DC motor separately and individual transfer functions for every single motor are calculated.

For the first motor;

No-load working current is measured to be 0.088 amperes during the experiment. The transfer function (TF) is found out by using no load current (input) and motor angular velocity (output) and it is presented in Equation 4.25. This test is done with a sampling rate of 0.01 second. Figure 4.8 shows the measured motor response and velocity output of the identified transfer function for comparison.

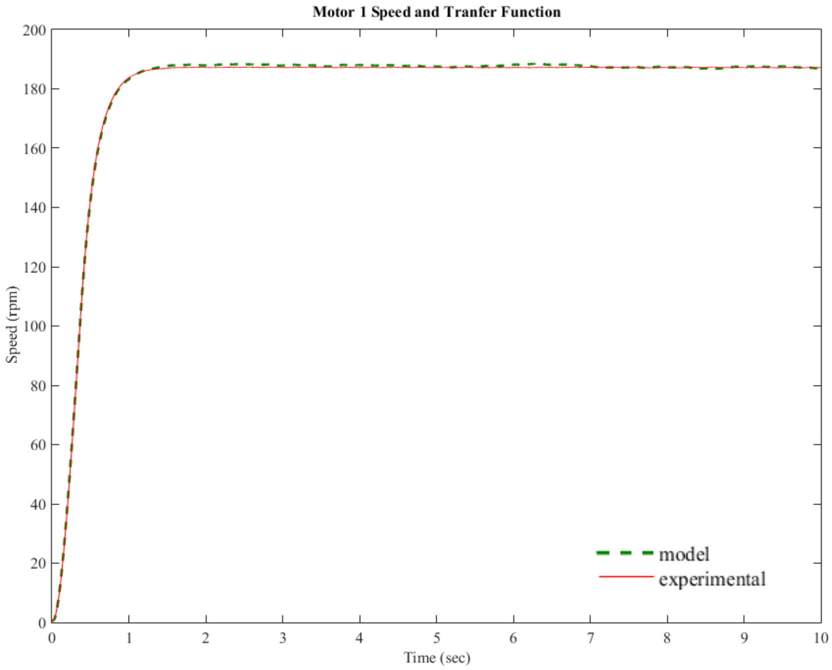


Figure 4.8 Motor 1 identification result

$$TF_1 = \frac{9.227e04}{s^2 + 12.77s + 42.97} \qquad (4.25)$$

This transfer functions response fits the experimental data a value of 96.65%.

For the second motor;

No-load working current is measured to be 0.090 amperes during the experiment. The transfer function (TF) is found out by using no load current (input) and motor angular velocity (output) and it is presented in Equation 4.26. This test is done with a sampling rate of 0.01 second. Figure 4.9 shows the measured motor response and velocity output of the identified transfer function for comparison

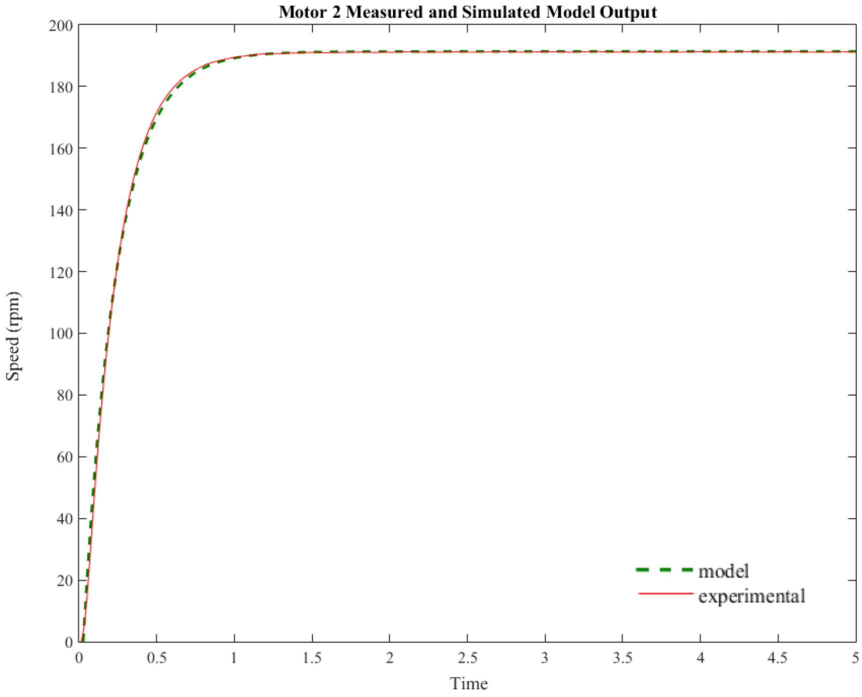


Figure 4.9 Motor 2 identification result

$$TF_2 = \frac{3{,}764e05}{s^2 + 40{,}45s + 177{,}2} \tag{4.26}$$

This transfer functions response fits the experimental data a value of 95.06%.

For the third motor,

No-load working current is measured to be 0.105 amperes during the experiment. The transfer function (TF) is found out by using no load current (input) and motor angular velocity (output) and it is presented in Equation 4.27. This test is done with a sampling rate of 0.01 second. Figure 4.10 shows the measured motor response and velocity output of the identified transfer function for comparison

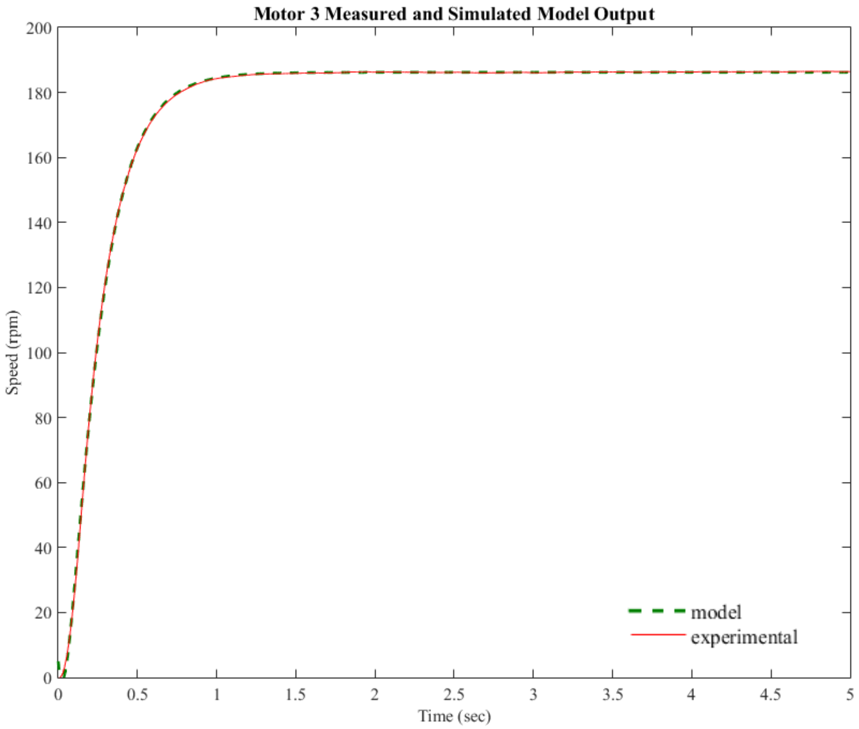


Figure 4.10 Motor 3 identification result

$$TF_3 = \frac{1{,}513e05}{s^2 + 21{,}64s + 85{,}34} \tag{4.27}$$

This transfer functions response fits the experimental data a value of 98.75%.

For the fourth motor,

No-load working current is measured to be 0.093 amperes during the experiment. The transfer function (TF) is found out by using no load current (input) and motor angular velocity (output) and it is presented in Equation 4.28. This test is done with a sampling rate of 0.01 second. Figure 4.11 shows the measured motor response and velocity output of the identified transfer function for comparison

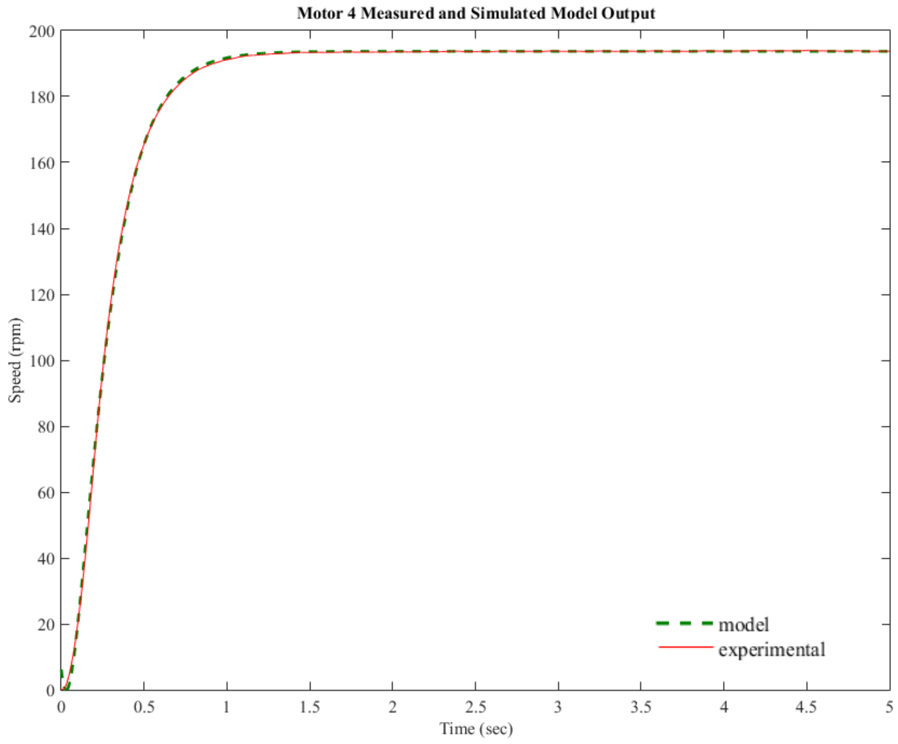


Figure 4.11 Motor 4 identification

$$TF_4 = \frac{1{,}282e05}{s^2 + 16{,}95s + 62{,}18} \tag{4.28}$$

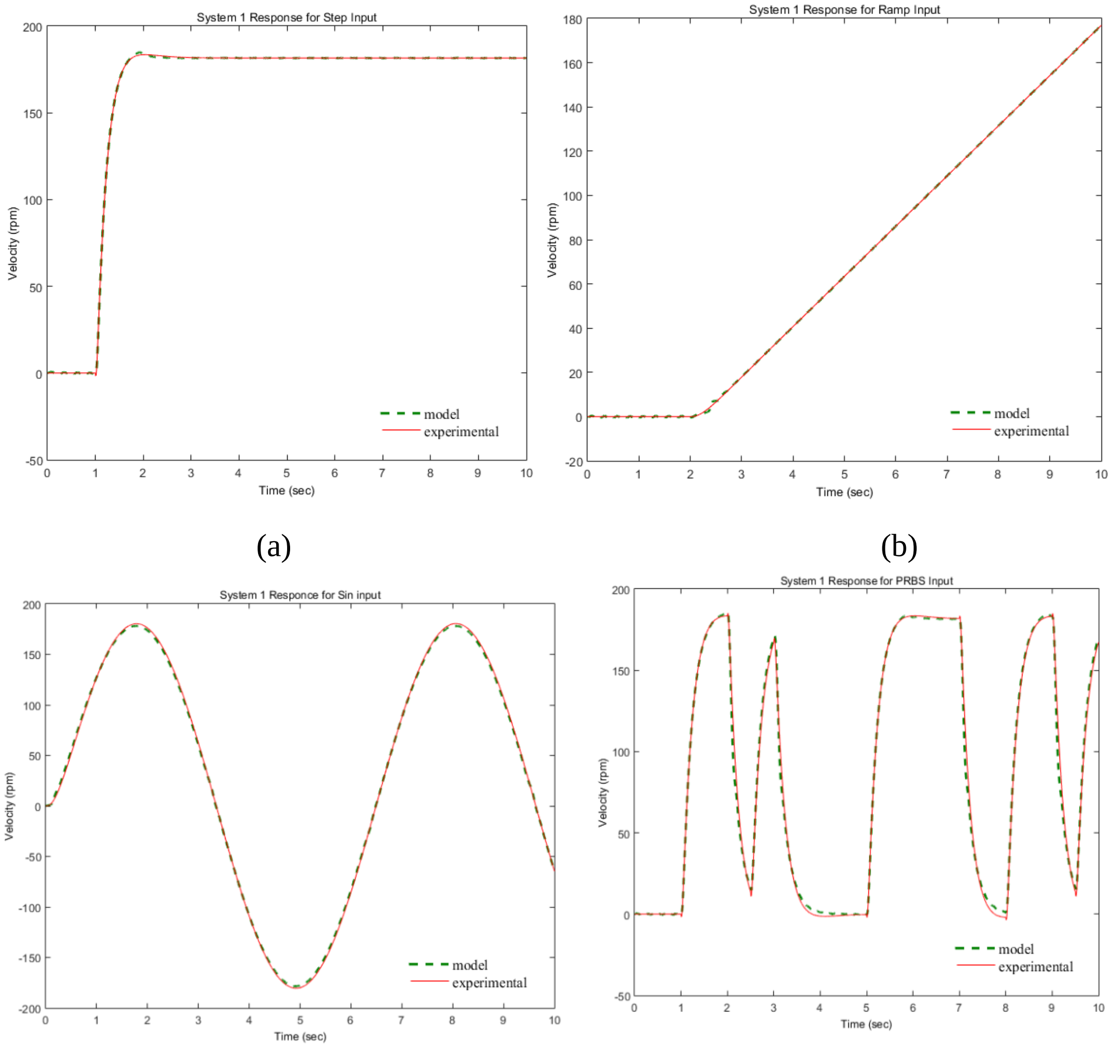
This transfer functions response fits the experimental data a value of 98.85%.

In conclusion, the response of the obtained for open loop transfer functions of each motor fit the measured data with a %95 to %99 accuracy. The transfer function of each motor differs from each other and the main reason for this is the wear of the gearboxes of the motors. Transfer functions reach the steady-state condition in about 1 second for all actuators.

4.4.3 Close-loop Transfer Function Identification

The actuation system consists of an omnidirectional wheel, DC motor, and motor driver. The close-loop transfer function gives us the relations between set velocity and output velocity. The system identification process is completed by using a step input and a transfer function for each actuation system is obtained and presented in Equations 4.29, 4.30, 4.31 and 4.32. The accuracy of the identified transfer function is checked by using four different types of inputs, which are step, ramp, sinusoidal and PRBS (Pseudo Random Binary Signal) inputs. The close-loop model identification results are illustrated in Figure 4.12, 4.13, 4.14 and 4.15. The amplitudes of the input signals are 180 rpm which is applied as a step, sinusoidal and PRBS inputs. The slope of the ramp input is 22.5 rpm/s and it is triggered at second 2. The sinusoidal input frequency is 1 rad/s.

For system 1,



(a)

(b)

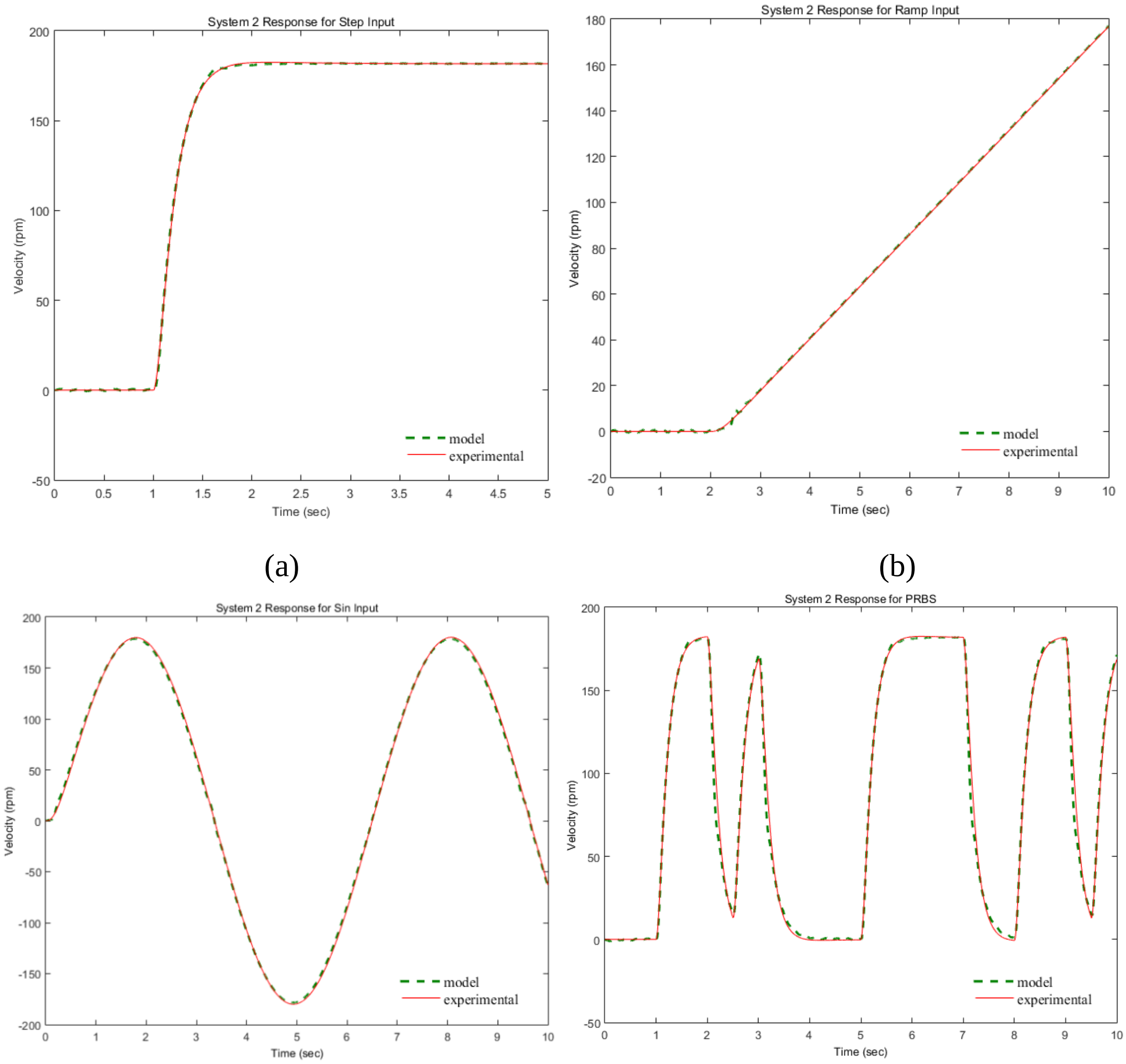
(c)

(d)

Figure 4.12 Actuation system 1 response for step, ramp, sinusoidal and PRBS inputs, respectively

$$TF_1 = \frac{-55{,}4s^2 + 5222s + 1{,}465e04}{s^3 + 55{,}99s^2 + 362{,}7s + 645{,}5} \tag{4.29}$$

For system 2,



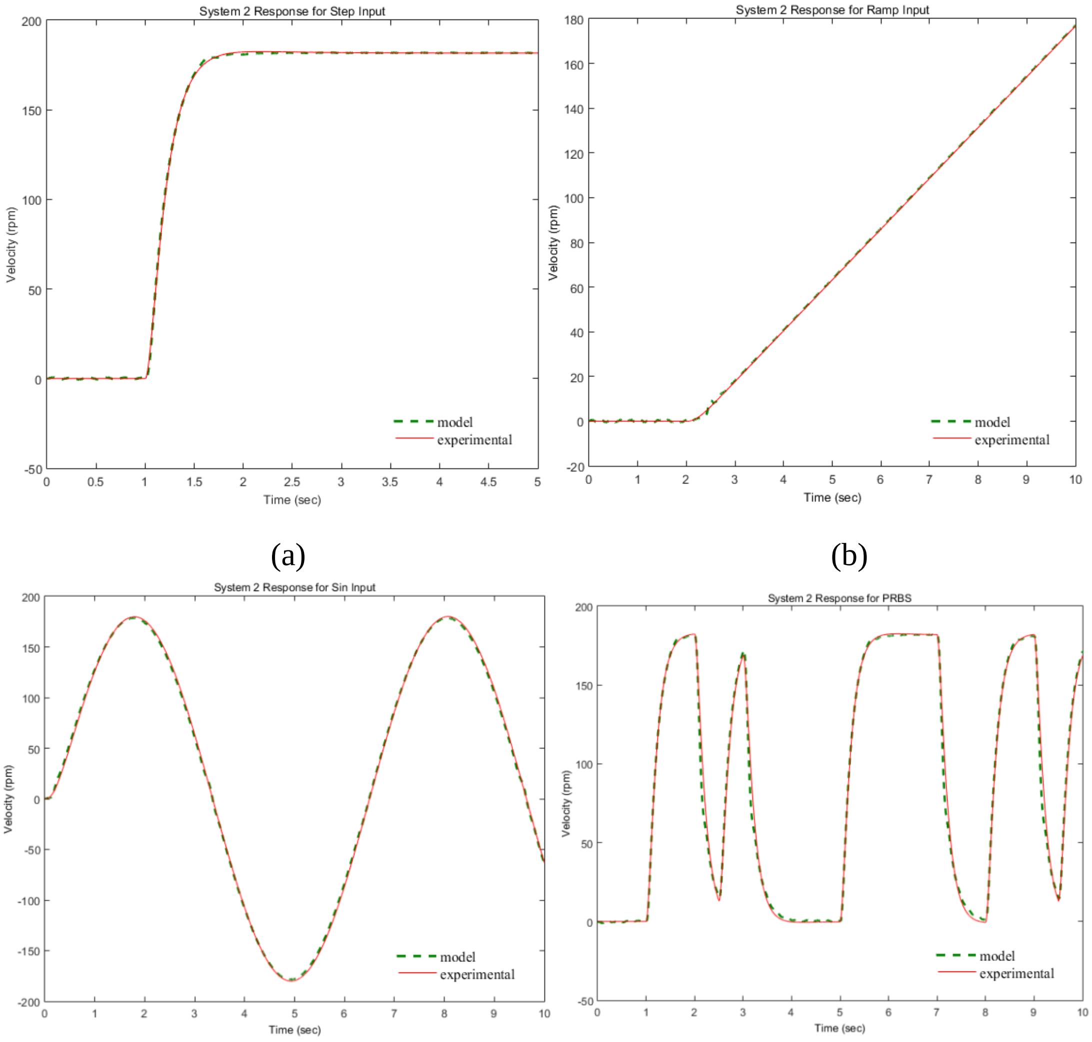
(a) (b)

(c) (d)

Figure 4.13 Actuation system 2 response for step, ramp, sinusoidal and PRBS inputs, respectively

$$TF_2 = \frac{-43,66s^2 + 3446s + 3,062e04}{s^3 + 47,02s^2 + 447s + 1349} \qquad (4.30)$$

For system 3,



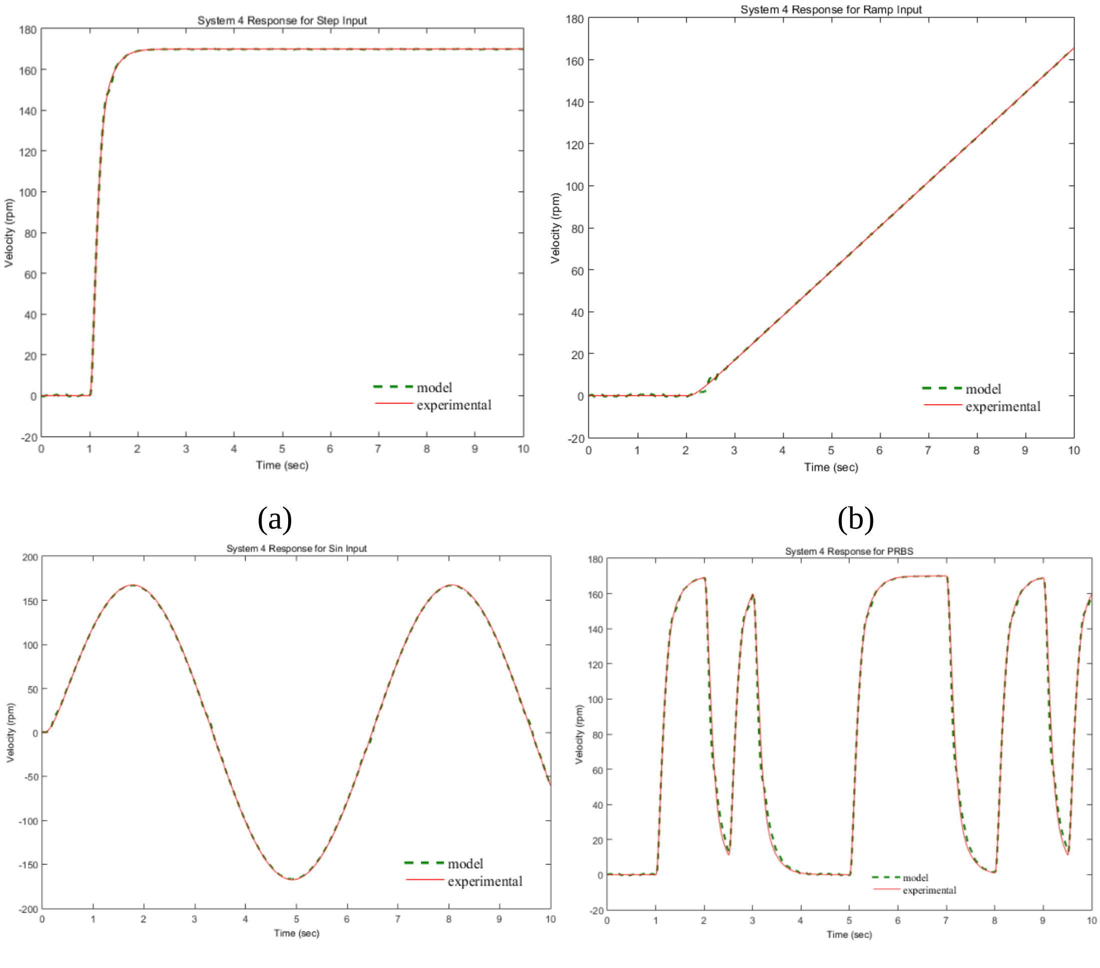
(a) (b)

(c) (d)

Figure 4.14 Actuation system 3 response for step, ramp, sinusoidal and PRBS inputs, respectively

$$TF_3 = \frac{-13{,}29s^2 + 2687s + 3221}{s^3 + 346{,}5s^2 + 8186s + 3{,}969e04} \tag{4.31}$$

For system 4,



(a)

(b)

(c)

(d)

Figure 4.15 Actuation system 4 response for step, ramp, sinusoidal and PRBS inputs, respectively

$$TF_4 = \frac{-4{,}965s^2 + 2197s + 2{,}155e04}{s^3 + 26{,}73s^2 + 313{,}2s + 1014} \tag{4.32}$$

The figures above show the correlation between identified transfer function and real system's response. In order to prove the reliability of the identification of the system, four different behavioral inputs are executed which have the same amplitude. As a consequently, all the outputs of the transfer functions follow the real system's response for the same type of input closely.

4.5 Obstacle Avoidance Algorithm

An obstacle avoidance algorithm is illustrated in Figure 4.16. The obstacle avoidance algorithm consists two section which are a virtual spring-damper system between obstacle and a virtual proxy of the mobile robot, and virtual coupling between virtual proxy and the mobile robot.

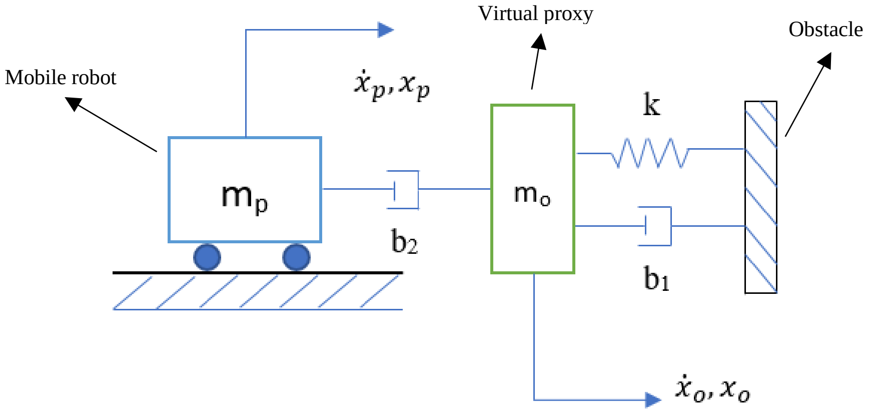


Figure 4.16 Obstacle avoidance algorithm

In the first section of the obstacle avoidance system, the virtual spring-damper system is attached in between the obstacle and the virtual proxy, which is illustrated in Figure 4.17.

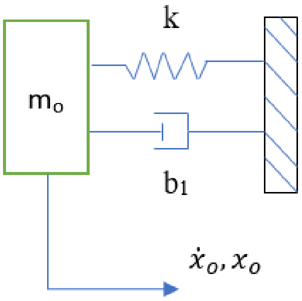


Figure 4.17 virtual obstacle avoidance system

The formula of the virtual force generated between the virtual proxy and the obstacle is given in Equation 4.33. It should be noted that a threshold value for the distance between the mobile robot and the obstacle is to be selected to activate the virtual force calculation. The coefficient of spring (k) and damper (b) are to selected experimentally.

$$F_{virtual} = kx_o + b_1\dot{x}_o \tag{4.33}$$

The mobile robot is connected to the virtual proxy with a virtual damper coupling which is illustrated in Figure 4.18.

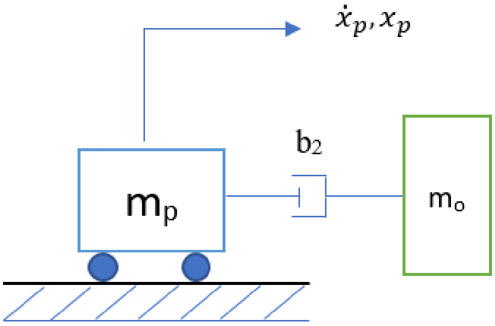


Figure 4.18 The relation between the mobile robot and the virtual object

The coupling force between virtual object and the mobile robot is calculated as Equation 4.34

$$F_{coupling} = b\dot{x_2} \; ; \; \dot{x}_2 = \dot{x}_p - \dot{x}_o \tag{4.34}$$

The coupling force between the virtual proxy and the mobile platform is required to be kept at zero for instantaneous reaction of the mobile robot to the obstacle in its working environment. Therefore, mobile platform's velocity is scheduled to be the same as the virtual proxy's velocity, $\dot{x}_p = \dot{x}_o$.

4.6 Fault Tolerance Algorithm

In case of a failure in the hardware or software of a system, fault tolerance algorithms can be utilized to accomplish the task. In this thesis study, fault tolerance scenarios are developed in case of failure in one of the actuators namely the motors. Hence, fault tolerance is adapted to the redundant system in case of losing power in one motor completely or decrement in motor efficiency. For this purpose, the failure is generated artificially in control interface by canceling motor control or decreasing its control input signal.

In Equation 4.35, the motor velocities are calculated from task space motion demand using weighted pseudo-inverse of the Jacobian matrix, $\hat{J}_w^{+}$, which is given in Equation 4.36.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \hat{J}_w^{+} \begin{bmatrix} V_x \\ V_y \\ -\omega_v L \end{bmatrix} \qquad (4.35)$$

$$\hat{J}_w^{+} = \widehat{W}^{-1}.\hat{J}^{T}.\left(\hat{J}.\widehat{W}^{-1}.\hat{J}^{T}\right)^{-1} \qquad (4.36)$$

$\widehat{W}$ is a diagonal weighing matrix. W_i represents the weight of actuator i in the calculation of performance index for minimum norm.

$$\widehat{W} = \begin{bmatrix} W_1 & 0 & 0 & 0 \\ 0 & W_2 & 0 & 0 \\ 0 & 0 & W_3 & 0 \\ 0 & 0 & 0 & W_4 \end{bmatrix} \qquad (4.37)$$

Increasing any W_i with respect to other three, in Equation 4.38, the corresponding motor's contribution in the total motion is decreased. If a weight is to be chosen infinite, related motor stops working however, the desired motion of the mobile robot is carried out with the other three since the system is already redundant. The calculation of the linear speed of the actuators for a given mobile robot velocity profile making use of the weighing matrix are given in Equation 4.38, 4.39, 4.40 and 4.41.

$$v_1 = -(0.5/(W_1 + W_2 + W_3 + W_4))(\omega_V.L(W_2 + W_4) + V_Y(Sin\theta\ (W_2 - W_4) + Cos\theta(W_2 + 2W_3 + W_4))\ + V_x\big(Cos\theta(W_2 - W_4) - Sin\theta(W_2 + 2W_3 + W_4)\big)) \qquad (4.38)$$

$$v_2 = -(0.5/(W_1 + W_2 + W_3 + W_4))(\omega_V.L(W_1 + W_3) + V_x(Sin\theta\ (W_3 - W_1) + Cos\theta(W_1 + W_3 + 2W_4)) + V_y\big(Cos\theta(W_1 - W_3) +\ Sin\theta(W_1 + W_3 + 2W_4)\big)) \qquad (4.39)$$

$$\begin{aligned} v_3 = -(0.5/(W_1 + W_2 + W_3 + W_4))(\omega_V.L(W_2 + W_4) \\ + V_x(2Sin\theta\ W_1 + (Cos\theta + Sin\theta)W_2 + (Sin\theta - Cos\theta)W_4) \\ + V_y(Sin\theta(W_2 - W_4) - Cos\theta(2W_1 + W_2 + W_4))) \end{aligned} \qquad (4.40)$$

$$
\begin{aligned}
v_4 = (0.5/(W_1 + W_2 + W_3 + W_4))(-\omega_V.L(W_1 + W_3) + V_x(Sin\theta\ (W_1 - W_3) \\
+ Cos\theta(W_1 + 2W_2 + W_3)) \\
+ V_y\big(Cos\theta(W_3 - W_1) - Sin\theta(W_1 + 2W_2 + W_3)\big)\big) \qquad (4.41)
\end{aligned}
$$

4.7 Summary of the Chapter

In this chapter, the overview of the control architecture for the redundant mobile robot is presented initially. Then, redundant mobile robot's kinematic models are examined. Following, estimation of the dynamic model parameters is represented and viscous and Coulomb friction coefficients are estimated experimentally. It is observed that the friction values for the linear motion in different axes are not the same since the actuation systems for each direction of motion has different wears in their gearboxes. Additionally, mobile robot's component models are identified experimentally. The identified models are the DC motor model (open-loop) and the actuation system (close-loop). According to results, it can be seen that each motor and actuation system have slightly transfer functions due to the wears in the gearboxes. Finally, in this Chapter, obstacle avoidance and fault tolerance algorithms are explained to be tested in experiments. The results of these experiments are provided in the next Chapter.

CHAPTER 5

TESTS OF THE CONTROL ALGORITHMS

In this chapter, experimental test results are presented for top-level controller tuning, obstacle avoidance algorithm, fault tolerance algorithm and teleoperation system. In order to set the control parameters of the top-level controller, frequency response of the system with the controller is investigated via bode plots. In obstacle avoidance tests and fault tolerance tests, system responses and characteristics are analyzed for various conditions. In teleoperation test, the communication delay between the master and the slave systems is measured.

5.1 Experimental Tuning of the Top-Level Controller

Prior to the subtask control applications on the robot such as obstacle avoidance and fault tolerance, motion of mobile robot has to be controlled with a fast responding and stable controller. The overall of controller algorithm of the mobile robot consists of a low-level controller of the motor driver and the top-level controller. The top-level controller is designed for regulating the robot’s angular velocity about the normal to the working surface.

The top-level controller designated for the robot to regulate its angular velocity is a PI controller. The controller gains, which are proportional (P) and integral (I) gains, are tuned experimentally. Initially, the controller gains are selected and the control algorithm is compiled into the control unit of the robot. Planned motion for the experimentation is a step angular velocity input about the z axis. During the tests, P and I gains are tuned using time response of the system. The steady state error obtained in the time response graph is tried to be minimized. Once the tuned gains provide a smaller steady state error, the controller with these gains are used in another test to observe the usable bandwidth of the controlled system.

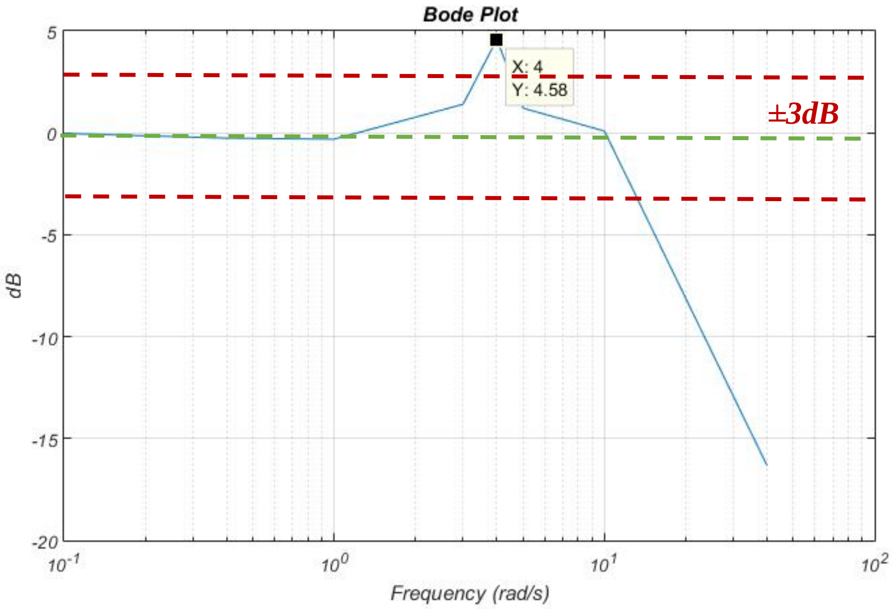
The control algorithm is designed in Simulink and the feedback signal to be used in the top-level controller obtained by using an external sensor, a gyroscope attached to the robot frame. The sampling time of the controller is selected complying with the external sensor’s working frequency, which is 140 Hz. This eliminates any noise created by the mismatch of the controller and the gyroscope frequencies.

The bandwidth of the system is measured using the frequency domain analysis. Sinusoidal velocity inputs at different frequencies are used as inputs to the system and the system’s output is acquired via a gyroscope. Then this data is used to calculate logarithmic magnitude ratio and phase delay of the system to develop Bode plots. Formulation for finding logarithmic magnitude ratio is provided in Equation 5.1 and the phase angle between the demand and the system response is calculated using the Equation 5.2. In Equation 5.1 the magnitude, $|G(\omega)|$, at each frequency is calculated by finding the ratio of peak values of input, u_{peak}, and output, y_{peak}. The phase $\emptyset(\omega)$ is calculated by using the time difference between the peaks of the input and output sinusoidal signals, t_{delay} and the period of the sinusoidal input, *T*.

$$20log_{10}|G(\omega)| = 20log_{10}\left|\frac{y_{peak}}{u_{peak}}\right| \tag{5.1}$$

$$\frac{\emptyset(\omega)}{360} = \frac{t_{delay}}{T} \tag{5.2}$$

Magnitude difference and phase delay are calculated using responses of the system for selected sine wave frequencies, ω, that are 0.1, 0.4, 1, 3, 4, 5, 10, 20 and 40 rad/s. Once they are calculated, bode diagram of the system is drawn and presented in Figure 5.3. As it can be seen in Figure 5.3, the magnitude plot has a peak value higher than 3dB approximately at 4 rad/s which is the resonance frequency. At the phase plot, the phase delay is near -90° at resonance frequency. Also, it is observed from the plot that after the resonance frequency, there is a decreasing rate of the magnitude ratio approximately at 40dB per decade. These clearly identify that the system shows a second-order system response in this frequency range. It should be noted that the peak value at resonance frequency increases as damping ratio decreases which make the system more oscillatory. In order to have a less oscillatory behavior, the damping ratio should be increased and the peak value is required to be held below to 3 dB.



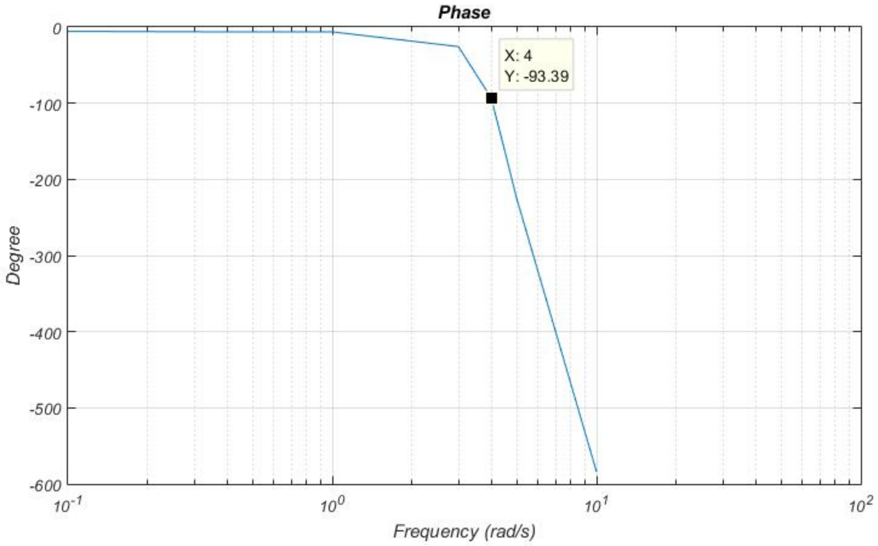


Figure 5.3 Bode Diagram of system with the initial controller

To have a less oscillatory response, PI gains are tuned to decrease the peak value beneath 3 dB at the resonance frequency. By decreasing proportional gain, Kp, and increasing the integral gain, T_i, of the PI control, final parameters of the PI controller are obtained and the Bode diagram of this system, which is again experimentally obtained, is illustrated in Figure 5.4.

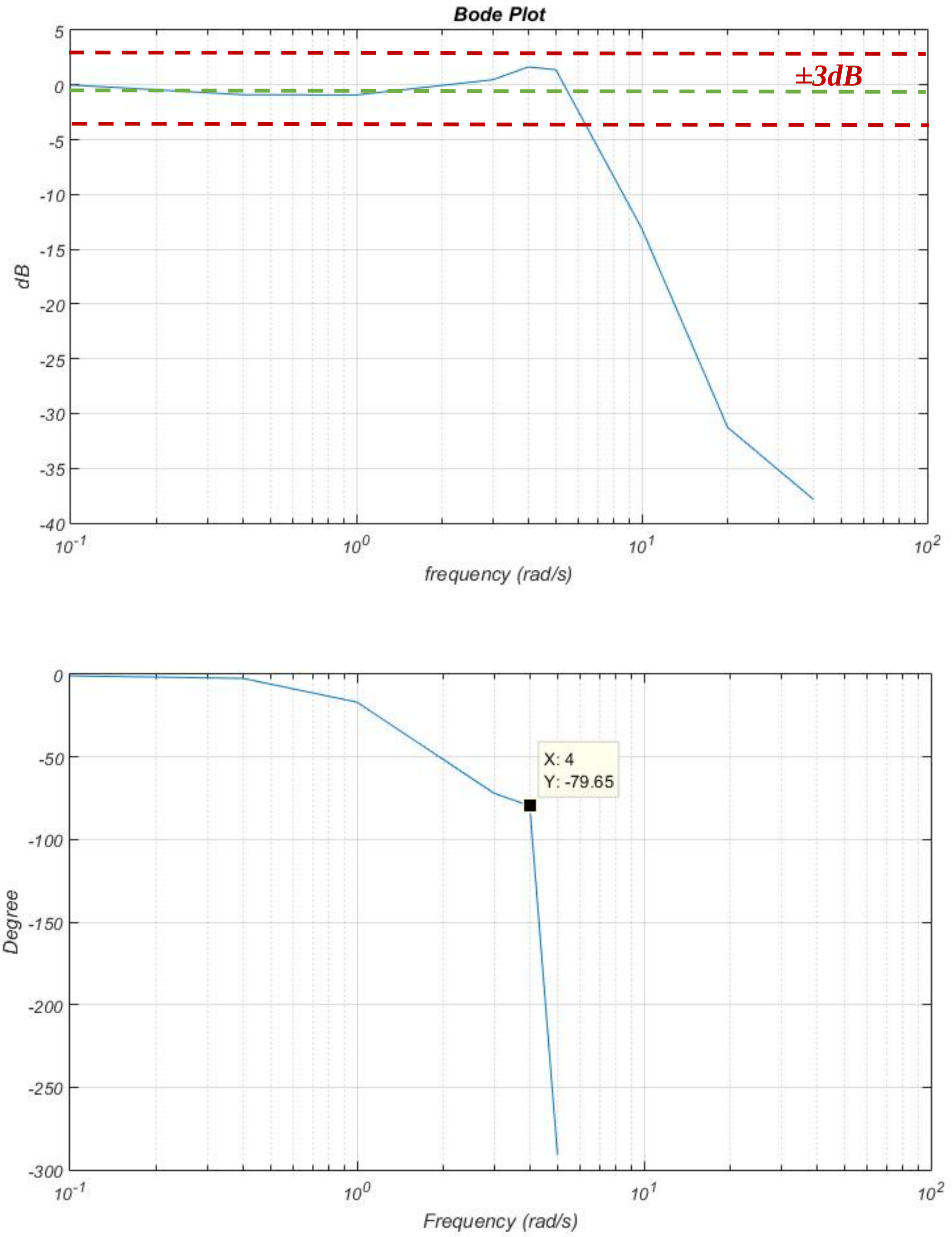


Figure 5.4 Bode Diagram of PID response for final controller

According to this graph, the system with final controller has a resonance frequency at 4 rad/s and the usable bandwidth of the system is about 6 rad/s. The time-domain characteristics of the top-level controller is given by the parameters: rise time (t_r), peak time (t_p), settling time (t_s) and percent overshoot. The step response of the system after the final tuning accomplished in frequency domain is given in Figure 5.5 and the time-domain characteristics are listed in Table 5.1.

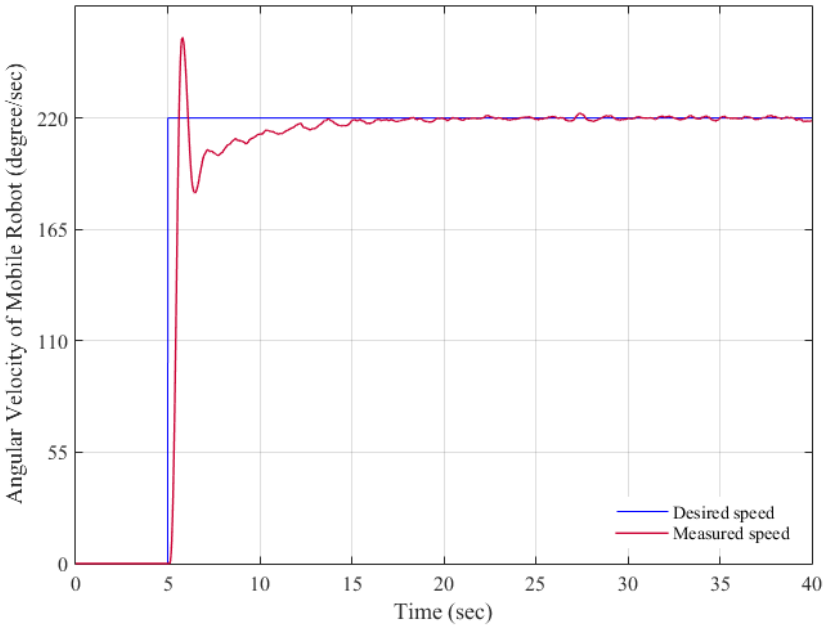


Figure 5.5 Step input response of PI controller

Table 5.1. Top-Level Controller Characteristics.

Parameters	
t_r	0.614 sec
t_p	0.796 sec
t_s	8 sec
Percent Overshoot	18
Steady-State Error	%2

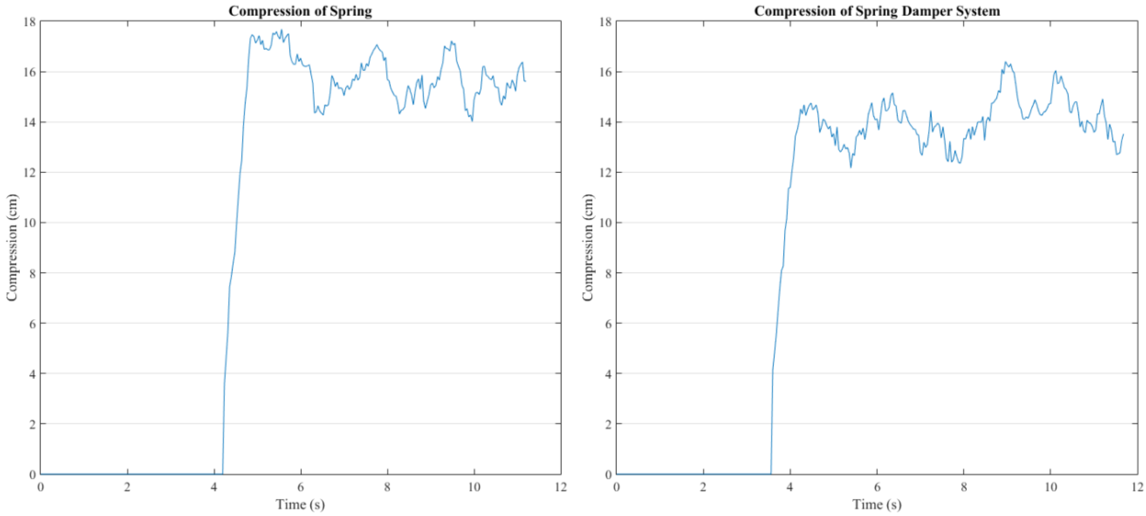
5.2 Obstacle Avoidance Test Results

The principle of the obstacle avoidance algorithm is given in Chapter 4. Before the using this algorithm, the coefficients of the virtual spring and the virtual damper are chosen by experimental evaluation. The virtual spring-damper system that is used in obstacle avoidance is tuned to ensure the system to provide maximum force at maximum penetration that is 25 cm away from the actual obstacle. The maximum force that can be created by the virtual spring-damper system is chosen to be equal to the maximum traction force that can be provided by the wheels of the mobile robot in one direction. The virtual coupling between the virtual proxy and the mobile platform is chosen to be unity so that the mobile robot’s velocity is directly regulated with respect to the virtual force.

The penetration depth of the mobile robot into spring-damper system connected to between the proxy and the obstacle is acquired by the infrared sensors placed on each side of the octagonal shaped body frame of the mobile robot. The penetration speed of the mobile robot is calculated using a discrete derivation of the distance read from the infrared sensor. However, this derivation is saturated at the maximum value of 10 in the controller since at the instance where the mobile robot penetrates the virtual wall for the first time, derivation goes to infinity.

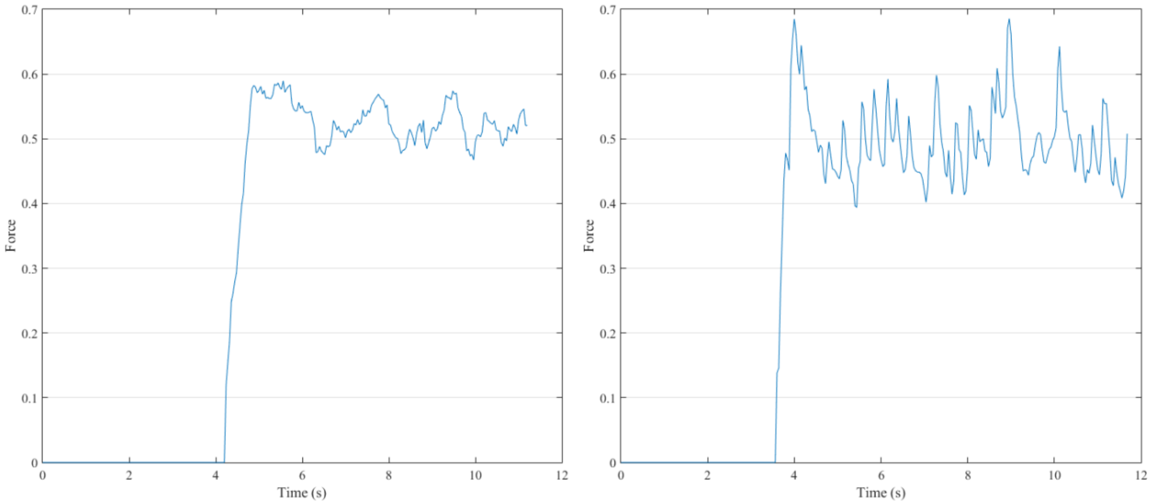
The compression of the virtual spring-damper system, the distance between the proxy and the obstacle, and calculated virtual force are shown in Figures 5.6 and 5.7, respectively. The system responses are provided for both using a single virtual spring and a virtual spring-damper system, respectively.

Figure 5.6 are the plots of measured distance between the proxy, which is equal to the mobile robot position, and the obstacle when the mobile robot is subjected to a constant velocity demand of 0.5m/s. In Figure 5.6 the distance from obstacle is measured using infrared sensors. In Figure 5.7, the virtual forces calculated as a result of obstacle avoidance algorithm are presented with respect to time for the same experiment shown in Figure 5.6. In Figure 5.6 (a) and 5.7 (a), the outcome of obstacle avoidance algorithm when only using a virtual spring are shown. On the other hand, in Figure 5.6 (b) and 5.7 (b), results for the system with the virtual spring-damper are shown in means of distance to obstacle and calculated virtual forces. The effect of the addition of a virtual damper is observed in compression as the penetration of the mobile platform into the virtual spring-damper is less with respect to the single virtual spring system. The calculated virtual force as a result of this penetration in both systems is almost the same. However, higher magnitude ripples can be observed which is the result of discrete derivation to calculate mobile platform velocity to be used in virtual force calculation with the virtual damper.



(a) (b)

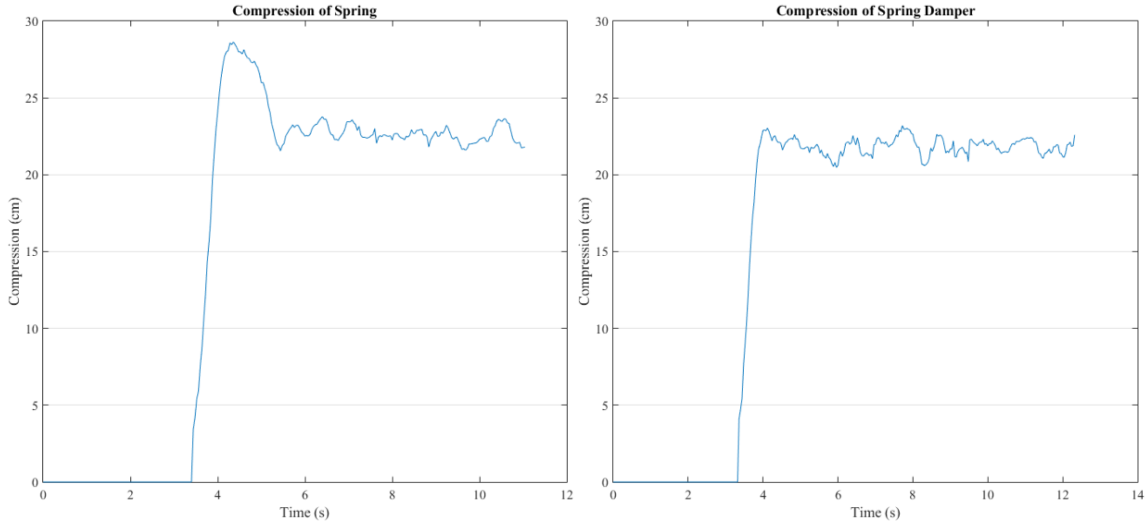
Figure 5.6 For a mobile robot speed of 0,5 m/s, compression of the virtual system with (a)-virtual spring, (b)-virtual spring-damper



(a) (b)

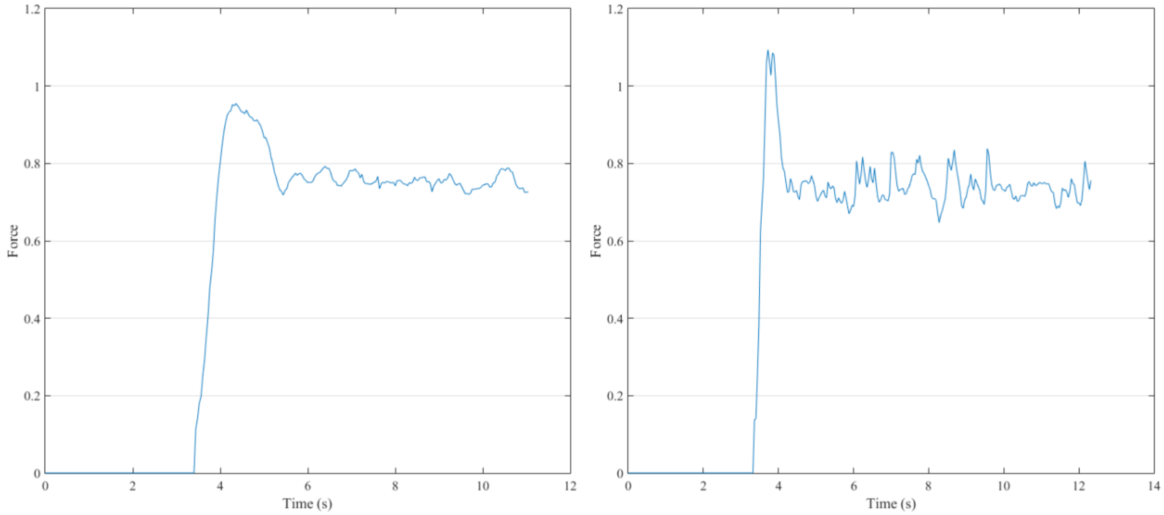
Figure 5.7 For a mobile robot speed of 0,5 m/s, calculated virtual force of the system with (a)-virtual spring, (b)-virtual spring-damper

Similar to Figure 5.8 and 5.7, Figure 5.8 and 5.9 are drawn in this case when the robot is moving at a speed of 0.75 m/s. It can be clearly observed that the damper effect prevents the overshoot which is the extra compression before reaching the steady-state. Also, the damper effect reduced the time to the reach steady-state as it can be observed in the virtual force graph.



(a) (b)

Figure 5.8 For a mobile robot speed of 0,75 m/s, compression of the virtual system with (a)-virtual spring, (b)-virtual spring-damper



(a) (b)

Figure 5.9 For a mobile robot speed of 0,75 m/s, calculated virtual force of the system with (a)-virtual spring, (b)-virtual spring-damper

As a result, when compression graphs are compared for each case, the damper effect can be observed clearly that it produces better results in avoiding obstacle by keeping robots away from the obstacle even in before reaching the steady-state condition. After validating the effect of the additional virtual damper in the obstacle avoidance algorithm, obstacle avoidance algorithm tests are carried out for multiple obstacle avoidance.

Case Study: Multiple Obstacle Avoidance Trial

The mobile robot having the previously described virtual spring-damper system with tuned parameters is tested on a track that has obstacles. Control algorithm within the mobile robot generates forces in two directions, x- and y-axes. The flow chart of implemented obstacle avoidance algorithm is given in Figure 5.12.

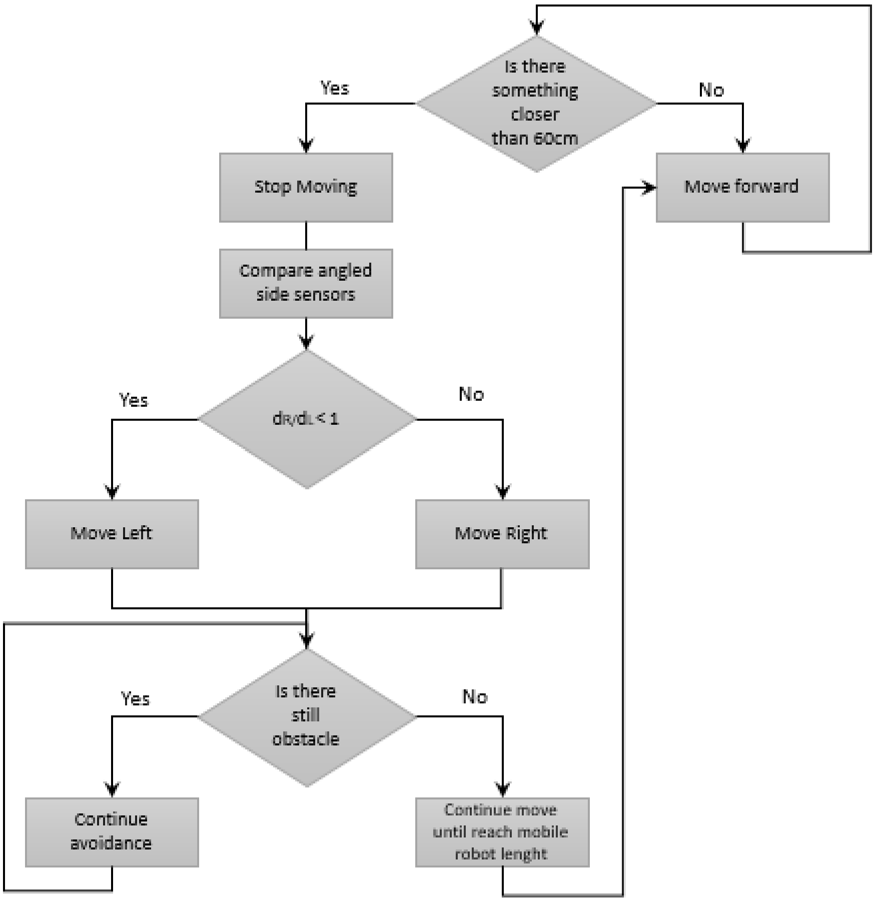

Is there something closer than 60cm
Yes
No
Stop Moving
Move forward
Compare angled side sensors
Yes
dR/dL < 1
No
Move Left
Move Right
Yes
Is there still obstacle
No
Continue avoidance
Continue move until reach mobile robot lenght

Figure 5.12. Flow chart for obstacle avoidance algorithm

Primary force is generated in the opposite direction of the motion when an obstacle is detected along the motion path. The detection of the obstacle happens when the distance sensor along the motion path issues that there is an obstacle closer than 60 cm to the mobile robot. This force is calculated using the virtual spring-damper system. A secondary force is used to avoid obstacles on the path and the magnitude of this force is calculated using only the virtual spring model that has the same spring coefficient used in the spring-damper system to calculate the primary force vector. The direction of this force is determined according to the ratio of the distance measured by the right and left distance sensors ($d_R/d_L<1$). The right and left directions are decided with respect to the motion path direction of the robot and they are the adjacent sensors to the one looking towards the obstacle. When the sensor, looking directly towards the obstacle, cannot sense the obstacle anymore, the mobile robot continues executing the avoiding motion for a certain amount of distance which is equal to the width of the mobile robot. This ensures the safe avoidance of the obstacle.

In the Figure 5.13, the test result of the obstacle avoidance algorithm is illustrated. In order to draw these graphs, a marker pen is attached to the mobile robot to mark its path during the experiment. In this experiment, mobile robot is initiated its motion from (0,0) point. The planned constant speed of the robot is 0.5 m/s along the (+) y-axis. The obstacles are placed sequentially along the y-axis and in the right and left sides respectively. The obstacles placed on the right and the left sides are required to make the robot move in the other direction to avoid the obstacle.

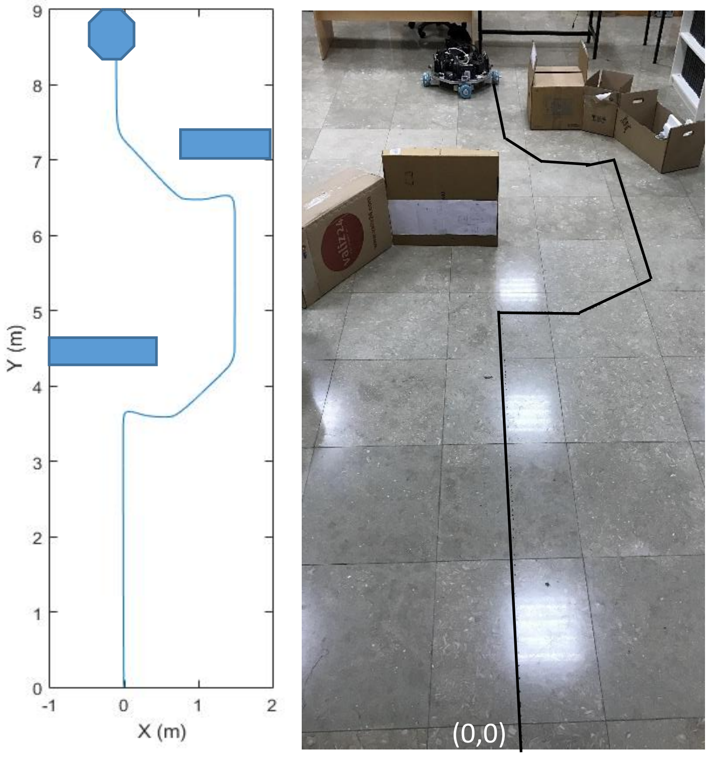


Figure 5.13. Experimental result for the two obstacles case

The same test methodology is applied for a different maximum speed of 0.8 m/s and the result is provided in Figure 5.14 with the blue lines against the initially selected speed response which is denoted with the black lines. It can be seen that the orientation of the mobile robot in the second experiment is not the same as it is in the first experiment. However, it can be clearly observed that the results with the 0.8 m/s speed produces an overshoot in y-direction and gets closer to the wall compared to the previous test result.

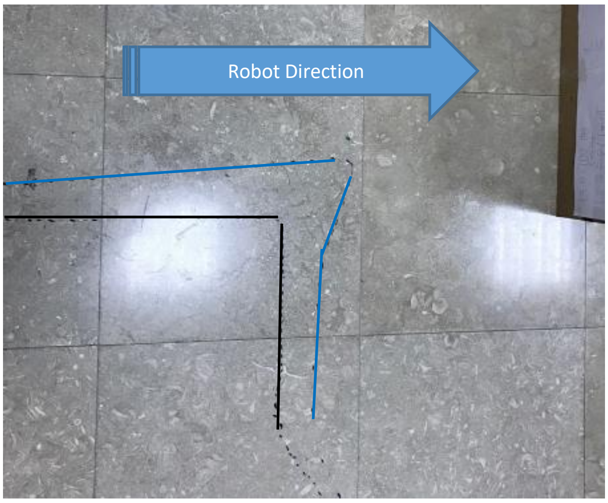


Figure 5.14 Test results of the obstacle avoidance algorithm for mobile robot speeds of 0.5 m/s and 0.8 m/s

5.3 Fault Tolerance Algorithm Test Results

The fault tolerance algorithm is designed for maintaining the mobile robot's movements in case of a performance loss in one of its actuator. This algorithm makes use of the weighted pseudo-inverse of the Jacobian matrix, which is explained in Chapter 4, to obtain joint space motion from workspace motion demands. In the case of the faulty actuator, all actuators motion demands are rescheduled by changing the weight of that actuator in the weighing matrix. During the experiments, faults are created artificially in the algorithm as performance drops in one of the actuators. The objective of this thesis does not include the development of a fault detection algorithm therefore, it is assumed that fault is detected instantaneously and the time of occurrence is known.

The fault is defined as proportional performance loss. During the experiments, the actuator is subjected to performance loss up to %70. According to test scenario, while the mobile robot is following a linear path, one of its actuators is subjected performance loss at 5th second.

Initially, the fault tolerance algorithm is tested in simulations. Simulation results show that algorithm is able to overcome all proportional performance losses. The actuator velocities and the mobile robot's motion are monitored and compared with respect to a desired motion input for a 10 second period. Moreover, simulation studies for fault tolerance are executed only for one axis since the mobile robot model in the simulation has identical actuation systems in both directions however, in the real mobile robot, the actuation systems in both directions are not identical. The difference in the actuation systems of the mobile robot can be listed as skew actuator axes, friction coefficient difference for the wheels and unequal distance of the wheels with respect to the mass center. For this reason, independent tests are run for observing fault tolerance algorithm for motion along the x- and y-directions. The simulations and experiments are run for different levels of performance loss in which the loss is increased by 10 percent until the actuators are completely disabled. In each simulation and experiment, the motion is started at 1 and 2 second, respectively and the performance loss for the first actuator is given at 5th second. The figures given in this section shows the simulation and experiment results only for one of the performance loss level, 50%. The results with the other levels of performance loss are not illustrated in this section with figures but they are compared with common metrics which are settling time, overshoot and steady-state error of the wheels' angular speeds. The resultant performance metrics are given in Table 5.1 and 5.2, and the figures of the performances resulted from the tests with the other performance loss levels are given in Appendix E.

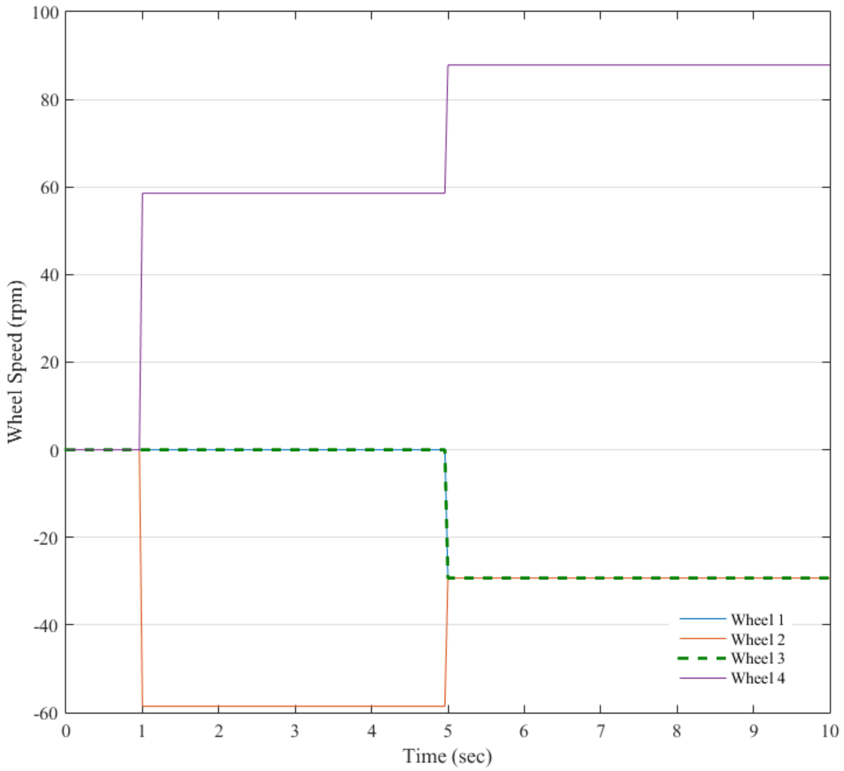


Figure 5.15 50% performance lose simulation test result

In Figure 5.15, the angular speeds of the wheels are calculated to provide a 0.6 m/s linear speed of the mobile robot in x-direction. Initially, the wheels 1 and 3, which are given by green dashed and blue lines, runs at 0 rpm while wheels 2 and 4 runs at -60 and +60 rpm, respectively. Until 5 seconds the mobile robots works without any fault in any of its actuators. However, at the 5th second, 50% of performance loss is issued to wheel 2, which can be observed as its velocity drops to 30 rpm. When this occurs, the pairing wheel that is placed on the opposite site, which is wheel 4, increases its angular velocity 50% that is 90 rpm approximately. This increase in rpm is executed by the controller to secure the tracking of the demanded linear velocity of the mobile robot. The angular velocity change in wheel 1 and 3 is issued by the algorithm to compensate the moment generated as a result of this fault. The angular velocities of wheel 1 and 3 rises to 30rpm in their (-) directions.

In the experimental study for the 50% performance loss in wheel 2 when the motion is specified to be in the x-direction, the motion to the mobile robot is given at 2nd second and the fault is issued at the 5th second. Before explaining the result of this test, it is should be recalled that wheels 2 and 4 has a relatively larger friction coefficients and they do not slip even in fast angular velocity changes. In this test, when the fault is given to wheel 2 and the controller triggers fault tolerance algorithm. The result of this test is presented in Figure 5.16.

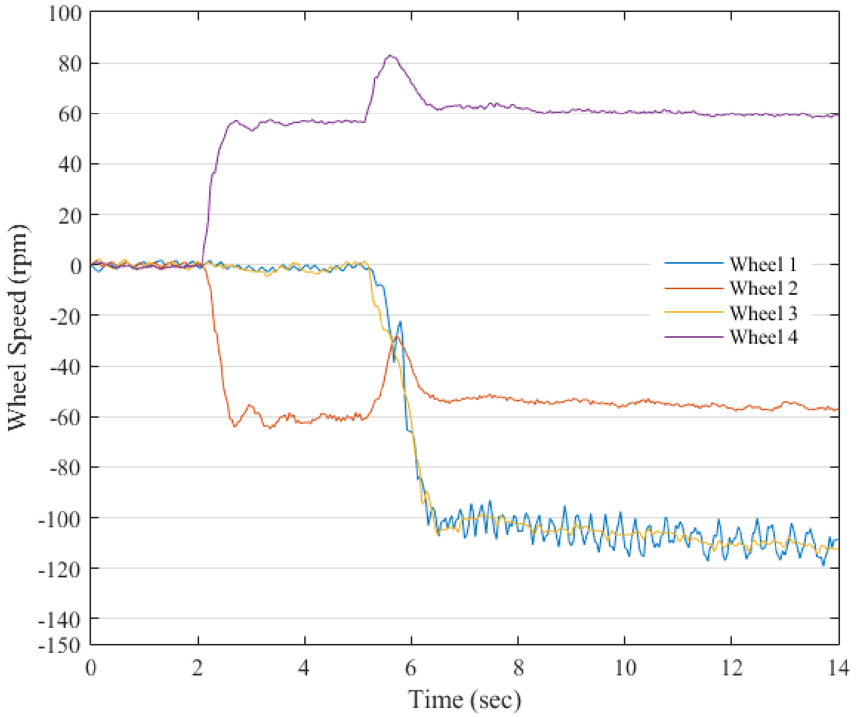


Figure 5.16 50% performance loss experiment result during the motion in x-direction

In Figure 5.16, it is observed that the wheel 1 and 3 speeds rise to 140 rpm in (-) direction to maintain the angular velocity of the mobile robot at a zero angular velocity about the z-direction. Although there is a fault in the actuator driving the second wheel, the angular velocity of wheel 2 does not drop to 50% of its normal angular velocity. This occurs due to the control of the angular velocity of the mobile robot with wheels 1 and 3. When the angular velocity of the mobile robot settles down to zero, then the faulty wheel 2 rolls freely at the same speed with its pair, wheel 4, due to the friction without any slip. When this happens in wheel 2 side, wheel 4 tries to maintain the linear velocity as demanded. The transient state until the mobile robot settles to the steady-state is clearly observed after the 5th second from the angular velocity of wheel 4.

Figure 5.17 provides the gyroscope feedback signal that is used in the top-level controller of the mobile robot. The transient state can be observed as a large overshoot just after the 5th second. As a result of this overshoot, the orientation of the mobile robot changes however, after the transient state, the angular velocity of the mobile robot is kept around zero, which proves that the fault tolerance algorithm was successful in maintaining the demanded angular velocity profile.

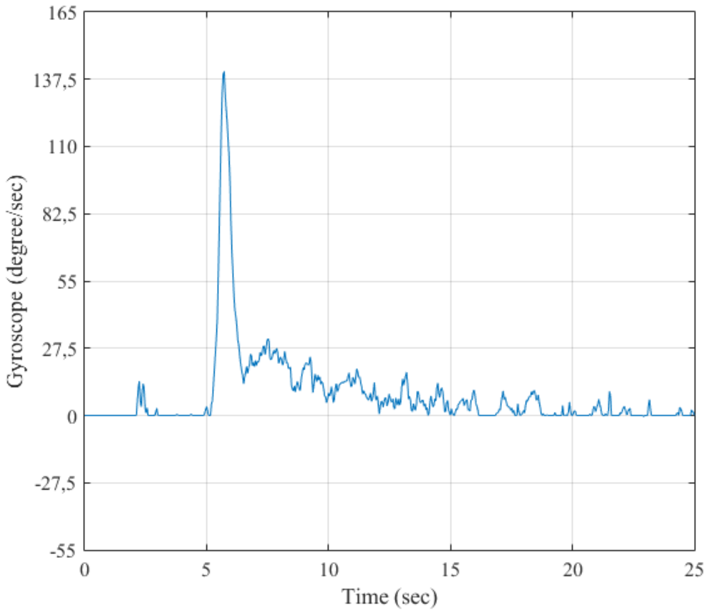


Figure 5.17 Gyroscope data acquired for 50% performance loss in the x-direction motion

Figure 5.18 gives the results of the test executed for 50% performance loss in wheel 1 during the motion along the (+) y-direction. The time of occurrence of initialization of motion and the fault for this test is identical to the one executed for x-direction. In contrary to the previously given test, the wheels providing the motion along the primary direction have relatively low friction coefficients and the wheels can slightly hold the ground even in slow start conditions. The results of this test show that wheels 1 and 3 show similar performance changes with simulation but the wheels 1 and 3 cannot transfer the motion to the ground adequately that result in smaller moment to be compensated for compared to the previous case. Because of the reduced amount of moment, mobile robot’s orientation changes slightly which can be compensated by wheel 2 and 4 with smaller modifications on their angular velocity.

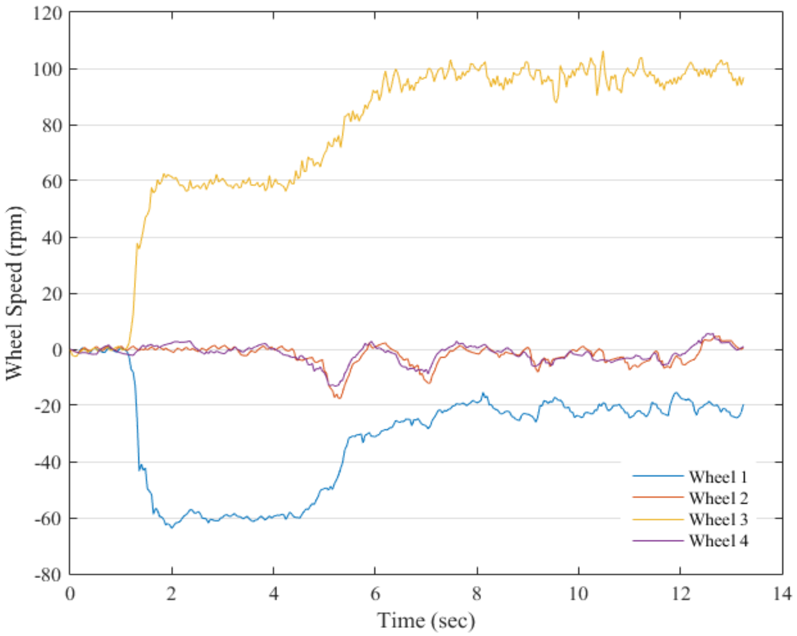


Figure 5.18 50% performance loss experiment result during the motion in y-direction

Gyroscope output during this test is shown in Figure 5.19. The transient state is observed similar to the previous case however, the overshoot is slightly less in this one since the faulty wheel does not generate as much moment as it was generated in the previous case.

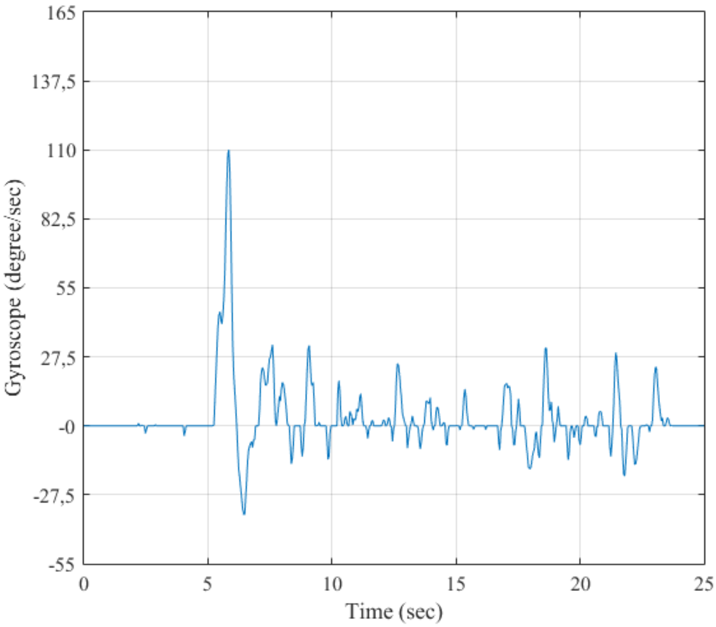


Figure 5.19 Gyroscope data acquired for 50% performance loss in the x-direction motion

The resultant paths for the tests for the 50% performance loss in x- and y-directions are illustrated in Figures 5.20 and 5.21, respectively. The figures are drawn by mounting a marker pen on the mobile robot to mark the path on the floor. Later, in order to make these paths more visible, the path is re-drawn in a drawing software.

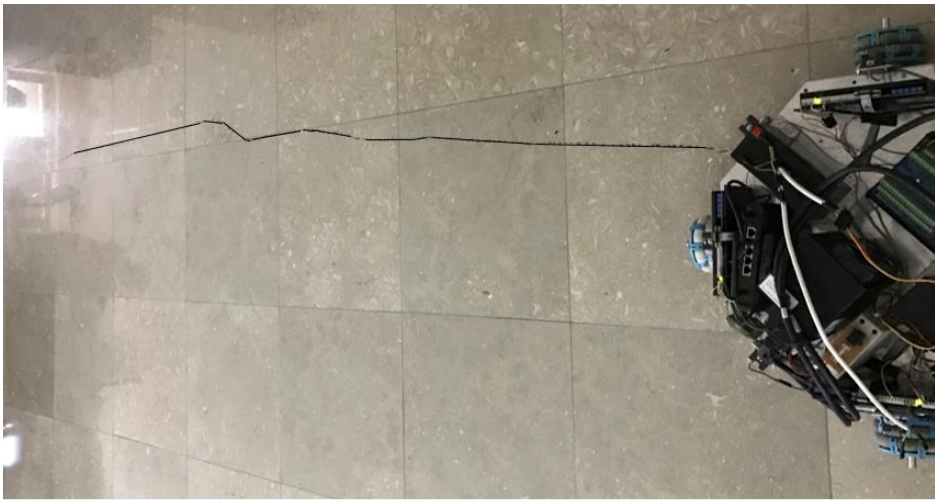


Figure 5.20 Mobile robot's path when 50% performance loss occurs in wheel 2 during the motion in (+) x-direction

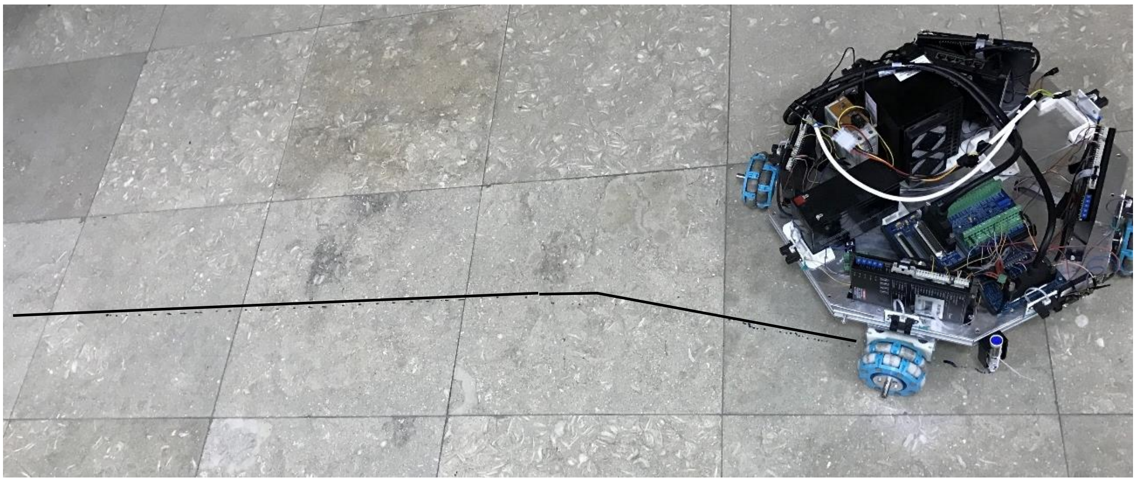


Figure 5.21 Mobile robot's path when 50% performance loss occurs in wheel 1 during the motion in (+) y-direction

In both figures, it is clearly seen that mobile robot is able to recover a linear path after a fault is introduced. Since the angular velocity is controlled, the mobile robot's orientation is changed during the transient state and kept constant after the transient state is over.

The experiment described for the 50% performance loss is repeated for 10%-70% performance losses by an increment of 10% in each direction. In order to compare the results, certain performance metrics are chosen as percent overshoot, settling time and average error. The results are obtained by processing the gyroscope data and they are tabulated in Tables 5.2 and 5.3 for the motion in x- and y-directions, respectively. The tables represent the values for experiments carried out with 10% to 70% performance loss. Settling time is found when the response of the system settles inside the specified error band for each experiment which are represented as steady-state error band in the Tables 5.2 and 5.3. Overshoots are given in deg/s which top peak values of the transient response which is realized when the fault is introduced in the experiments.

Table 5.2 Performance metrics of the fault tolerance algorithm with respect to the proportional performance losses in x-direction

Percentage	Overshoot (deg/s)	Settling Time (s)	Steady-State Error Band (deg/s)
10	13	3	1
20	70	3	3
30	88	6	5
40	117	8	7
50	145	8	9
60	157	8	10
70	187	6	10

Table 5.3 Performance metrics of the fault tolerance algorithm with respect to the proportional performance losses in y-direction

Percentage	Overshoot (deg/s)	Settling Time (s)	Steady-State Error Bandwidth (deg/s)
10	10	2	1.375
20	52	3	8.25
30	68	4	9.625
40	90.5	5	16
50	110	7	33
60	121	8	33
70	144	8	33

According to simulation study results, fault tolerance algorithm is able to overcome all proportional performance loss with bounded errors in trajectory tracking. On the other hand, in experimental studies, fault tolerance algorithm produced acceptable results up to 70% performance loss in trajectory tracking. Especially, after 40% of performance loss, the response of the system starts fluctuating right after the performance loss is started while the mobile robot is moving towards y-direction. After 70% performance loss, the mobile robot cannot sustain its motion in both directions in which fault tolerance could not be accomplish. For the fault tolerance tests, along with the x- direction, it can be said that the system shows relatively better performance when compared with test results obtained along y-direction. In Table 5.1 and 5.2, the response characteristics of wheel angular velocities are given in means of overshoots, settling time and error band in steady state for different percent of performance loss. From the tables, it can be seen that the errors is much lower and acceptable for the experimental tests done in the x-direction. As a result, the trajectory tracking with fault tolerance algorithm along x-direction shows better performance as can be seen in test results. The possible reasons for that could be;

- Nonlinear resistance due to difference between coefficients of friction for different wheels

- Shifted mass center of mobile robot

5.4 Teleoperation

Teleoperation can be explained as controlling a robot from a distance site. Teleoperation systems contain a master device which is operated by a human operator, and a slave device, the robotic tool, operating in the distance place. Teleoperation is commonly used for mobile robots which are used in hazardous areas, underwater robots, and aerial robots.

The User Datagram Protocol (UDP) is an alternative communications protocol to Transmission Control Protocol (TCP). Both UDP and TCP run on top of the Internet Protocol (IP). UDP provides a best-effort datagram service to an End System (IP host).

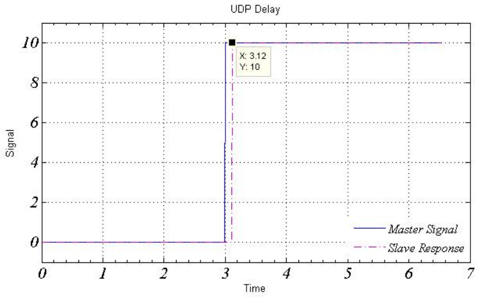
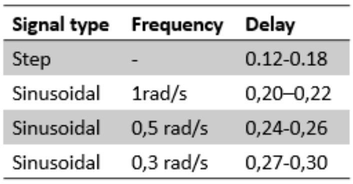


Figure 5.22 Identification of Delay from sent and received UDP data.

As provided in Figure 5.22, system delay of bilateral teleoperation is measured 0.12 second for a step input. The communication delay is identified regarding to variety input signals and the results are tabulated in Table 5.4

Table 5.4 Communication delay



Signal type	Frequency	Delay
Step	-	0.12-0.18
Sinusoidal	1rad/s	0,20–0,22
Sinusoidal	0,5 rad/s	0,24-0,26
Sinusoidal	0,3 rad/s	0,27-0,30

CHAPTER 6

CONCLUSIONS

This thesis aims to improve the previously built mobile robot in IRL by providing better overall control performance and subtask controllers. For this study, firstly the redundant mobile robot is redesigned and rebuilt with new components. The major change in the mechanical hardware is the integration of a suspension system into wheels' assembly for better road-holding and reduced vibrations transmitted to the electronic hardware. The new suspension system is supported with spring elements and a bedding system. Another improvement on the mobile robot is the change of the DAQ system with new and more capable hardware in terms of more signals I/O and better performance with increased analog resolution.

For control studies, literature is reviewed according to the subject area of the thesis, which is top-level control of omnidirectional mobile robots, obstacle avoidance and fault tolerance in mobile robots. After the literature is reviewed, description of hardware and software used in the designed system are presented. Next, the control of mobile robot and control subtasks, obstacle avoidance and fault tolerance, are explained. The method for control of the mobile robot is selected among possible methods in the literature. Top-level controller is applied to regulate the angular velocity of the mobile robot about the z-axis. As a result of having a top-level controller, it was possible to compensate for the orientation shifts due to manufacturing errors and also transient state during the fault tolerance case studies.

The obstacle avoidance algorithm is devised so that virtual forces are generated using virtual spring-damper by measuring the distance between the obstacles and the mobile robot via IR sensors. The IR sensors used in the obstacle avoidance case has measuring ranges of 25 cm to 60 cm. Therefore, the maximum virtual force generated is scheduled so that the 25 cm distance will not be penetrated. Also, the virtual force calculations are initiated as soon as the mobile robot is in the 60 cm range with respect to the obstacle. The test results indicated that the obstacles along the path of the mobile robot are successfully avoided even under high speeds of the mobile robot.

The weighted pseudo-inverse of the Jacobian matrix method is applied to reschedule wheel performance in the case of a fault in one of the wheel actuators. Although the simulation tests indicate that the wheel performance scheduling work sufficiently, during the experimental tests, it was observed that the disturbance generated by the faulty robot could not be rejected with open-loop control. It should be noted that in the simulations, all the wheels are modeled to be identical. Therefore, top-level control is used along with the fault tolerance algorithm. The experimental result indicated that the fault tolerance algorithm with the top-level controller was successful until the fault in a wheel reached to a 70% of performance loss. It was also deduced that the different friction coefficients of the roller on the wheels resulted in different performances in x- and y-direction motions. This is also the main source of the disturbance which could not be rejected with the open-loop control strategy.

Teleoperation scenario is practiced for future studies and the communication delay is observed in between 0.12 -0.3 seconds. For future works, to create more accurate motion, the structure of the mobile robot can be enhanced by using omnidirectional wheels that have roller with the friction coefficients equal to each other. Using the same type of wheels will improve the control of the robot in the sense of motion tracking and fault tolerance. Moreover, in the future studies, the mobile robot can be employed as the slave system in a bilateral force-reflecting teleoperation system by using virtual force calculation capability which is already devised in the obstacle avoidance algorithm. Additionally, fault tolerance and obstacle avoidance algorithms can be used and tested together.

REFERENCES

Amazon (2016), Amazon website (Accession date: 30.11.2016) https://www.amazon.com/b?node=8037720011

Audi (2016), Audi Micro website (Accession date: 30.11.2016) http://microsites.audi.com/mission-to-the-moon/article/the-evolution-of-the-moon-rover/

Axenie C., Cernega D., Mobile Robot Fault Tolerant Control. Introducing ARTEMIC., Recent Advances in Signal Processing, Robotics and Automation, 2010, pp. 213-219

Batteryuniversity (2016), Batteryuniversity webpage (Accession date:30.11.2016) batteryuniversity.com/learn/archive/whats_the_best_battery

Conceicao A. S., Moreira P., Costa P. J., Model Identification of a Four Wheeled Omni-Directional Mobile Robot, IFAC, 2006

Connor F. W., Yukun L., Christopher M. C., Christoper G. L., Human vs robot: Comparing the viability and utility of autonomous underwater vehicles for the acoustic telemetry tracking of marine organisms, Journal of Experimental Marine Biology and Ecology, 2016, pp. 112-118

Cui Y., Voyles R. M., Lane J. T., Mahoor M. H., ReFrESH: A Self-Adaptation Framework to Support Fault Tolerance in Field Mobile Robots, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 1576-1582

Diegel O., Badve A., Bright G., Potgieter J., Tlale S., Improved Mecanum Wheel Design for Omni-directional Robots, Australasian Conference on Robotics and Automation, 2002, pp. 117-121

Dixon W. E., Walker I. D., Dawson D. M., Fault Detection for Wheeled Mobile Robots with Parametric Uncertainty

Evaluationengineering (2016), Evaluationengineering website (Accession date:30.11.2016) www.evaluationengineering.com

Eugenio Y., Jose L. V., Jheifer P., Design of a brushed DC motors PID controller for development of low-cost robotic applications, Engineering Mechatronics and Automation (CIIMA), 2014.

Evrimteknik (2016), Evrimteknik website (Accession date: 30.11.2016) http://www.evrimteknik.com.tr/Assets/Documents/INCREMENTALENCODERGENETEKNIKAIKLAMALARI.pdf

Ge S. S., Cui Y. J., New Potential Functions for Mobile Robot Path Planning, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL16, 2000, pp. 615-620

Granosik G., Borenstein J., Minimizing Air Consumption of Pneumatic Actuators in Mobile Robots, International Conference on Robotics and Automation, 2004, pp. 3634-3639

Guzman F., Hugo V., Maneuvering Mechanisms for Omnidirectional Robots using Semi-circular Mecanum Wheels, 2015

Hashemi E., Jadidi M. G., Jadidi N.G., Model-based PI-fuzzy control of four-wheeled omni-directional mobile robots, Robotics and Autonomous Systems 59, 2011, pp. 930-942

Huang H., Tsai C., Adaptive Trajectory Tracking and Stabilization for Omnidirectional Mobile Robot with Dynamic Effect and Uncertainties, IFAC 2008, pp. 5383-5388

Fierro R., Lewis F. L., Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics, Journal of Robotic Systems, 1997, pp. 149-164

Fujimori A., Nikiforuk P. N., Gupta M. M., Adaptive Navigation of Mobile Robots with Obstacle Avoidance, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 13, 1997, pp. 596-602

Jennifer H., Teleoperation, CSC338

Kawamura S., Underwater Robot Development for Manipulation Task and their Uses in Biwa Lake, International Federation of Automatic Control IFAC, 2015, pp.14-19

Kullaa J., Detection, identification, and quantification of sensor fault, Proceedings of ISMA2010 Including USD2010, pp. 893-908

Lichiardopol S., A Survey on Teleoperation, DCT report, Technische Universiteit Eindhoven, 2007, pp. 1-34

Lui Y., Wu X., Zhu J. J., Lew J, Omni-Directional Mobile Robot Controller Design by Trajectory Linearization, 2003

Lui Y., Zhu J. J., Williams R. L., Wu J., Omni-directional mobile robot controller based on trajectory linearization, Robotics and Autonomous Systems, 2008, pp. 461-479

Maximintegrated (2016), Maximintegrated website (Accession date: 30.11.2016) https://www.maximintegrated.com/en/app-notes/index.mvp/id/5830)

Mnn (2016), Mnn website (Accession date: 30.11.2016) http://www.mnn.com/lifestyle/arts-culture/blogs/the-great-deep-sea-sub-race

Nakazato T., Wyl S. V., Osumi H., Dessimoz J. D., Control of Omni-Directional Mobile Platform with Four Driving Wheels, DARH2005 Conference, 2005

Niku S. B., An Introduction to Robotic Analysis, Systems, Applications, 2001

Normalesup (2016), Normalesup website (Accession date: 30.11.2016) http://www.normalesup.org/~pham/docs/affine

Pallegedara A., Udawatta L., Jayathilake D., Fault Tolerance Sensor fusion approach to Mobile Robot Navigation, Proceeding of the International Conference on Information and Automation, 2005, pp. 298-303

Pai N.S., Hsieh H. H., Lai Y. C., Implementation of Obstacle Avoidance Control for an Autonomous Omni-Directional Mobile Robot Based on Extension Theory, Sensors, 2012 pp. 13947-13963

Ren C., Ma S., Generalized Proportional Integral Observer based control of an omnidirectional mobile robot, 2015,

Rocco P., Control of industrial robots, Politecnico di Milano

Rec (2016), Rec website (Accession date: 30.11.2016) http://education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/1.html

Rojas R., Förster A. G., Holonomic Control of a robot with an omni-directional drive, Künstliche Intelligenz, 2006

Romero L., Concha A., Control of Position/Velocity in a Mobile Robot Using DC Brushless Motors, Robotics and Automotive Mechanics Conference, 2006

Rusu P., Petriu E. M., Whalen T. E., Cornell A., Spoelder H. J. W., Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation, IEEE Transactions on Instrumentation and Measurement, 2003, pp. 1335-1340

Oddgifts (2016), Oddgifts website (Accession date: 30.11.2016) http://www.oddgifts.com/products/dji-phantom-drone

Omrcen D., Zlajpah L., Nemec B., Combined Torque and Velocity Control of a Redundant Robot System, Mobile Robots, Moving Intelligence, 2006, pp. 576

Salih J. E. M., Rizon M., Yaacob Z., Designing Omni-Directional Mobile Robot with Mecanum Wheel, American Journal of Applied Sciences, 2006, pp. 1831-1835

Shilpa K., The Application of Wireless Local Area Network Technology to the Control of Mobile Robots, International Journal of Computer Science and Information Security, Vol. 3, No. 1, 2009

Sidenbladh H., Kragic D., Christensen H. I., A person Following Behavior for a Mobile Robot, International Conference on Robotics & Automation, 1999, pp. 670-675

Simpson J., Jaconsen C. L., and Jadud M. C., 2006, Mobile Robot Control The Subsumption Architecture, Communicating Process Architectures 2006, IOS Press 2006, pp. 225-236

Song J., Byun K., Design and Control of an Omnidirectional Mobile Robot with Steerable Omnidirectional Wheels, Mobile Robots, Moving Intelligence, 2006, pp. 576

SpaceX (2016), SpaceX website (Accession date: 30.11.2016) http://www.spacex.com/falcon9

Swisslog (2016), Swisslog website (Accession date: 30.11.2016) http://www.swisslog.com/en/Products/HCS/Automated-Material-Transport/RoboCourier-Autonomous-Mobile-Robot

Şahin O. N., Dede M. İ. C., Dört Tekerlekli Çok-Yönlü Uzaktan Kontrol Edilen bir Mobil Robot için Hata Telafisi, Uluslararası Katılımlı 17. Makina Teorisi Sempozyumu, 2015,

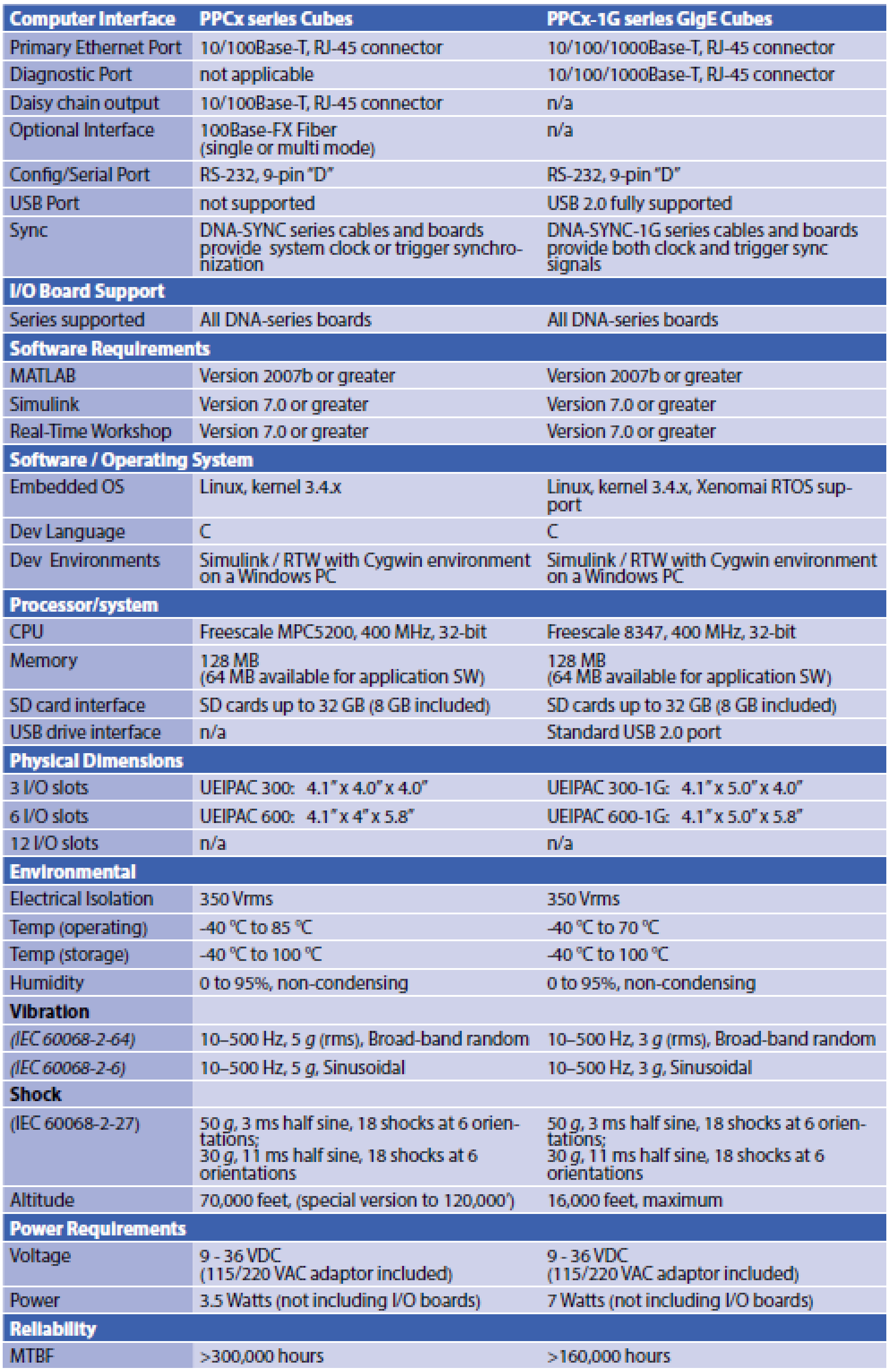
Tanaka J., Suzumori K., Takata M., Kanda T., Mori M., A mobile jack robot for rescue operation, IEEE, 2005

Wang T. Y., Tsai C. C., Wang D. A., Dynamic Control of An Omnidirectional Mobile Platform, Journal of Nan Kai, 2010, pp. 9-18

Watanebe K., Control of an Omnidirectional Mobile Robot, 1998 Second International Conference on Knowledge-Based Intelligent Electronic System, 1998, pp.51-60

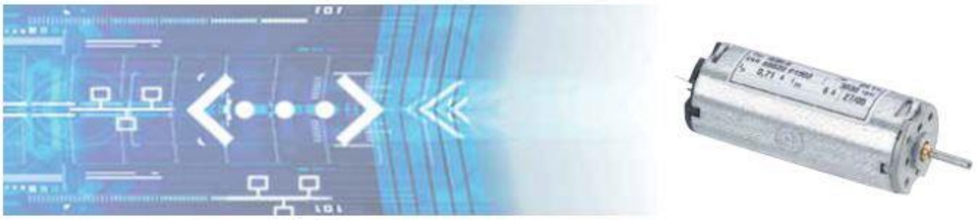
Yu J., Lan C., System Modelling and Robust Design of Microaccelerometer Using Piezoelectric Thin Film, International Conference on Multisensor Fusion and Integration for Intelligent Systems, 1999, pp. 99-104

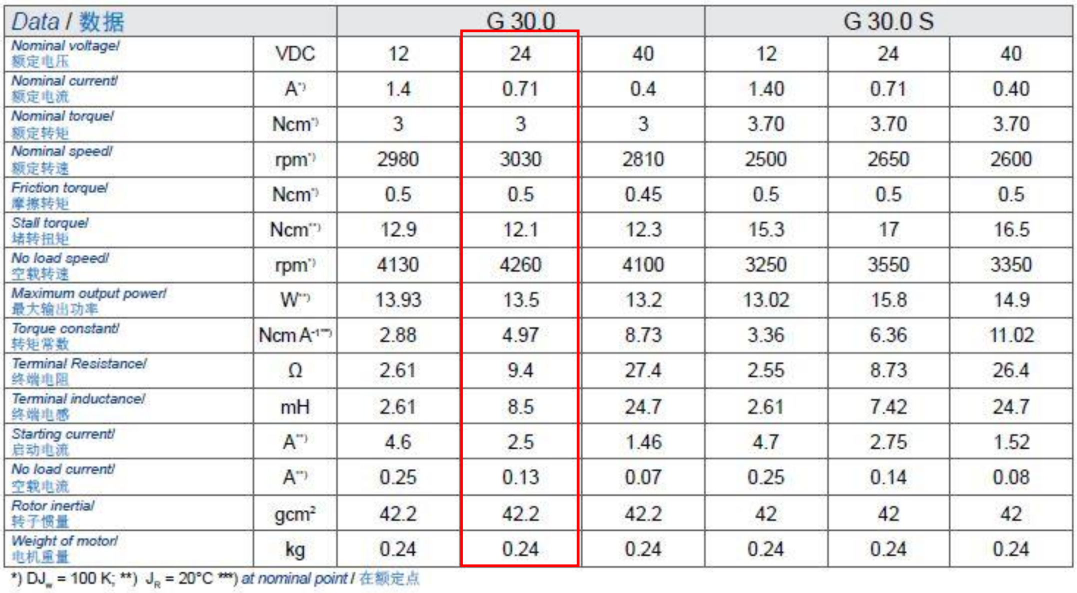
APPENDIX A-UEISIM TECHNICAL SPECIFICATIONS



Computer Interface	**PPCx series Cubes**	**PPCx-1G series GigE Cubes**
Primary Ethernet Port	10/100Base-T, RJ-45 connector	10/100/1000Base-T, RJ-45 connector
Diagnostic Port	not applicable	10/100/1000Base-T, RJ-45 connector
Daisy chain output	10/100Base-T, RJ-45 connector	n/a
Optional Interface	100Base-FX Fiber (single or multi mode)	n/a
Config/Serial Port	RS-232, 9-pin "D"	RS-232, 9-pin "D"
USB Port	not supported	USB 2.0 fully supported
Sync	DNA-SYNC series cables and boards provide system clock or trigger synchronization	DNA-SYNC-1G series cables and boards provide both clock and trigger sync signals
I/O Board Support		
Series supported	All DNA-series boards	All DNA-series boards
Software Requirements		
MATLAB	Version 2007b or greater	Version 2007b or greater
Simulink	Version 7.0 or greater	Version 7.0 or greater
Real-Time Workshop	Version 7.0 or greater	Version 7.0 or greater
Software / Operating System		
Embedded OS	Linux, kernel 3.4.x	Linux, kernel 3.4.x, Xenomai RTOS support
Dev Language	C	C
Dev Environments	Simulink / RTW with Cygwin environment on a Windows PC	Simulink / RTW with Cygwin environment on a Windows PC
Processor/system		
CPU	Freescale MPC5200, 400 MHz, 32-bit	Freescale 8347, 400 MHz, 32-bit
Memory	128 MB (64 MB available for application SW)	128 MB (64 MB available for application SW)
SD card interface	SD cards up to 32 GB (8 GB included)	SD cards up to 32 GB (8 GB included)
USB drive interface	n/a	Standard USB 2.0 port
Physical Dimensions		
3 I/O slots	UEIPAC 300: 4.1" x 4.0" x 4.0"	UEIPAC 300-1G: 4.1" x 5.0" x 4.0"
6 I/O slots	UEIPAC 600: 4.1" x 4" x 5.8"	UEIPAC 600-1G: 4.1" x 5.0" x 5.8"
12 I/O slots	n/a	n/a
Environmental		
Electrical Isolation	350 Vrms	350 Vrms
Temp (operating)	-40 °C to 85 °C	-40 °C to 70 °C
Temp (storage)	-40 °C to 100 °C	-40 °C to 100 °C
Humidity	0 to 95%, non-condensing	0 to 95%, non-condensing
Vibration		
(IEC 60068-2-64)	10–500 Hz, 5 *g* (rms), Broad-band random	10–500 Hz, 3 *g* (rms), Broad-band random
(IEC 60068-2-6)	10–500 Hz, 5 *g*, Sinusoidal	10–500 Hz, 3 *g*, Sinusoidal
Shock		
(IEC 60068-2-27)	50 *g*, 3 ms half sine, 18 shocks at 6 orientations; 30 *g*, 11 ms half sine, 18 shocks at 6 orientations	50 *g*, 3 ms half sine, 18 shocks at 6 orientations; 30 *g*, 11 ms half sine, 18 shocks at 6 orientations
Altitude	70,000 feet, (special version to 120,000')	16,000 feet, maximum
Power Requirements		
Voltage	9 - 36 VDC (115/220 VAC adaptor included)	9 - 36 VDC (115/220 VAC adaptor included)
Power	3.5 Watts (not including I/O boards)	7 Watts (not including I/O boards)
Reliability		
MTBF	>300,000 hours	>160,000 hours

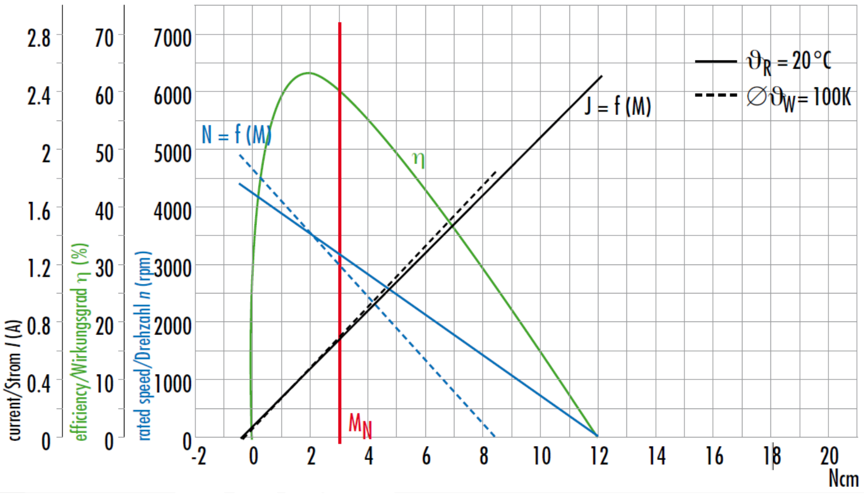
APPENDIX B- DUNKERMOTOREN TECHNICAL SPECIFICATIONS



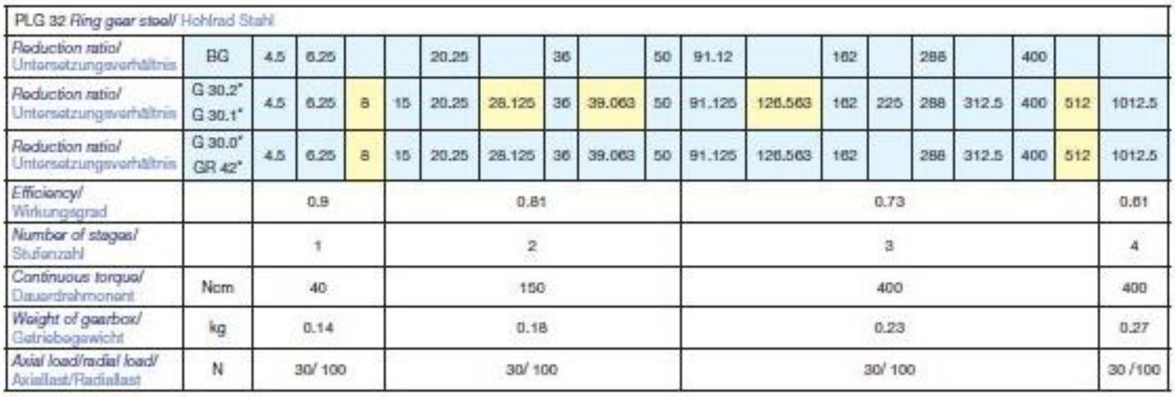


Data / 数据		G 30.0			G 30.0 S		
Nominal voltage/ 额定电压	VDC	12	24	40	12	24	40
Nominal current/ 额定电流	A*)	1.4	0.71	0.4	1.40	0.71	0.40
Nominal torque/ 额定转矩	Ncm*)	3	3	3	3.70	3.70	3.70
Nominal speed/ 额定转速	rpm*)	2980	3030	2810	2500	2650	2600
Friction torque/ 摩擦转矩	Ncm*)	0.5	0.5	0.45	0.5	0.5	0.5
Stall torque/ 堵转扭矩	Ncm**)	12.9	12.1	12.3	15.3	17	16.5
No load speed/ 空载转速	rpm*)	4130	4260	4100	3250	3550	3350
Maximum output power/ 最大输出功率	W**)	13.93	13.5	13.2	13.02	15.8	14.9
Torque constant/ 转矩常数	Ncm A^{-1}***)	2.88	4.97	8.73	3.36	6.36	11.02
Terminal Resistance/ 终端电阻	Ω	2.61	9.4	27.4	2.55	8.73	26.4
Terminal inductance/ 终端电感	mH	2.61	8.5	24.7	2.61	7.42	24.7
Starting current/ 启动电流	A**)	4.6	2.5	1.46	4.7	2.75	1.52
No load current/ 空载电流	A**)	0.25	0.13	0.07	0.25	0.14	0.08
Rotor inertial/ 转子惯量	gcm²	42.2	42.2	42.2	42	42	42
Weight of motor/ 电机重量	kg	0.24	0.24	0.24	0.24	0.24	0.24

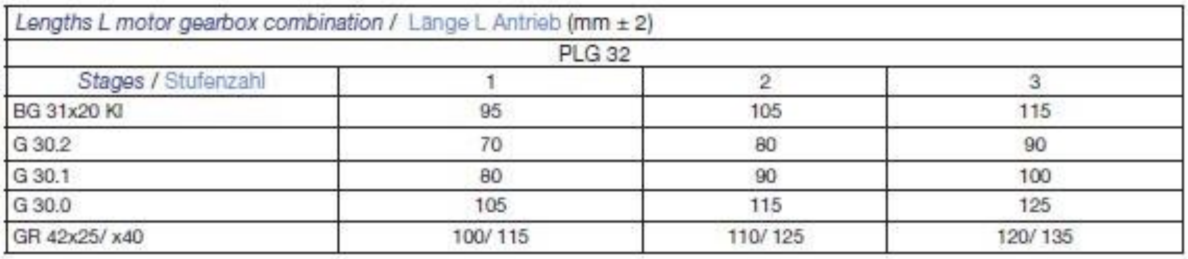
*) DJ_w = 100 K; **) J_R = 20°C ***) *at nominal point* / 在额定点



_ Data
Leistungsdaten

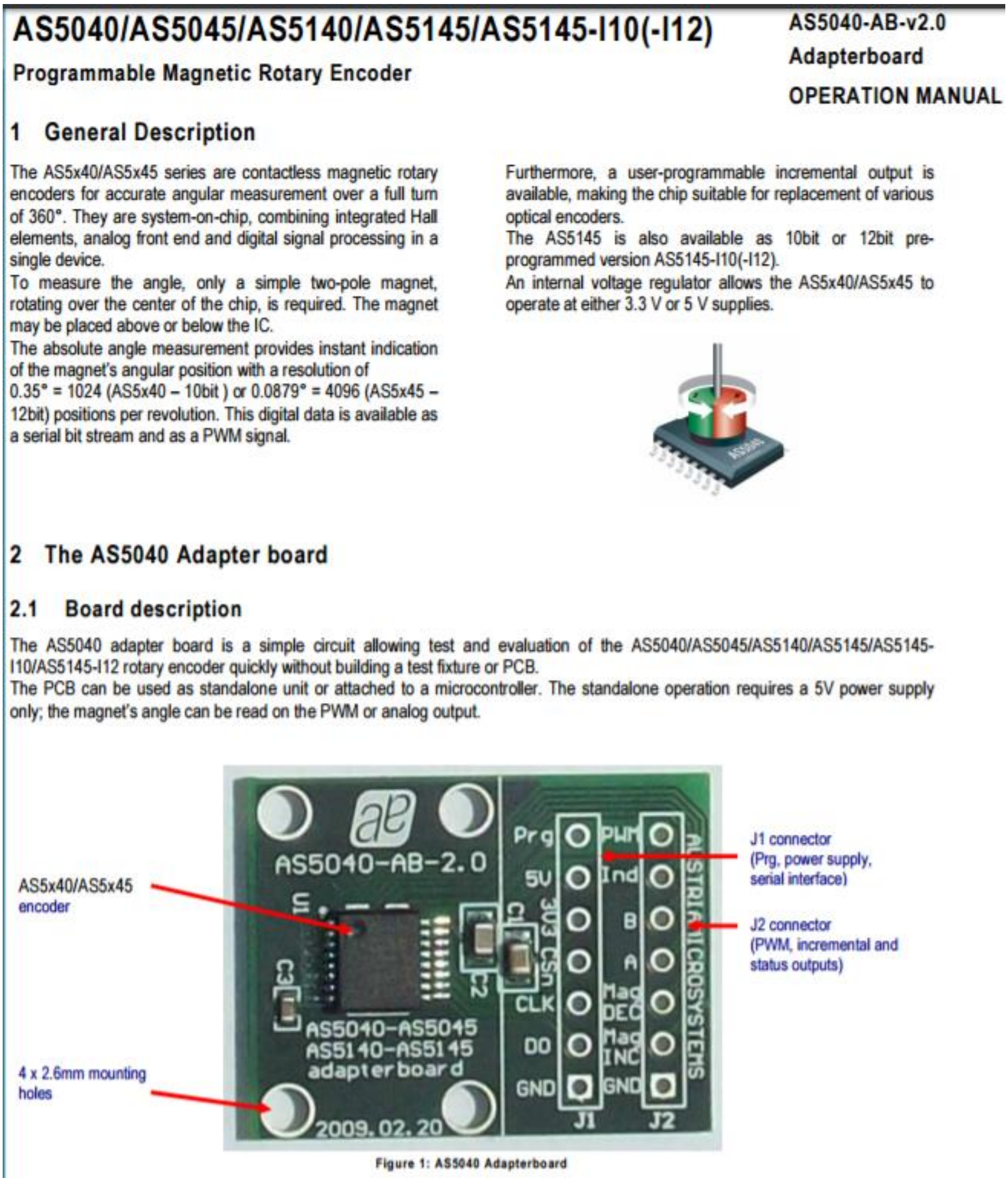


PLG 32 *Ring gear steel/* Hohlrad Stahl																			
Reduction ratio/ Untersetzungsverhältnis	BG	4.5	6.25			20.25		36		50	91.12		162		288		400		
Reduction ratio/ Untersetzungsverhältnis	G 30.2* G 30.1*	4.5	6.25	8	15	20.25	28.125	36	39.063	50	91.125	126.563	162	225	288	312.5	400	512	1012.5
Reduction ratio/ Untersetzungsverhältnis	G 30.0* GR 42*	4.5	6.25	8	15	20.25	28.125	36	39.063	50	91.125	126.563	162		288	312.5	400	512	1012.5
Efficiency/ Wirkungsgrad		0.9			0.81						0.73								0.61
Number of stages/ Stufenzahl		1			2						3								4
Continuous torque/ Dauerdrehmoment	Ncm	40			150						400								400
Weight of gearbox/ Getriebegewicht	kg	0.14			0.18						0.23								0.27
Axial load/radial load/ Axiallast/Radiallast	N	30/ 100			30/ 100						30/ 100								30 /100



Lengths L motor gearbox combination / Länge L Antrieb (mm ± 2)			
	PLG 32		
Stages / Stufenzahl	1	2	3
BG 31x20 KI	95	105	115
G 30.2	70	80	90
G 30.1	80	90	100
G 30.0	105	115	125
GR 42x25/ x40	100/ 115	110/ 125	120/ 135

APPENDIX C- MAGNETIC ENCODER DATASHEET



AS5040/AS5045/AS5140/AS5145/AS5145-I10(-I12)

Programmable Magnetic Rotary Encoder

AS5040-AB-v2.0
Adapterboard
OPERATION MANUAL

1 General Description

The AS5x40/AS5x45 series are contactless magnetic rotary encoders for accurate angular measurement over a full turn of 360°. They are system-on-chip, combining integrated Hall elements, analog front end and digital signal processing in a single device.

To measure the angle, only a simple two-pole magnet, rotating over the center of the chip, is required. The magnet may be placed above or below the IC.

The absolute angle measurement provides instant indication of the magnet's angular position with a resolution of 0.35° = 1024 (AS5x40 – 10bit) or 0.0879° = 4096 (AS5x45 – 12bit) positions per revolution. This digital data is available as a serial bit stream and as a PWM signal.

Furthermore, a user-programmable incremental output is available, making the chip suitable for replacement of various optical encoders.

The AS5145 is also available as 10bit or 12bit pre-programmed version AS5145-I10(-I12).

An internal voltage regulator allows the AS5x40/AS5x45 to operate at either 3.3 V or 5 V supplies.

2 The AS5040 Adapter board

2.1 Board description

The AS5040 adapter board is a simple circuit allowing test and evaluation of the AS5040/AS5045/AS5140/AS5145/AS5145-I10/AS5145-I12 rotary encoder quickly without building a test fixture or PCB.

The PCB can be used as standalone unit or attached to a microcontroller. The standalone operation requires a 5V power supply only; the magnet's angle can be read on the PWM or analog output.

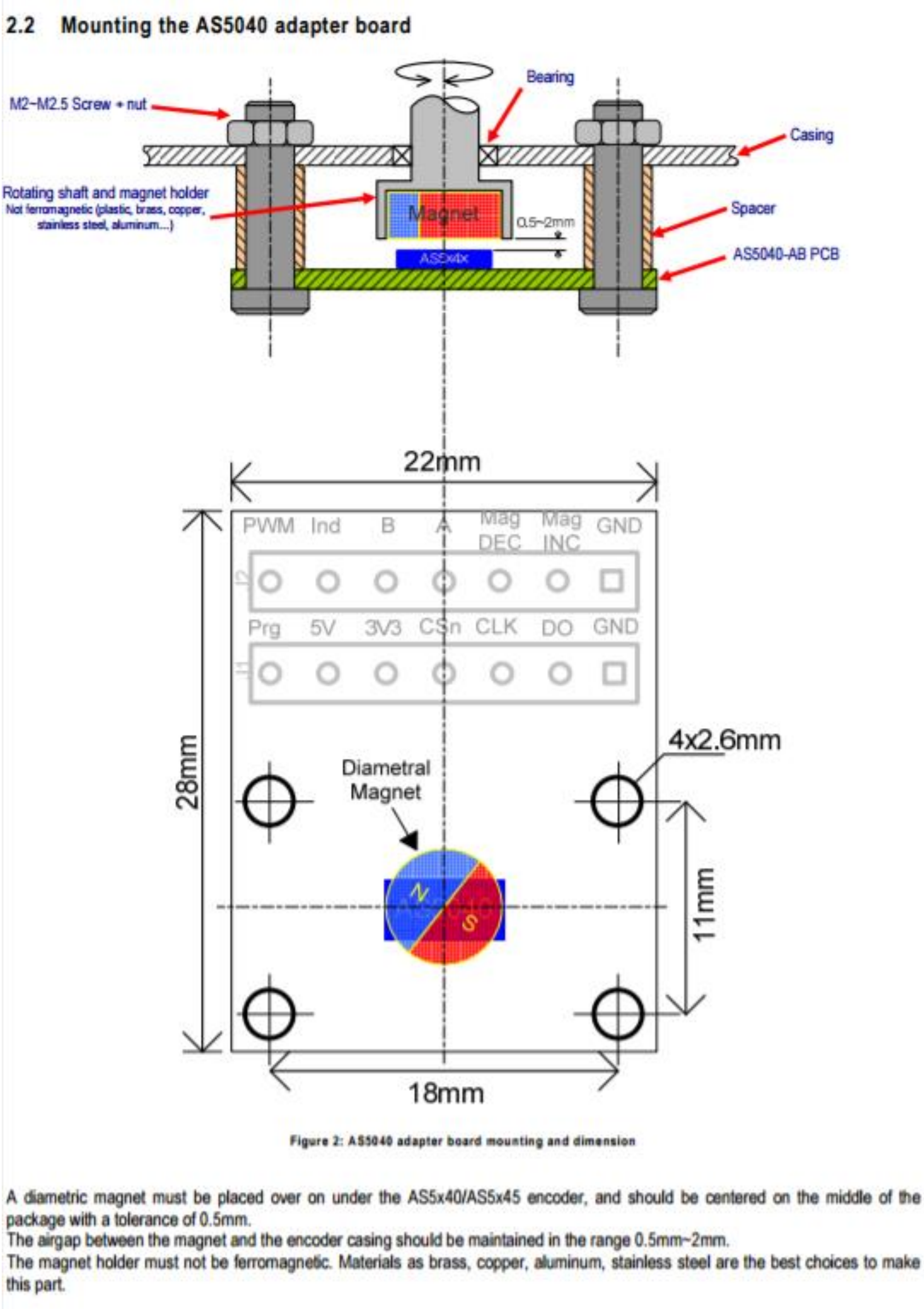
AS5x40/AS5x45 encoder

4 x 2.6mm mounting holes

J1 connector (Prg, power supply, serial interface)

J2 connector (PWM, incremental and status outputs)

Figure 1: AS5040 Adapterboard



2.2 Mounting the AS5040 adapter board

Figure 2: AS5040 adapter board mounting and dimension

A diametric magnet must be placed over on under the AS5x40/AS5x45 encoder, and should be centered on the middle of the package with a tolerance of 0.5mm.

The airgap between the magnet and the encoder casing should be maintained in the range 0.5mm~2mm.

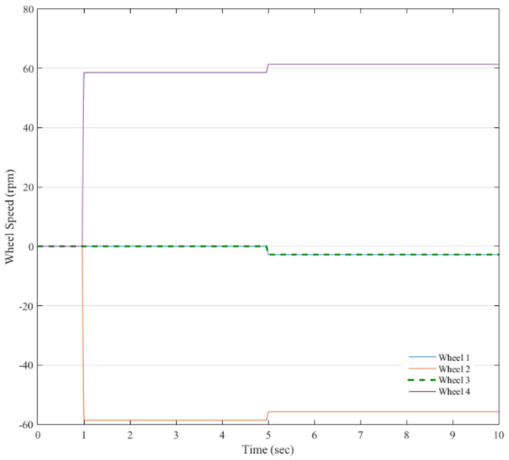
The magnet holder must not be ferromagnetic. Materials as brass, copper, aluminum, stainless steel are the best choices to make this part.

APPENDIX D- MOTOR DRIVER TECHNICAL SPECIFICATIONS

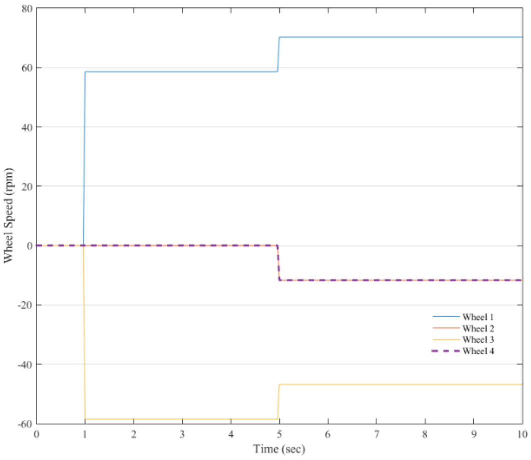
Specifications of Maxon ADS 50/10 motor amplifier

Electrical Data	
Nominal supplyy voltage +V_{cc}	12 to 50 VDC
Absolute minimum supply voltage +$V_{cc\ min}$	11.4 VDC
Absolute maximum supply voltage +$V_{cc\ max}$	52.5 VDC
Max. output voltage	0.9 . V_{cc}
Max. output current I_{max}	20A
Continuous output current I_{cont}	10A
Switching frequency	50kHz
Efficiency	95%
Band width current controller	2.5 kHz
Built in motor choke	75µH / 10A
Inputs	
Set value: -10 to +10V (R_i=20kΩ)	-10 to +10V
Enable	+4 to +50VDC
Input voltage DC tacho “Tacho Input”	Min. 2VDC, max 50 VDC
Encoder signals “channel A, A\, B, B\”	Max. 100kHz, TTL level
Output:	
Current monitor “Monitor I”	-10 ... +10VDC (R_O=100Ω)
Speed monitor “Monitor n”	-10 ... +10VDC (R_O=100Ω)
Status reading “READY” Open collector, short-circuit protected	Max. 30 VDC ($I_L \leq$20mA)
Voltage outputs:	
Aux. Voltage, short-circuit protected	+12 VDC, -12 VDC, max. 12 mA (R_O= 1kΩ)
Encoder supply voltage	+5 VDC, max 80 mA

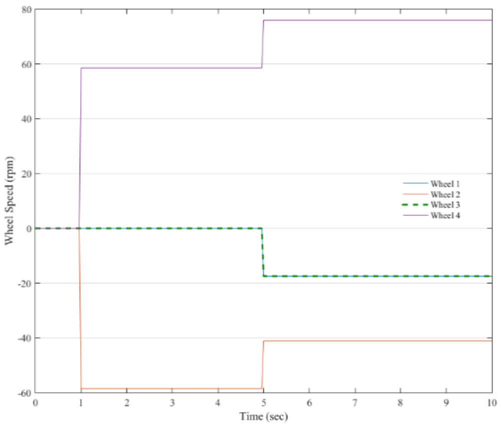
APPENDIX E – FAULT TOLERENCE SIMULATION AND EXPERIMENT RESULTS



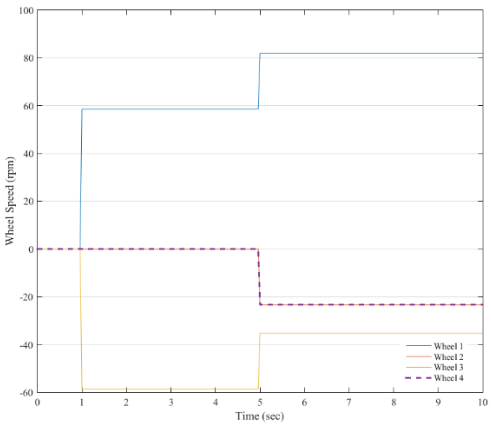
(a)



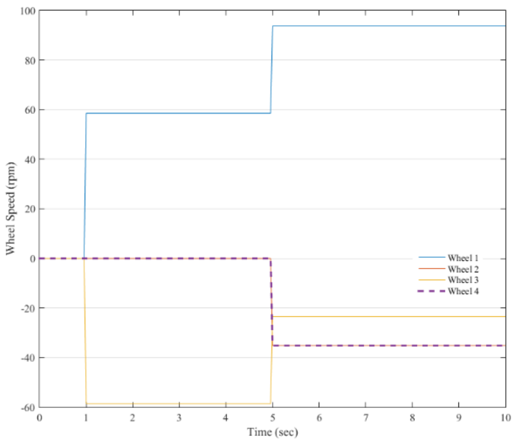
(b)



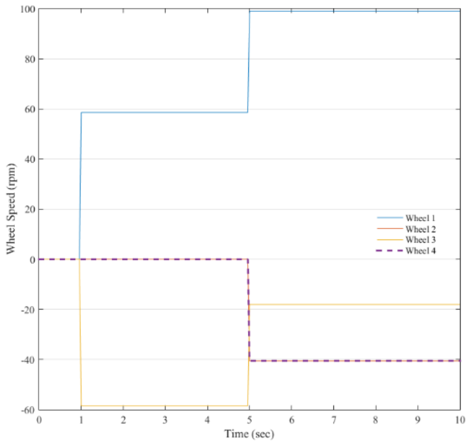
(b)



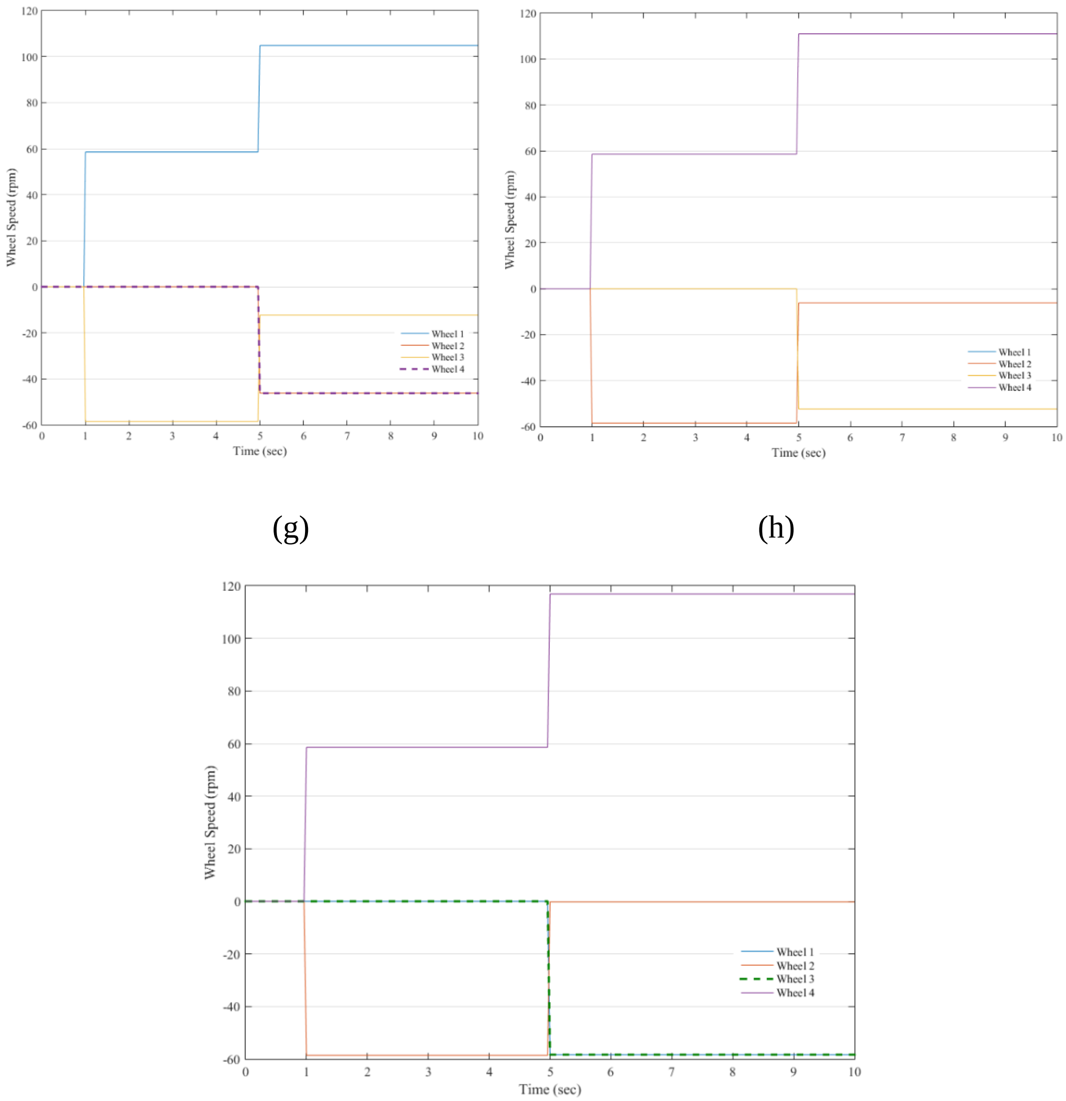
(d)



(e)



(f)

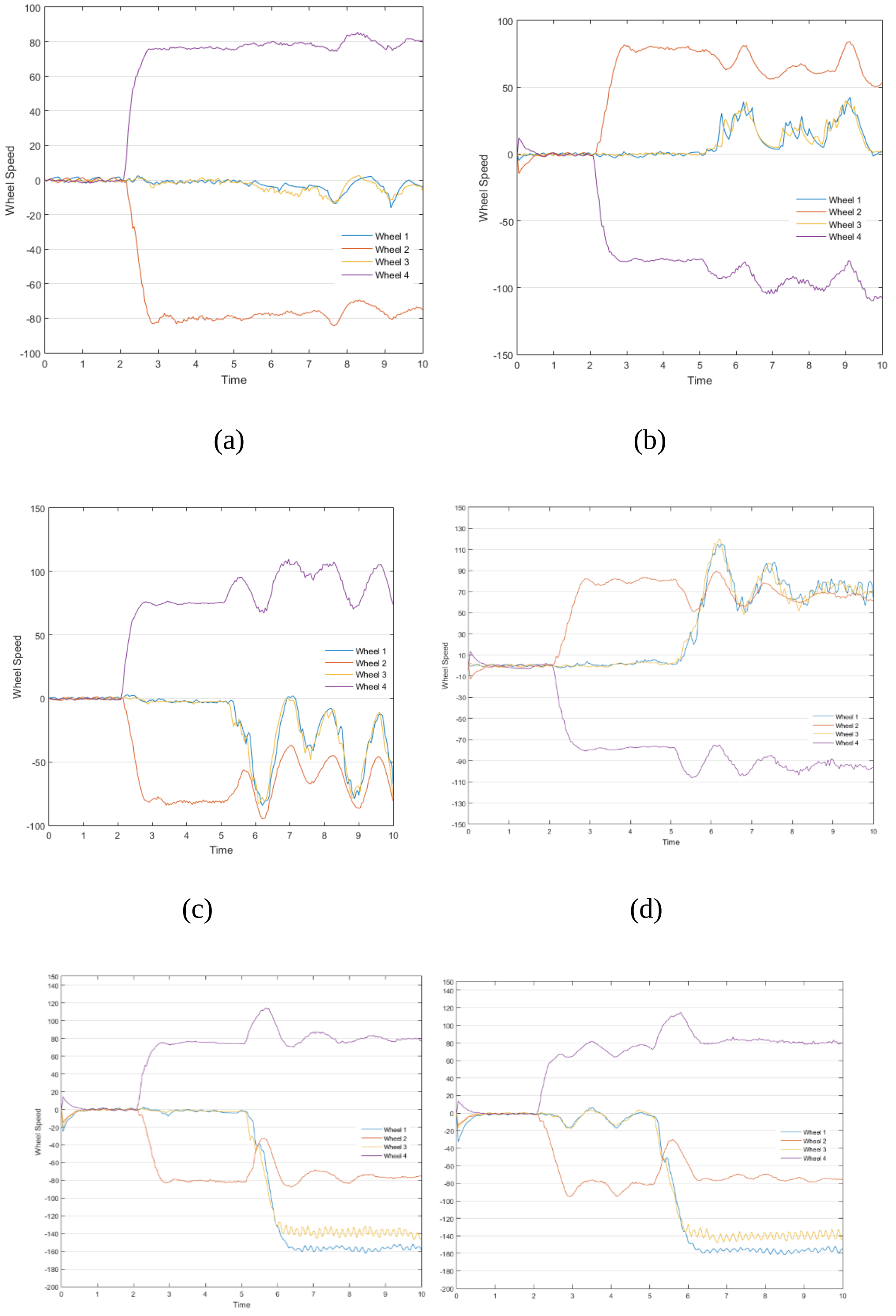


(g)

(h)

(i)

Figure E1. Simulation Results for performance loss (a)-%10, (b)-%20, (c)-%30, (d)-%40, (e)-%60, (f)-%70, (g)-%80, (h)-%90, (i)-%100

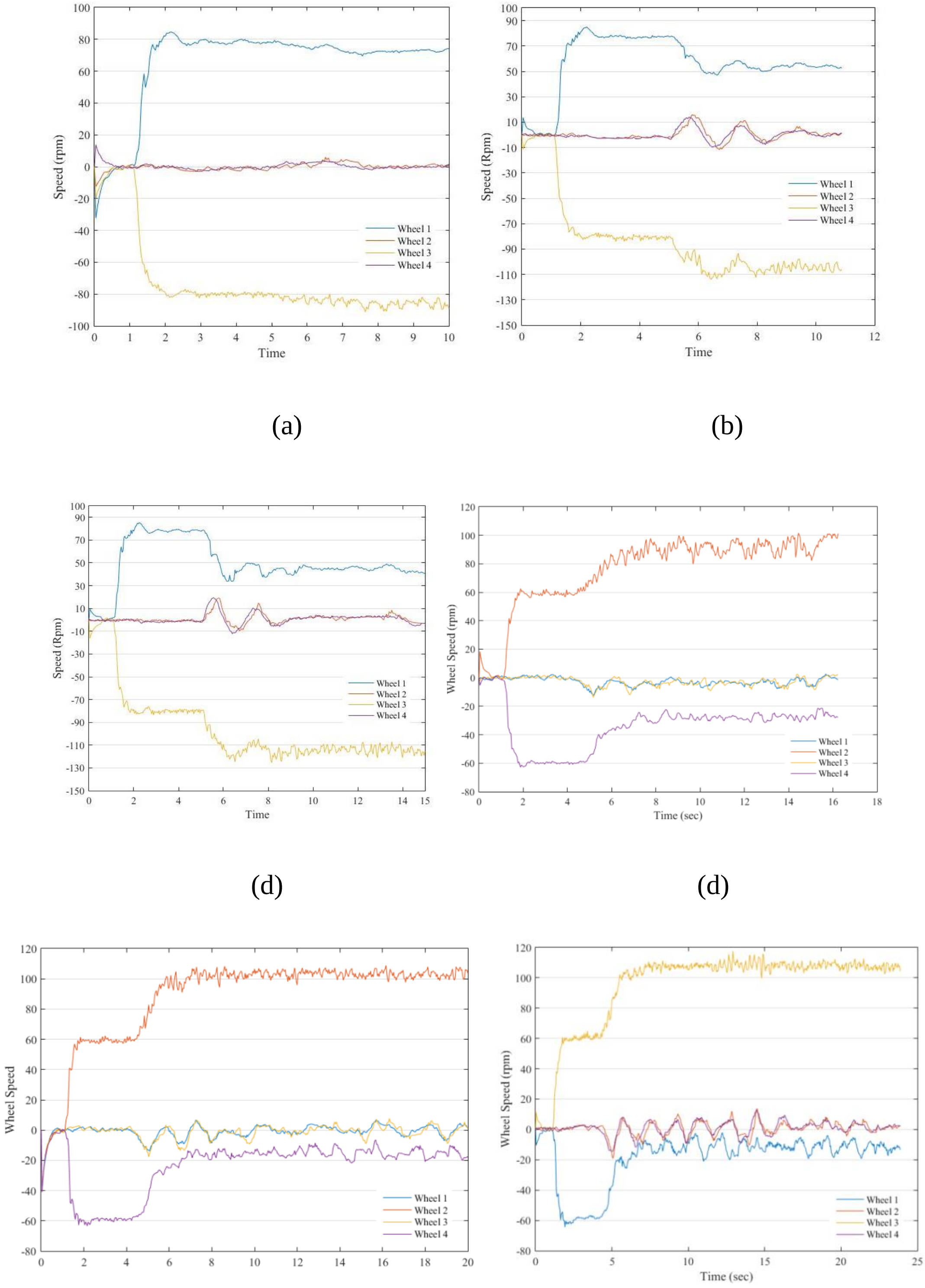


(a) (b)

(c) (d)

(e) (f)

Figure E2. Experimental study results in x direction for (a)-%10, (b)-%20, (c)-%30 (d)-%40, (e)-%60, (f)-%70 performance loss



(a)

(b)

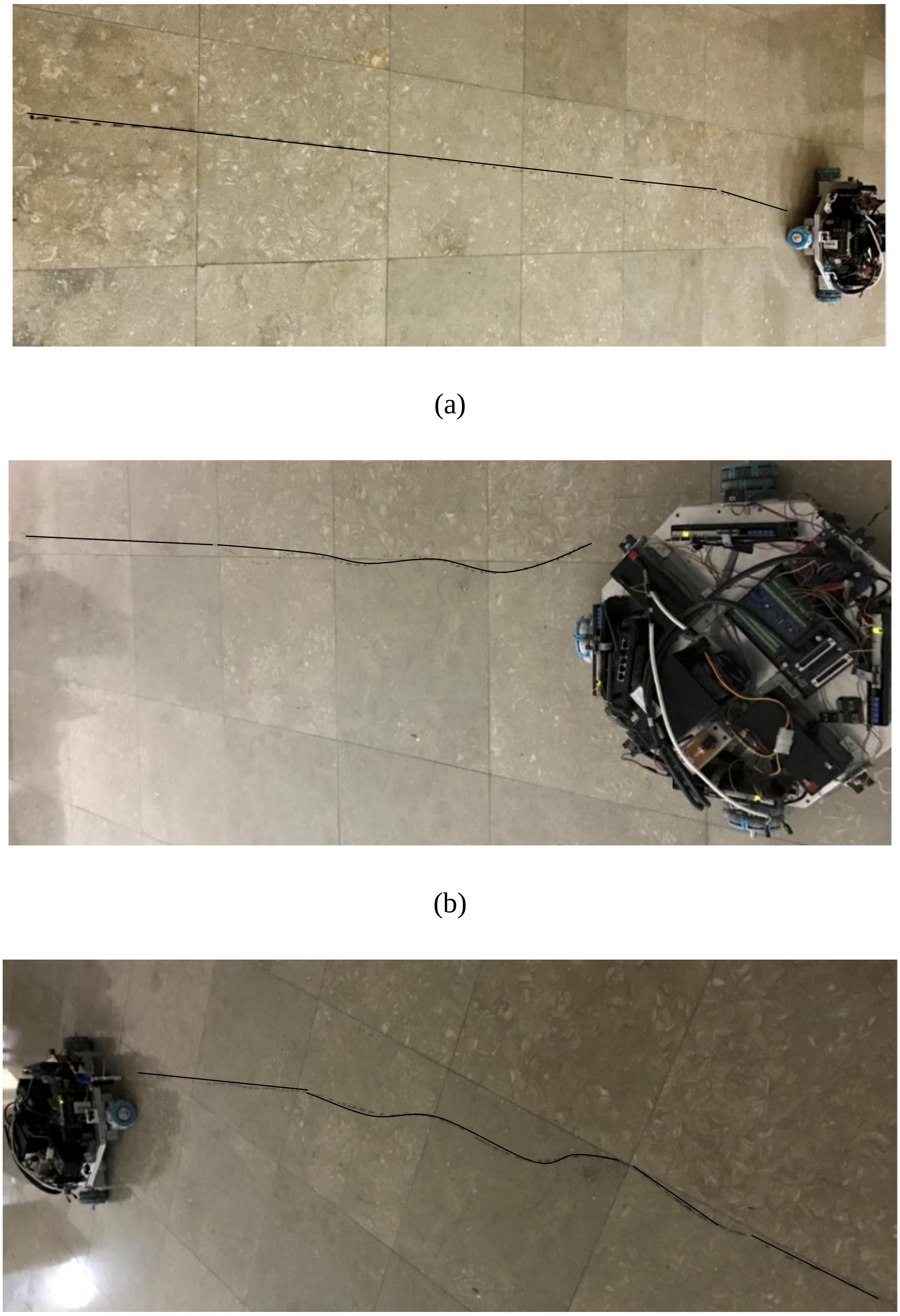
(d)

(d)

(e)

(f)

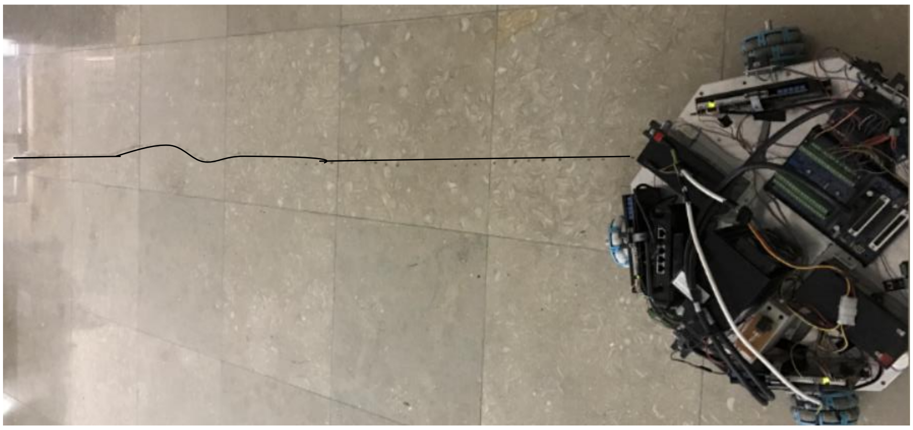
Figure E3. Experimental study results in y direction for (a)-%10, (b)-%20, (c)-%30, (d)-40, (e)-%60, (f)-%70 performance loss



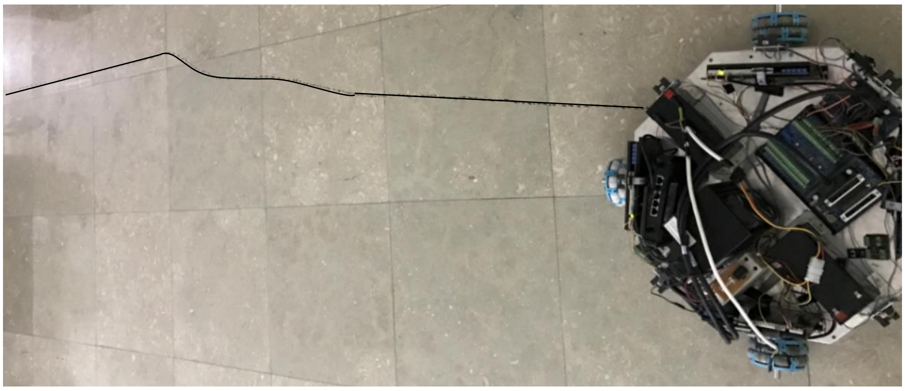
(a)

(b)

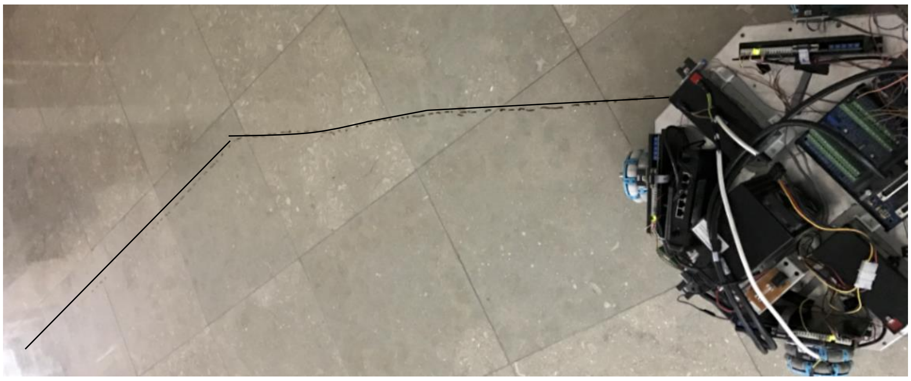
(c)



(d)

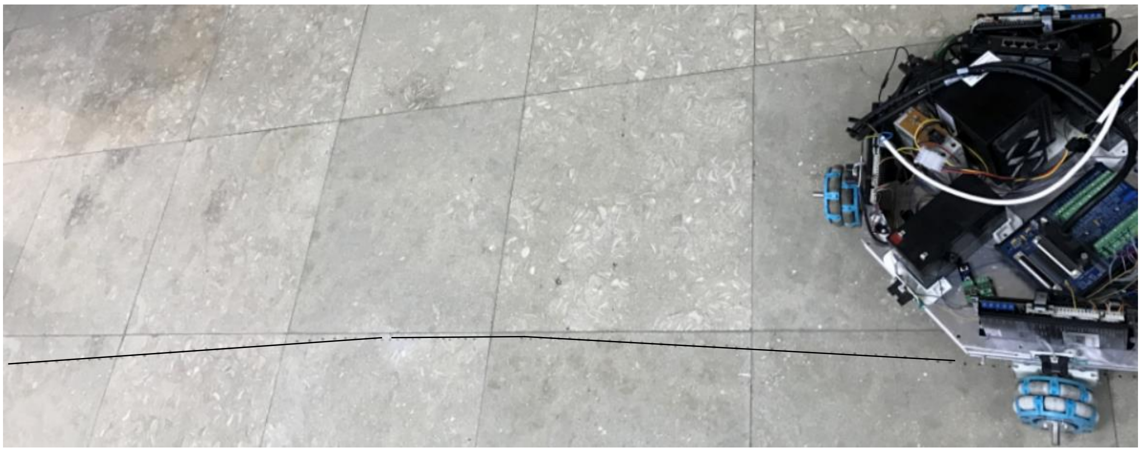


(e)

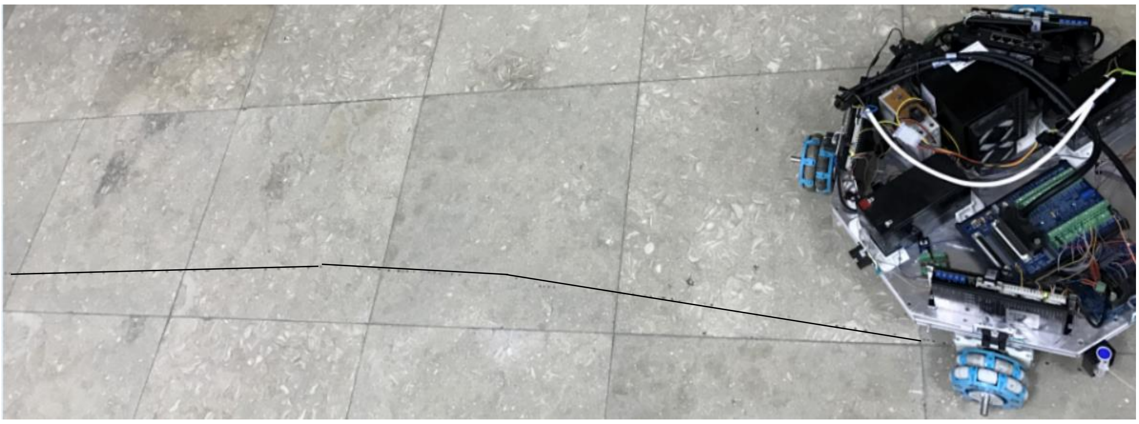


(f)

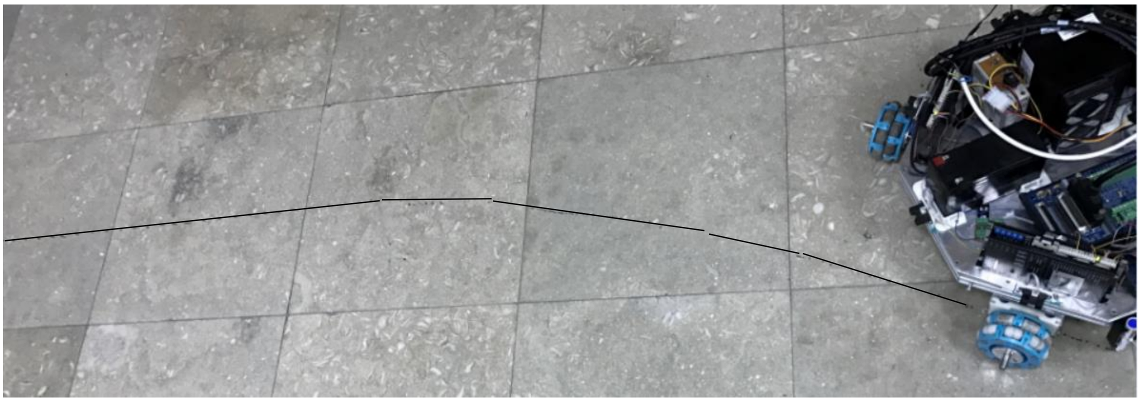
Figure E4 Robot following path in X direction (a)-%10, (b)-%20, (c)-%30, (d)-%40, (e)-%60, (f)-%70



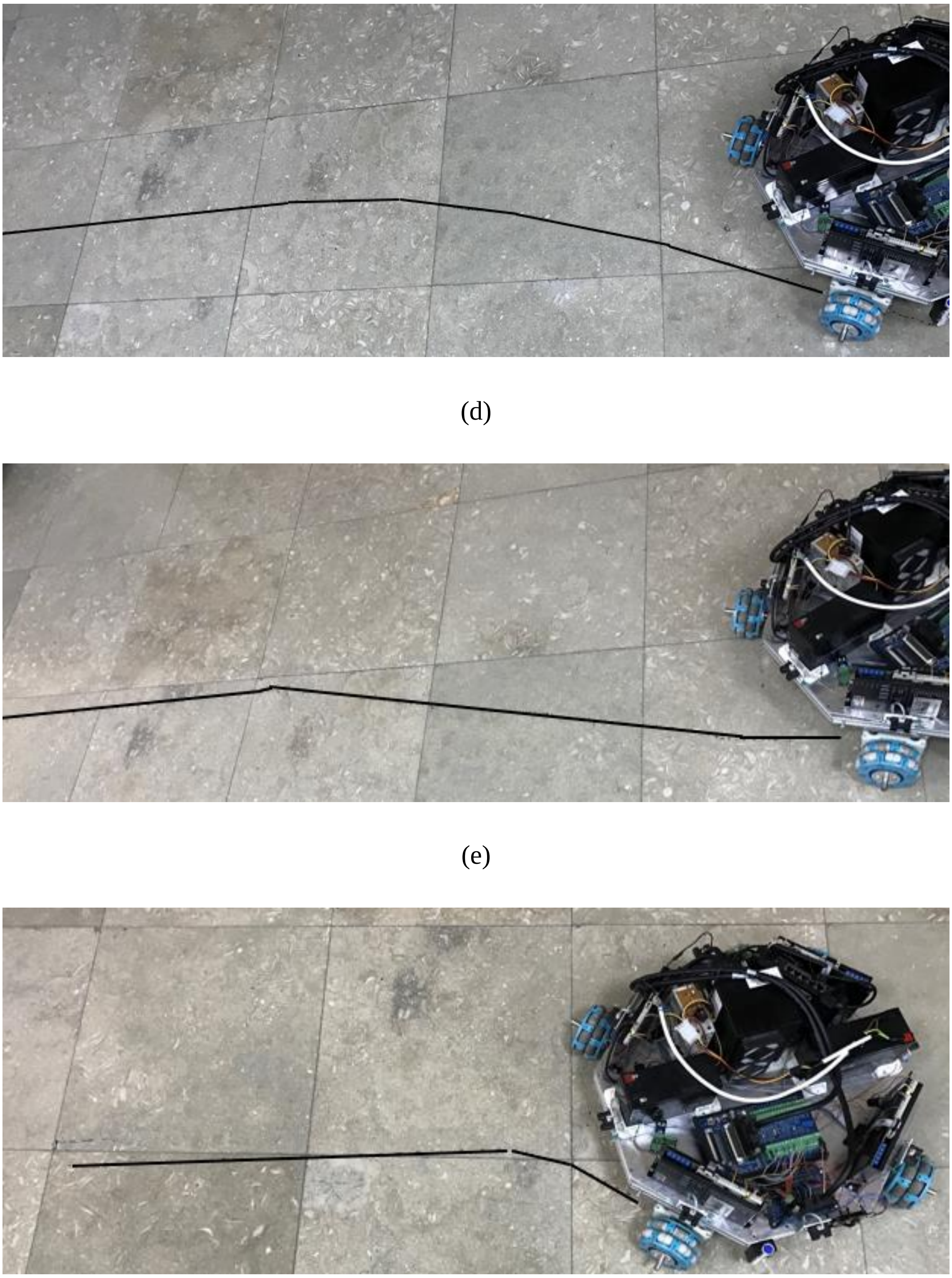
(a)



(b)



(c)



(d)

(e)

(f)

Figure E5 60 Robot following path in Y direction (a)-%10, (b)-%20, (c)-%30, (d)-%40, (e)-%60, (f)-%70