ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

DESIGN AND CONTROL OF AN AUTONOMOUS ELECTRICAL VEHICLE FOR INDOOR TRANSPORT APPLICATIONS

M.Sc. THESIS

Şükrü Yaren GELBAL

Department of Mechatronics Engineering

Mechatronics Engineering Programme

SEPTEMBER 2016

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE
ENGINEERING AND TECHNOLOGY

DESIGN AND CONTROL OF AN AUTONOMOUS ELECTRICAL VEHICLE FOR INDOOR TRANSPORT APPLICATIONS

M.Sc. THESIS

Şükrü Yaren GELBAL
(518141012)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Erdinç ALTUĞ

SEPTEMBER 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İÇ MEKANDA TAŞIMA UYGULAMALARINA YÖNELİK ELEKTRİKLİ OTONOM ARAÇ TASARIMI VE KONTROLÜ

YÜKSEK LİSANS TEZİ

Şükrü Yaren GELBAL
(518141012)

Mekatronik Mühendisliği Ana Bilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Erdinç ALTUĞ

EYLÜL 2016

Şükrü Yaren GELBAL, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 518141012 successfully defended the thesis entitled "DESIGN AND CONTROL OF AN AUTONOMOUS ELECTRICAL VEHICLE FOR INDOOR TRANSPORT APPLICATIONS", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Assoc. Prof. Dr. Erdinç ALTUĞ** ....
Istanbul Technical University

Jury Members : **Prof. Dr. Şeniz ERTUĞRUL** ....
Istanbul Technical University

Prof. Dr. Haluk KÜÇÜK ....
Marmara University

Date of Submission : **6 September 2016**
Date of Defense : **7 September 2016**

To my family,

FOREWORD

First, I would like to thank my advisor Assoc. Prof. Dr. Erdinç Altuğ for his continuous support. His valuable comments and remarks with the motivation he provided, helped me to take firm and faster steps forward.

Further, I would like to thank to TÜBİTAK for their financial support to this project, project number 115E345.

Finally, I would like to thank my family for their love and the support. This study is dedicated to them.

September 2016 Şükrü Yaren GELBAL
(Mechatronics Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

IPS : Indoor Positioning System
AGV : Automated Guided Vehicle
MABX2 : Microautobox 2
RS232 : Recommended Standard 232
LIDAR : Light Detection and Ranging
PC : Personal Computer
ECU : Engine Control Unit
GPS : Global Positioning System
OS : Operating System
RC : Radio Control
RTI : Real Time Interface
CAN : Controller Area Network
RPM : Revolution Per Minute
PID : Proportional Integral Derivative
PWM : Pulse Width Modulation
HIL : Hardware in the Loop
UDP : User Datagram Protocol
USB : Universal Serial Bus
SLAM : Simultaneous Localization and Mapping
FOV : Field of View
FRP : Freezing Robot Problem
HIL : Hardware in the Loop

SYMBOLS

u_1 : Forward velocity of the vehicle
l : Wheel base length of the vehicle
θ : Vehicle heading
φ : Steering angle
x, y : Position of the vehicle on horizontal and vertical axes
$U_{\text{att}}, U_{\text{rep}}$: Attractive and repulsive potential fields
$F_{\text{att}}, F_{\text{rep}}$: Attractive and repulsive forces acting on vehicle
ξ, η : Coefficients for attractive and repulsive forces
q : Position of the vehicle
q_0 : Position of the potential field source
d : Distance between vehicle and the potential field source
d_0 : Critical distance

LIST OF FIGURES

DESIGN AND CONTROL OF AN AUTONOMOUS ELECTRICAL VEHICLE FOR INDOOR TRANSPORT APPLICATIONS

SUMMARY

As the need of intelligent vehicles on our roadways emerges, there is an equally important need emerges as well: The need of intelligent vehicles on areas such as university campuses, airports or shopping malls. These intelligent vehicles can help elderly, disabled, or people with heavy luggage. This thesis describes an intelligent vehicle that can be used indoor areas where pedestrians exist. The vehicle is planned to carry luggages and transport humans.

Vehicle used is an electric golf cart, considering the significant advantages of less noise, no toxic gas emission and higher maneuverability. Firstly, vehicle is modified for unmanned drive. Drivers are added to control actuators on steering wheel and brake pedal. Then, main controller, dSpace MABX2 is placed. This device runs a MATLAB simulink model embedded in itself. While running, this model communicates with real world through input and output pins on the device, which are related to RTI blocks placed inside simulink model. Controllers are constructed in this simulink model and actuators were ready to control by connecting this in/out pins to related elements with cable. Other than the main controller, a separate controller, an Arduino board is used for braking, for emergency purposes. If an emergency situation occurs, if brake signal is cut off from main controller or if button on the related RC transmitter is pressed, this controller applies full braking independent of the main controller. PID controllers are preferred for steering wheel, brake and throttle unmanned drive subsystems.

Indoor positioning is one of the most important problems when it comes to autonomous vehicles.There are studies proposing several computer vision based, wireless signal based etc. methods. Most accurate method is (IPS) but it is costly to set up and because of wireless signals gets weaker while passing through walls, it is not the best solution for every indoor environment. In this study, an encoder is used as main sensor for calculating position. Error caused by tire slip is very small because of the flat surface and slow move speed of the vehicle. But because of the error being accumulative, on long distance travel, real position and calculated position differ slightly. A computer vision based method similar to landmarking could be implemented in future phases to correct this difference.

Environment identification and decision making is necessary for autonomous drive. For detecting obstacles and pedestrians in front of the vehicle, a LIDAR sensor is used. 3d cloud data consisting of 4 plane, can be obtained from this sensor. With the 4 plane LIDAR sensor used, it is possible to separate pedestrians from static obstacles and measure their movement speed. A second 1 plane scanning LIDAR with wide scan angle added to detect objects falling out of the 4 plane LIDAR scan angle, for the purpose of achieving more stable and safer obstacle avoidance. Avoiding obstacles is

first priority for the vehicle. Some path following algorithms had been experimented on. General path following logic is based on goal points. To travel between two destinations in a known map, vehicle is given a number of goal points in proper order. Vehicle follows this points using implemented path following algorithm until the last goal point is reached. Last goal point means vehicle arrived the destination.

On first experimental path following algorithm, vehicle calculates error of heading between itself and the goal point and rotates towards goal point by selecting the shortest direction, using a control logic. Moreover, vehicle constantly checks if the goal point is in vehicle's minimum turning radius. If it is, vehicle will never be able to reach it while trying to rotate towards it. Instead, vehicle maneuvers to opposite direction until the point is out of the minimum turning radius. Then rotates towards it. Second and final experiment is potential field method. A method including both path following and obstacle avoidance behaviors. Calculating pushing forces proportional to distances from objects in front of the vehicle and pulling force proportional to distance from next goal point, vehicle is able to maneuver between obstacles and reach the point.

In this thesis, various stages of design and production of an autonomous vehicle, which is planned to operate in indoor environment where pedestrians exists, is explained. Sensors and mechatronic systems used for unmanned drive were presented, hardware and software used for control are discussed. Moreover, algorithms used for the vehicle to travel autonomously and and sensors used for receiving environmental data are explained. Finally, the real world driving tests performed are shown and the results were discussed.

İÇ MEKANDA TAŞIMA UYGULAMALARINA YÖNELİK ELEKTRİKLİ OTONOM ARAÇ TASARIMI VE KONTROLÜ

ÖZET

Günümüzde, sensörlerin ve işlem gücü yüksek cihazların kolay üretilebilir ve ulaşılabilir olması sayesinde, mekatronik ile ilgili birçok alanda, insanların hayatını kolaylaştıracak araştırmalar hızlanmıştır. Sürücüsüz araçlar da bu alanların en çok ilgi görenlerindendir. Sürücüsüz hava araçları, askeri faaliyetler, hafif eşyaların taşınması, afetlerde alan keşifleri vb. konularda görev almaktadır. Sürücüsüz kara araçları ise, gelecekte hem askeri alanda kullanılabilecek, hem de sivil düşünüldüğünde, insan faktöründen kaynaklanan kazaları sıfıra indirebilecek ve gerek insan, gerek yük ve eşya taşınmasını, ulaşımı oldukça kolaylaştıracaktır.

Sürücüsüz araçlar arasında en yaygın araştırmaları bulunan çeşit, karayollarında ilerleyebilecek, trafik içinde hareket edebilecek, uzun mesafede insan taşıyabilecek araçlardır. Bu araçların geliştirilmesinde genellikle standart otomobiller modifiye edilerek kullanılmakta, bu otomobillere çeşitli mekatronik sistemler ve sensörler entegre edilerek, sürücüsüz hareket edebilecek hale getirilmektedir. Bunun yanında, son birkaç yılda, alışveriş merkezleri, havaalanları gibi geniş alanları kullanan insan sayısının artması sebebiyle, bu alanlarda insanların gidecekleri yerleri rahat bulabilmesi için kolay ulaşılabilir sanal haritalar, rehber robotlar gibi ürünler ortaya çıkmıştır. Dolayısıyla, bu konuya yönelik olan iç mekanda insan taşıyan otonom araç ilgili çalışmalar da önem kazanmıştır. Bu tezde, havaalanları, alışveriş merkezleri gibi yayaların yoğun olarak bulunduğu iç mekanlarda kullanılabilecek bir otonom aracın tasarımı anlatılmaktadır.

Araç iç mekanda çalışacağı için, zararlı gazlar açığa çıkaran ve gürültü kirliliğine yol açan benzinli araçlar yerine, elektrikli bir araç tercih edilmiştir. Manevra kabiliyetinin yüksek olmasına gerek duyulduğundan, boyutları küçük bir golf aracı tercih edilmiştir. Öncelikle golf aracı, sürücüsüz hareket edebilmesi için modifiye edilmiş, direksiyon ve fren pedalına daha önce yerleştirilen ve bunları fiziksel olarak hareket ettiren çeşitli mekanik aktüatörleri kontrol edecek sürücüler yerleştirilmiştir. Aracı hızlandırmak için ise aracın motor kontrolünü yapan ECU ünitesine analog gerilim olarak sinyal verilmesi gerekmektedir. Daha sonra, bu sürücülere ve ECU'ya referans sinyali gönderecek olan ana kontrolcü yerleştirilmiştir.

Ana kontrolcü olarak, kullanım kolaylığı ve güvenilirliği açısından, otonom araçlar önde gelmek üzere birçok mekatronik araştırmada yaygın olarak kullanılan, dSpace MABX2 tercih edilmiştir. Simulink ile, MABX2'nin simulink için geliştirdiği RTI blokları kullanılarak bir tümleşik model hazırlanıp, cihaza gömülmektedir. Cihaz çalışırken bu simulink modelini sürekli olarak koşturmakta, modeldeki bloklarla ilişkili giriş ve çıkış pinlerinden, gerçek dünya ile sinyal alışverişi yapmaktadır. Bu simulink modeli üzerinden kapalı çevrim kontrolcüler oluşturulup, sensörlerden gelen geri besleme sinyalleri ile sürücülere gidecek olan referans sinyallerini taşıyan kablolar,

cihaza uygun şekilde bağlanarak kontrol sağlanmaktadır. Ayrıca, ana kontrolcüye ek olarak, fren sistemi için güvenlik amaçlı bir kontrolcü daha yerleştirilmiştir. Bu kontrolcü için Arduino kart kullanılmış, ana kontrolcüden sinyal gelmediği zamanlarda frene basacak şekilde ayarlanmıştır. Bunun yanında bir de kablosuz alıcı bağlanmış, acil bir durumda, uzaktan kumandadan ilgili düğmeye basıldığında, ana kontrolcüden bağımsız olarak fren pedalına tamamen basılmasını sağlamaktadır.

İnsansız sürüş için kullanılan direksiyon, fren ve gaz sistemlerindeki kontrolcüler için PID kontrolcüler tercih edilmiştir. Kontrolcü katsayılarının ayarlanması için aracın ön ve arka akslarının altlarına destekler konularak yer ile teması kesilmiş ve denemeler yapılmıştır. Daha sonra ana kontrolcüye RC sinyal alıcı bağlanarak, bu sistemlerin kararlılığını ve kontrolcülerin uygunluğunu test etme amacıyla, laboratuvar içinde ve koridorda RC kumanda ile sürüş denemesi yapılmıştır. Bu testlerde aracın hızlanma ve yavaşlama kararlılığı, manevra kabiliyeti ölçülmüştür. Kontrolcülerin kararlı olduğu görüldükten sonra otonom sürüş için sensör entegrasyonu çalışmalarına başlanmıştır.

İç mekana yönelik geliştirilen otonom araçlarda, sorun teşkil eden en önemli konulardan biri, aracın mekan içindeki konumunun bulunmasıdır. Dış mekanda çalışan otonom araçlarla GPS sensörü ile cm hassasiyetinde konum bilgisi alınabilirken, iç mekanda çalışan araçlarda GPS sensörü uydu sinyali alamadığından, bu mümkün olmamaktadır. Bu sorunu çözmek için çeşitli çalışmalar yapılmış, görüntü işleme tabanlı, kablosuz sinyal tabanlı(IPS) vb. çeşitli yöntemler denenmiştir. Bunlardan en stabil ve isabetli olanı, mekana kablosuz sinyal verici cihazlar, araç üzerine bir alıcı cihaz yerleştirip, bu cihazlardan alınan sinyaller kullanılarak triangulasyon yöntemi ile konumun hesaplanmasıdır. Ancak böyle bir sistemin kurulması sinyal noktası sayısına bağlı olarak maliyetli olmakla birlikte, kablosuz sinyaller duvarlardan geçerken zayıfladığından her alan için en iyi seçim değildir. Bu çalışmada konum hesaplanması için temel sensör olarak enkoder kullanılmıştır. Enkoder'dan alınan hız verisi, direksiyon açısı verisinden elde edilen araç doğrultusu verisi ile birlikte kinematik denklemlerden geçirilmekte ve aracın konumu bu şekilde sürekli olarak hesaplanmaktadır. Tekerlek kayması sebebiyle meydana çıkan hatanın oranı, aracın düz zeminde ve düşük hızda ilerlemesinden kaynaklı olarak çok düşüktür. Yine de uzun mesafeler kat edildiğinde, kümülatif hatadan dolayı, gerçek konumla ölçülen konum arasında farklar oluşabilmektedir. Bu sorunun çözümü için ise gelecek çalışmalarda, mekanın çeşitli yerlerine yerleştirilmiş veya mekanın kendisinden önceden elde edilmiş özgün görüntüler referans alınıp, araç üzerine yerleştirilecek bir kamera sisteminden alınan görüntü ile karşılaştırılarak aracın konum ölçümünün düzeltilmesi hedeflenmektedir.

Aracın yayaların yoğun bulunduğu ortamlarda çalışması, hareket eden veya edemeyen engellerin ayırt edilmesi, dar hareket alanı ve insan davranışı gibi faktörlerden kaynaklanan problemleri de beraberinde getirmektedir. Bu konuda daha önceden küçük robotlarla birçok araştırma yapılmış, insanların davranışlarını önceden tahmin edebilen ve insanlardan mümkün olduğunca uzak durmaya yönelik kontrolcü ve teknikler geliştirilmiştir. Bu çalışmada engellerin algılanması için, kendi gönderdiği gözle görülmeyen ışınların yüzeylerden yansıma sürelerini hesaplayan bir LIDAR sensör kullanılmıştır. Bu sensör gerek hava, gerek kara için üretilen sürücüsüz araçlarda yaygın olarak kullanılmakta, ışınların geri dönüş sürelerinden, ışığın değdiği yüzeyin uzaklığını hesaplayabilmektedir. Bunun yanında 4 katmanlı tarama yaparak, gördüğü ortamı 4 düzlem bazında üç boyutlu nokta bulutu şeklinde sunabilmektedir.

Aynı zamanda içindeki algoritma sayesinde, baktığı ortamdaki objeleri de boyutlarıyla ayırt edebilmekte, hareket hızlarını ölçebilmektedir. Bu sayede yayaları diğer engellerden ayırabilmek, dolayısıyla hareket edebileceklerini önceden tahmin etmek ve hareketlerini ölçmek kolaylaşmaktadır. Bu sensörün üzerine 1 düzlem ve daha geniş tarama açısına sahip bir LIDAR daha eklenmiştir. Bu sensör 4 düzlem LIDAR kadar ayrıntılı veri vermese de, geniş tarama açısı kör noktalar için kullanıldığında, daha kararlı ve güvenli engelden kaçma davranışı sağlamaktadır.

Otonom sürüş için, sensörlerden alınan verilerin işlenerek, belirli karar ve planlama mekanizmalarından geçirilmesi gerekmektedir. Bu çalışmada, haritası bilinen bir mekanda aracın çeşitli durak noktaları arasında hareket etmesi planlanmıştır. Bu hareket için rota planlaması, aracın başlangıç konumundan istenilen durak noktasına kadar sıralı hedef noktalar belirlenmesi ile yapılmaktadır. Aracın bu noktalardan belirli bir uzaklık töleransı ile geçmesi ve hedef noktalar bitene kadar, noktadan geçtikçe bir sonraki noktayı hedef alması ile, araç son hedef noktaya ulaştığında, istenilen durak noktasına ulaşacaktır. Söz konusu hedef noktalar arasında hareket için, önce sadece nokta takibine yönelik bir algoritma denenmiş, daha sonra hem nokta takibini, hem engelden kaçma kabiliyetini içerek potansiyel alan methodu kullanılmıştır.

Kullanılan ilk algoritmada aracın hedef noktaya yönelmesinin kontrolü, aracın doğrultusunun ve hedef noktanın sabit x ekseni ile yaptığı açının karşılaştırılmasıyla yapılmaktadır. Bu karşılaştırmanın sonucu hata olarak alınmakta ve direksiyon bu hataya göre kontrol edilerek, minimum dönüş miktarı ile hedef noktaya yönelecek şekilde manevra yapılmaktadır. Bunun yanında, hedef noktaya yönelmeden önce, bu noktanın aracın dönebileceği en küçük dönüş yarıçaplı çemberin içinde olup olmadığı kontrol edilmektedir. Eğer nokta bu çemberin içindeyse, araç tam bir dönüş hareketi yapsa da noktaya ulaşamayacağından, önce aksi yönde manevra yaparak noktayı bu çemberin dışına çıkarmakta, daha sonra noktaya doğru yönelmeye çalışmaktadır. İkinci ve son algoritmada ise potansiyel alan methodundan yola çıkarak, engellerin uzaklıklarıyla orantılı olarak itme kuvveti ve hedef noktasına olan uzaklıkla orantılı olarak çekme kuvveti hesaplanıp, bu kuvvetler kullanılarak aracın yönelmesi gereken doğrultu belirlenmektedir. Araç bu şekilde engellerin arasından manevra yaparak geçebilmekte ve hedef noktasına ulaşabilmektedir.

İç mekana yönelik geliştirilen otonom araçlarda karşılaşılan bir diğer problem de, ortamdaki yayalar sebebiyle meydana gelen kaza veya hedef noktaya zamanında ulaşamama durumlarıdır. Yayaların davranışlarının tahmin edilebilmesi için çeşitli çalışmalar yapılmaktadır. Geliştirilen robotlar bu davranışları öngörerek hareketli yayaların gideceği yolu tahmin edebilmekte ve yoldan çekilebilmekte, duran yayaların ise etrafından dolanabilmektedir. Ancak sadece davranışları öngörmek ve yayalardan uzak durmak bazı durumlarda yeterli olmamaktadır. Bir alışveriş merkezinde ilerleyen bir otonom aracın yolunun, yolda duran bir grup insan tarafından kesilmesi ve aracın geçebileceği bir yer bulamaması, bu yetersiz durumlara bir örnek olarak verilebilir. Bu ve benzer durumların en efektif çözümü, aracın insanlarla iletişim kurmasıdır. Çalışmanın ilerleyen aşamalarında planlanan düzenlemelerle, araca insanlarla iletişim kurabileceği donanımlar entegre edilmesi ve buna yönelik yeni davranışlar tanımlanması mümkündür. Önceki örnek üzerinden gidilirse, bu iletişim, aracın topluluktan geçme izni isteyerek, yayaların kenara çekilmesi ile kendisine bir yol açması olarak düşünülebilir.

Bu çalışmada, yayaların bulunduğu bir iç mekanda çalışması planlanan bir otonom aracın tasarımının çeşitli aşamalarının üzerinde durulmuştur. İnsansız hareket için kullanılan mekatronik sistemler ve sensörler sunulmuş, kontrol için kullanılan donanım ve yazılım açıklanmıştır. Bunun yanında aracın otonom şekilde seyahat etmesi için kullanılan algoritmalar gösterilmiş, bu algoritmalar için gerekli verilerin ortamdan alınması için kullanılan sensörlerden bahsedilmiştir. Son olarak da gerçek dünyada yapılan sürüş testlerine değinilmiş ve sonuçlar irdelenmiştir.

1. INTRODUCTION

The latest advances in sensing and control technologies made the dream of self-driving autonomous vehicles, a reality. Especially within the last decade, there is a considerable effort on research community on developing autonomous vehicles [1–6]. These efforts lead to multiple successful autonomous vehicle prototypes on the roads worldwide. The autonomous vehicles are expected to improve our life quality by reducing road accidents [2], traffic stress, travel time and improving efficiency on roadways with cooperative driving [7]. One of the most well known examples of the autonomous vehicles on road, is the google car (Figure 1.1). Utilizing high technology sensors and large amount of collected data, it is one of the most successful prototypes in this field.

Figure 1.1 : Google car autonomously driving on road.

Although the roadways take the biggest share of our transportation, the need of transportation is not limited to roadways. There is a considerable need of intelligent transportation systems on environments such as university campuses, shopping malls, and airports. Most of the recent work on autonomous vehicles is either on roadways, or on restricted areas such as laboratories or parking lots. Few work in literature deals with operating robots and transportation vehicles in indoor, public, crowded environments.

AGVs that are used for transportation of materials on factories [8–10] are an example of having automated vehicles with close proximity to pedestrians. Running AGVs on predetermined paths, with speed lower than person's walking, and using collision sensors were sufficient to get a collision free operation in these environments. Similarly, assistive robots such as [11] find and escort elderly in a retirement facility, require accurate planning under uncertainty of the environment.

Most of the public, crowded environments involve low or none vehicle traffic. Some may include outdoor as well as indoor environments. Although, navigation of autonomous vehicles on outdoor areas [12] is under consideration of multiple research projects, navigation within indoor areas such as buildings, tunnels are not very common. Moreover, the operational need of transportation inside buildings requires electric vehicles due to hazardous internal combustion engine residue gasses (Figure 1.2). Having an autonomous vehicle inside a building poses additional problems that need to be addresses. Firstly, GPS sensor is one of the most relied on sensor for obtaining down to cm level accuracy on localization of autonomous vehicles. Unfortunately, it cannot be used inside buildings. Secondly, moving crowd possess great safety and navigation problems to be solved.

Figure 1.2 : An indoor passenger transportation system used on many airports.

Accurate and fast detection of pedestrians is critical for safety. In previous studies detection of pedestrians and other static or dynamic obstacles is performed with vision, sonar and laser sensors [13]. Recently, problems related to operating autonomous vehicles in dense, interacting crowds were proposed in [14, 15]. In some cases, multiple simultaneous pedestrian movements lead to a vehicle that is unable to move. Researchers labeled this problem as the freezing robot problem (FRP). In order to accurately predict the motion of the crowd behavior, some researchers investigated the dynamics of crowds [16–20].

This study proposes an intelligent autonomous transporting vehicle that can be used by elderly, disabled or anyone who requires to be transported with adequate sensor capabilities and control strategies to safely work in known indoor and outdoor public environments with multiple pedestrians moving. Unlike recent studies, the proposed vehicle will be used in indoor public environments with multiple pedestrians. This feature makes the project unique.

2. DESIGN OF THE SYSTEM

In order to perform effectively in indoor environments, it is required to have an all-electric vehicle with enough battery to perform multiple transportations. A vehicle with small turning radius is advantageous to navigate in tight areas. For these reasons, a Yamaha electric golf cart G19E was selected as the vehicle platform.

For autonomous drive, achieving the unmanned drive and then, obtaining information about both the environment and itself is necessary. Designed system chart symbolizing system elements and their connections, is shown in figure 2.1.

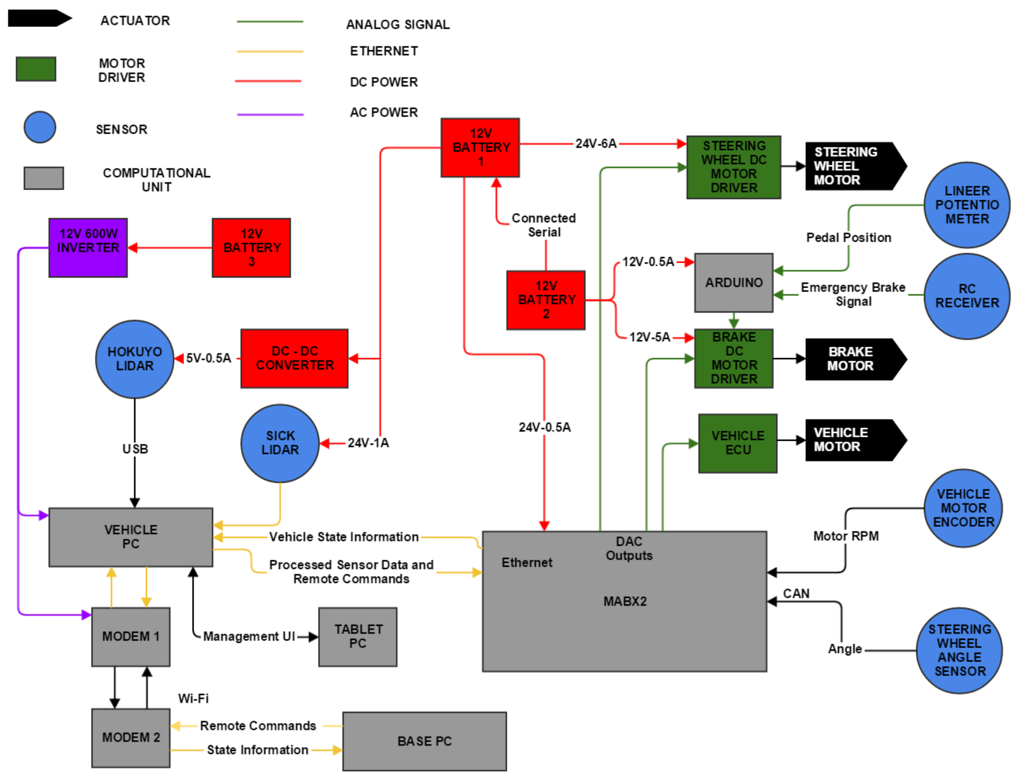


Figure 2.1 : Designed system elements and connections.

Mechanical actuators were connected to vehicle. There are electronic systems to control these actuators. Also to sense its environment in different ways and understand its state effectively, several kinds of sensors were implemented on vehicle, some of them having both a unique role and cooperative role with another sensor. Briefly, these sensors are including a LIDAR to sense and classify obstacles in front of it, a rotational encoder to measure displacement and decide location, several sensors where some of them being related to actuators to measure and control vehicle state. In this chapter, all of these components will be explained in detail.

2.1 Unmanned Drive Subsystems

To achieve unmanned drive, vehicle's steering, throttle and brake should be controlled by automated systems. Steering wheel and brake system is fully mechanical on the original vehicle, where throttle is not. Because of this, steering and brake subsystems has mechanical parts connected to vehicle, controlled by electronic parts. In order not to eliminate manual driving capability totally, these subsystems were integrated parallel to current systems on the vehicle. These three main subsystems are explained in this section, along with the sensors, actuators and drivers they include.

2.1.1 Steering subsystem

Mechanical component of this subsystem, a DC electric motor, is connected to the steering wheel with sprocket and chain. System is shown in figure 2.2.

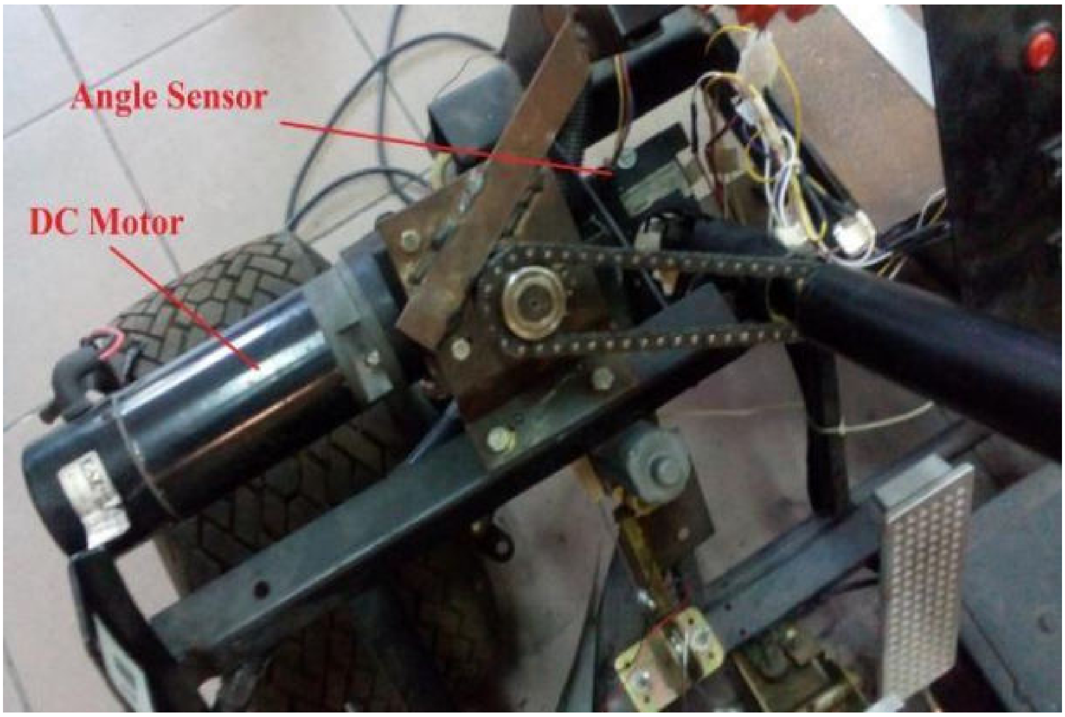


Figure 2.2 : DC motor connected to steering wheel.

This motor is controlled by a DC motor driver, thus plays a role as an actuator to control vehicle steering by rotating the steering wheel mechanically. A steering angle sensor is connected to steering wheel rod. This Bosch steering angle sensor is able to measure steering angles -780 to 780. Measurement from the steering angle sensor is used to control position of the steering wheel.

2.1.2 Brake subsystem

Second subsystem is for braking. Purpose of this subsystem is to slow down the vehicle but mainly to ensure safety. A window lifter DC motor system is used for braking. Wire is connected to brake pedal through a pulley. With the rotation of DC motor, wire is wrapped inside or freed to outside, pulling or releasing the brake pedal. The system is shown in figure 2.3.

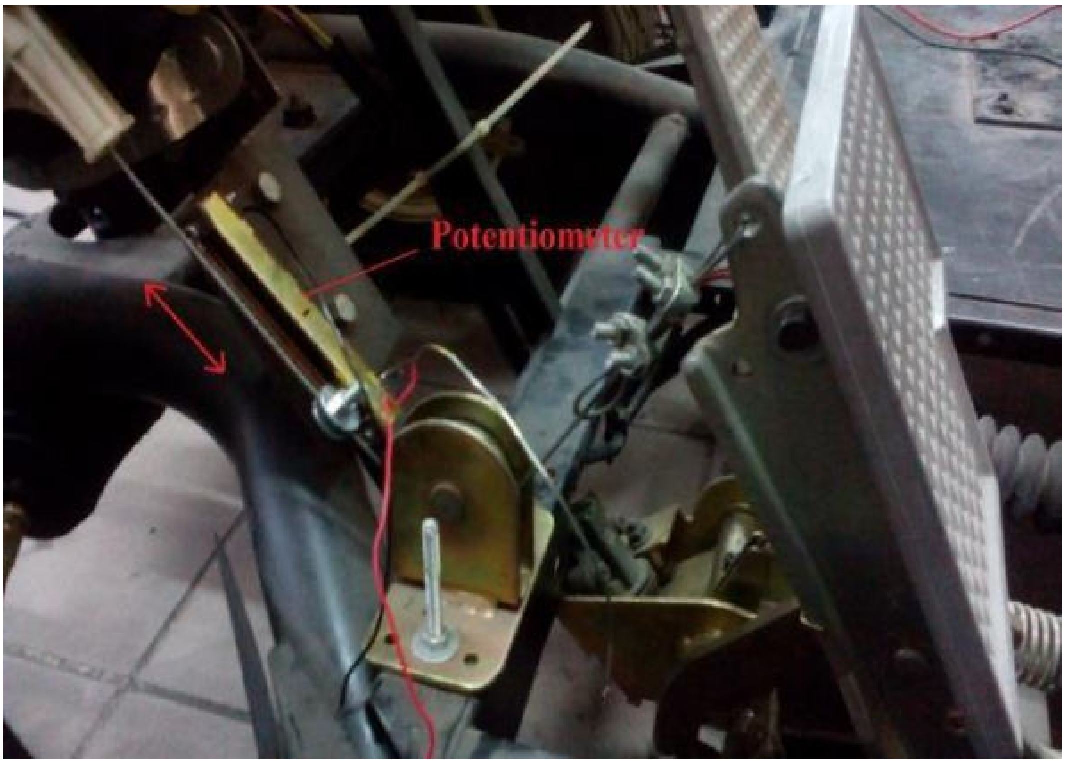


Figure 2.3 : Brake mechanism.

For driving the motor, analog servo drive board from advanced motion controllers was used. A linear potentiometer is connected to system by physically fixing on wire, sensing the wire movement, thus the brake pedal position. This sensor is used to control the pedal position.

2.1.3 Throttle subsystem

On default setting, vehicle's throttle is controlled by a signal from potentiometer on gas pedal. ECU powers this potentiometer and reads the pedal position by reading the analog voltage signal out of this sensor. Then provides equivalent power to electric motor. Thus, vehicle moves to selected direction. Direction selection can be made via a 3-state switch on the front panel of the vehicle.

This subsystem does not include any mechanical actuators. An encoder was connected to motor shaft, which is used to measure angular velocity of motor shaft. To control speed of vehicle, a voltage signal generated by controller is sent to vehicle's ECU. Angular velocity is measured by wheel encoder and used in throttle control closed loop as feedback. Also to trick vehicle as if it is moving all the time, a PWM signal is sent to ECU tachometer probe continuously. This signal prevents time consuming on/off behavior of electric valve of the vehicle motor.

2.2 Environmental and State Sensors

When it comes to the word autonomous, sensors are a must for vehicle to measure its own state, communicate with the environment and make decisions. There are variety of sensors shown in this section, each for different uses, different parts of the environment identification or measurement process. Their purpose and physical implementation are discussed.

2.2.1 Encoder

Autonomous vehicles which are being researched nowadays, mostly use GPS sensor to measure position. But they are able to use it because their working environment is outdoor, meaning GPS sensor is able to get satellite signal and determine the position. This vehicle is designed mainly for indoor environment. Because of this, using a GPS sensor is not enough. Instead of using GPS sensor as main position measuring sensor, an encoder connected to motor shaft is used. It is shown in figure 2.4. Encoder measures the angular speed of the shaft, using this information with steering wheel angle, translation and position of the vehicle is calculated.

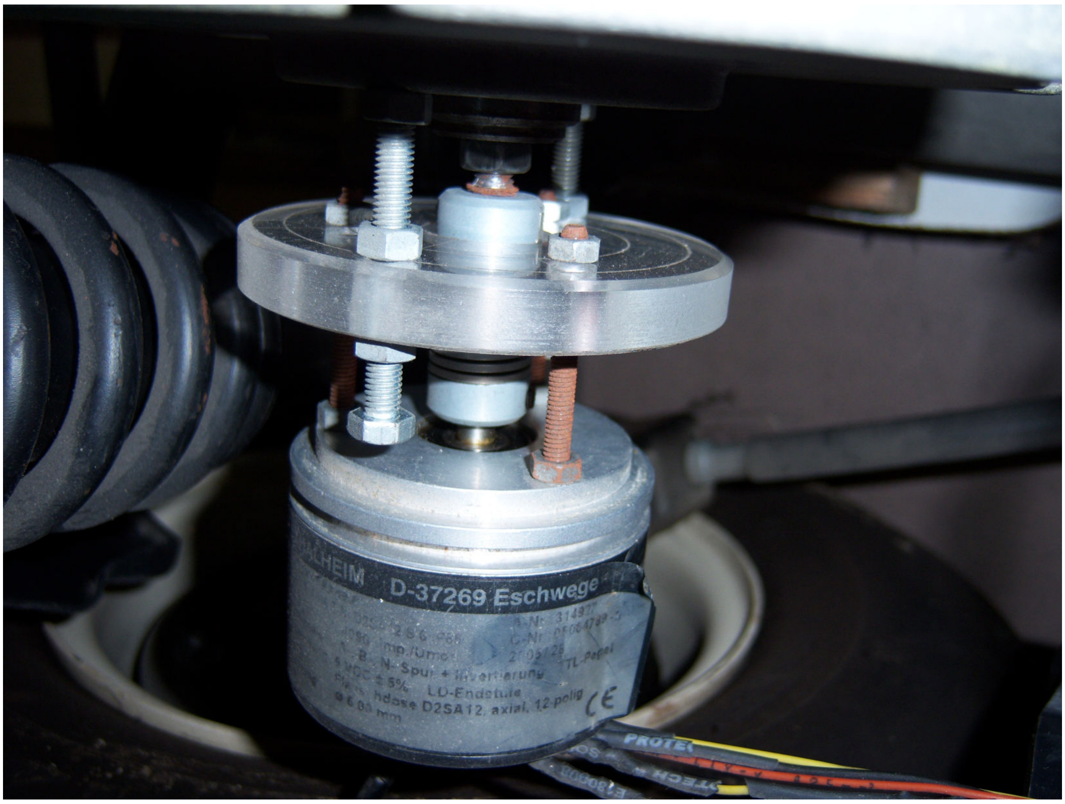


Figure 2.4 : Encoder connected to motor shaft.

Encoder sensor has an accumulative error because of the slip on tires. On low speed, this error is negligible if the distance traveled is short. But further corrections from other sensor/sensors is necessary for eliminating this error on long distance. For this purpose, a stereo camera system is considered. Which will be using landmarking technique to determine position of the vehicle on every mark and correct the position calculated from encoder measurement. This camera system will be implemented in autonomous system on future phases.

2.2.2 LIDAR sensors

LIDAR is a remote sensing method that uses light to detect range. LIDAR scanner is a sensor that visually scans the environment using LIDAR method. Sends invisible laser rays, this rays bounces from surfaces and return back to the device. Device collects this rays and calculates distance of each point hit by ray, using the time passed between the ray went forward, bounced and collected back.

Two LIDAR sensors are mounted on the vehicle to scan the environment. With this LIDAR scan data, vehicle is able to detect obstacles and manuever away from them. Moreover, further development can be done to gain capabilities namely object classification, 3D mapping, SLAM algorithms etc. Figure 2.5 is an illustration for positions and FOV of the LIDAR sensors on the vehicle. Main LIDAR sensor is the SICK LIDAR with high resolution. Its FOV is shown in red color. Second is Hokuyo LIDAR with low resolution but wider FOV. It supports the main LIDAR by detecting obstacles at the front left and front right of the vehicle.

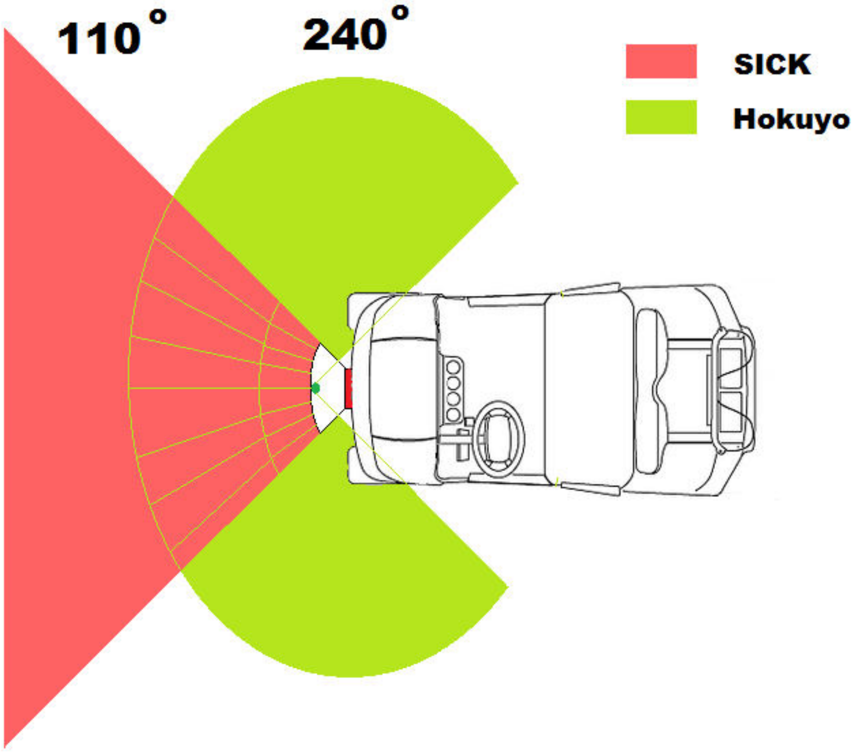


Figure 2.5 : Mounting position and FOV illustration for LIDAR scanners.

2.2.2.1 SICK LIDAR

The device we use is an SICK LIDAR scanner, shown in figure 2.6. Scans points in 4 layers,each layer being a horizontal line,so from view of the scanner, it scans 4 different horizontal planes and this planes are lined up vertically. Using 4 plane layering, unlike just scanning 1 plane, vehicle will be able to classify objects easier. Layer color and angle information given in figure 2.7.

LIDAR scanner has 2 sockets on it. Test setup has been constructed without mounting on the vehicle. One socket is connected to 12 V supply, the other sockets connected to laptop for receiving Ethernet data. First of all, network configuration of PC has been done, changing IP address to see the LIDAR scanner on the network. After that the program named SICK Laser View, installed, configured and the device was tested.

Figure 2.6 : SICK LIDAR scanner.

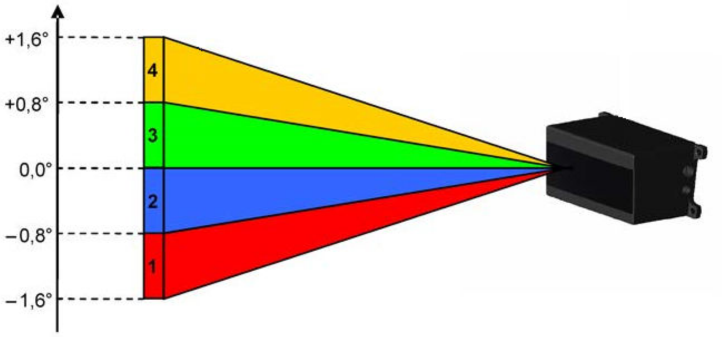
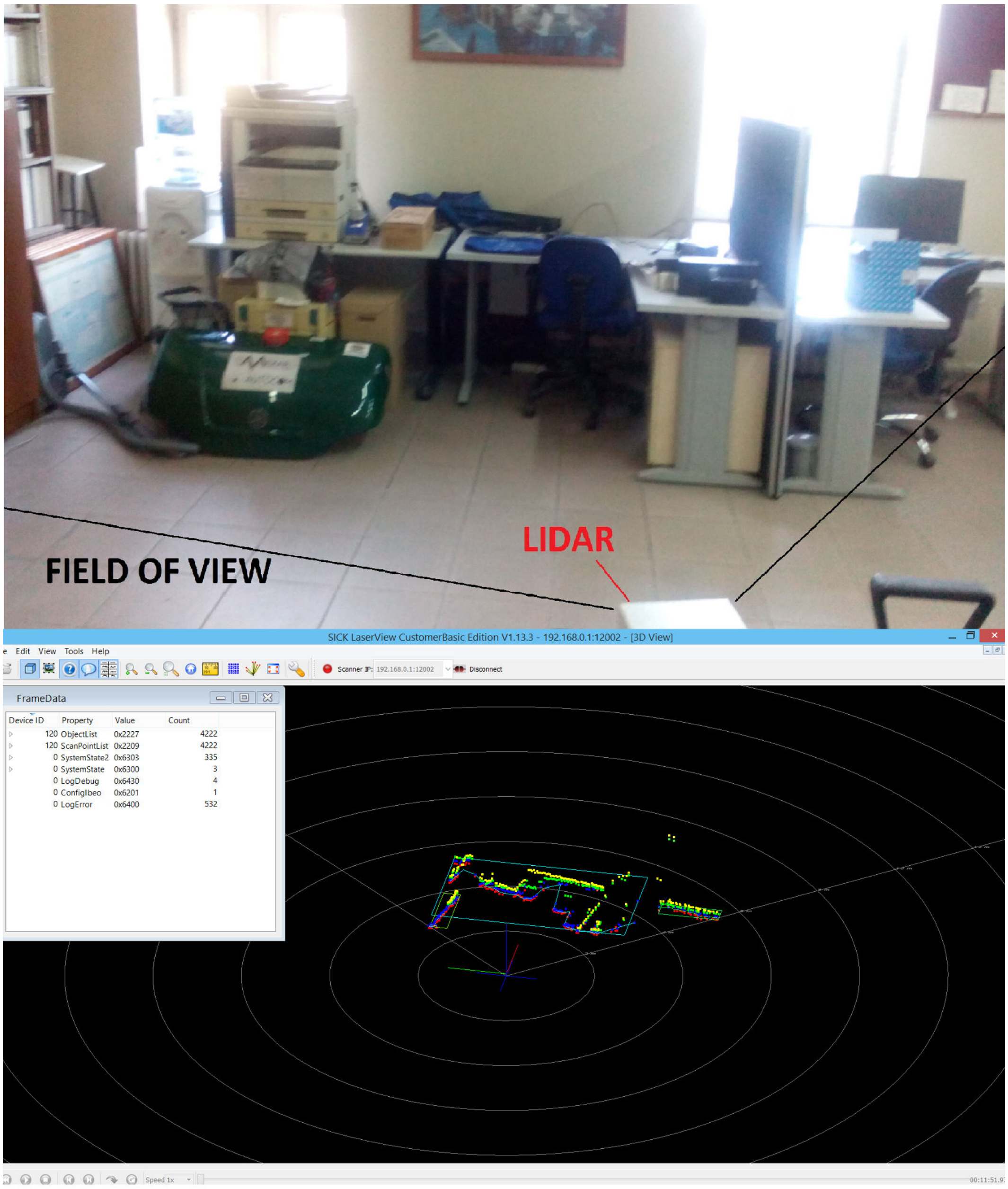


Figure 2.7 : Scan plane layering.

Both the real world scan environment and a screenshot taken from SICK Laser View program shown in figure 2.8. On Laser View part, each circle represents a distance of 1 meter. At the center of all circles, there is the device and straight lines from the center, are the limits of field of view of the device. Points are colored according to layers. The device also groups points and forms objects. Using this feature, device can draw contours and object boxes. This can be seen looking at figure 2.8.

After the testing setup and scan experiments are done for the sensor, it was mounted in front of the vehicle. After that, 4 holes were drilled on the metal plate in front of the vehicle. Sensor is fixed on this metal plate with its own brackets. Brackets were configured to make sensor look forward with 90 degree angle with ground and tilt was configured parallel to ground. Power input was connected to 24V power line. Then, ethernet connection was done with the PC on vehicle, through a network switch. Thus, sensor was ready to use in autonomous drive. A picture of the sensor mounted with the protective cover, is shown in figure 2.9.



Device ID	Property	Value	Count
120	ObjectList	0x2227	4222
120	ScanPointList	0x2209	4222
0	SystemState2	0x6303	335
0	SystemState	0x6300	3
0	LogDebug	0x6430	4
0	ConfigIbeo	0x6201	1
0	LogError	0x6400	532

Figure 2.8 : LIDAR scanner testing.

2.2.2.2 Hokuyo LIDAR

As an addition to 4 layer SICK LIDAR, Hokuyo LIDAR was integrated into system. This LIDAR scans 1 layer but has wider scan angle compared to other LIDAR. It is used to scan objects which fall out of the scan range of main LIDAR, while vehicle is moving. Using the data in desired regions of the 240° scan, steering to the direction of avoided obstacles before reaching the proper maneuver angle is prevented. More stable and safer obstacle avoidance behavior is achieved.

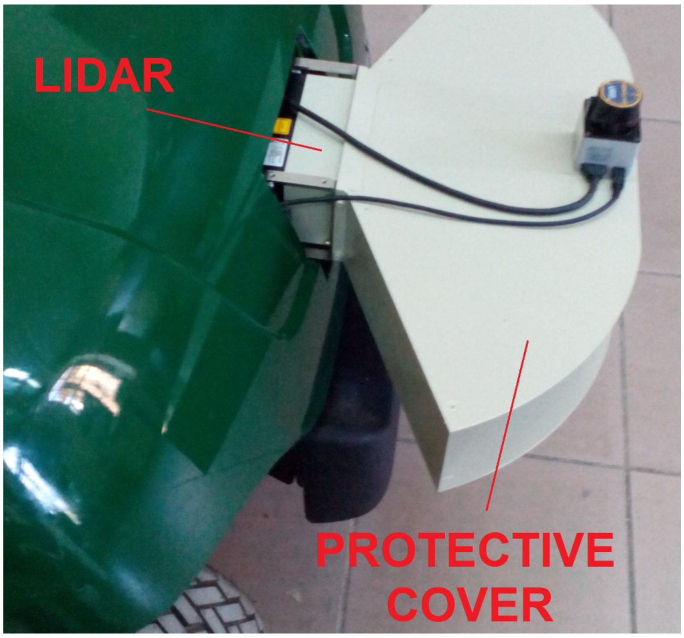


Figure 2.9 : SICK LIDAR scanner mounted.

Figure 2.10 : Hokuyo LIDAR scanner mounted.

Hokuyo LIDAR sensor was fixed on the protective cover of the other LIDAR with 2 holes drilled. When determining the position of the sensor, field of view of the other LIDAR was taken into account for this sensor to cover blind spots as much as possible. To supply power with 5V voltage, a DC-DC converter was connected to 24V power line of the vehicle and its output was connected to sensor. Reading of the sensor data was done by connecting the USB output of the sensor to PC directly. LIDAR sends data with serial communication via USB port. Mounted picture of the sensor is shown in figure 2.10. Also a scan received by sensor via its own scanning software is shown in figure 2.11.

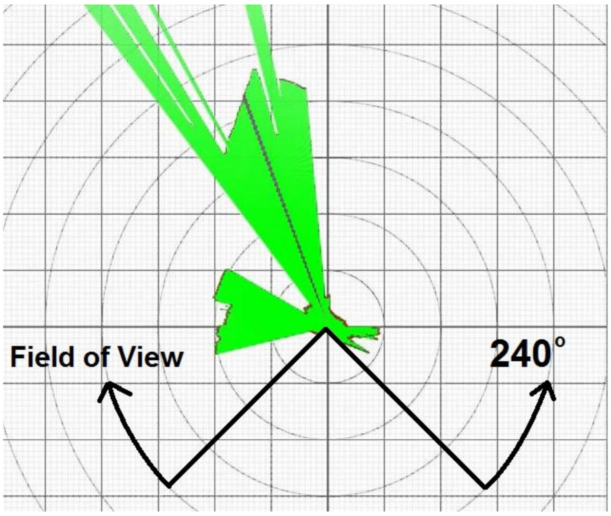


Figure 2.11 : Hokuyo LIDAR scan.

2.3 Processing Units

Autonomous vehicle is a complex mechatronic system which has numerous types of sensors and actuators. On design, numerous types of processing units are included with different roles. These roles includes controlling, data extraction, emergency measures, state calculation. Roles and the units are explained in this section.

2.3.1 Microautobox 2

MABX2 is a robust and easy to use stand-alone prototyping unit produced by dSpace. Allows user to run a simulink model in real time, on a hardware. Has many applications in numerous areas. Device is shown in figure 2.12.

Main control unit of the system is the dSpace Microautobox 2. All controllers are embedded in this system as a simulink model. There are digital/analog input/output pins, communication pins for RS232, CAN communication on the device. These pins are used for communicating with sensors and controlling drivers. Desired signals to actuators are placed in the simulink model and connected from MABX2 output pins to drivers, accordingly.

Figure 2.12 : dSpace Microautobox 2.

Device has two Ethernet sockets which we use both of them. One is for the PC in the vehicle, sending processed sensor data. Other one is for embedding simulink model into hardware, simulation parameter tuning and data analysis with dSpace ControlDesk program. General overview of the operation of MABX in this system is, it obtains processed/raw sensor data from PC/sensors, makes calculations using decision algorithms and controller blocks in simulink model embedded into it. Then sends signals to drivers through output pins, for controlling actuators.

2.3.2 PC

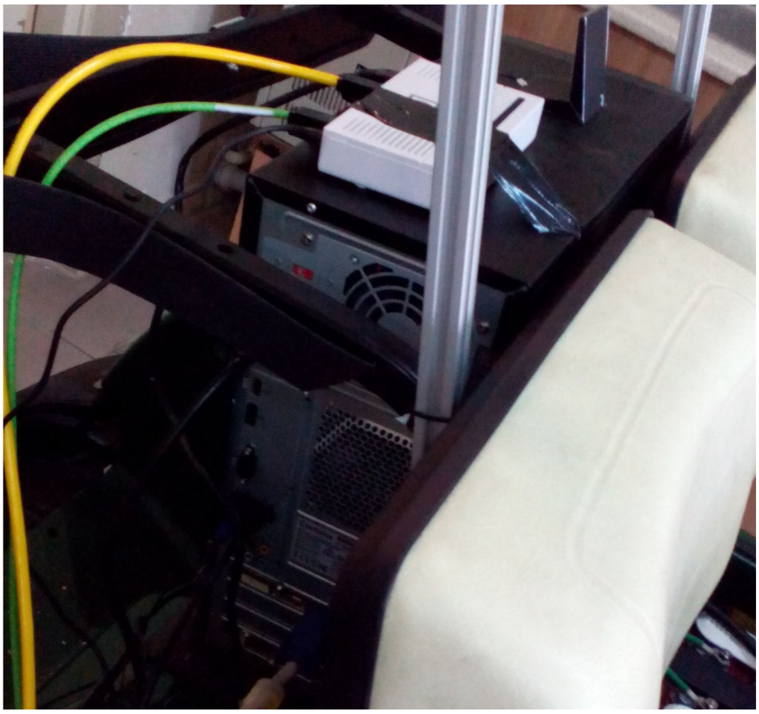


Figure 2.13 : PC on back side of vehicle.

PC is the unit that obtains data from sensors, process it and obtains the data that MABX2 will use. It is shown in figure 2.13. It is needed for image processing, sensor data extraction, data recording and it plays bridge role between different components. It runs Ubuntu OS. Sends the processed data to MABX2 via Ethernet. Also saves the collected data for data analyzing will be done later on, if necessary. The main program communicating with sensors and devices, is coded in Java. PC is mounted on back side of the vehicle, behind the passenger seats. There is also a network switch on PC for it to communicate with MABX2 and LIDAR at the same time and a wireless antenna.

2.3.3 Arduino

Arduino in the system is mainly placed for separating the brake pedal control into two brains for emergency purposes. It receives reference signal from MABX2 and communicates with brake motor driver. Measures the pedal position using a linear potentiometer. It has a closed loop control with this potentiometer feedback and sends control signals to brake motor driver.

It also has some features about emergency situations. When the signal from MABX2 is completely cut off, arduino commands brake pedal to move braking position. In addition, a RC receiver is connected to arduino, to use for emergency full brake when a specific switch on RC transmitter is activated.

2.3.4 RC emergency stop

Even with the stable algorithms and properly working devices, it is possible to see abnormal behaviors that leads to accidents. It is possible even if it has a small probability and a precaution is necessary. A remote controlled board consisting of relays was connected to 24V power line, to prevent accidents if unwanted behavior of the vehicle occurs.

When operator pushes the button on the remote controller, corresponding relay on the boards changes state. Board is connected such way which when relay is pulled, power supply is cut down. This power supply includes the power of the main controller, throttle and steering drive. This cut down causes electric motor of the vehicle to stop suddenly and causes the brake driver to apply brakes, stops the vehicle in short time as a result. Also enables manual steering of the vehicle for the cases where throttle is on fault and emergency maneuvering is needed. Board is shown in figure 2.14.

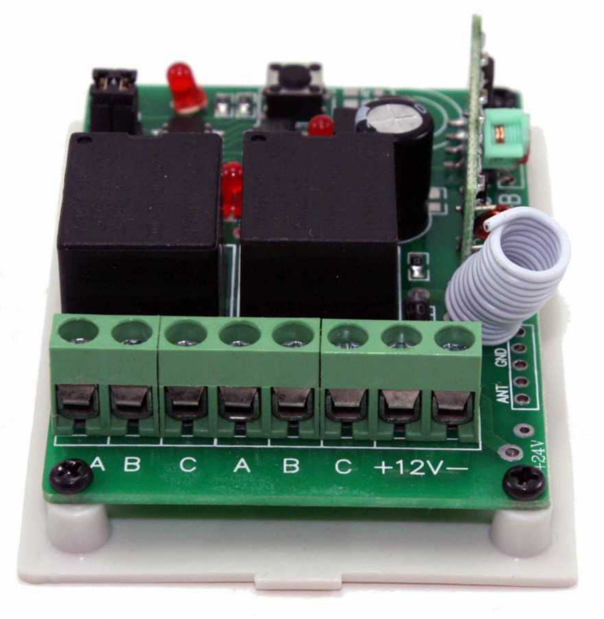


Figure 2.14 : RC emergency stop relay board.

2.4 Power Supply

The vehicle has numerous sensors and actuators with power supply need. It is an electric vehicle, therefore has its own 6 x 8V - 48V battery pack to drive the motor.

But besides the fact that one can only obtain voltage as multiple of 8V without using any converter, there are fluctuations of voltage and current on these batteries in the situations such as sudden motor start or stop. This fluctuations were considered harmful to supplied devices, especially electronic boards, causing them to reset while running. For these reasons, separate power supplies were placed on the vehicle to power devices. These power supplies includes two 12V dry batteries to supply 24V and 12V devices and 12V gel battery to supply the inverter which is producing AC power for PC on the vehicle.

To place these expansive power supplies on the vehicle, a metal plate was manufactured and placed in free space under the passenger seats. Then, plate was fixed on the vehicle body from two sides. An inverter and batteries except vehicle's own battery pack were placed and fixed on this plate. A picture of battery packs under the passenger seats is shown in figure 2.15. After placing the batteries, three separate fuses were placed and fixed. First fuse is connected to 12V battery directly. Then, batteries were connected serial and end cables were connected to second fuse. 12V gel battery was connected to third fuse. These 3 fuses were connected to a 3 channel cam switch which is fixed on the plastic surface in front of the passenger seats. Then ground and separate power line cables were connected to corresponding devices. With this setup, when the switch is turned on, all devices are supplied power.

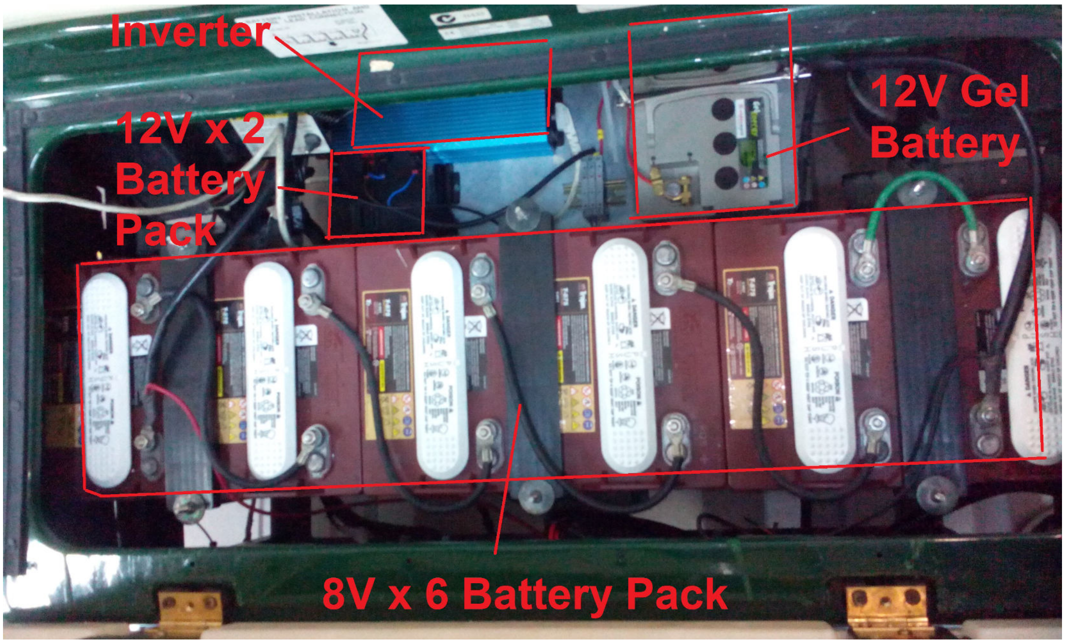


Figure 2.15 : Power supply.

3. SOFTWARE AND CONTROL

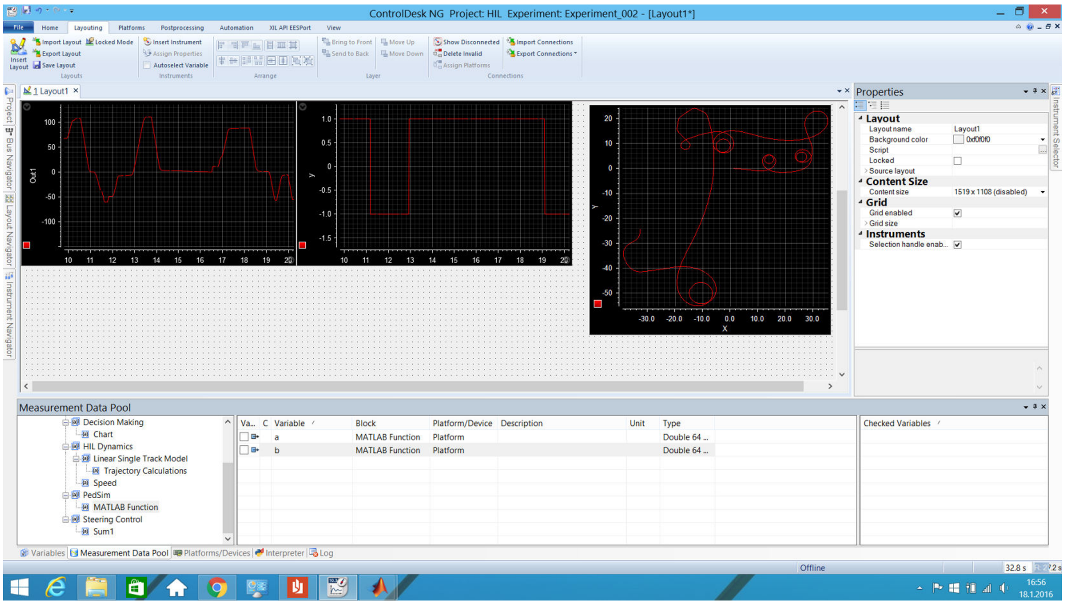
3.1 Software

3.1.1 MATLAB simulink

MATLAB and simulink are used for modeling the controllers, dSpace MABX interface and embedding the model into the device. Control algorithms and some of the calculation functions are coded using MATLAB.

3.1.2 Controldesk

Controldesk is a program, that we use to reach memory of the dSpace hardware, when a simulation is running on it, analyze and change the values of variables in the running model (Figure 3.1). It also allows recording of selected variables, makes collecting data easier. Program has a simple interface allows user to create his own layout with numerous types of tools. Displays, sliders, bars etc. can be easily placed on the layout and used after assigned to a proper variable



Layout
Layout name: Layout1
Background color: 0xf0f0f0
Script
Locked: ☐
Source layout
Content Size
Content size: 1519 x 1108 (disabled)
Grid
Grid enabled: ☑
Grid size
Instruments
Selection handle enab...: ☑

Va...	C	Variable	Block	Platform/Device	Description	Unit	Type
☐		a	MATLAB Function	Platform			Double 64 ...
☐		b	MATLAB Function	Platform			Double 64 ...

Figure 3.1 : Real time experiment on Controldesk.

3.1.3 Ubuntu

Ubuntu is a Linux based free operating system. Actually it's the most widely used free operating system. Ubuntu is preferred because it's more effective to do networking and communication and to install and use programming related software, compared to Windows.

3.2 Autonomous Drive Simulink Model

Main controller MABX2 runs a simulink model in itself. This model handles controlling the actuators, making decisions, calculating necessary information for movement algorithms. The model consists of several main subsystems, namely steering, brake, throttle control subsystems and localization, communication sections. Control blocks has closed loops with dSpace RTI blocks in them, where they need to communicate with world through MABX2 inputs and outputs. Autonomous drive is achieved by controlling actuators after processing data coming from communicate block. Subsystems are explained in this section.

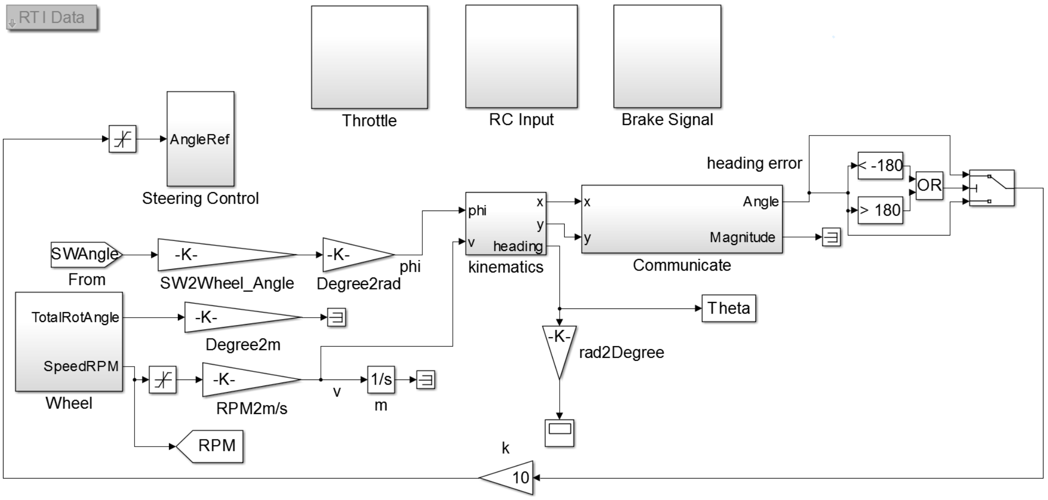


Figure 3.2 : Autonomous drive overall model.

3.2.1 Steering control

Subsystem has a closed loop control with steering reference, CAN angle, controller, boundary and analog output blocks. Steering control subsystem is shown in figure 3.3. CAN angle block has MABX2 CAN communication block in it. Steering wheel angle sensor communicates with MABX2 via CAN. Message structure and byte offset of angle data was manually set in communication block. Subsystem takes current angle

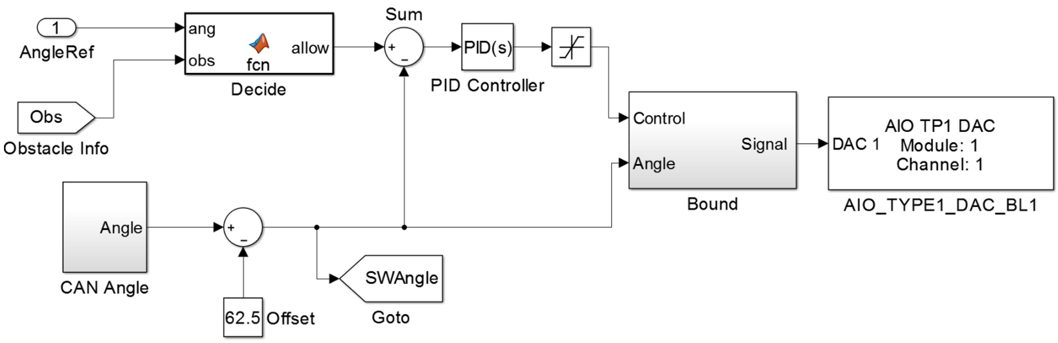


Figure 3.3 : Steering control subsystem model.

and calculates the error by subtracting it from reference angle. Steering angle reference is limited by decision block. When there is an obstacle at the left or right of the vehicle, vehicle is not allowed to steer that direction, until it gets past the obstacle. Error is fed to PID controller and a saturation block limits the controller output signal, therefore limiting the steering wheel motor rotating speed. This signal enters to a bounding block with switch conditions, made for caution purpose. To avoid physical overturning of steering wheel, steer control range is limited by this block. Finally, signal moves to analog output block, where the related output pin on MABX2 is connected to steering wheel motor driver's speed reference.

3.2.2 Brake control

Brake control is mainly done by Arduino by a closed loop control as mentioned on previous chapter. This subsystem is for taking the brake reference signal and transferring that to Arduino as an analog signal via MABX2 analog output block. When autonomous drive is on and reference speed is zero, brakes are applied. Brake control subsystem is shown in figure 3.4.

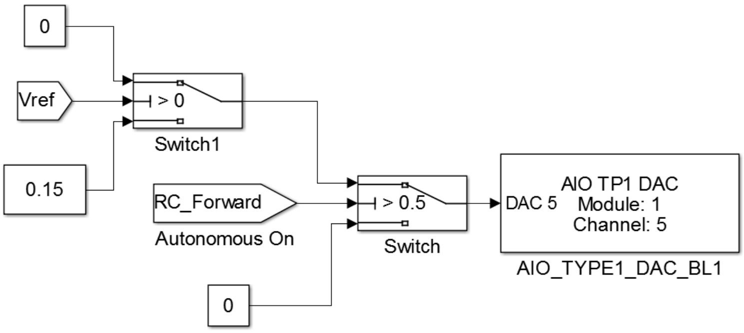


Figure 3.4 : Brake control subsystem model..

3.2.3 Throttle control

Subsystem model has a closed loop control with wheel speed reference as RPM, current wheel speed as RPM, controller and analog output blocks. Current speed is subtracted from reference speed and error is calculated. Reference speed is multiplied by zero if autonomous drive is not activated or vehicle arrived at the goal point. Error is fed to PID controller. Controller output is scaled into voltage range and sent to analog output block. Related analog output pin on MABX2 is connected to ECU throttle signal cable. Second analog output block is for mimicking manual gas pedal potentiometer behavior. Third one is for sending constant PWM to ECU Tachometer sensor to trick it as if motor is rotating. This causes the electric valve for the motor to stay on, preventing the time consuming on/off behavior of electric valve of the vehicle motor. Throttle control subsystem is shown in figure 3.5.

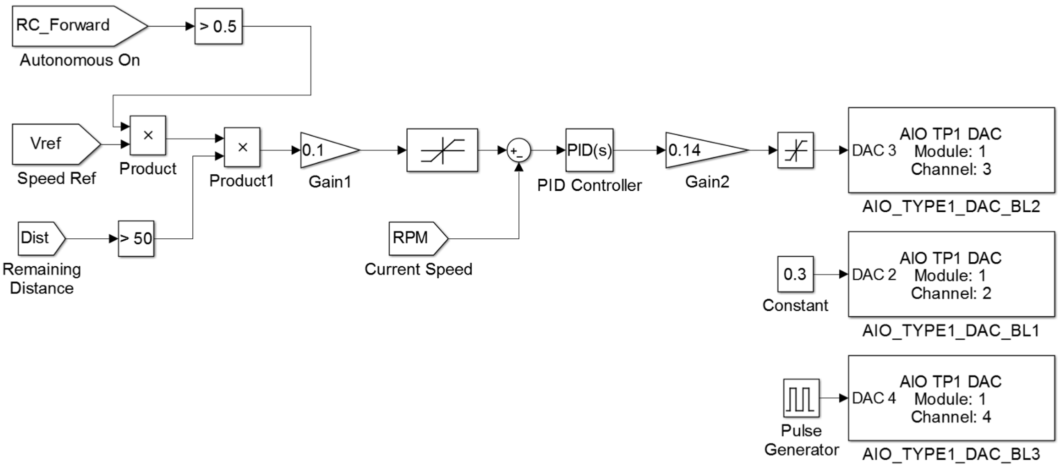


Figure 3.5 : Throttle control subsystem model.

3.2.4 Communication with PC

Some of the sensor data is processed in PC and sent to MABX2. Within this subsystem, first MABX2 communicates to communicate with the PC via ethernet and receives the data sent to it. After receiving, converts the data into integer array. Speed reference and obstacle information are directly moved into the other subsystems in model. Potential field calculation data, magnitude, angle and sign of the repulsive force are converted into x and y components. Total force is calculated in function block by adding attractive force. Angle of total force goes through a low pass filter and used as heading error. Subsystem is shown in figure 3.6.

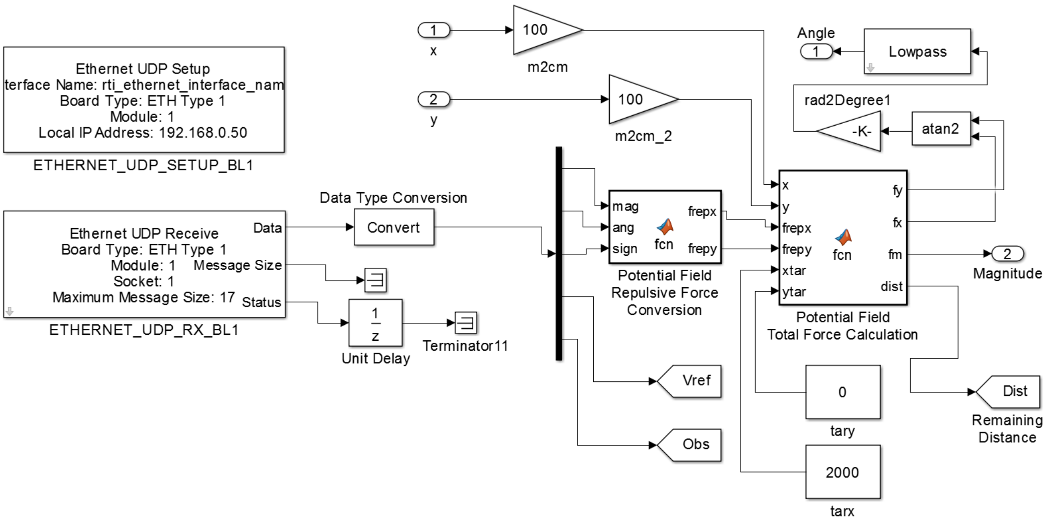


Figure 3.6 : Subsystem model to receive data from PC and calculate heading error.

3.3 Algorithms and Decision Mechanisms

In order to travel between two points, autonomously, vehicle needs to know position of itself, position of the goal and the positions of obstacles. Also different types of states and behaviors are included in this traveling process. The most important part of this project is to have a reliable and safe system to deal with static and dynamic obstacles. The most important dynamic obstacles are moving pedestrians. LIDAR sensor is effective and reliable to detect and track pedestrians.

3.3.1 Localization

In indoor environment, one of the most serious problems is localization. On this vehicle, an encoder is used as main sensor for localization calculations. RPM speed is obtained accurately from encoder. This information is used together with angle information from steering wheel and constant wheel circumference value, to calculate position.

$$\dot{\theta} = \frac{u_1}{l} \tan\varphi \tag{3.1}$$

Steering wheel angle is obtained from angle sensor. Using steering wheel angle, steering angle φ is calculated. Using steering angle φ, forward velocity u_1 and wheel base length l, change in vehicle heading θ is calculated.

$$\dot{x} = u_1 \cos\theta \tag{3.2}$$

$$\dot{y} = u_1 \sin\theta \tag{3.3}$$

After heading is calculated, using sine and cosine of heading together with forward velocity, change in vehicle x and y positions are calculated. Illustration of the components for calculation is shown in figure 3.7.

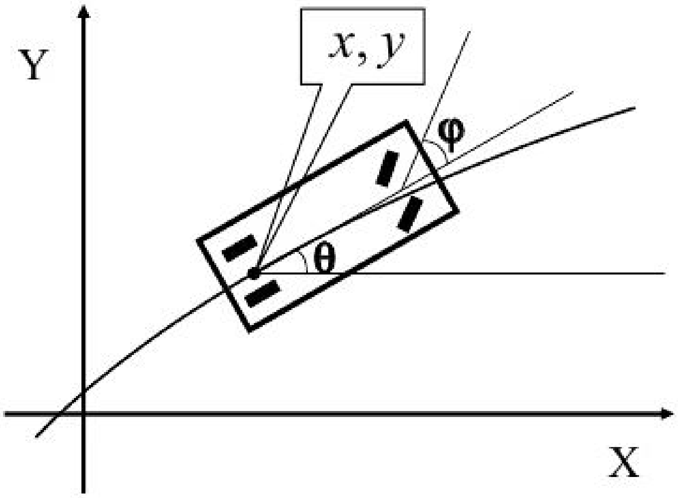


Figure 3.7 : Vehicle position, heading and steering angle.

Localization subsystem is included in main simulink model. It receives wheel angle information from steering wheel control subsystem, using "from" and "go to" blocks. Wheel speed is received from encoder in wheel block, using encoder block of dSpace RTI and making conversion. Forward velocity is calculated from RPM and fed to kinematics block with steering wheel angle. Position is calculated in kinematics block using the method mentioned above. Subsystem is shown in figure 3.8.

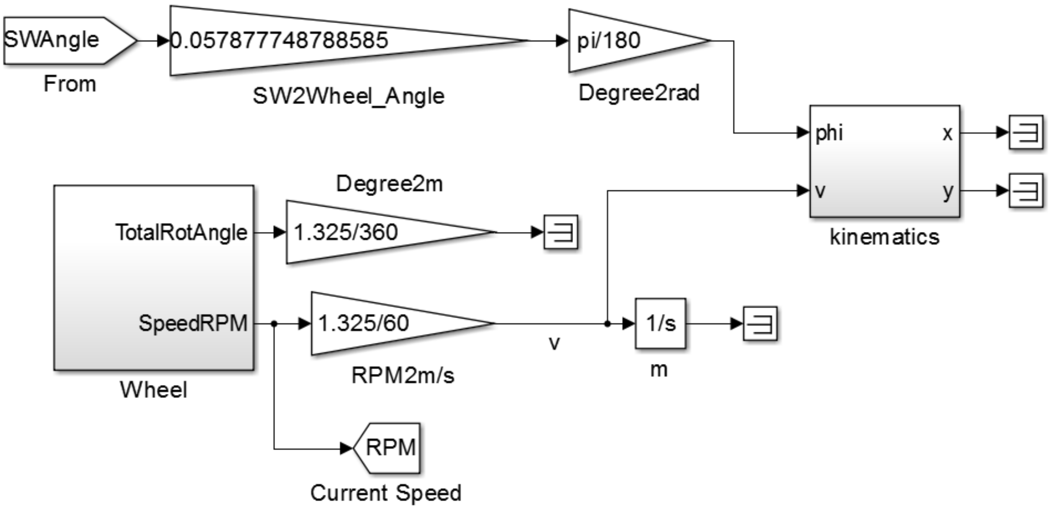


Figure 3.8 : Localization model.

3.3.2 Path following

With position information provided, there are many different ways of traveling between two destinations while avoiding obstacles. For this study, preferred policy is creating virtual goal points between start position and destination, then continuously moving towards the next goal point while avoiding obstacles. Using this policy, autonomous drive is dynamic and doesn't need any fully predefined paths or heavy calculations.

3.3.3 Heading error based path following

First experiments were done using a simple heading based path following algorithm. A simulink model was constructed to implement the algorithm and simulate path following behavior. Model consists of steering and engine parts to simulate vehicle dynamics, blocks to calculate position, next goal point, control logic blocks and a block for checking if the point is inside the minimum turning radius of the vehicle. Model is shown in figure 3.9.

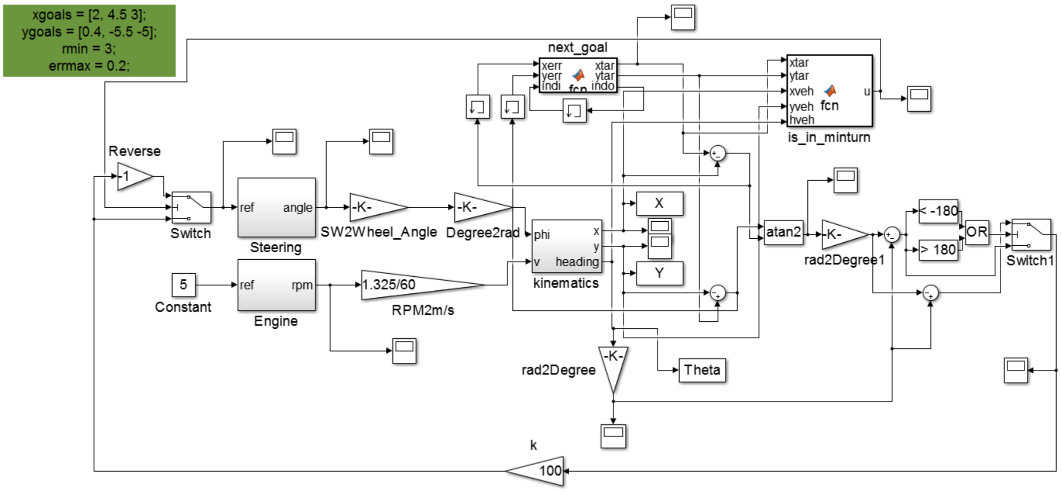


Figure 3.9 : Path following simulation model.

Steering and engine blocks were made using first order linear systems with controllers, to simulate the delay and dynamics on the real vehicle. Speed is constant on the simulation. Minimum turning radius and goal point arrival tolerance are adjustable. Goal point coordinates are defined as arrays before simulation. Next goal point is decided continuously by next goal block while vehicle is moving. If distance between vehicle and current goal point is smaller than tolerance, next goal point is selected as current.

Control part makes decisions and calculates desired steering angle for moving towards to goal point. Heading of the vehicle received and subtracted from the angle between goal point and x axis. This heading error is fed to control logic. Control logic checks the position of the goal point relative to vehicle, deciding the shortest rotation to turn vehicle towards goal point. Is in minimum block calculates if the goal is inside the minimum turning radius. If it is, the vehicle will never be able to reach there and continuously turn in a circle. Control logic detects this and makes the vehicle maneuver to opposite direction of the goal point for making it reachable, until it is out of minimum turning radius.

One of the simulation results is shown in figure 3.10. Three goal points were defined. First point and second point is outside of the minimum turning radius, vehicle turns towards them directly. Third point is inside the minimum turning radius and vehicle maneuvers to opposite direction to be able to reach it. It is seen vehicle is able to follow the points accurately with this heading control based algorithm.

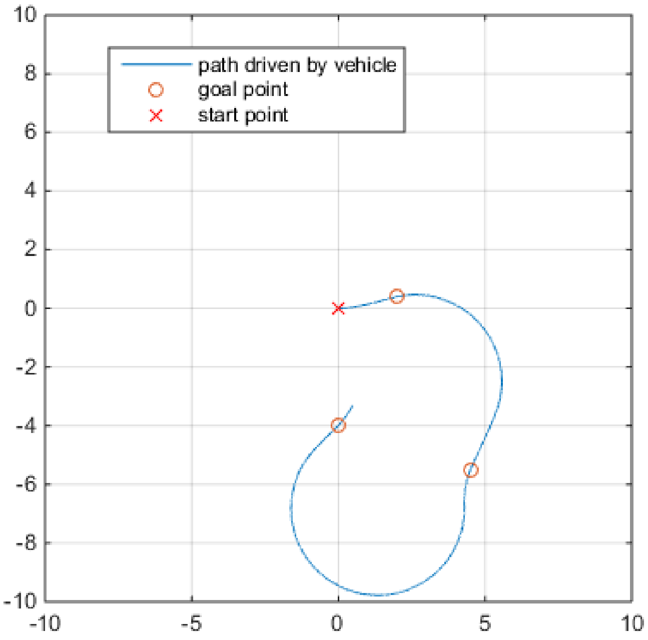
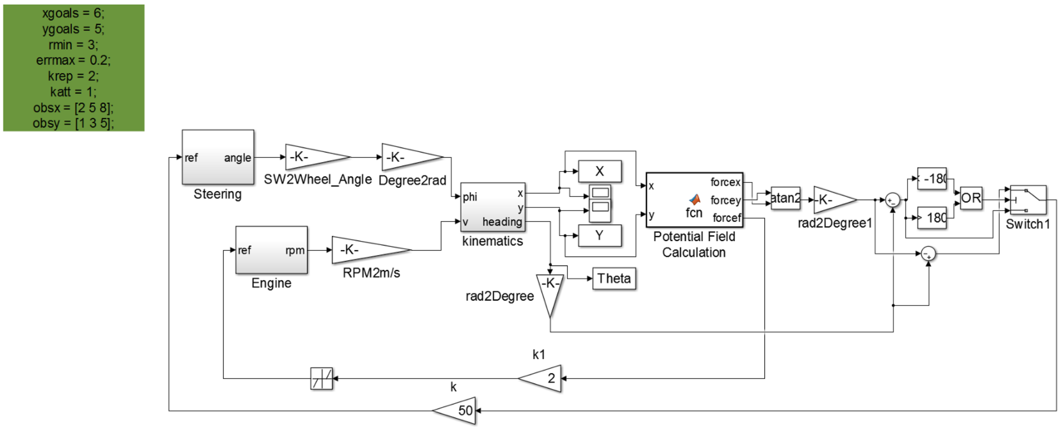


Figure 3.10 : Path following simulation results.

Even though this algorithm is computationally low cost and ideal for path following, it doesn't include obstacle avoidance. For this reason, there was a need to switch the algorithm to a method which includes path following and obstacle avoidance together.

3.3.4 Potential field method based path following

After first experiments are done by not considering the need of obstacle avoidance, simulation model and controllers are optimized, we started to work on more capable autonomous driving algorithm. A method which includes both obstacle avoidance and path following behavior, potential field method was chosen. Some of the limitations of the method and solution proposals are present in the literature [21–23], but are not directly concerning this phase of the study, therefore are not discussed. New system model is constructed using the vehicle model and heading control model in first experiments. Potential field method algorithm is implemented in this new system model using MATLAB code. Resulting system model is shown in figure 3.11.



```
xgoals = 6;
ygoals = 5;
rmin = 3;
errmax = 0.2;
krep = 2;
katt = 1;
obsx = [2 5 8];
obsy = [1 3 5];
```

Figure 3.11 : Potential field method simulation model.

Basics of the potential field method lies on the assumption as if the actor, in this case it is vehicle, is a point object and effected by repulsive forces from obstacles and attractive forces from goal points. Total force acting on the actor is calculated and robot is moved by that force. Illustration of this basic idea is shown in figure 3.12.

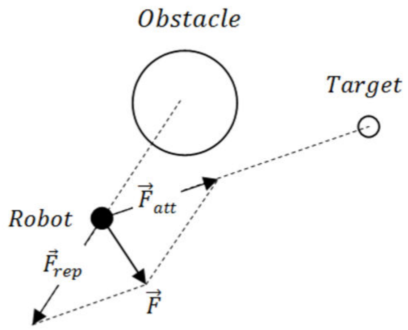


Figure 3.12 : Potential field method illustration of the basic idea.

This repulsive and attractive forces are summed separately, multiplied by constants and vector sum gives the total force acting on vehicle. When multiplication constant of repulsive force is increased, vehicle becomes more beware of the obstacles. On the other hand, when multiplication constant of attractive force is increased, vehicle becomes more aggressive about reaching the goal point and becomes less sensitive about the obstacles and avoiding them. Basic equation of the method is shown below.

$$\vec{F} = \vec{F}_{att} + \vec{F}_{rep} \tag{3.4}$$

Where $\vec{F}_{att}$ is attractive force and $\vec{F}_{rep}$ is repulsive force acting on the vehicle. Which can be defined in some different forms depending on the potential field definitions in literature. The potential field form used in this study is as suggested by [25], shown below. d_0 is the critical distance which obstacles closer are allowed to apply forces.

$$U_{att} = \xi \frac{d^2}{2} \tag{3.5}$$

$$U_{rep} = \begin{cases} \frac{1}{2}\eta(\frac{1}{d} - \frac{1}{d_0})^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases} \tag{3.6}$$

Where $d = |q - q_o|$; q is the position of the vehicle, q_o is position of the other obstacle or goal point. U_{rep} is repulsive potential field and U_{att} is attractive potential field. ξ and η are adjustable parameters. Forces associated with the potential fields are defined as negative gradients of the fields. Therefore;

$$\vec{F}_{att} = \xi(q - q_o) \tag{3.7}$$

$$\vec{F}_{rep} = \begin{cases} \eta(\frac{1}{d} - \frac{1}{d_0})\frac{(q-q_o)}{d^3} & d \leq d_0 \\ 0 & d > d_0 \end{cases} \tag{3.8}$$

Using the model created by this method, a goal point and some obstacles are defined and several simulations are done with both static and moving obstacles. Also parameter effect is experimented. Some simulation results with different coefficients of repulsion and attraction is shown in figure 3.13. It is seen that this coefficients can effect the vehicle path heavily. Depending on the obstacle positions, the difference between these parameters can even cause vehicle to take entirely different path.

It is seen that the virtual vehicle is capable of avoiding obstacles and reaching the goal point. After experimenting with simulations, algorithm is integrated into main program running on PC. Implementation and working principles are discussed in next section.

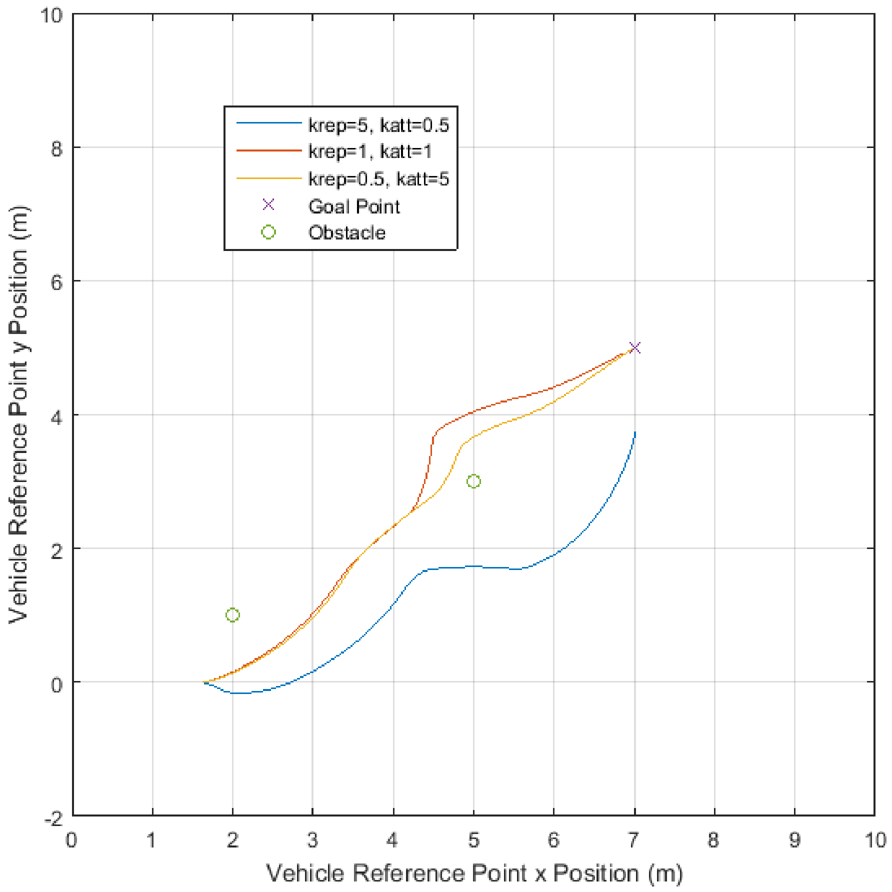


Figure 3.13 : Potential field method simulation results.

3.3.5 Processing on PC

The PC discussed on the Processing Units section, has an important role on autonomous drive. It receives data from two LIDARs and processes that data, passes the data through some algorithms and makes calculations. Sends the results of this calculations to main control unit MABX2 to be used in control of the vehicle. For this purpose, two programs runs in parallel on PC. One is written in Java, other is in Python. This programs communicates with each other while running. General flow of the Java program is shown in figure 3.14.

First, program reads the data from SICK LIDAR and parses it. To parse the data, program finds and separates arrays in data, starting with magic number which is determined by the manufacturer of the sensor. Then, iterates between these messages and finds the correct message which has the correct type. After finding the message

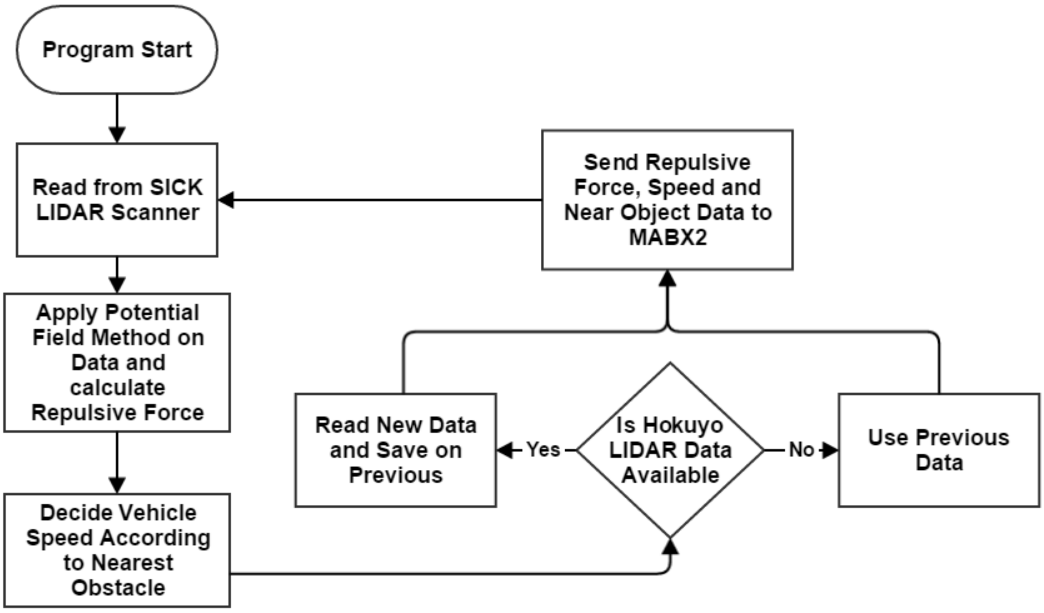


Figure 3.14 : General flow of Java program.

of proper type, using the byte offset and byte length of the desired information is found from the ethernet protocol document, provided by the manufacturer, data is extracted and converted into corresponding data type. If every step is coded in separate functions, all information in data buffer can be extracted easily, using the ethernet protocol document as reference.

After parsing the SICK LIDAR data, program finds distance of each point from the vehicle, by iterating on every point scanned. Then, it calculates repulsive force for every point, assuming as if each point is an obstacle, using a similar approach with [24]. Total repulsive force is calculated using these forces. Thus, obtaining the repulsive force mentioned in the potential field method explanation. After that, program determines desired speed for the vehicle considering distance from the closest obstacle. Stopping the vehicle if there is an obstacle too close. Then, program receives the information of if there is an obstacle at front left or front right of the vehicle by communicating with Python program. This information is determined by the Hokuyo scanner sensor. If there is no data available, program uses the previous scan data. Lastly, program sends repulsive force, desired speed and this obstacle information to MABX2 via ethernet connection.

The other program running in parallel with Java program on the vehicle, is written in Python and handles the data coming from Hokuyo scanner. There is no disadvantage for using Python, since there is no need for heavy calculations in this program. It reads and processes the serial communication data coming through USB port, determines the distances of points in scan range. Then, checks if there are any close obstacles in the predetermined left and right regions in its scan range. Lastly, sends the information to Java program through a local port by UDP. Transferring data with UDP protocol prevents freezing while waiting for data and allows asynchronous communication. General flow of the Python program is shown in figure 3.15.

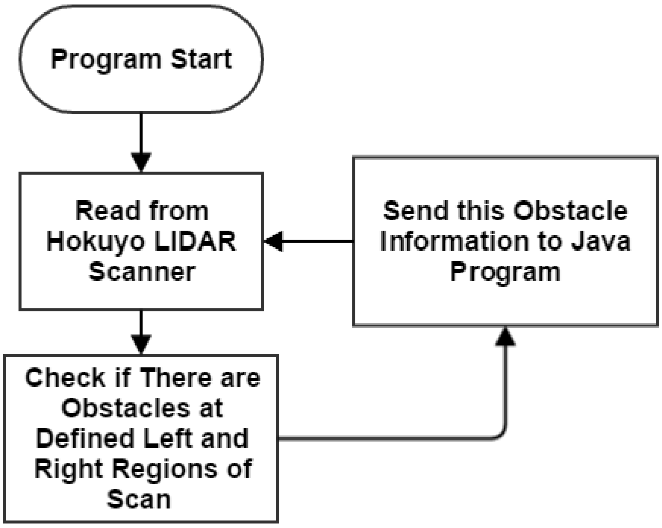


Figure 3.15 : General flow of Python program.

4. EXPERIMENTS

Real world experiments are necessary to collect data and test controllers and algorithms. Real world experiments and testing of some algorithms created in this study are explained in this section.

4.1 RC Driving

Before driving the vehicle autonomously, unmanned drive test were done by controlling the vehicle with RC transmitter. RC receiver device is connected to MABX2. RC command receive subsystem were made in simulink model to receive signals from transmitter and extract meaningful commands. Subsystem is shown in figure 4.1.

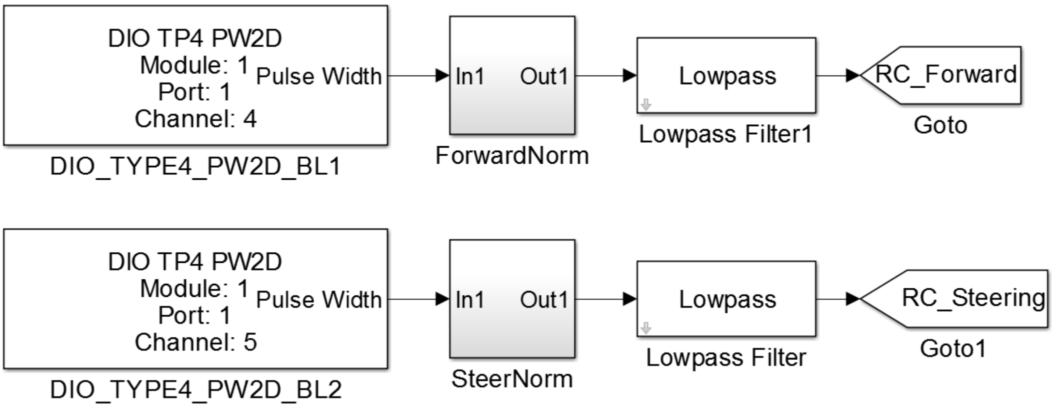


Figure 4.1 : RC command model.

Subsystem consists of signal receive, normalizing and filtering blocks. Signal receive blocks receives the pulse width of the signal sent by transmitter channels. These two signals are normalized between -1 and 1 using maximum and minimum pulse width values received from transmitter. After normalizing, signals goes through filter blocks to avoid noise and chattering caused RC communication. Finally, this filtered command signals are fed to reference speed, brake and steering signals in other control subsystems, via "go to" and "from" blocks.

RC driving experiment has taken place at the corridor in front of the laboratory. RC controlled vehicle starts from one end of the corridor and passes between box and wall. Stops when it reaches to person who controls it. Frames taken from video is shown in figure 4.2.

Figure 4.2 : RC driving experiment.

4.2 Autonomous Path Following

After making real world test for controllers and vehicle movement, path following algorithm was implemented in main simulink model and tested in laboratory.

Approximate top down map of laboratory was constructed by measuring. Two goal points were selected to make the vehicle turn the narrow corner. Results of experiment and simulation both are shown in figure 4.3.

It is seen that vehicle is able to follow the goal points. But rotation of the real vehicle is lagging compared to virtual vehicle. Meaning configured steering wheel rotation speed is not enough to meet vehicles forward speed on large rotations and should be reconfigured. Except this lag, path driven is same. Also real world experiment data matches the measured ground truth.

It is seen that vehicle is able to follow the goal points. But rotation of the real vehicle is lagging compared to virtual vehicle. It may be concluded that configured steering wheel rotation speed is not enough to meet vehicles forward speed on large

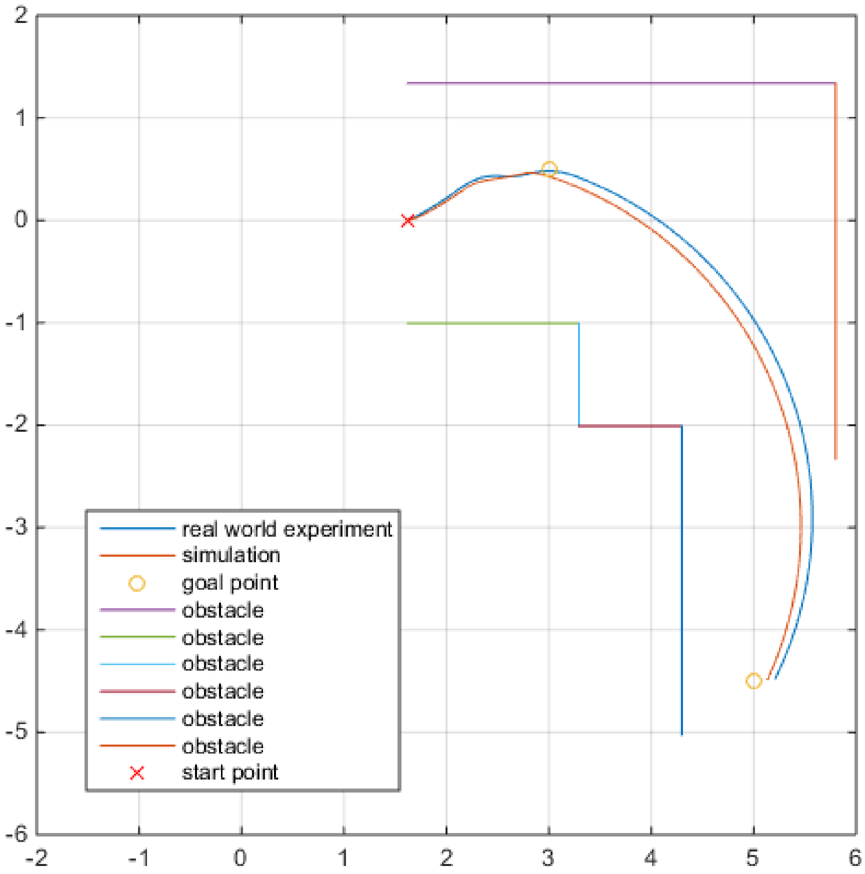


Figure 4.3 : Autonomous path following experiment.

rotations and can be reconfigured. Except this lag, path driven is same. Also real world experiment data matches the measured ground truth.

4.3 Vehicle Human Interaction Experiments

Indoor environments has many kinds of obstacles for the vehicle to avoid. The most important objects are dynamic obstacles and they are mainly moving pedestrians. The proposed approach relies heavily on LIDAR sensor for pedestrian detection. The vehicle should anticipate the human activity, should avoid from all pedestrians. On the other hand, the system should perform its goal of transporting people and goods to necessary locations within estimated time. These two goals might sometimes conflict and lead to the so-called freezing robot problem (FRP) [14]. The robot may be stuck and unable to find an alternative collision-free trajectory. One solution to this problem is to introduce pedestrian interaction to improve vehicle navigation. If an alternative collision-free path is not detected, or following current path is more cost effective (e.g., shorter), then it is inevitable to ask for a permission to pass along, as we humans normally do. The proposed flowchart of the navigation algorithm is described in figure 4.4.

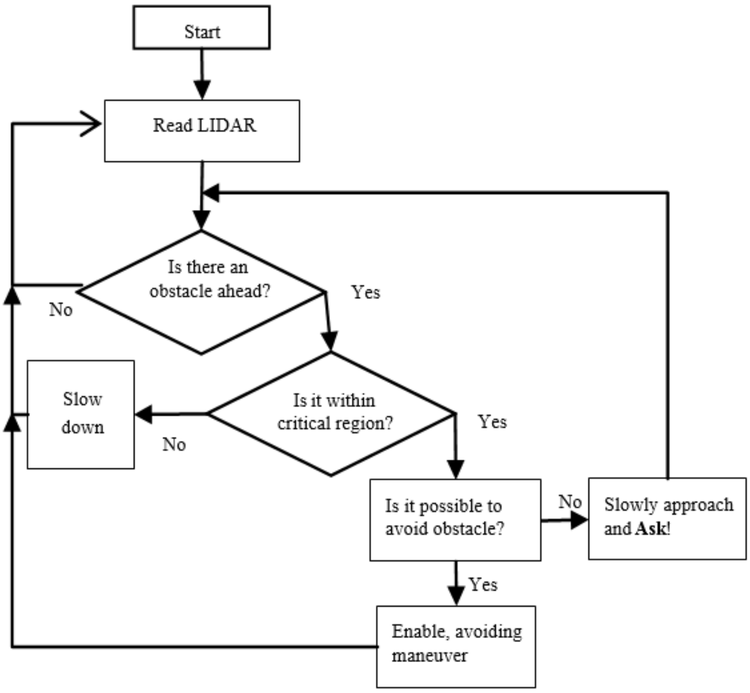


Figure 4.4 : Decision making flowchart for avoiding pedestrian obstacles.

To validate this approach a Simulink model was constructed, which consists of linear vehicle dynamics model, pedestrian model, virtual LIDAR, driver and destination seek main blocks, to simulate the crowded environment figure 4.5.

Simulink stateflow toolbox is very useful for both implementing a decision-making algorithm in Simulink model and visualizing it. A decision-making, a stateflow chart which has the algorithm proposed was constructed in Simulink. Stateflow block is shown in the figure 4.6. Algorithm used in the simulation, has 4 behaviors on 4 main states, being normal state, slow down state, avoid state and communication state.

Vehicle moves in these states according to a virtual LIDAR sensor mounted on front of the vehicle, detecting objects in sight and their distances. For decision-making, calculations about this sensor are made inside "Read LIDAR" block. Overall sight of the sensor divided into 3 virtual region separated by 60 degrees. For each region, sensor checks if any obstacle is inside or not. If there is an obstacle, calculates its distance.

Comparing this distance with critical distance parameter, sensor decides if obstacle is in critical distance. LIDAR sends this region data to decision-making stateflow.

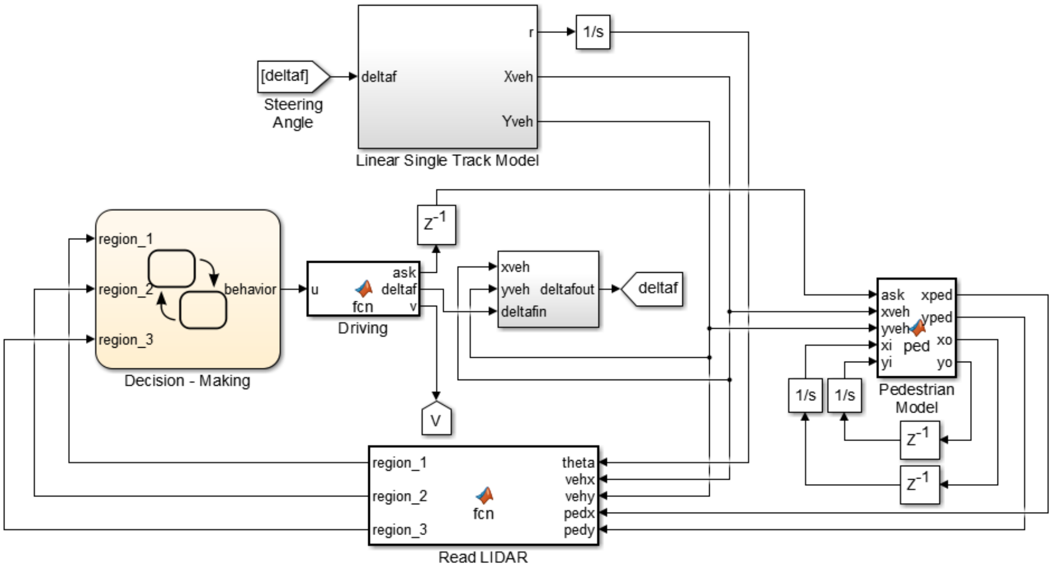


Figure 4.5 : Pedestrian interaction and avoidance simulink model.

For each state in stateflow block, vehicle has different driving routines. These routines are mainly controlled by "Driver" block which does this by determining desired speed and desired steering angle. Destination seeking block takes control after, if it is allowed by "Driver" block. It has a simple “heading error” based goal seeking algorithm implemented in it so it can turn the vehicle towards the destination if there are no obstacles on sight.

In the Simulink model, "Pedestrian" block includes the environment and pedestrian mathematical models. It is responsible for interaction between pedestrians themselves and their interaction with vehicle, resulting all of pedestrian movement in simulation. Pedestrian movement is decided by an agent control method named steering. Each agent has its own velocity and this velocity is controlled by steering vector. Steering vector can be controlled by various behaviors like seeking, separation, cohesion, wandering etc. In this simulation, seeking is implemented for the pedestrians to find their way to where they should stand and separation is implemented for them to not collide with another pedestrian or the vehicle.

Two different scenarios simulated to illustrate the benefits of the proposed approach. In first scenario, small group of people can be avoided by a basic maneuver. When vehicle detects there is an obstacle in critical region, it makes a maneuver and avoids

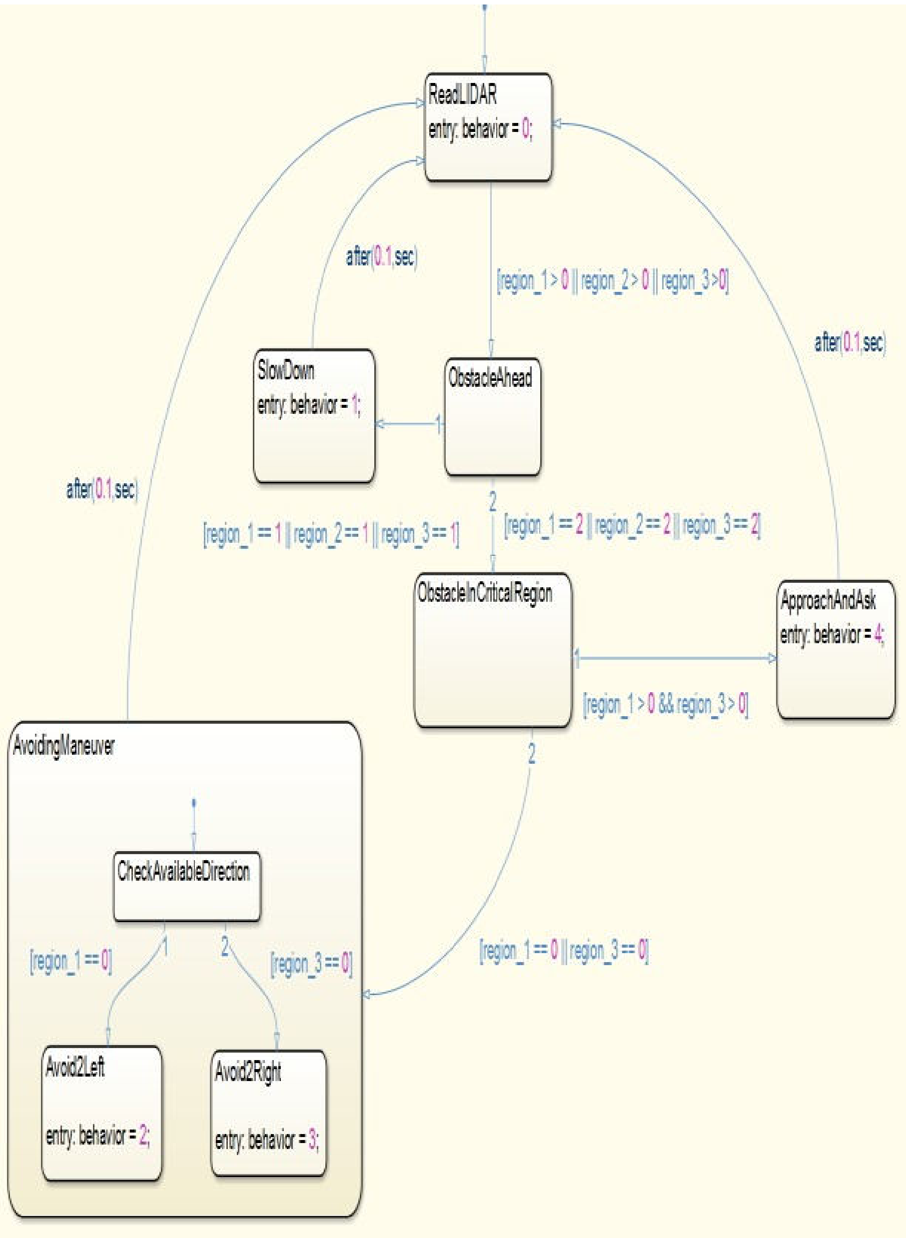


Figure 4.6 : Decision making simulink stateflow block.

it. The simulation results are plotted in MATLAB and some frames are shown in figure 4.7. Necessary information is plotted for several different time values. Rectangle is representing the vehicle, 3 lines representing the regions and circles representing the pedestrians. Region color changes according to obstacle detection and critical distance. The vehicle behavior is written upper left corner of each image.

Moving around the crowd consumes time; moreover, as the crowd is moving it might be a challenge to perform an avoiding maneuver. In second scenario however, when vehicle detects obstacle is within the critical region, it is in a situation it can't avoid the group of people by maneuvering left or right. There can be also two walls, blocking

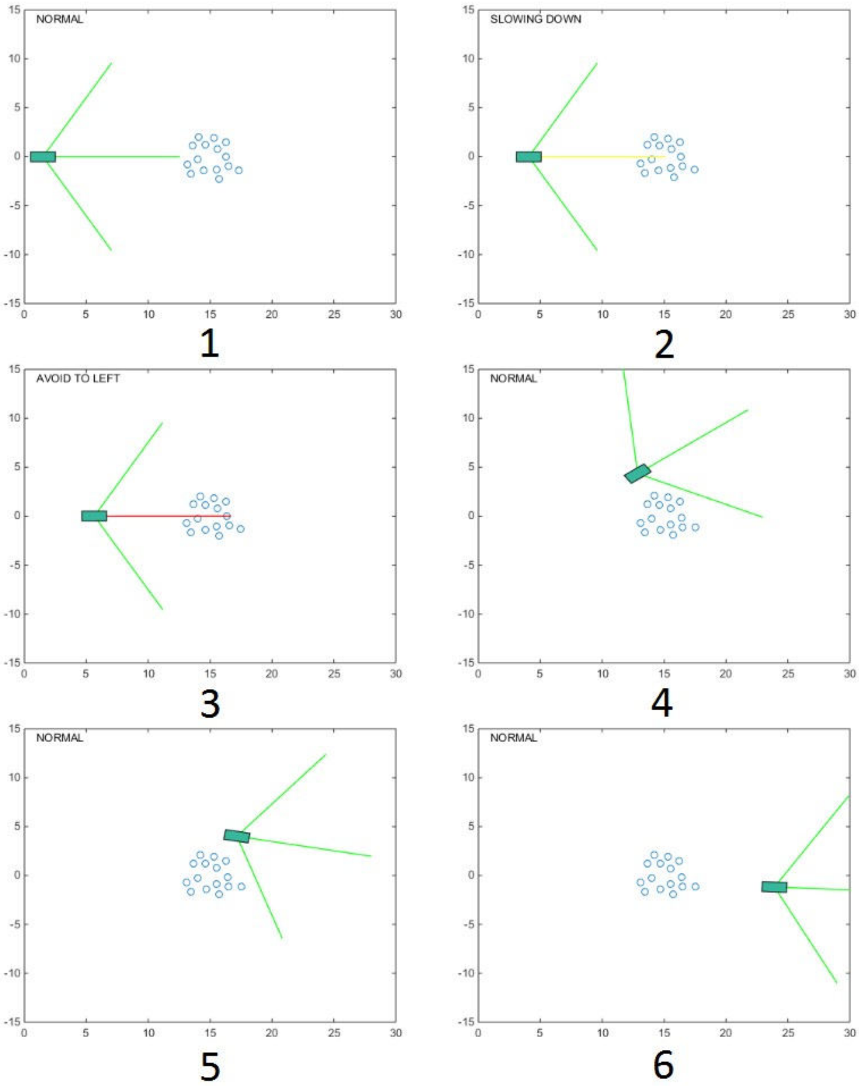


Figure 4.7 : Frames from the first scenario.

the vehicle such as maneuvering left or right is not an option for avoidance. The vehicle should go-backwards and search for an alternative collision-free trajectory. But there are such cases which there is no possible route even vehicle went backward and searched for it. To avoid the freezing robot problem, alternatively, vehicle communicates with the group. It slowly approaches people on the direction of destination and asks for permission to pass. With the people cooperating with the vehicle and walking out of its way, vehicle moves to desired location without stopping or going back. Frames from the simulation is shown in figure 4.8. The red vehicle color means that the vehicle is communicating with nearby pedestrians. Comparing first and second simulations, one can easily see the advantages of the Asking behavior. Vehicle can go to the desired goal location in less time and also easily avoid the freezing robot problem.

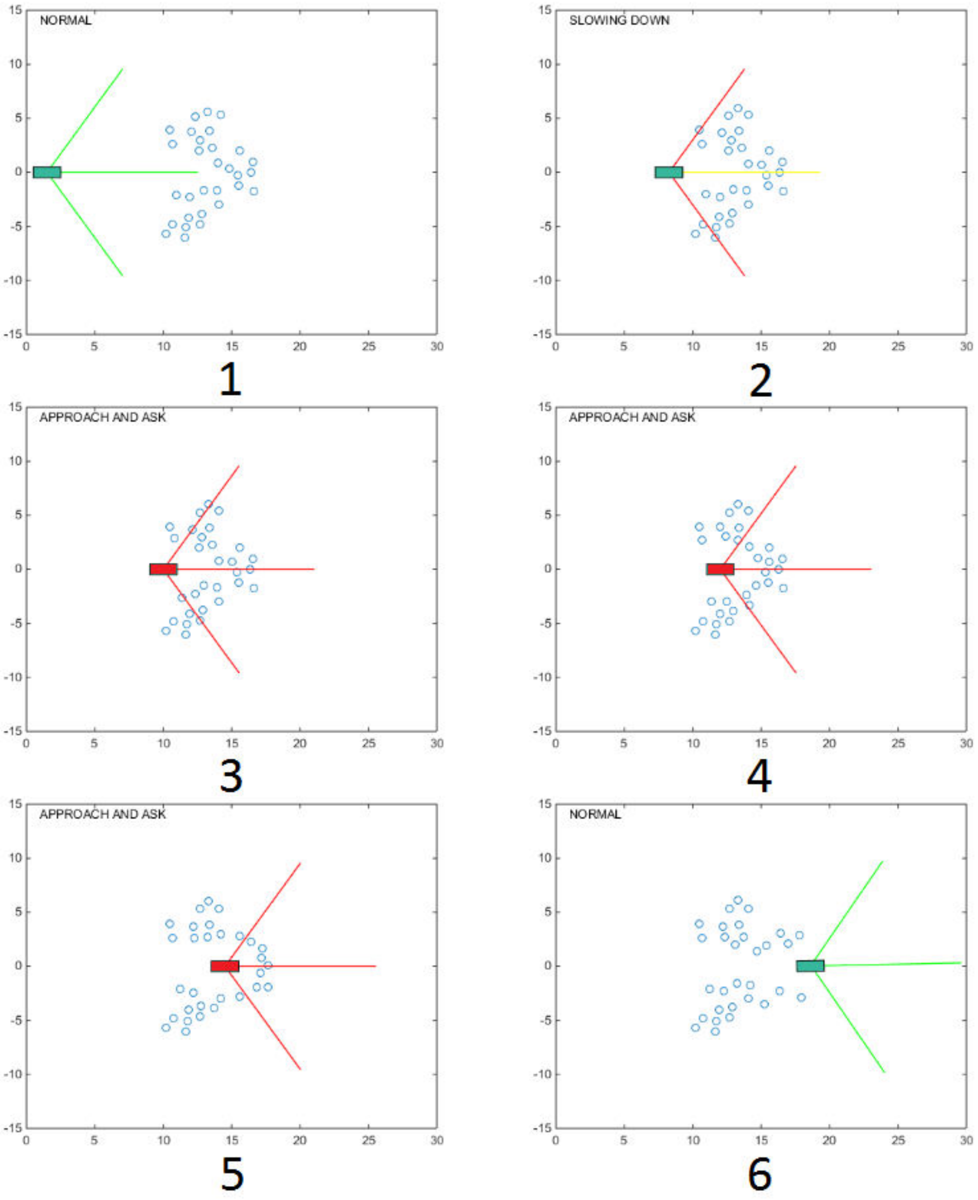


Figure 4.8 : Frames from the second scenario.

When more than one vehicle is available at the environment, priority concept similarly to road vehicles can be used. Any vehicle (or passenger) with higher priority can be given the right of the passing, and lower priority vehicles (and passengers) can be requested to move aside.

4.3.1 HIL simulation

Hardware-in-the-loop (HIL) simulation is a very effective technique to develop and optimize controllers and algorithms. First scenario mentioned above is coded along with vehicle model and the low level controllers in MATLAB. HIL simulation setup was mainly constructed for the purpose of testing different scenarios on the real vehicle's dynamics and controllers. Hence, being able to see performance of the vehicle and controllers more approximate to real world and being able to test and optimize controllers more effectively. HIL simulation was achieved with running the pedestrian simulation in real time, on MABX2. Steering and speed of vehicle dynamics model was connected to real vehicle's steering and speed control system in simulink. Vehicle is lifted up in order to prevent tires from touching the ground, making the vehicle stay stationary even when the wheels are moving.

Simulink model for controlling the vehicle's steering wheel is discussed in chapter 2. Using standard PID control in closed loop with steering angle sensor feedback. The only modification is the reference angle comes from the simulation part of the model and measured angle goes to that simulation part. Control output goes to MABX2 analog output, which is connected to DC motor driver. Speed control simulink model for the vehicle is also discussed in chapter 2. Modification for HIL simulation is similar to steering wheel, it takes the reference speed from simulation and sends the actual speed data from encoder to virtual environment in simulation.

After HIL simulation was done with the first scenario, data was recorded by dSPACE ControlDesk software and exported to MATLAB for comparing with the no hardware simulation (Figure 4.9).

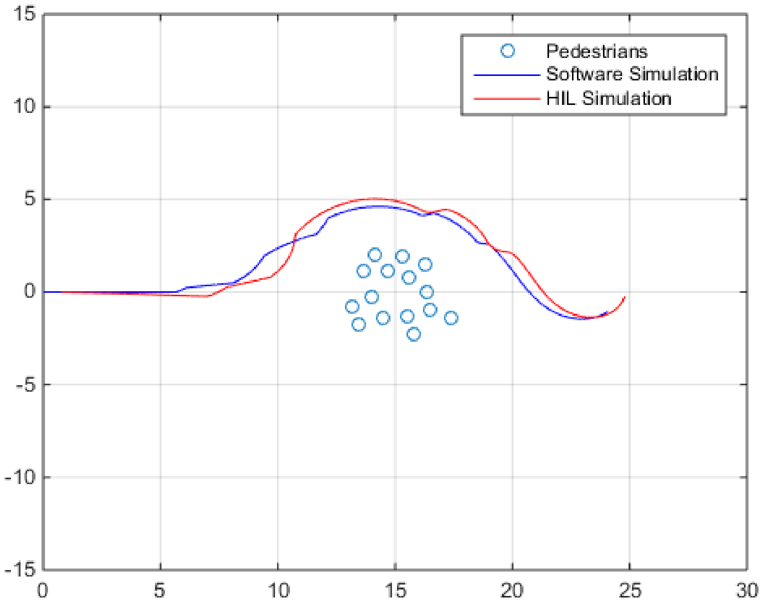


Figure 4.9 : Autonomous path following experiment.

It can be seen that the vehicle in software simulation with linear dynamic model follows a slightly different path than the vehicle in HIL simulation. Difference is due to effects like delay and laziness, and comes from real world steering wheel and engine dynamics. Also the qualification of the controllers can be done looking at the difference.

Even though algorithm development, HIL setup and simulations are done, this pedestrian interaction element is not implemented on the final form of the vehicle in this study. The reason is there are no classification algorithms implemented for LIDAR yet, meaning the vehicle cannot separate pedestrians from other static obstacles, rendering the proposed algorithm very problematic.

4.4 Autonomous Path Following and Obstacle Avoidance

After all of the sensors and autonomous driving algorithm are implemented, real world tests are done for fully autonomous driving. Corridor out of the laboratory is chosen for the test environment. Several experiment drives are done with the vehicle trying to reach a predefined goal point. With static obstacles, with dynamic obstacles. On figure 4.10, some frames from the video are shown.

In the experiment shown below, vehicle starts at near the entrance of the laboratory. A goal point of 20 m forward is defined and autonomous driving is activated. Vehicle successfully goes past two narrow passages, avoids two obstacles and travels towards the goal point.

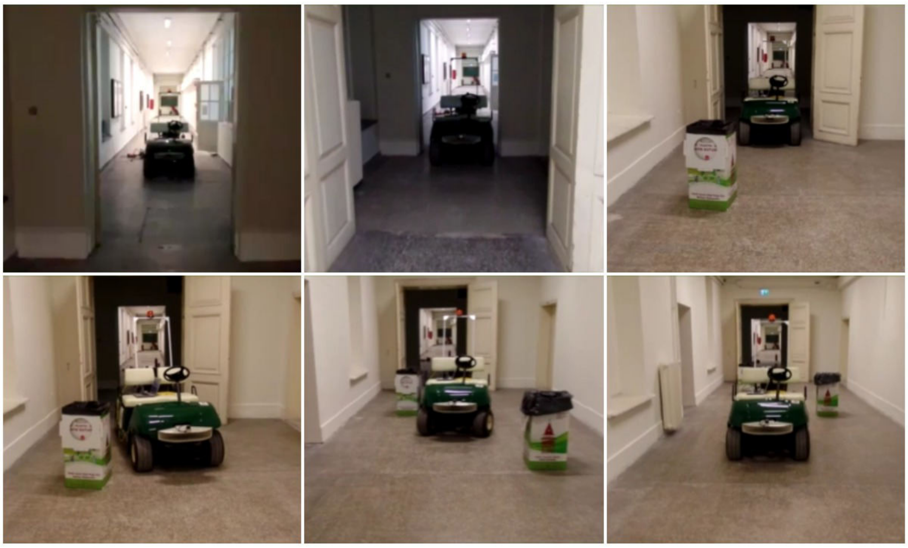


Figure 4.10 : Fully autonomous drive experiment.

Graph in the figure 4.11 shows the calculated position of the vehicle. Position reference point is the middle of the front axle on the vehicle. Although the vehicle doesn't move in y direction as much as it moves in x direction, graph y axis scale is readjusted to make the path more recognizable.

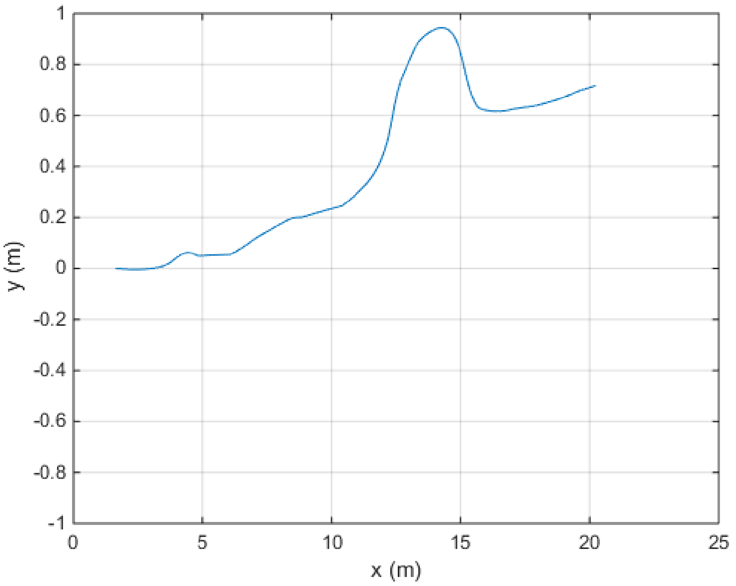


Figure 4.11 : Vehicle position data.

Graph in the figure 4.12 shows the vehicle heading while it is doing autonomous drive in this experiment. There is no significant lateral movement until it drives past two narrow passages.

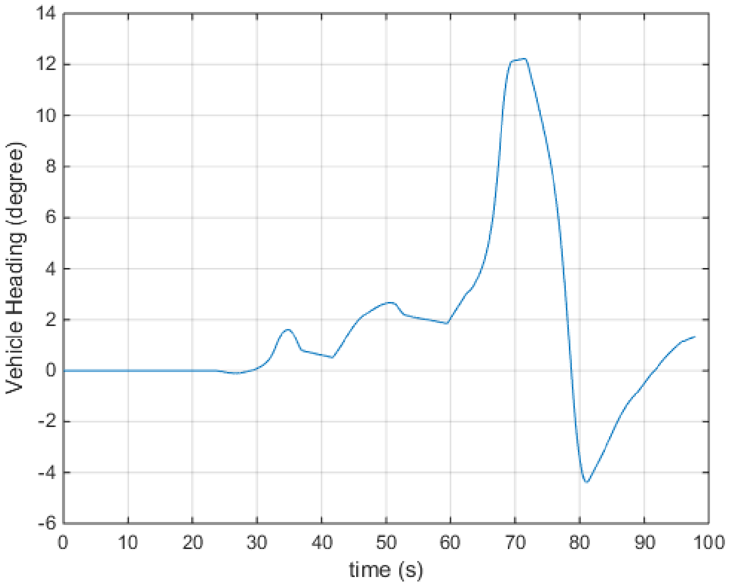


Figure 4.12 : Vehicle heading data.

As soon as it comes close to first obstacle, it does a sharp manuever to left. After it goes past the first obstacle, it tries to align itself to goal point. After a small time, it comes close to second obstacle and does a manuever to right. Finally, drives past the last obstacle and turns straight to the goal point.

The last graph in the figure 4.13 shows the obstacle warning information obtained by processing the data received from Hokuyo LIDAR. It is discussed in the chapter 2, that Hokuyo LIDAR is used to detect obstacles on the front left and front right of the vehicle. The information shown in the graph is used for obstacle avoidance. If the value is 1 it means there is an obstacle very close on the front right. If the value is 2 it means there is an obstacle very close on front left. If it is 3, there is an obstacle on both front left and front right. When the obstacle gets out of the FOV of SICK LIDAR, this data stops vehicle from maneuvering onto that obstacle.

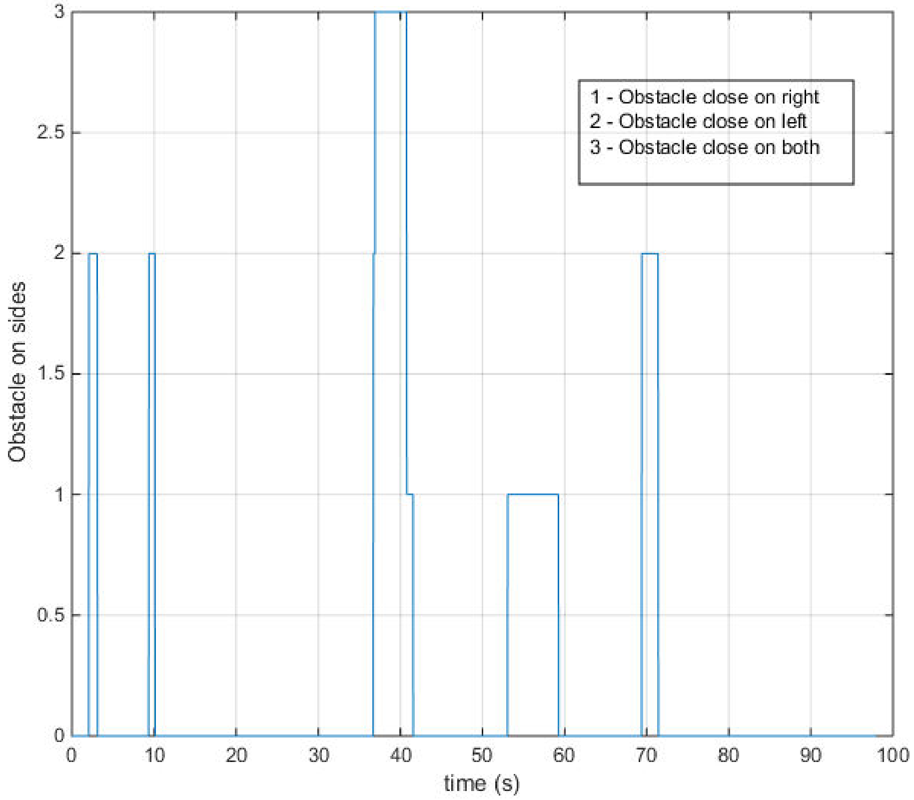


Figure 4.13 : Hokuyo LIDAR obstacle information on sides.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Need of autonomous vehicles grows continuously. Hence, researches on autonomous vehicles are constantly increasing in number and quality. Different variations are developed with different operation environments or using different equipment, both on roads and indoor environments such as airports or shopping malls. These indoor autonomous vehicles can help elderly, disabled, or people with heavy luggage.

Small electric vehicles have high maneuverability, low noise and no gas emission. Therefore they are best choice for operating in indoor environment. Considering this, a golf cart is modified to move unmanned. Several actuators, drivers and a main controller is integrated into system. Both software and hardware components used in the design and implementation process are explained. Control methods for these actuators and component placement are discussed and simulink models are explained.

Power needed for all these actuators, and a PC are provided by three main batteries and an inverter placed under the passenger seat. Two 12V batteries are connected in series to obtain 12V and 24V. Third 12V battery is connected to inverter for powering up the PC. Connections are made through fuses and a main switch. Moreover, as an emergency measure, an RC Relay board is implemented on the power line. This relay provides remote emergency stop capability to the vehicle.

With the pedestrians in operating environment, problems such as separating moving and static obstacles, narrow paths, unpredictable human behavior should be considered. To identify the environment and obstacles ahead of vehicle, LIDAR sensor is used. LIDAR sensor has advantages such as high precision, object detection, object movement speed measurement. On top of the high precision LIDAR sensor, an additional low precision and wider scan angle LIDAR sensor is used for more stable and safer maneuvering behavior.

Navigation within indoor environments with pedestrians possesses additional problems to be handled. Experiments about interactions between vehicle and pedestrians are done in order to propose efficient ways for navigating between crowds. Decision making and specific tools for implementing this algorithm into the simulation are discussed. Two different scenarios are simulated and results are illustrated to support the proposed navigation method. Moreover, a HIL setup created is discussed for optimizing vehicle controllers and algorithms in desired scenarios. Real vehicle's dynamics and controllers result in the closest resemble of real world experimenting data, leading to effective optimization of controllers for different types of problems possible to encounter when moving in crowded environments.

Several path following and decision algorithms are discussed and simulation results are shown. Implemented path following algorithm, potential field method is fully dynamic and based on goal points. Using this method, vehicle can move between goal points and avoid both static and moving obstacles. In the potential field method, vehicle calculates total force acting on it and moves according to direction of the force. Speed is determined by the distance of the nearest object and increases when the distance increases. Vehicle stops when an object is too close or it arrives the destination.

In this thesis, the design and control of a human transporter that is planned to work in indoor environments is presented. Various stages of development are shown with methods and equipment used.

5.2 Future Work

Position calculation is one of the most critical steps of autonomous drive. Sensors used in this study, will not be enough to travel in large indoor environments because of the long distance travel error. For this reason, future improvements should be done in position calculation, using computer vision based landmarking techniques.

Another topic to make improvements on, is object classification for navigation in crowded environment. HIL simulations and algorithm development are done but the proposed vehicle pedestrian interaction method is not implemented. Reason is the vehicle has no object classification capability yet. Separating pedestrians from other obstacles will result more effective obstacle avoidance.

In terms of sensing the environment, the vehicle has at most 240 degree view in front of it. 240 degree LIDAR is 1 layer and cannot be used in the middle of the vehicle because its scanning angle is straight. There are better but more expensive LIDAR sensors on the market with shifted scanning angle in order to provide 360 degree view around the vehicle. They scan up to 64 layer and this point cloud data is very close to 3d scan of the environment which object classification can be easily and reliably done. Also, vehicle having the capability of seeing 360 degree around it, leads to much safer and more reliable autonomous drive. As an alternative solution, one can also place multiple cheap small LIDAR sensors around the vehicle. But cheap LIDAR sensors scan 1 layer and because of that, object classification will not be reliable.

REFERENCES

[1] **Khatib, O.** (1985). Real-time obstacle avoidance for manipulators and mobile robots, *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 5, pp.90–98.

[2] **Buehler, M.**, **Lagnemma, K. and Singh, K.S.** (2007). *The 2005 DARPA Grand Challenge: The Great Robot Race. Springer Tracts in Advanced Robotics.*, Springer.

[3] **Thrun, S.**, **Montemerlo, M.**, **Dahlkamp, H.**, **Stavens, D.**, **Aron, A.**, **Diebel, J.**, **Fong, P.**, **Gale, J.**, **Halpenny, M.**, **Hoffmann, G.**, **Lau, K.**, **Oakley, C.**, **Palatucci, M.**, **Pratt, V.**, **Stang, P.**, **Strohband, S.**, **Dupont, C.**, **Jendrossek, L.**, **Koelen, C.**, **Markey, C.**, **Rummel, C.**, **van Niekerk, J.**, **Jensen, E.**, **Alessandrini, P.**, **Bradski, G.**, **Davies, B.**, **Ettinger, S.**, **Kaehler, A.**, **Nefian, A. and Mahoney", P.** (2006). Winning the darpa grand challenge, *Journal of Field Robotics*.

[4] **Urmson, C.**, **Anhalt, J.**, **Bagnell, D.**, **Baker, C.**, **Bittner, R.**, **Clark, M.N.**, **Dolan, J.**, **Duggins, D.**, **Galatali, T.**, **Geyer, C.**, **Gittleman, M.**, **Harbaugh, S.**, **Hebert, M.**, **Howard, T.M.**, **Kolski, S.**, **Kelly, A.**, **Likhachev, M.**, **McNaughton, M.**, **Miller, N.**, **Peterson, K.**, **Pilnick, B.**, **Rajkumar, R.**, **Rybski, P.**, **Salesky, B.**, **Seo, Y.W.**, **Singh, S.**, **Snider, J.**, **Stentz, A.**, **Whittaker, W.**, **Wolkowicki, Z.**, **Ziglar, J.**, **Bae, H.**, **Brown, T.**, **Demitrish, D.**, **Litkouhi, B.**, **Nickolaou, J.**, **Sadekar, V.**, **Zhang, W.**, **Struble, J.**, **Taylor, M.**, **Darms, M. and Ferguson, D.** (2008). Autonomous driving in urban environments: Boss and the Urban Challenge, *Journal of Field Robotics*.

[5] **Broggi, A.**, **Medici, P.**, **Cardarelli, E.**, **Cerri, P.**, **Giacomazzo, A. and Finardi, N.** (2010). Development of the control system for the Vislab Intercontinental Autonomous Challenge, *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp.635–640.

[6] **Wuthishuwong, C.**, **Silawatchananai, C. and Parnichkun, M.** (2009). Navigation of an intelligent vehicle by using stand-alone GPS, compass and laser range finder, *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pp.2121–2126.

[7] **Guvenc, L.**, **Uygan, I.M.C.**, **Kahraman, K.**, **Karaahmetoglu, R.**, **Altay, I.**, **Senturk, M.**, **Emirler, M.T.**, **Karci, A.E.H.**, **Guvenc, B.A.**, **Altug, E.**, **Turan, M.C.**, **Tas, .S.**, **Bozkurt, E.**, **Ozguner, .**, **Redmill, K.**, **Kurt, A. and Efendioglu, B.** (2012). Cooperative Adaptive Cruise Control Implementation of Team Mekar at the Grand Cooperative Driving

Challenge, *IEEE Transactions on Intelligent Transportation Systems*, *13*(3), 1062–1074.

[8] **Ferrara, A.**, **Gebennini, E. and Grassi, A.** (2014). Fleet sizing of laser guided vehicles and pallet shuttles in automated warehouses, *International Journal of Production Economics*, *157*, 7 – 14, the International Society for Inventory Research, 2012.

[9] **Martínez-Barberá, H. and Herrero-Pérez, D.** (2010). Autonomous navigation of an automated guided vehicle in industrial environments, *Robotics and Computer-Integrated Manufacturing*, *26*(4), 296 – 311.

[10] **Ji, M. and Xia, J.** (2010). Analysis of vehicle requirements in a general automated guided vehicle system based transportation system, *Computers and Industrial Engineering*, *59*(4), 544 – 551.

[11] **Roy, N.**, **Gordon, G. and Thrun, S.** (2003). Planning under uncertainty for reliable health care robotics., *In FSR*.

[12] **Chong, Z.J.**, **Qin, B.**, **Bandyopadhyay, T.**, **Wongpiromsarn, T.**, **Rankin, E.S.**, **Ang, M.H.**, **Frazzoli, E.**, **Rus, D.**, **Hsu, D. and Low, K.H.** (2011). Autonomous personal vehicle for the first- and last-mile transportation services, *2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)*, pp.253–260.

[13] **Leibe, B.**, **Seemann, E. and Schiele, B.** (2005). Pedestrian detection in crowded scenes, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp.878–885 vol. 1.

[14] **Trautman, P. and Krause, A.** (2010). Unfreezing the robot: Navigation in dense, interacting crowds, *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp.797–803.

[15] **Liu, W.**, **Weng, Z.**, **Chong, Z.**, **Shen, X.**, **Pendleton, S.**, **Qin, B.**, **Fu, G.M.J. and Ang, M.H.** (2015). Autonomous vehicle planning system design under perception limitation in pedestrian environment, *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pp.159–166.

[16] **Al-nasur, S.J.** (2006). New Models for Crowd Dynamics and Control, *Ph.D. thesis*, Virginia Polytechnic Institute and State University.

[17] **Parry, G.W.**, (2007), The Dynamics of Crowds.

[18] **Markowski, M.J.** (2008). Modeling Behavior in Vehicular and Pedestrian Traffic Flow, *Ph.D. thesis*, University of Delaware.

[19] **Pritikana Das, M. Parida, V.K.K.** (2014). Review of Simulation Techniques for Microscopic Mobility of Pedestrian Movement, *Trends in Transport Engineering and Applications*.

[20] **Shin, S.**, **Suh, J. and Yeo, H.** (2013). Development of ECM-based microscopic pedestrian movement model, *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp.249–254.

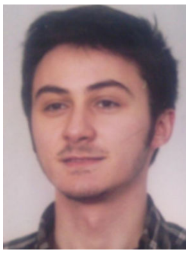
[21] **Wilschut , T.** (2011). An Obstacle Avoidance Algorithm for a Mobile Robot Based upon the Potential Field Method, *Ph.D. thesis*, Eindhoven University of Technology.

[22] **Koren, Y. and Borenstein, J.** (1991). Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation, *IEEE Conference on Robotics and Automation*, pp.1398–1404.

[23] **Hellstrom, T.** (2011). Robot Navigation with Potential Fields , Umea University Department of Computing Science.

[24] **Ladha, S. and Kumar, D. K.and Bhalla, P.and Jain, A.and Mittal , R. K.** (2011). Use of LIDAR for Obstacle Avoidance by an Autonomous Aerial Vehicle , Birla Institute of Technology and Science.

[25] **Khatib, O.** (1985). Real-time obstacle avoidance for manipulators and mobile robots, *IEEE Conference on Robotics and Automation*, pp.500–505.



CURRICULUM VITAE

Name Surname: Şükrü Yaren Gelbal

Place and Date of Birth: Istanbul 13.02.1993

E-Mail: sygelbal@gmail.com

B.Sc.: Okan University - Department of Electrical and Electronics Engineering, Mechatronics Engineering Programme (2010-2014)

List of Publications and Patents:

- Emirler, M. T., Uygan, I. M. C., Gelbal, S. Y., Gozu, M., Boke, T. A., Aksun Guvenc, B., Guvenc, L., 2015, ”Vehicle Dynamics Modelling and Validation for Hardware-in-the-Loop Testing of Electronic Stability Control”, International Journal of Vehicle Design (Accepted)
- Gelbal, S. Y., Kececi, E. F., Altug, E., 2016 , ”Design and HiL Setup of an Autonomous Vehicle for Crowded Environments”, IEEE International Conference on Advanced Intelligent Mechatronics - AIM, Banff, Alberta, Canada, July 12-15.
- Senkul, A. F., Gelbal, S. Y., Altug, E., 2016, ”Manufacturing and Flight Tests of a Quadrotor UAS with Tiltable Rotors”, International Conference on Unmanned Aircraft Systems - ICUAS, Arlington, Virginia, USA, June 7-10.
- Senkul, A. F., Gelbal, S. Y., Altug, E., 2015, ”Tiltable Rotor System Design and Control for Aerial Robots” (in Turkish), Turkey Robotics Conference - TORK, Istanbul, October 26-27.
- Gelbal, S. Y., Bodur, K., Altinpinar, O. V., Sahin, K., Hot, M., Sezer, V., 2015, ”Design of a Tracked Land Vehicle with Autonomous Driving and Aiming Capability” (in Turkish), Automatic Control National Meeting - TOK, Denizli, September 10-12.
- Emirler, M. T., Uygan, I. M. C., Gelbal, S. Y., Gozu, M., Boke, T. A., Aksun Guvenc, B., Guvenc, L., 2014, ”Vehicle Dynamics Modelling and Validation for a Hardware-in-the-Loop Vehicle Simulator” (in Turkish), 7th Automotive Technologies Congress - OTEKON, Bursa, May 26-27.