

**A NOVEL OBSTACLE AVOIDANCE APPROACH
FOR NONHOLONOMIC GROUND VEHICLE AUTONOMY**

Ph.D. THESIS

Volkan SEZER

Department of Control and Automation Engineering

Control and Automation Engineering Programme

JULY 2012

**A NOVEL OBSTACLE AVOIDANCE APPROACH
FOR NONHOLONOMIC GROUND VEHICLE AUTONOMY**

Ph.D. THESIS

**Volkan SEZER
(504072109)**

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Metin GÖKAŞAN

JULY 2012

**HOLONOM OLMAYAN KARA TAŞITINDA OTONOMLUĞU
SAĞLAMAK İÇİN YENİ BİR ENGELDEN SAKINMA YAKLAŞIMI**

DOKTORA TEZİ

**Volkan SEZER
(504072109)**

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Metin GÖKAŞAN

TEMMUZ 2012

Volkan SEZER, a Ph.D. student of ITU Graduate School of Science 504072109 successfully defended the thesis entitled “**A NOVEL OBSTACLE AVOIDANCE APPROACH FOR NONHOLONOMIC GROUND VEHICLE AUTONOMY**”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Metin GÖKAŞAN**
Istanbul Technical University

Jury Members : **Prof. Dr. İbrahim EKSİN**
Istanbul Technical University

Prof. Dr. Galip CANSEVER
Yıldız Technical University

Prof. Dr. Ramazan Nejat TUNCAY
Okan University

Prof. Dr. Seta BOGOSYAN
Istanbul Technical University

Date of Submission : **13 June 2012**

Date of Defense : **27 July 2012**

To my son,

FOREWORD

I would like to gratefully acknowledge the supervision of Prof. Dr. Metin Gökaşan for his important support and constructive comments throughout this work. His belief in me during this thesis highly encouraged my research enthusiasm. I would like to express my deep and sincere gratitude to Prof. Dr. Seta Bogosyan and Assoc. Prof. Dr. Pinar Boyraz for their constructive criticism and excellent advices during the preparation of this thesis. I would also like to thank the members of my thesis committee; Prof. Dr. İbrahim Eksin and Prof. Dr. Galip Cansever for their guidance and valuable advices. My sincere thanks are due to Prof. Dr. Ata Mugan for his effort and support throughout the development phase of our autonomous ground vehicle .

My special thanks go to my colleagues; Hasan Heceoğlu, Çağrı Dikilitaş and Ziya Ercan for their friendliness and support. Without them, it would not be possible to develop such an autonomous vehicle. I would also like to thank Mechatronics Education and Research Center (MEAM) family, especially to Alev Keskin and Reyhan Kenan, who are the very talented technicians, for their practical contributions.

Finally, I would like to thank deeply my wife Tuba Sezer, my mother Ayşe Sezer, my father İsmail Sezer and my sister Sibel Sezer for their unconditional love, support and belief in me throughout my PhD. I would also like to thank my son, who will be born in a few months, for his beautiful energy flowing through every cell of my body.

July 2012

Volkan SEZER
(Mechatronics Engineer, MSc)

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD.....	ix
TABLE OF CONTENTS.....	xi
ABBREVIATIONS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xix
SUMMARY	xxiii
ÖZET	xxv
1. INTRODUCTION	1
1.1 Motivation.....	1
1.2 Semi-Autonomous Ground Vehicle.....	2
1.3 Fully Autonomous Ground Vehicle.....	3
1.4 Purpose of the Thesis.....	6
1.5 Contributions of the Thesis.....	7
1.6 Thesis Outline.....	8
2. KINEMATIC AND DYNAMIC VEHICLE MODELING	9
2.1 Introduction	9
2.2 Model Parameters	9
2.3 Kinematic Vehicle Model	11
2.4 Dynamic Vehicle Model	12
2.4.1 Pacejka tire model	13
2.4.2 Steering angle projection block	15
2.4.3 Vehicle dynamics block.....	16
2.4.4 Kinematics and geometry block	16
2.4.5 Longitudinal wheel slip calculation block.....	16
2.5 Conclusions	17
3. DEVELOPMENT OF THE EXPERIMENTAL PLATFORM "OTONO- BİL"	19
3.1 Introduction	19
3.2 Minimum Design Requirements for an Unmanned Ground Vehicle Development.....	21
3.3 Electrical Modifications.....	22
3.3.1 Sensors and controllers.....	22
3.3.2 Additional power system	23
3.3.3 Vehicle interface system	24
3.4 Mechanical Modifications	25
3.4.1 Using computer environment in mechanical design process.....	25
3.4.2 Automatic braking mechanism.....	26

3.4.3 Automatic steering mechanism	27
3.5 Communication and Interface Software	27
3.6 Conclusions	30
4. A NOVEL OBSTACLE AVOIDANCE ALGORITHM: "FOLLOW THE GAP METHOD"	33
4.1 Introduction	33
4.2 Problem Definition	35
4.2.1 Point robot approach.....	36
4.2.2 Calculation of "distance to obstacle"	37
4.3 Follow the Gap Method	38
4.3.1 Calculation of the gap array and finding the maximum gap	38
4.3.2 Calculation of the gap center angle	41
4.3.3 Calculation of the final heading angle	44
4.4 Simulations	47
4.4.1 Artificial potential field theory	47
4.4.2 Comparisons	50
4.5 Experimental Tests	53
4.6 Conclusions	55
5. A NEW FUZZY SPEED PLANNING METHOD FOR SAFE NAVIGATION	59
5.1 Introduction	59
5.2 Speed Planning Methods for APF	61
5.2.1 Method A.....	61
5.2.2 Method B	61
5.3 Fuzzy Decision Making for Speed Planning	62
5.3.1 Method overview	62
5.3.2 Design of Fuzzy-1 block	63
5.3.3 Design of Fuzzy-2 block	64
5.4 Simulations and Comparisons	66
5.5 Conclusions	69
6. A NEW FUZZY SPEED CONTROL STRATEGY CONSIDERING THE LATERAL VEHICLE DYNAMICS	71
6.1 Introduction	71
6.2 Fuzzy Speed Controller	73
6.2.1 Design of Fuzzy-1 block	73
6.2.2 Design of Fuzzy-2 block	75
6.3 Simulations and Comparison.....	77
6.4 Experimental Results	80
6.5 Conclusions	81
7. SIMULATIONS AND EXPERIMENTAL RESULTS OF DESIGNED STRATEGIES TOGETHER	83
7.1 Simulations	83
7.1.1 Simulation scenario 1 - static obstacles.....	83
7.1.2 Simulation Scenario 2 - dynamic obstacles.....	84
7.2 Experimental Results	86

7.2.1 Experimental test scenario 1 - static obstacles	87
7.2.2 Experimental test scenario 2 - dynamic obstacles.....	88
7.3 Conclusions	90
8. CONCLUSIONS.....	91
REFERENCES.....	93
APPENDICES.....	99
APPENDIX A.1	101
APPENDIX A.2	101
CURRICULUM VITAE.....	105

ABBREVIATIONS

ACC	: Adaptive Cruise Control
APF	: Artificial Potential Field
CAbS	: Collision Avoidance by Steering
CAD	: Computer Aided Design
CAN	: Controller Area Network
CC	: Cruise Control
CMbB	: Collision Mitigation by Braking
CoG	: Center of Gravity
DARPA	: Defense Advanced Research Projects Agency
FGM	: Follow the Gap Method
FIS	: Fuzzy Inference System
GPS	: Global Positioning System
IMU	: Inertial Measurement Unit
LIDAR	: Light Detection and Ranging
MABX	: Microautobox
NI	: National Instruments
PCB	: Printed Circuit Board
PRM	: Probabilistic Roadmap
RADAR	: Radio Detection and Ranging
RF	: Radio Frequency
RNDF	: Route Network Definition File
RRT	: Rapidly Exploring Random Tree
UGV	: Unmanned Ground Vehicle
VFF	: Virtual Force Field
VFH	: Vector Field Histogram

LIST OF TABLES

	<u>Page</u>
Table 2.1 : Vehicle Parameters.	10
Table 3.1 : Sensor List of Otonobil for State Estimation and Mapping.	22
Table 3.2 : List of Computational Components of Otonobil.	23
Table 4.1 : APF Parameters.	49
Table 4.2 : Average of Simulation Results ($d_0 = 25m$).	52
Table 5.1 : Rule Base for Fuzzy-1 Block. (<i>For Inputs-VN: Very Near, N: Near, M: Medium, F: Far, VF: Very Far, FL: Far Left, L: Left, OF: On Front, R: Right, FR: Far Right. For Output-VS: Very Small, S: Small, M: Medium, L: Large, VL: Very Large</i>).....	64
Table 5.2 : Rule Base for Fuzzy-2 Block. (<i>For Inputs- L: Little, M: Medium, H: High. For Output- VS: Very Small, S: Small, M: Medium, L: Large, VL: Very Large</i>	66
Table 5.3 : Average Cost Comparison.....	69
Table 6.1 : Rule Base for Fuzzy-1 Block.	75
Table 6.2 : Rule Base for Fuzzy-2 Block.	77
Table 6.3 : Normalizing Gain Factors.	78

LIST OF FIGURES

	<u>Page</u>
Figure 1.1 : Percentage of the Traffic Accidents of USA [1].	1
Figure 1.2 : Roadmap of Active and Passive Safety Systems and their Safety Potential [2].	2
Figure 1.3 : Authority Sharing in a Semi-Autonomous Vehicle [3].	4
Figure 1.4 : Different UGV Implementations for Specific Aims.	4
Figure 1.5 : System Architecture for Team MIT Vehicle for DARPA GC 2007 [4].	5
Figure 1.6 : Sub Modules of the Designed UGV.	6
Figure 2.1 : Base Vehicle Used for Modeling and Simulations.	9
Figure 2.2 : Vehicle Parameters for Kinematic Model.	11
Figure 2.3 : Vertexes and Related Parameters for Kinematic Model.	12
Figure 2.4 : Kinematic Vehicle Model Simulation Results ($X_{init} = 0, Y_{init} = 0, \theta_{init} = 0$).	13
Figure 2.5 : Single Track Vehicle Model Parameters.	13
Figure 2.6 : Dynamic Vehicle Model Block Diagram.	14
Figure 2.7 : Electric Motor Toque-Speed Curve.	17
Figure 2.8 : Dynamic Vehicle Model Simulation Results. ($X_{init} = 0, Y_{init} = 0, \theta_{init} = 0, v_{init} = 0$).	18
Figure 3.1 : Designed UGV, "Otonobil".	19
Figure 3.2 : Otonobil's Additional Components from Front and Rear Side.	20
Figure 3.3 : Computational Components of Otonobil.	24
Figure 3.4 : Additional Batteries and Power Box.	24
Figure 3.5 : Interface Circuit.	25
Figure 3.6 : Human Interface System.	25
Figure 3.7 : Mechanical Modifications.	26
Figure 3.8 : CAD Drawing of the Vehicle.	26
Figure 3.9 : Automatic Brake System for Otonobil.	27
Figure 3.10 : Automatic Steering System for Otonobil.	27
Figure 3.11 : Sensor and Computer System and Their Communication in Otonobil.	28
Figure 3.12 : Outside Communication Scheme of Otonobil.	28
Figure 3.13 : UMD-B12P25 RF Communication Module.	29
Figure 3.14 : Interface Software for Inside of the Vehicle.	29
Figure 3.15 : Interface Software for Outside of the Vehicle.	30
Figure 4.1 : Obstacle Enlargement for Point Robot Representation.	37
Figure 4.2 : Distance to Obstacle Parameters and Geometry.	37
Figure 4.3 : Steps of the Follow the Gap Method.	38

Figure 4.4 : Obstacle Representation.	39
Figure 4.5 : Gap Border Evaluation.	40
Figure 4.6 : Gap Border Parameters.	41
Figure 4.7 : Gap Center Angle Representation.	42
Figure 4.8 : Gap Center Angle Parameterization.	42
Figure 4.9 : Comparison of Φ_{gap_c} and $\Phi_{gap_c_basic}$ Gap Angle Calculation.	44
Figure 4.10 : Obstacles with Goal Point.	45
Figure 4.11 : Final Angle with Respect to Minimum Distance and α Coefficient ($\Phi_{gap_c} = \pi/4$ radian and $\Phi_{goal} = -\pi/4$ radian).	46
Figure 4.12 : Simulation Results of FGM for Different Obstacle Configurations..	48
Figure 4.13 : APF Forces.	48
Figure 4.14 : Trajectories of APF and FGM in Local Minimum Scenarios.	50
Figure 4.15 : Comparison of FGM, APF and A* Shortest Path Trajectories.	52
Figure 4.16 : Collision Avoidance Norm and Traveled Distance Values for Follow the Gap, Follow the Gap-basic and Artificial Potential Fields Methods.	53
Figure 4.17 : Test Field and First Obstacle Configuration.	54
Figure 4.18 : Experimental Results of Test-1.	54
Figure 4.19 : Experimental Results of Test-2.	55
Figure 4.20 : Experimental Results of Test-3.	56
Figure 5.1 : Role of Desired Speed Determination in a UGV.	60
Figure 5.2 : Cascade Fuzzy Connection for Speed Planning.	63
Figure 5.3 : Membership Functions for Inputs and Output of Fuzzy-1 Block.	64
Figure 5.4 : Fuzzy Surface for Fuzzy-1 Block.	65
Figure 5.5 : Membership Functions for Inputs and Output of Fuzzy-2 Block.	66
Figure 5.6 : Fuzzy Surface for Fuzzy-2 Block.	67
Figure 5.7 : Simulation Snapshot from Simulation Environment.	68
Figure 5.8 : PI Controller for Low Level Speed Control.	68
Figure 5.9 : Trajectory Comparison of Three Algorithms.	69
Figure 5.10 : Total Cost Values for 20 Monte Carlo Simulations.	70
Figure 6.1 : Role of Speed Control in a UGV.	73
Figure 6.2 : Cascade Fuzzy Connection for Speed Control.	74
Figure 6.3 : Membership Functions for Inputs and Output of Fuzzy-1 Block.	75
Figure 6.4 : Fuzzy Surface for Fuzzy-1 Block.	76
Figure 6.5 : Membership Functions for Inputs and Output of Fuzzy-2 Block.	77
Figure 6.6 : Fuzzy Surface for Fuzzy-2 Block.	78
Figure 6.7 : Vehicle Speed Tracking Comparison of Structure-1 and Structure-2 (Simulation 1).	79
Figure 6.8 : Vehicle Speed Tracking Comparison of Structure-1 and Structure-2 (Simulation 2).	79
Figure 6.9 : Unstable Vehicle Trajectory of Structure-1 (Simulation 2).	80
Figure 6.10 : Unstable Vehicle Trajectory of Structure-1 (Simulation 2).	80
Figure 6.11 : Experimental Results of Proposed Low Level Speed Controller.	81
Figure 7.1 : Vehicle Trajectory for Simulation Scenario-1.	84
Figure 7.2 : Reference Speed and Real Speed with Steering and Pedal Inputs for Simulation Scenario-1.	84

Figure 7.3 : Vehicle Trajectory in 2D, for Simulation Scenario-2.	85
Figure 7.4 : Vehicle Trajectory in 3D, with Additional Time Axis for Simulation Scenario-2.....	85
Figure 7.5 : Reference Speed and Real Speed with Steering and Pedal Inputs for Simulation Scenario-2.	86
Figure 7.6 : Test Field for Experiments.	87
Figure 7.7 : Vehicle Trajectory for Experimental Test Scenario-1.....	87
Figure 7.8 : Reference Speed and Real Speed with Steering and Pedal Inputs for Experimental Test Scenario-1.	88
Figure 7.9 : Vehicle Trajectory for Experimental Test Scenario-2.....	89
Figure 7.10 : Vehicle Trajectory in 3D, with Additional Time Axis for Experimental Test Scenario-2.	89
Figure 7.11 : Reference Speed and Real Speed with Steering and Pedal Inputs for Experimental Test Scenario-2.	90
Figure A.1 : Cosine Theorem Notation.....	101
Figure A.2 : Apollonius Theorem Notation	101

A NOVEL OBSTACLE AVOIDANCE APPROACH FOR NONHOLONOMIC GROUND VEHICLE AUTONOMY

SUMMARY

The majority of vehicular accidents are attributed to human error. Active safety systems which are designed to eliminate the human errors in ground vehicles have been growing year by year. Most of these systems are about vehicle stability. Besides this, studies about collision warning and collision avoidance systems have been very popular in recent years. These kind of semi-autonomous applications are the sub-packages of the self driving full autonomous cars of the future. Obstacle avoidance algorithms have a very critical task in both semi-autonomous and full-autonomous applications.

In this thesis, a novel obstacle avoidance algorithm, "Follow the Gap Method" (FGM) is designed for nonholonomic ground vehicles. Obstacle avoidance ability is one of the most important subsystems of autonomous robots. These algorithms are different from the classical path planning algorithms since no prior information is given to the robot. Nature of the obstacle algorithm should be reactive because, coordinates of any obstacle may change at any time and can not be known previously. This prevents the use of the classical optimization techniques in obstacle avoidance problems. The algorithm must compute just the next action in every instant, based on the current context. There are too many studies about this subject in literature.

The most important property of the designed FGM algorithm is maximizing the distance to obstacle value as much as possible. Before starting the algorithm, robot dimension is added to obstacle dimensions in order to convert the problem into a point robot obstacle avoidance problem. The FGM has 3 main parts. First part of the algorithm calculates the gaps around the robot. Nonholonomic constraints and robot field of view constraints are considered in this calculation. In second part, maximum gap is selected and heading angle to the center of the maximum gap is calculated. Two different methods are illustrated for this calculation. In the first method, Apollonius and Cosinus theorems are used while a basic average is calculated in the second method. In the last part of the FGM, a final reference angle is calculated by using both the center of the maximum gap angle and the goal angle. This calculation is done by using a fusing function which uses the distance to obstacle parameter inside. Gap center angle and the goal angle may be in different directions. However, the final angle approaches the gap center angle when the distance to obstacle approaches zero. This guarantees the obstacle avoidance if there is at least one gap around the robot. The fusing function is designed specifically to reflect this property.

Comparison of the algorithm is done with the Artificial Potential Fields (APF) method and the A* shortest path algorithm. A safety metric which depends on the distance to obstacle value is defined and used for fair comparison between methods. Series of Monte Carlo simulations where the obstacles and the goal coordinate values are

given randomly to each model are performed for a fair comparison. According to the simulation results, the FGM results in safer trajectories than the others. This safety is measured numerically using the designed safety metric. The FGM is free from the local minimum problem which can be seen in the APF and APF based algorithms such as the virtual force field (VFF) method. The FGM has only one tuning parameter α , which adjusts the ratio between obstacle avoidance and arriving at the goal point. The reactive nature of the algorithm provides successful results even in dynamic obstacle scenarios.

Most of the obstacle avoidance algorithms in literature calculate only the reference heading vector for obstacle avoidance. This can be enough in low speed values but in higher values, the dynamic properties of the robot become important. This time, performance of the obstacle avoidance algorithm depends on not only the maneuver but also the speed of the robot. For this reason, speed decision is studied in a separated section and a new speed planning strategy is developed. Two cascade connected fuzzy inference systems (FIS) are used for speed planning. First block calculates a risk factor by using the distance to obstacle and angle to obstacle values while the second FIS is used for stabilizing the yaw dynamics. A safety metric is designed for Monte Carlo simulations which shows that this new method is safer than the classical methods.

Low level speed controller also affects the avoidance performance. The aim of the low level speed controller is to calculate the throttle and brake pedal signals in order to track the desired speed. A new low level speed controller using fuzzy logic is designed in this thesis. Similar cascaded FIS structure is used here as in speed planning algorithm. Three input values for FIS are; speed error, integral of the speed error and the steering angle values. According to the simulation results, the new method results in safer yaw dynamics especially for the aggressive maneuver scenarios.

All designed algorithms are simulated and compared with the existing methods. Kinematic and dynamic vehicle modeling issues and the used vehicle parameters are explained in a separated section. Each algorithm is coded using C programming language into the S-functions using Matlab/Simulink environment.

All designed algorithms are tested using experimental autonomous ground vehicle platform. The experimental platform is a full autonomous ground vehicle which is converted from a conventional electric vehicle during this thesis work. Conversion procedure is divided into two groups as electrical and mechanical modifications. Sensors, computer system, additional batteries and interface circuit of the vehicle are explained in electrical modifications part. Automatic steering, automatic braking, computer aided drawing of the vehicle are explained in mechanical modifications part. Apart from these, two different interface softwares are developed for in-vehicle and out-vehicle communication for this thesis. "Control Desk" tool of Dspace company is used for in-vehicle interface software which is designed for testing the in-vehicle communication, data recording and in-vehicle safety. C++ programming language with Visual Studio platform is used for out-vehicle communication software, which runs in a host PC. This software sends the desired coordinate values or emergency signal to the vehicle, can drive the vehicle remotely and can visualize the critical data of the vehicle.

Each of the designed algorithms runs real-time in Microautobox 1401/1501 hardware. Algorithms are coded using C programming language, integrated in C-S functions with 10ms cycle time.

HOLONOM OLMAYAN KARA TAŞITINDA OTONOMLUĞU SAĞLAMAK İÇİN YENİ BİR ENGELDEN SAKINMA YAKLAŞIMI

ÖZET

Günümüzde yapılan trafik kazalarının önemli bir kısmı insan hatalarından kaynaklanmaktadır. Kazaları önlemeye yönelik geliştirilen aktif güvenlik sistemleri giderek çoğalmaktadır fakat hala yeterli seviyede değildir. Şu anda var olan güvenlik sistemleri daha çok, araç stabilitesini sağlamaya yöneliktir. Bunun yanında, olası bir çarpışmayı önceden tahmin edip uyarı veren veya kısa süreliğine otoriteyi sürücüden devralan yarı otonom sistemler üzerindeki çalışmalar son yıllarda oldukça hız kazanmıştır. Bu çalışmalarla varılması öngörülen son nokta ise, kendi kendini sürebilen tam otonom araçlardır. Her iki yapıda da engelden kaçma algoritmalarının önemi oldukça fazladır.

Bu tezde, holonom olmayan otonom kara taşıtları için "Boşluğu Takip Et" (BTE) isimli yeni bir engelden kaçma algoritması tasarlanmıştır. Engelden kaçma algoritmaları, otonom yapıların en önemli alt sistemlerinden biridir. Bu algoritmalar, klasik yörünge planlama algoritmalarından farklı olarak, ortamdaki engellerin hareket esnasında algılanmasıyla dinamik olarak çalışmak zorundadır. Yani önceden belirli bir harita yoktur ve hızlı cevap verme gereksinimi, klasik optimizasyon tekniklerinin uygulanmasını engellemektedir. Konuyla ilgili günümüze kadar pek çok çalışma mevcuttur.

Bu tezde tasarlanan BTE algoritmasının en önemli özelliği, engellerden kaçarken, engellerle robot arasındaki mesafeyi mümkün olduğunca fazla kılmasıdır. Algoritma hesaplama aşamasına geçmeden önce, robotun boyutunu, etrafındaki engellerin boyutuna ekleyerek, kendisini bir nokta haline getirmektedir. Bu hazırlık aşamasından sonra, BTE algoritması üç ana kısımdan oluşmaktadır. İlk kısımda öncelikle aracın etrafındaki boşlukların listesi çıkartılır. Bu boşluklar hesaplanırken, aracın holonom olmamaktan dolayı gelen kısıtları ve sensörlerin görüş açıları da hesaplara katılmaktadır. BTE algoritmasının ikinci kısmında ise robotun çevresindeki en geniş boşluk seçilir ve bu boşluğun merkezine doğru yönelen açı hesaplanır. En geniş boşluğun merkezi hesaplanırken iki farklı metod gösterilmiştir. Bu metodların ilkinde, Apollonius ve Kosinüs teoremlerinden faydalanılmış, diğerinde ise basitçe ortalama alınmıştır. Her ikisinin de yol açtığı yörüngeler simülasyonlarla gösterilmiştir. Algoritmanın üçüncü ve son kısmında ise, en geniş boşluğun merkezine yönelen açı ile, hedefe giden açı değerleri kullanılarak bir referans yönelim açısı hesaplanır. Bu iki açı değeri, aracın en yakın engelle olan mesafesine bağlı bir birleştirme fonksiyonuna sokularak sonuca gidilir. En geniş boşluğun merkezi ve varmak istenilen hedef noktası, birbiriyle çelişen doğrultularda olabilir. Fakat tasarlanan birleştirme fonksiyonu, robot ile engel arasındaki mesafe sıfıra giderken, matematiksel olarak boşluğun merkezine yakınsadığından, etrafta boşluk olduğu sürece engellerden sakınmayı garanti eder.

Algoritmanın diğer metodlarla güvenlik anlamında adil bir şekilde karşılaştırılabilmesi için, araçla engeller arasındaki mesafeye bağlı bir fonksiyonun birinci normu

karşılaştırma metriği olarak kullanılmıştır. Karşılaştırmalar, en yaygın bilinen engelden kaçma algoritmalarından olan yapay potansiyel alan (YPA) metodu ve en kısa yol stratejisi olan A* algoritmalarıyla, engellerin ve hedefin rastgele belirlendiği Monte Carlo simülasyon tekniğiyle yapılmıştır. Yapılan karşılaştırmalarda, diğer metodlara oranla daha güvenli yörüngeler elde edilmiştir. Tasarlanan algoritma, YPA ve YPA türevli sanal kuvvet alanları metodundaki lokal minimum probleminden bağımsızdır. Ayrıca, aracın holonomik olmayan kısıtları ve aracın görüş açısı da boşlukların hesabında kullanılarak, araç görebildiği ve holonomik olmayan kısıtlarına rağmen gidebileceği yönler doğru yönlendirilir. Bunun yanında, algoritmanın ne kadar hedef odaklı, ne kadar engelden kaçma odaklı olacağı da kullanıcı tarafından dışarıdan girilen bir ayar parametresiyle belirlenebilmektedir. Ayar parametre sayısının 1 olması da kullanım için kolaylık sağlamaktadır. Algoritmanın reaktif yapısı, dinamik engellerden kaçma konusunda da başarılı sonuçlar elde edilmesini sağlamıştır. BTE algoritması, sadece holonom olmayan kara taşıtlarına değil, engelden kaçma gereksinimi duyulan tüm robotik uygulamalarda da kullanılabilir.

Literatürdeki engelden kaçma algoritmalarının en büyük eksikliklerinden biri, stratejinin sadece robotun manevrasına yönelik geliştirilmesidir. Bunun temel nedeni, algoritmaların genellikle düşük hızlarda çalışan mobil robotlar üzerinde geliştirilmiş olması ve kinematik çözümlerin yeterli sayılmasıdır. Fakat robot hızı arttıkça, robotun dinamik etkileri de artmaktadır.

Örneğin bu tezde kullanılan otonom kara aracı platformunda, engelden kaçmak için hesaplanan direksiyon açısının uygulanması esnasında aniden yüksek bir hız referansı girilip gaza basılırsa, araç stabilitesi bozulabilir ve araç savrulabilir. Yoğun engelli bir ortamda aracın daha yavaş bir şekilde manevrasını gerçekleştirilmesi istenir. Engelden kaçma algoritmasının performansı, aracın dinamik özellikleri de göz önüne alındığında, aracın hızıyla da doğrudan alakalıdır. Bu nedenle bu tezde, sadece aracın yöneliminin belirlenmesinin değil, aracın referans hızına karar verilmesi konusunda da yeni bir metod geliştirilmiştir. Bu metodda, bulanık mantık yöntemi kullanılarak, aracın en yakın engelle olan mesafesi ve araca olan açısı bilgileriyle bir risk faktörü hesaplanır. Bir başka bulanık çıkarıcıda da bu risk faktörüyle direksiyon açısı bilgisi birlikte kullanılarak yeni bir hız karar mekanizması tasarlanmıştır. İki bulanık çıkarıcı birbirlerine kaskat bağlanmıştır. Bu sayede, üç giriş için karar mekanizmasının tasarım zorluğu ortadan kaldırılmış ve iki adet iki girişli karar mekanizması tasarlanmıştır. Bu sayede toplam kural sayısı da önemli ölçüde azaltılmıştır. Aracın engelle olan mesafesi ve savrulma oranını içeren bir fonksiyon kullanılarak karşılaştırma için bir güvenlik metriği oluşturulmuştur. Yapılan Monte Carlo simülasyonları sonucunda, klasik metodlara göre daha güvenli sonuçlar elde edilmiştir.

Aracın alt seviye hız kontrolcüsü de engelden kaçma performansında doğrudan rol oynamaktadır. Alt seviye kontrolcüsünün görevi, verilen hız referansına göre aracın pedal girişlerine verilmesi gereken değerleri hesaplamaktır. Bu kontrolcü için de yeni bir bulanık çıkarım yapısı tasarlanmış ve bu yapının içerisine de hız hatası, hız hatasının integrali ve hız direksiyon açısı bilgileri yerleştirilmiştir. Hatanın türevi yerine integralinin kullanılmasının nedeni, literatürde var olan ve hatanın kendisi ve integralini kullanan fakat direksiyon sinyalini göz önüne almayan benzer yöntemle adil olarak karşılaştırma yapılabilmesi içindir. Tasarlanan metodda, iki bulanık çıkarıcı birbirlerine kaskat bağlanmıştır. Bu sayede üç giriş için karar mekanizmasının tasarım zorluğu ortadan kaldırılmış ve iki adet iki girişli karar mekanizması tasarlanmıştır. Bu

sayede toplam kural sayısı da önemli ölçüde azaltılmıştır. Alt seviye hız kontrolörünün performansı için, dinamik araç modeli kullanılarak, karşılaştırmalı simülasyonlar yapılmıştır. Yapılan karşılaştırmalarda yeni metodun direksiyon açısından bağımsız çalışan klasik metoda göre, özellikle agresif manevra yapılan bölümlerde daha güvenilir sonuçlar verdiği gösterilmiştir.

Tasarlanan tüm yeni algoritmalar, simülasyon ortamında denenip, diğer metodlarla karşılaştırılmıştır. Bu simülasyonlar esnasında kullanılan araç kinematik ve dinamik modelleri ve araç parametreleri tez içerisinde ayrı bir bölümde açıklanmıştır. Modelleme için, Matlab/Simulink® ortamı kullanılmıştır. Tüm algoritmalar, C programlama dili kullanılarak, Simulink blokları içerisinde, S-fonksiyonlar yardımıyla entegre edilmiştir.

Tezde tasarlanan tüm algoritmalar ayrıca deneysel sistem üzerinde de gerçekleştirilmiştir. Deneysel sistem, tez çalışması esnasında tam otonom bir kara aracına dönüştürülen klasik bir elektrikli araçtır. Bu otonom aracın dönüştürülmesiyle ilgili yapılan çalışmalar da tezde ayrıntılı olarak gösterilmiştir. Yapılan dönüştürme işlemi, elektriksel ve mekanik modifikasyonlar olmak üzere, iki ana başlıkta incelenmiştir.

Elektriksel modifikasyonlar kısmında, araca otonomluk için sonradan takılan tüm sensörler ve bilgisayar sistemleri yer almaktadır. Tez kapsamında tasarlanan tam otonom kara aracına; etrafın algılanabilmesi, aracın konumlandırılması ve araç durumlarının ölçülebilmesi için çeşitli sensörler yerleştirilmiştir. Ayrıca sinyal seviyelerindeki uyumsuzluk ve kullanıcı arayüzü için de basit bir elektronik devre tasarlanmıştır. Son olarak, ekstra takılan tüm sensör, motor ve bilgisayar sistemleri için harici bir güç sistemi oluşturulmuş ve bu sistemler aracın kendi bataryalarından bağımsız olarak beslenmiştir.

Mekanik modifikasyonlar kısmında ise, aracın fren ve direksiyon sisteminin otomatik hale getirilmesi yer alır. Her iki sistem için bazı mekanik tasarımlar yapılmış ve uygun tork-hız karakteristiğini sağlayan elektrik motorlarıyla bu sistemler otomatik hale getirilmiştir. Bunun yanında, sensör yerleşimi ve koruyucu parçaların tasarımı için aracın bilgisayar ortamındaki çizimleri tasarımlarda kolaylık sağlamıştır.

Bunların dışında, aracın iç haberleşmesi ve dış haberleşmesi için iki ayrı arayüz yazılımı geliştirilmiştir. Araç içi arayüz için Dspace firmasının "ControlDesk" isimli yazılım paketi kullanılmıştır. Bu yazılımın amacı, araç içindeki haberleşmeyi test etmek, veri toplamak ve araç içi güvenliği sağlamaktır. Aracın dışarıyla haberleşmesi içinse, C++ programlama diliyle Visual Studio ortamında yazılan bir arayüz programı geliştirilmiştir. Bu yazılım ile araca gitmesi istenen koordinatlar kablosuz olarak gönderilebilmekte, araç uzaktan sürülebilmekte, kritik bilgiler araçtan alınıp görselleştirilmekte ve araca acil durum sinyali gönderilebilmektedir.

Sonuç olarak tez kapsamında, 3 temel konuda yenilikler getirilmiştir. İlki, yeni bir engelden kaçma algoritmasının tasarımı, ikincisi yeni bir hız referans belirleme yaklaşımı ve sonuncusu da yeni bir alt seviye hız kontrol yaklaşımının geliştirilmesidir. Bu tez kapsamında tasarlanan algoritmalar, araç içerisinde yer alan Microautobox 1401/1501 donanımı içerisinde gerçek zamanlı olarak çalışmaktadır. Algoritmaların koşturulması için, C programlama diliyle yazılan S-fonksiyonlar, Simulink bloklarıyla entegre bir şekilde çalışmaktadır. Bütün çevrim için 10ms'lik bir örnekleme zamanı kullanılmıştır.

1. INTRODUCTION

1.1 Motivation

The majority of vehicular accidents are attributed to human error. Based on British and American crash data, 93 percent of crashes were either wholly or partly caused by human factors such as driver error, intoxication, or inattention. Semi-autonomous and fully autonomous vehicle guidance would be one of the best approaches to prevent this massive cost and human tragedy. Figure 1.1 shows the percentage of the traffic accident sources of USA from [1]. As it can be seen, most of the accidents are directly related with driver mistakes.

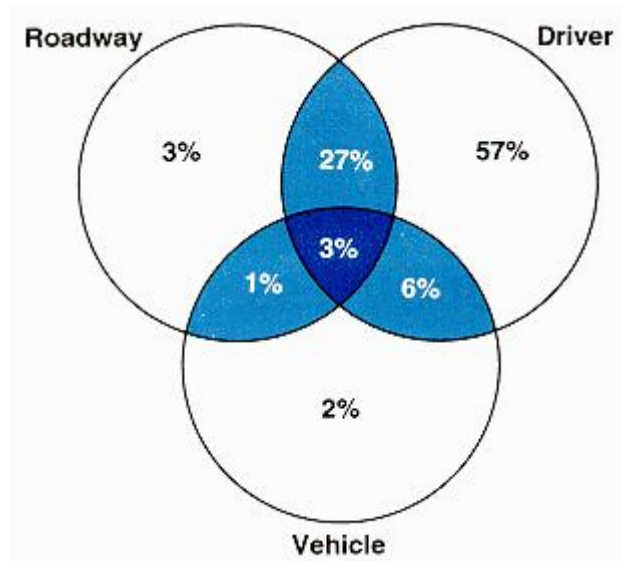


Figure 1.1: Percentage of the Traffic Accidents of USA [1].

Autonomy of ground vehicles is not a novel idea and it has been studied for a long time. Most of the conventional vehicles cover active safety systems which take over the authority from driver for safer operation. Today's safety systems work mostly on the stability level and therefore rely on sensors that measure the states of the vehicle such as the yaw rate [5]. Future developments will focus on active guidance systems that link the vehicle to its environment. This trend mainly originates from the rapid development of environmental sensor systems such as Radio Detection And Ranging

(RADAR), Light Detection And Ranging (LIDAR), or video and effective algorithms for sensor fusion, object detection and tracking.

A roadmap of driver assistance and safety systems and their safety potential are illustrated in Figure 1.2.

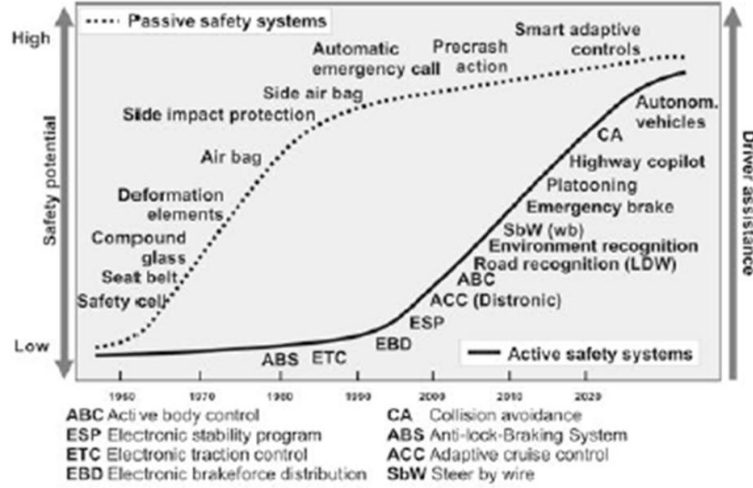


Figure 1.2: Roadmap of Active and Passive Safety Systems and their Safety Potential [2].

According to the Figure 1.2, driver assistance ratio is growing each year and the autonomous vehicles are the latest predicted technology for ground vehicles.

1.2 Semi-Autonomous Ground Vehicle

The conventional vehicles which are equipped with some of the above active safety systems can be thought as semi-autonomous vehicles. Semi-autonomous vehicles take the authority partially or fully from the human driver in dangerous situations. The decision to use automation to assist or replace a human operator in safety-critical tasks must account with the technological capabilities of the sensor control subsystems, the autonomy capabilities and preferences of the human operator.

Collision avoidance is a system for warning and avoidance of a pending collision [6]. This term is generally used in active safety area. The equivalent version of this word in autonomous vehicles is “obstacle avoidance”. Both these two words can be used for the same meaning in different sources.

Collision avoidance system has an extended functionality compared to the Obstacle and Collision Warning. An autonomous intervention takes over the control of the

vehicle in critical situations in order to avoid an accident. Longitudinal and lateral control is done by the system during the defined time while the dangerous event takes place [7].

Collision Avoidance systems, as a subsequent step to collision mitigation, are one of the great challenges in the area of active safety for road vehicles. Collision mitigation functions usually intervene when a collision is unavoidable, and actuate e.g. the brake system in order to reduce the consequences of a collision by reducing the impact speed. For collision mitigation functions it is generally true that the safety benefit becomes larger when the impact speed reduction is larger. The mitigational effect is achieved by reducing the amount of collision energy by reducing the collision speed. The fundamental mechanism for collision avoidance is, however, different, as here the trajectory of the involved vehicle(s) is changed in way to avoid an impact. Principally, this can be done in two separate ways, which are addressed within the two sub projects Collision Mitigation by Braking (CMbB) and Collision Avoidance by Steering (CAbS). For Collision Mitigation Systems a special situation has been identified where Collision Avoidance potential exist, when the motion of surrounding traffic is studied and taken into account in the decision making. In that manner, CMbB will judge a threat situation even from an "escape path" point of view and thereby exhibits collision avoidance potential [8].

Collision avoidance systems in a semi-autonomous vehicle should share the authority with the driver. A risk estimation for decision making for activating the collision avoidance algorithm should also be done. [9] illustrates a risk estimation method for collision avoidance system. In [3], optimal trajectory is calculated with Model Predictive Control and this trajectory is then used for calculating the authority management between driver and controller. Figure 1.3 illustrates the diagram of this approach in which the "K" value determines the ratio of driver and controller commands.

1.3 Fully Autonomous Ground Vehicle

A fully autonomous ground vehicle or in other words an unmanned ground vehicle UGV is essentially an autonomous robot but is specifically a vehicle that operates on

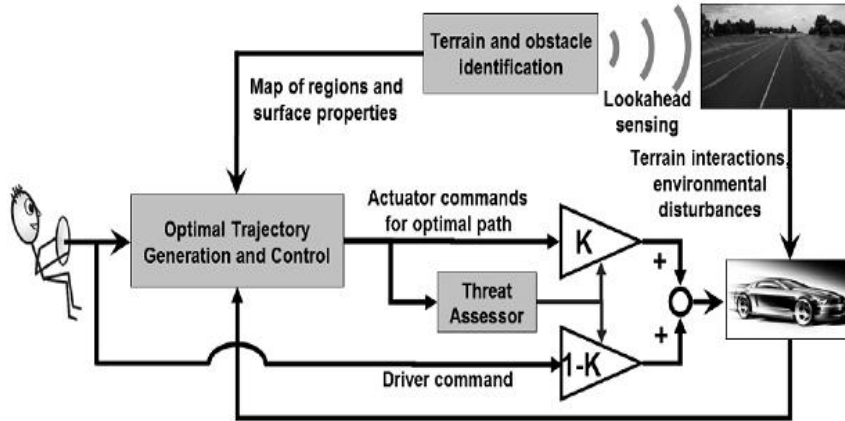


Figure 1.3: Authority Sharing in a Semi-Autonomous Vehicle [3].

the surface of the ground. A fully autonomous robot in the real world has the following abilities [10].

- Perception - Viewing the world and interpreting what it sees.
- Localization - Keeping track of the robot's position.
- Global Path Planning - Finding the fastest and safest way to get from start to goal.
- Local Navigation - Making sure the robot doesn't tip, drive into holes or bump into obstacles.

The list can be extended with different capabilities but these are the minimal abilities that a UGV must have. Figure 1.4 illustrates some UGV examples for different aims.



(a) Carnegie Mellon's UGV Winner of DGC07 [11]. (b) "Guardium" Military UGV [12]. (c) Irobot's Autonomous Robotic Vacuum [13].

Figure 1.4: Different UGV Implementations for Specific Aims.

The biggest competition for UGV's is the DARPA (Defense Advanced Research Projects Agency) Grand Challenge. The DARPA Grand Challenge is a prize

competition for unmanned vehicles, funded by the DARPA, the most prominent research organization of the United States Department of Defense. A lot of teams from universities or technology companies [11], [4], [14], [15] have designed and produced UGV's for this competition. Every team has different hardware and software architectures for their systems. Figure 1.5 illustrates the team MIT's system architecture.

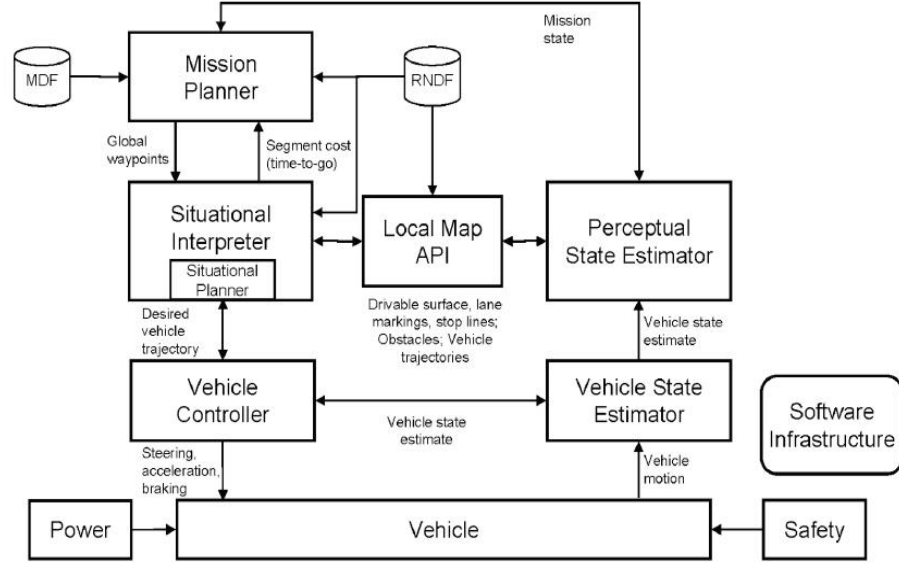


Figure 1.5: System Architecture for Team MIT Vehicle for DARPA GC 2007 [4].

The main two blocks of the figure above which are directly related to this thesis are Situational Planner and Vehicle Controller. Global planning is done in Mission Planner block which includes predetermined waypoints that are given by competition in a Route Network Definition File (RNDF). This file specifies accessible road segments and provides information such as waypoints, stop sign locations, lane widths, checkpoint locations, and parking spot locations. Situational planner can be thought as local planner in order to calculate the trajectory for arriving the way points with its obstacle and collision avoidance capability.

The Situational Planner identifies and optimizes a kino-dynamically feasible vehicle trajectory that moves towards the RNDF waypoint selected by the Mission Planner and Situational Interpreter using the constraints given by the Situational Interpreter. Uncertainty in local situational awareness is handled through rapid replanning and constraint tightening. The Situational Planner also accounts explicitly for vehicle

safety, even with moving obstacles. The output is a desired vehicle trajectory, specified as an ordered list of waypoints (each with position, velocity, and heading) that are provided to the Vehicle Controller.

The Vehicle Controller uses the inputs from the Perceptual State Estimator to execute the low-level control necessary to track the desired paths and velocity profiles issued by the Situational Planner.

1.4 Purpose of the Thesis

Title of this thesis is "A Novel Obstacle Avoidance Approach for Nonholonomic Ground Vehicle Autonomy". Obstacle avoidance is different from global planning since it is activated when the environment is unknown. Obstacle avoidance algorithms should work cooperatively with global planners. Global planners commands are disabled and obstacle avoidance is activated when an unexpected obstacle scenario is met. Goal point to the obstacle avoidance algorithm is given by global planner. Figure 1.6 illustrates the sub modules of the designed UGV. Gray boxes show the new designed algorithms in this thesis.

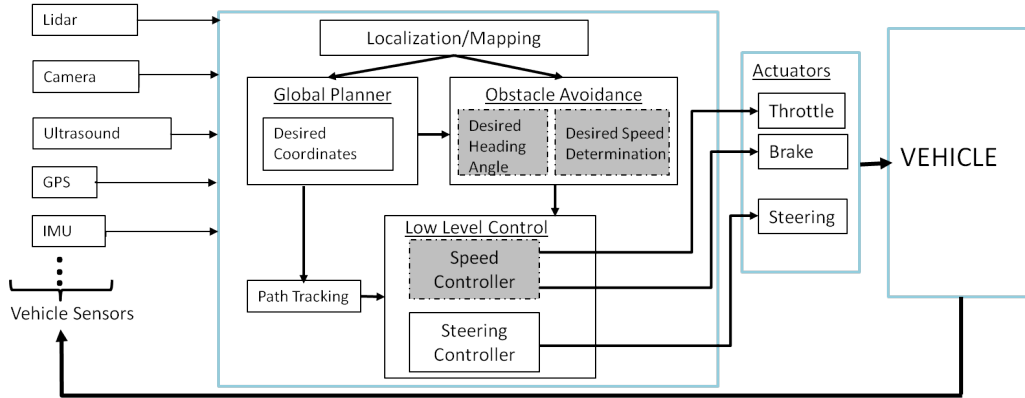


Figure 1.6: Sub Modules of the Designed UGV.

Purpose of the thesis is to design a new and effective obstacle avoidance method for a nonholonomic ground vehicle. Formal definition of the problem for heading reference calculation for obstacle avoidance is given in Section 4.2. Many of the obstacle avoidance algorithms are studied in mobile robotics field where the robot's speed is relatively low. For this reason the researchers generally concentrates on finding the desired heading angle of the robot. However, especially in higher speeds, robot dynamics becomes more important and robot's speed affects the obstacle avoidance

performance along with the robot maneuver. Since the obstacle avoidance performance depends not only the steering method but also the vehicle velocity and velocity tracking capability, designing new speed planning and speed control methods are the other purposes of the thesis. Literature overview for each problem is given in related sections.

1.5 Contributions of the Thesis

Several contributions have been made within this thesis. Most of the results have been published or accepted to be published in scientific journals or conference proceedings. All designed algorithms are tested in simulations and experimental platform. Contributions can be given in three parts.

First part is the calculation of the desired heading vector for obstacle avoidance which is explained in Chapter 4 in detail. Main contributions about the newly designed obstacle avoidance algorithm can be listed as;

- The algorithm results in safer trajectories in comparison with the other tested methods.
- The algorithm has no local minimum problem.
- Nonholonomic constraints of the robot is taken into account and feasible trajectories are generated while the other methods do not have this property.
- The field of view of the robot is taken into account and the robot is not forced to move towards unperceived directions.
- Follow the Gap algorithm is easy to tune with only one tuning parameter, “alpha”.

The second part of the contributions are made within the new designed speed planning algorithm in Chapter 5. Following contributions can be listed;

- Three parameters are used for speed planning by using two cascaded Fuzzy Inference Systems (FIS),
- Danger level of the obstacle is calculated using both the distance to obstacle and obstacle angle values.

- Steering angle value is considered along with the danger level of the obstacles in order to provide a stable yaw dynamics.

Finally, low level speed control is the third main part for the contributions which explained in Chapter 6. Contributions about this subject are listed as;

- Two cascaded Fuzzy Inference Systems (FIS) are used for low level speed control.
- Steering angle value is considered for stable yaw dynamics.

1.6 Thesis Outline

This thesis is organized as follows. Chapter 2 illustrates the kinematic and dynamic vehicle modeling . Chapter 3 introduces the design of the experimental platform which is a UGV and used for the experimental tests of the designed algorithms. Chapter 4 explains the novel obstacle avoidance algorithm called as the "Follow the Gap Method" in detail. Chapter 5 illustrates the new fuzzy speed planning strategy. The new low level speed control method is given Chapter 6. Finally, simulation and experimental results of the designed strategies together, are illustrated in Chapter 7.

2. KINEMATIC AND DYNAMIC VEHICLE MODELING

2.1 Introduction

In this chapter, kinematic and dynamic vehicle modeling subjects are explained. Vehicle modeling is very important for realistic simulations of designed algorithms. In low speed values, kinematic vehicle model is enough for obstacle avoidance tests. When the speed value increases, dynamic vehicle model becomes necessary for realistic results. Both of these two models are used for the simulations in this thesis. Matlab/Simulink environment is used for modeling calculations. Detailed information about vehicle modeling concept can be found in [16], [17], [18] and [19].

2.2 Model Parameters

Model parameters which are used during the simulations of this thesis represent the 'Otonobil' which is a pure electric two-seater vehicle with 6.5kW electric motor. It is also used for the road tests of the designed algorithms. Figure 2.1 illustrates the used EC-2 model electric vehicle from Bestar Company [20].



Figure 2.1: Base Vehicle Used for Modeling and Simulations.

Parameters that are used for both kinematic and dynamic vehicle model are illustrated in Table 2.1. Pacejka tire parameters are taken from the appendix chapter of [21].

Table 2.1: Vehicle Parameters.

Parameter	Value	Unit
Mass(M)	857	kg
Distance from c.g. to front axle(l_f)	0.92	m
Distance from c.g. to rear axle(l_r)	0.78	m
Tire Inertia(J_{tire})	6	kgm^2
Tire Diameter(R)	0.28	m
Rolling Resistance Coefficient(f_r)	0.016	
Atmospheric Density(ρ)	1.2	kg/m^3
Frontal Area(A)	2.11	m^2
Aerodynamic Drag Coefficient(ξ)	3	
Distance from front axle to front side(n)	0.314	m
Distance from rear axle to rear side(m)	0.277	m
Gear Ratio(g_r)	10	
Gear Efficiency(η_g)	0.9	
Camber Angle(γ)	0	deg
a0	1.6929	
a1	-55.21	
a2	1271.3	
a3	1601.8	
a4	6.4946	
a5	$4.8 \cdot 10^{-3}$	
a6	-3.87510^{-1}	
a7	1	
a8	$-4.54 \cdot 10^{-2}$	
a9	$4.28 \cdot 10^{-3}$	
a10	$8.65 \cdot 10^{-2}$	
a111	0	
a112	0	
a12	0	
a13	0	
b0	1.65	
b1	-7.62	
b2	1122.6	
b3	-7.3610^{-3}	
b4	144.82	
b5	$-7.66 \cdot 10^{-2}$	
b6	-3.8610^{-3}	
b7	8.50510^{-2}	
b8	$7.57 \cdot 10^{-2}$	
b9	$2.36 \cdot 10^{-2}$	
b10	$2.36 \cdot 10^{-2}$	

2.3 Kinematic Vehicle Model

Kinematic modeling is the study of the mathematics of motion without considering the forces that affect the motion. Figure 2.2 shows the vehicle parameters for kinematic model.

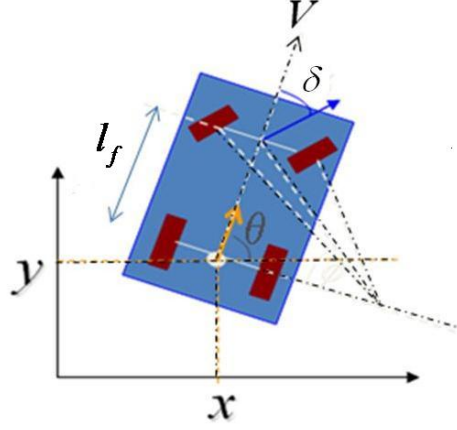


Figure 2.2: Vehicle Parameters for Kinematic Model.

Kinematic vehicle modeling equations are illustrated as follows.

$$\frac{dx}{dt} = V \cos(\Theta) \quad (2.1)$$

$$\frac{dy}{dt} = V \sin(\Theta) \quad (2.2)$$

$$\frac{d\Theta}{dt} = \frac{V}{l_f} \tan(\delta) \quad (2.3)$$

Inputs of the model are vehicle speed(V) and steering angle (δ). Outputs of the model are the X-Y coordinates of the center of gravity (CoG) and the heading angle (Θ). In order to calculate the vertex coordinates of such a rectangular shaped ground vehicle in Cartesian space, basic trigonometric relations are used. Figure 2.3 shows the vertexes and used parameters for this calculation.

Following equations are for calculating the midpoint coordinates of front and rear side of the vehicle (A and B).

$$\begin{aligned} A_x &= x - m \cos(\Theta) \\ A_y &= y - m \sin(\Theta) \end{aligned} \quad (2.4)$$

$$\begin{aligned} B_x &= x + (l_f + n) \cos(\Theta) \\ B_y &= y + (l_f + n) \sin(\Theta) \end{aligned} \quad (2.5)$$

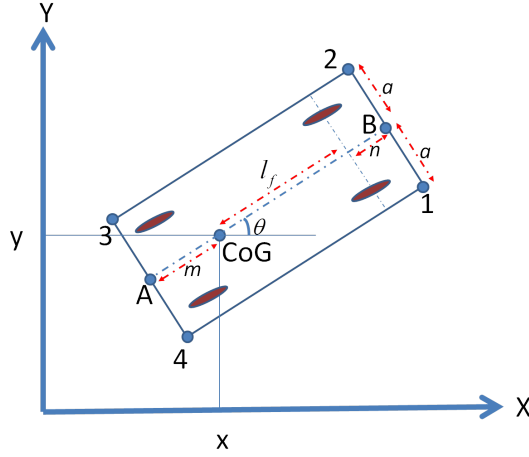


Figure 2.3: Vertices and Related Parameters for Kinematic Model.

Vertexes are labeled as 1, 2, 3 and 4 in Figure 2.3. Equations for the vertex coordinates are illustrated below.

$$\begin{aligned} 1_x &= B_x + a \cos\left(\frac{\pi}{2} - \Theta\right) \\ 1_y &= B_y - a \sin\left(\frac{\pi}{2} - \Theta\right) \end{aligned} \quad (2.6)$$

$$\begin{aligned} 2_x &= B_x - a \cos\left(\frac{\pi}{2} - \Theta\right) \\ 2_y &= B_y + a \sin\left(\frac{\pi}{2} - \Theta\right) \end{aligned} \quad (2.7)$$

$$\begin{aligned} 3_x &= A_x + a \cos\left(\frac{\pi}{2} - \Theta\right) \\ 3_y &= A_y - a \sin\left(\frac{\pi}{2} - \Theta\right) \end{aligned} \quad (2.8)$$

$$\begin{aligned} 4_x &= A_x - a \cos\left(\frac{\pi}{2} - \Theta\right) \\ 4_y &= A_y + a \sin\left(\frac{\pi}{2} - \Theta\right) \end{aligned} \quad (2.9)$$

A sample simulation of the kinematic model is illustrated in Figure 2.4 for the given speed and steering angle profiles. Red dots in Figure 2.4c show the center of gravity point which can be seen in Figure 2.3.

2.4 Dynamic Vehicle Model

Single track bicycle model with Pacejka tire equations are used for dynamic vehicle modeling. This model includes both the longitudinal and the lateral dynamics. The single-track model is obtained by lumping the two front wheels into one wheel in the center line of the car, the same is done with the two rear wheels [17]. Figure 2.5 illustrates the single track vehicle parameters.

The augmented single track model consists of Pacejka tire model, steering angle projection, lateral and longitudinal vehicle dynamics, kinematics and geometry and

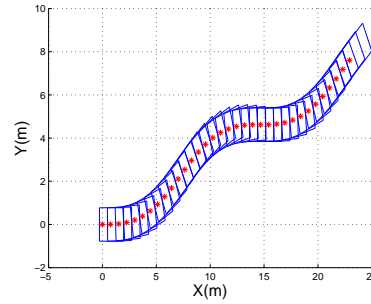
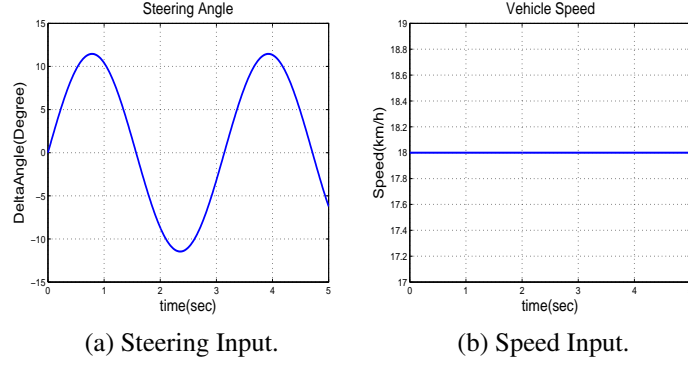


Figure 2.4: Kinematic Vehicle Model Simulation Results ($X_{init} = 0, Y_{init} = 0, \theta_{init} = 0$).

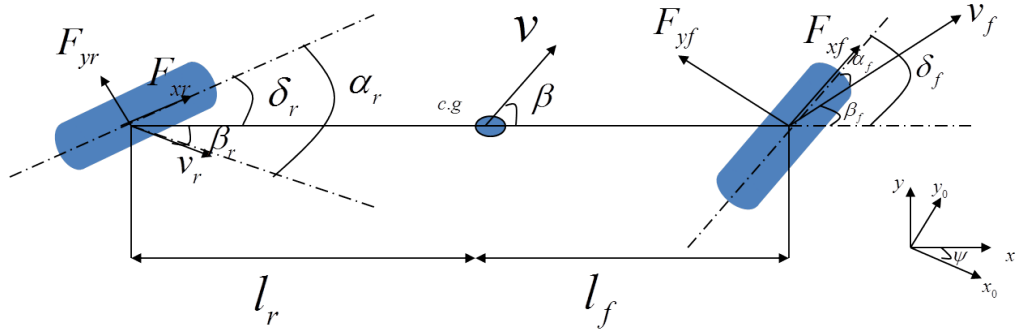


Figure 2.5: Single Track Vehicle Model Parameters.

wheel velocity calculation subsystems. Block diagram of the dynamic vehicle model is illustrated in Figure 2.6.

Explanations about each block are illustrated in following sections.

2.4.1 Pacejka tire model

Tires enable vehicle motion by providing for traction, braking, steering, and load support. Tires are inflated with air, which provides a flexible cushion between the vehicle and the road that smoothes out shock and provides for a comfortable ride. Pacejka tire model is used for modeling tires of the vehicle. This model is based on the

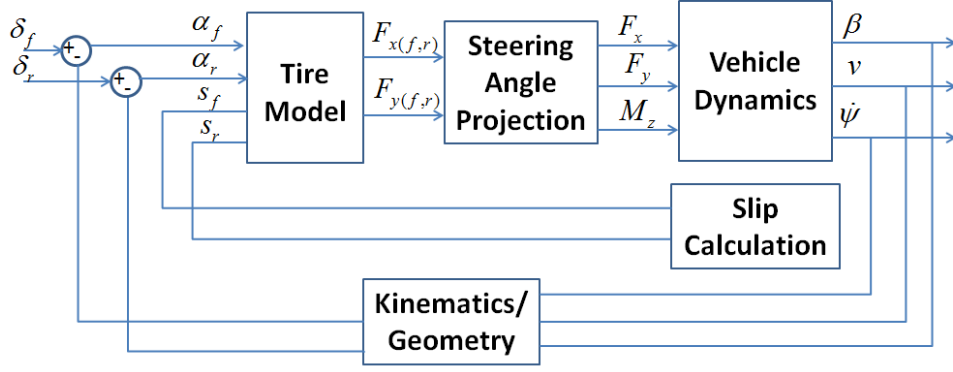


Figure 2.6: Dynamic Vehicle Model Block Diagram.

Pacejka Magic Formula which is obtained by empirical results. More information can be found in [19]. Equation for calculating the longitudinal tire force is given below.

$$F_{x(f,r)}(s) = D \sin(C \arctan(B(s_{f,r} + S_h) - E(B(s_{f,r} + S_h) - \arctan(B(s_{f,r} + S_h)))))) + S_v \quad (2.10)$$

where B, C, D, E, S_v, S_h are six coefficients which depend on the vertical tire force F_z and the camber angle γ . Calculation of these coefficients are illustrated below.

$$C = b_0 \quad (2.11)$$

$$D = \mu_p F_z \quad (2.12)$$

$$BCD = (b_3 F_z^2 + b_4 F_z) e^{-b_5 F_z} \quad (2.13)$$

$$E = b_6 F_z^2 + b_7 F_z + b_8 \quad (2.14)$$

$$S_h = b_9 F_z + b_{10} \text{ (Horizontal Shift Value)} \quad (2.15)$$

$$S_v = 0 \text{ (Vertical Shift Value)} \quad (2.16)$$

Equation for calculating the lateral tire force is given below.

$$F_{y(f,r)}(\alpha) = D_l \sin(C_l \arctan(B_l(\alpha + S_{hl}) - E(B_l(\alpha + S_{hl}) - \arctan(B_l(\alpha + S_{hl})))))) + S_{vl} \quad (2.17)$$

where $B_l, C_l, D_l, E_l, S_{vl}, S_{hl}$ are six coefficients which depend on the vertical tire force F_z and the camber angle γ . Calculation of these coefficients are illustrated below.

$$C_l = a_0 \quad (2.18)$$

$$D_l = \mu_{yp} F_z \quad (2.19)$$

$$\mu_{yp} = a_1 F_z + a_2 \quad (2.20)$$

$$E_l = a_6 F_z + a_7 \quad (2.21)$$

$$a_{11} = a_{111} F_z + a_7 \quad (2.22)$$

$$B_l C_l D_l = a_3 \sin(2 \arctan(\frac{F_z a_4}{a_3})) (1 - a_5 |\gamma|) \quad (2.23)$$

$$S_{hl} = a_8 \gamma + a_9 F_z + a_{10} \quad (2.24)$$

$$S_{vl} = a_{11} \gamma F_z + a_{12} F_z + a_{13} \quad (2.25)$$

2.4.2 Steering angle projection block

Steering angle projection block takes the front wheel longitudinal force F_{xf} , front wheel lateral force F_{yf} , rear wheel longitudinal force F_{xr} , rear wheel lateral force F_{yr} and steering angle values for rear and front wheels δ_f , δ_r to calculate the corresponding forces at the vehicle center of gravity F_x , F_y and M_z . Here, wheel longitudinal and lateral forces are projected on chassis coordinate axes. Forces and moment that affect vehicle chassis are calculated below in terms of wheel forces and steering angles.

$$F_x = -\sin(\delta_f) F_{yf} - \sin(\delta_r) F_{yr} + \cos(\delta_f) F_{xf} + \cos(\delta_r) F_{xr} - F_{aero} - F_{roll} \quad (2.26)$$

$$F_y = \cos(\delta_f) F_{yf} + \cos(\delta_r) F_{yr} + \sin(\delta_f) F_{xf} + \sin(\delta_r) F_{xr} \quad (2.27)$$

$$M_z = l_f \cos(\delta_f) F_{yf} - l_r \cos(\delta_r) F_{yr} + l_f \sin(\delta_f) F_{xf} - l_r \sin(\delta_r) F_{xr} \quad (2.28)$$

In Equation 2.26, F_{aero} and F_{roll} are aerodynamic resistive force and rolling resistance force values respectively. They are calculated as illustrated below.

$$F_{aero} = \frac{\xi \rho A}{2} v^2 \quad (2.29)$$

$$F_{roll} = Mg F_r \quad (2.30)$$

where A is the frontal area, ξ is the aerodynamic drag coefficient, ρ is the atmospheric density and F_r is the rolling resistance coefficient.

2.4.3 Vehicle dynamics block

In this block vehicle side slip angle β vehicle speed V and yaw rate $\dot{\psi}$ are calculated using chassis forces F_x , F_y and yaw moment M_z . Equations are given below.

$$\dot{\psi} = \int_{t_0}^t \frac{M_z}{J} dt. \quad (2.31)$$

$$V = \int_{t_0}^t \frac{\cos(\beta)F_x + \sin(\beta)F_y}{M} dt. \quad (2.32)$$

$$\beta = \int_{t_0}^t \frac{-\sin(\beta)F_x + \cos(\beta)F_y}{Mv} - \dot{\psi} dt. \quad (2.33)$$

2.4.4 Kinematics and geometry block

The angle between wheel speed and the chassis for the front and rear wheels β_f , β_r are calculated in this block. Equations are illustrated below.

$$\beta_f = \arctan(\tan(\beta) + \frac{l_f \dot{\psi}}{v \cos(\beta)}) \quad (2.34)$$

$$\beta_r = \arctan(\tan(\beta) - \frac{l_r \dot{\psi}}{v \cos(\beta)}) \quad (2.35)$$

2.4.5 Longitudinal wheel slip calculation block

Wheel slip is the measure of the difference between the rotational speed of the wheel and the translational velocity of the wheel. In order to calculate the slip values for the rear and front wheels s_f , s_r , longitudinal speed vector of front and rear wheels v_{xf} , v_{xr} and wheel angular velocity values w_f and w_r must be calculated first. Equations are illustrated below.

$$v_{xf} = \sqrt{(v \cos(\beta))^2 + (v \sin(\beta) + \dot{\psi} l_f)^2 \cos(\delta_f - \beta_f)} \quad (2.36)$$

$$v_{xr} = \sqrt{(v \cos(\beta))^2 + (v \sin(\beta) - \dot{\psi} l_r)^2 \cos(\delta_r - \beta_r)} \quad (2.37)$$

$$w_f = \int_{t_0}^t \frac{T_f - F_{xf} R}{J_{tire}} dt. \quad (2.38)$$

$$w_r = \int_{t_0}^t \frac{T_r - F_{xr} R}{J_{tire}} dt. \quad (2.39)$$

T_f and T_r are the input torque values given to the vehicle. Since this is a front driven electric vehicle, T_f is calculated by using the electric motor torque T_m , gear ratio g_r and gear efficiency η_g values as illustrated below.

$$T_f = T_m g_r \eta_g \quad (2.40)$$

The torque-speed characteristic of the electric motor in nominal region with throttle input th_{in} is illustrated in Figure 2.7.

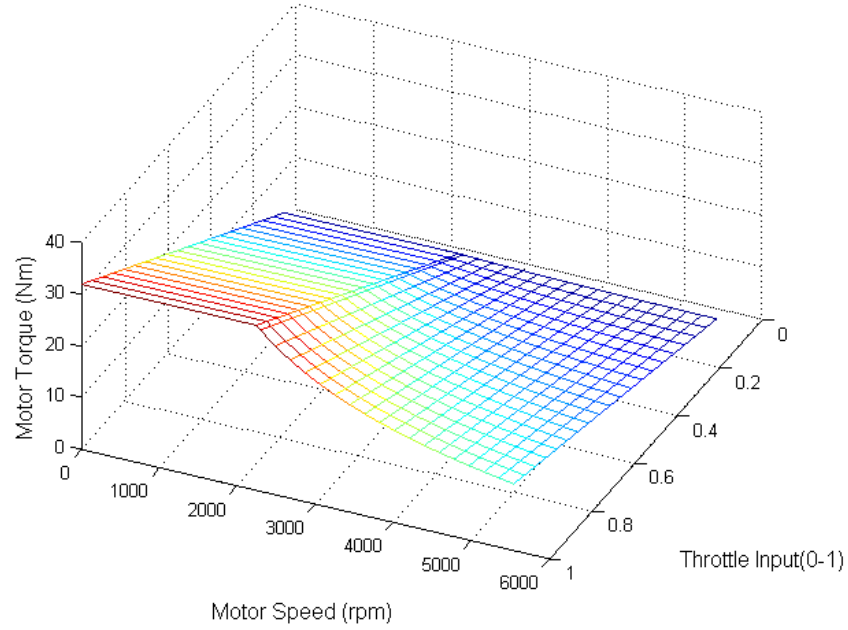


Figure 2.7: Electric Motor Toque-Speed Curve.

After the calculation of the v_{xf} , v_{xr} , w_f and w_r , wheel slip values are calculated as illustrated below.

$$s_f = \frac{w_f R - v_{xf}}{v_{xf}} s_r = \frac{w_r R - v_{xr}}{v_{xr}} \Leftrightarrow w_{f,r} R < v_{x,r} (\text{Braking}) \quad (2.41)$$

$$s_f = \frac{w_f R - v_{xf}}{w_f R} s_r = \frac{w_r R - v_{xr}}{w_r R} \Leftrightarrow w_{f,r} R > v_{x,r} (\text{Accelerating}) \quad (2.42)$$

A sample simulation of the dynamic model is illustrated in Figure 2.8 for the given speed and steering angle profiles. Red dots in Figure 2.8c show the center of gravity point.

2.5 Conclusions

In this chapter, kinematic and dynamic vehicle modeling subjects are explained in detail. Derivation of the equations in this chapter can be found in related references. Developed models are used in order to simulate the newly designed algorithms for obstacle avoidance, desired speed determination and low level speed control in the rest of the thesis. All simulations are performed in Matlab/Simulink environment based on

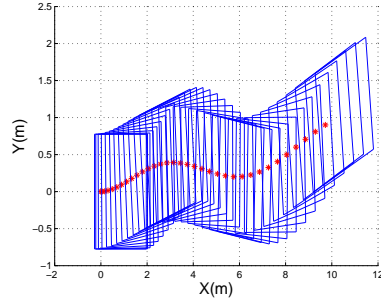
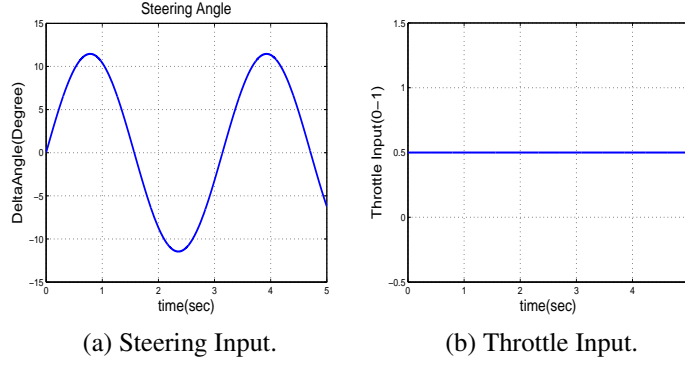


Figure 2.8: Dynamic Vehicle Model Simulation Results. ($X_{init} = 0, Y_{init} = 0, \theta_{init} = 0, v_{init} = 0$)

the developed kinematic and dynamic vehicle models with the vehicle parameters that are shown in Table 2.1.

3. DEVELOPMENT OF THE EXPERIMENTAL PLATFORM "OTONOBİL"

3.1 Introduction

An experimental platform, 'Otonobil' is developed in order to test the designed algorithms. Experimental platform is an unmanned ground vehicle (UGV) and converted from a conventional electric vehicle EV-2 from Bestarmotor company [20]. EV-2 is a pure electric vehicle with 6.5 kW electric motor and 72V 150Ah battery system. Autonomous ground vehicle after the conversion process is illustrated in Figure 3.1. Otonobil is one of the the first fully autonomous ground vehicles of Turkey. This section illustrates the minimum design requirements for a UGV development, electrical modifications, mechanical modifications and finally the communication and interface software respectively.



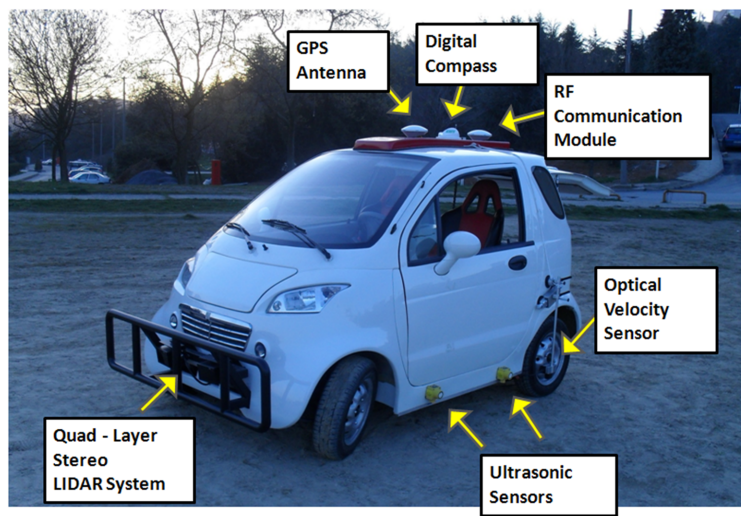
Figure 3.1: Designed UGV, "Otonobil".

There are some special reasons for choosing the EV-2 as a base vehicle. These reasons can be summarized as follows:

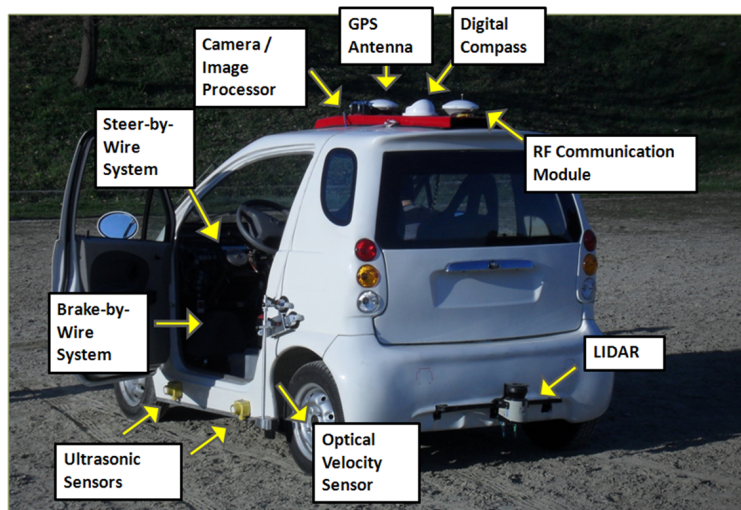
- Its minimal dimensions make the experimental tests easy on the road.

- An internal combustion engine causes vibration to the system because of its working principle. These vibrations may cause extra noise to sensors such as; inertial measurement unit, laser scanners and etc. Filtering this noise causes the loss of the original sensor data. But in a pure electric vehicle vibrations are lower than combustion engine based vehicle.
- Since the vehicle does not have a transmission, there is no need to design an additional mechanical system for automatic gear shift.

Otonobil's additional components from rear and front side are illustrated in Figure 3.2.



(a) Front View.



(b) Rear View.

Figure 3.2: Otonobil's Additional Components from Front and Rear Side.

Even though the first aim is to convert the base vehicle into a UGV, some additional features are also considered. First of all, human drivability feature is protected. Additional information about this will be given in Mechanical Modifications part. Another feature is drivability by a joystick from outside of the vehicle. Finally, the main feature of the vehicle is its autonomy which is directly related with the scope of thesis. So, the vehicle has 3 operation modes:

- Classic Mode: Drive by human in vehicle.
- Remote Control Mode: Drive by human outside the vehicle.
- Autonomous Mode: Drive autonomously to a given desired location.

3.2 Minimum Design Requirements for an Unmanned Ground Vehicle Development

There is a minimal set of hardware components that reside on almost all UGV. When designing a UGV testbed, it is important to consider attributes such as cost, size, and weight while ensuring modularity [22] . Following items summarize the additional hardware systems of a typical UGV.

- Processing System: The processing system is the central point of a UGV, which is responsible for doing all the data processing and decision making operations. This hardware gathers data from environment, vehicle and interface software. Generally, one computer is not enough and more than one computers are used simultaneously in order to be sufficient for all of the calculations.
- Actuators/Controllers: The actuators and their controllers are responsible for actuating the related components of the UGV such as steering wheel, brake pedal or throttle pedal. Accurate positioning of these actuators is very important for getting higher performance from a UGV.
- Sensors: Sensors are responsible for measuring the all necessary information from the environment and the vehicle itself. GPS (Global Positioning System), IMU(Inertial Measurement Unit), LIDAR (Light Detection and Ranging), camera and encoders can be given as examples for UGV sensors.

- **Communication:** Ultimately a UGV needs to communicate with an external device. This may be as simple as sending progress data to a human interface or as complex as interfacing with a group of UGVs.

3.3 Electrical Modifications

This section illustrates the electrical modifications that are made for Otonobil. These modifications are; selection of the sensors and controllers, design of the additional power system and design of the signal interface electronic circuit.

3.3.1 Sensors and controllers

In a UGV, there should be specialized sensors to perform mapping and state estimation. More information about sensors that are used in intelligent vehicles can be found in [23]. The sensors that are used in Otonobil for mapping and state estimation are listed in Table 3.1. Gray colored sensors are for mapping of the environment & obstacles and the others are for state estimation.

Table 3.1: Sensor List of Otonobil for State Estimation and Mapping.

Sensor Type	Quantity	Brand/Model
Laser Scanner	2	IBEO-LUX
Laser Scanner	1	SICK LMS 151
Camera	1	SONY-XCI-SX100
Ultrasonic Sensor	6	Banner-QT50ULB
Differential GPS	1	Trimble SPS851-SPS551H
Digital Compass	1	KVH Azimuth1000
IMU	1	Crossbow VG700AB-201
Optic Speed Sensor	1	Corsys-Datron LF II P
Potentiometer for Steering Wheel Position	1	Spectrol(5 kOhm)

These sensor data should be processed for localization and mapping. Trajectory planning, tracking and obstacle avoidance operations are performed according to these processed data. All these tasks need huge amount of processing power. That's why the tasks are distributed between 3 main computers on board. These computers and their duties are given in Table 3.2.

Figure 3.3 illustrates the computational components of the Otonobil.

Table 3.2: List of Computational Components of Otonobil.

Computer	Description	Duty
NI PXI-8110RT processor PXI-7954R FPGA module on PXI1000B chassis	It has 2.26Ghz quad-core processor and a powerful FPGA module with several I/O cards.	Localization, mapping and global path planning
Dspace Mi-croautobox (MABX) 1401 /1501 /1507	It has 800Mhz processor and several I/O interfaces.	Local Trajectory Planning and tracking, low level control (throttle, steering, brake), wireless communication
SONY-XCI-SX100	It is a smart camera, which can process images with its own processor and sends processed data via UDP/IP.	Image Processing
IBEO ECU	It can give both object and raw data. It has tracking algorithms inside and can classify the objects around it.	LIDAR Data Processing

3.3.2 Additional power system

An extra isolated power supply is needed in order to supply the power for additional sensors, actuators and computers. When calculating the additional power requirement, the worst case (all the sensors, computers and actuators are using maximum power at the same time) is considered. For this aim 4 piece of 12V/50Ah lead-acid batteries are used by connecting them serial and parallel. 12V and 24V voltage references are enough for all the additional components. Appropriate fuses and connectors are also used for each new electrical component. Figure3.4 shows the new battery pack and power module which includes the fuses and the connector box in the rear side of the vehicle.

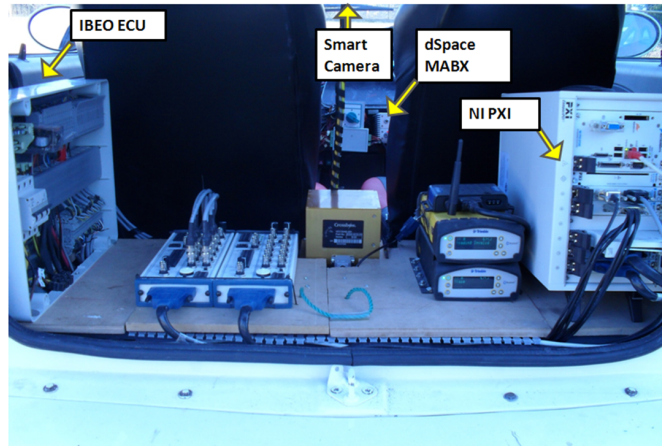


Figure 3.3: Computational Components of Otonobil.

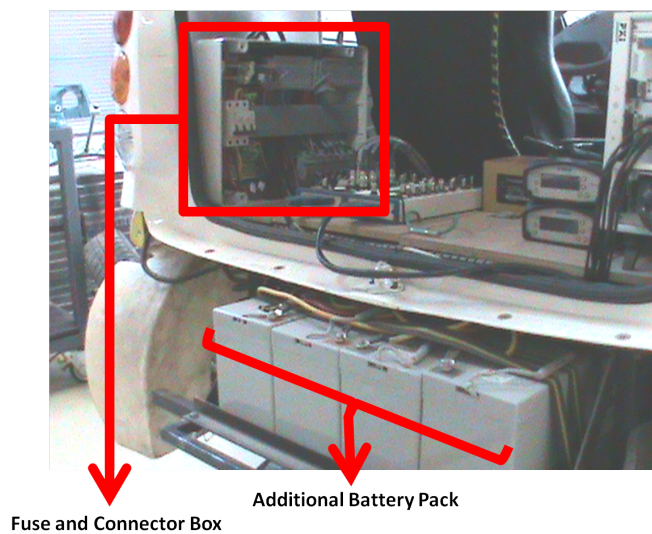
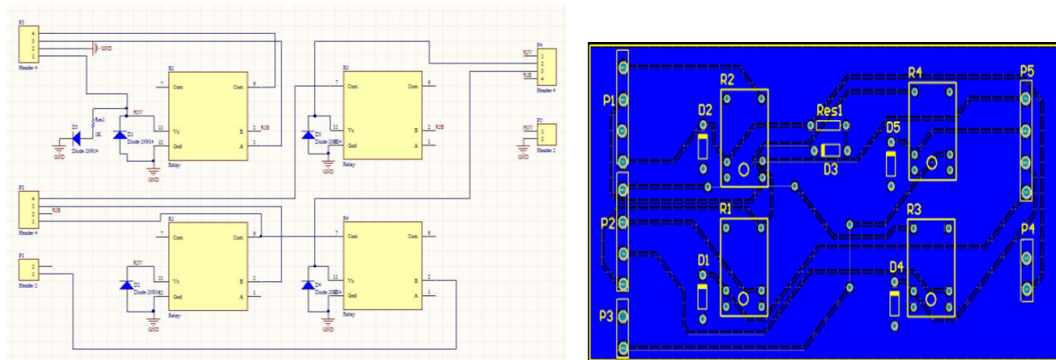


Figure 3.4: Additional Batteries and Power Box.

3.3.3 Vehicle interface system

One of the most important properties of a UGV is its drive-by-wire capability. An electric circuit was designed and produced for this aim. Schematic and PCB layout of the circuit are illustrated in Figure 3.5.

The aim of this circuit is to switch the acceleration signal to electric motor and brake lamp signals between pedals and low level controller (MABX). According to the position of the mode selection switch which determines the mode of the vehicle (autonomous or classic mode), desired electric signals are given from the pedals or MABX. Installed components for the interface system are illustrated in Figure 3.6.

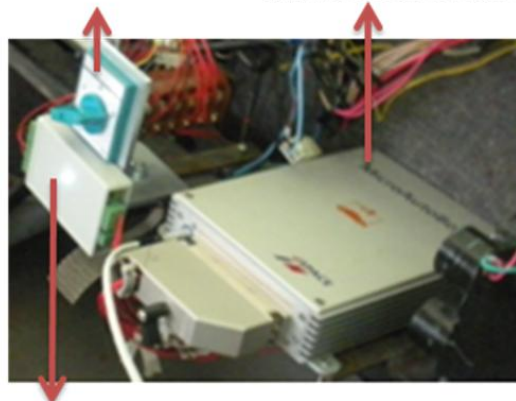


(a) Interface Circuit's Schematic.

(b) Interface Circuit's Printed Circuit Board.

Figure 3.5: Interface Circuit.

Mode Selection Switch MABX Hardware



Vehicle Interface Circuit

Figure 3.6: Human Interface System.

3.4 Mechanical Modifications

In order to convert a conventional car into an autonomous one, some mechanical modifications must be done as well as the electrical modifications. There are two main mechanical modifications shown in Figure 3.7; steer by wire and brake by wire systems.

Figure 3.7a illustrates the original vehicle steering and brake system while Figure 3.7b illustrates them after modification. Beside these, mounting of the sensors needs some mechanical modifications too. All these works are explained in the following subsections.

3.4.1 Using computer environment in mechanical design process

Computer aided design (CAD) drawing of the vehicle is illustrated in Figure 3.8 which is used for establishing the additional components to the vehicle.

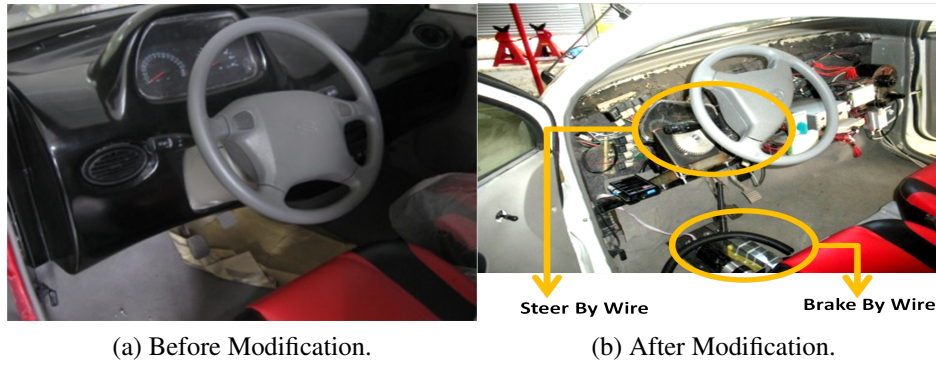


Figure 3.7: Mechanical Modifications.

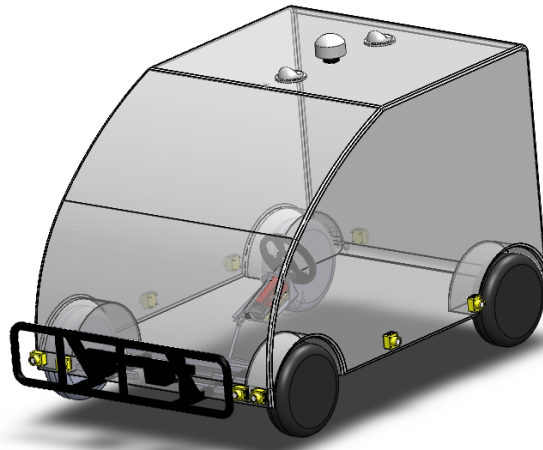


Figure 3.8: CAD Drawing of the Vehicle.

Basic measurements, consisting of simple dimensions of the vehicle help to build a CAD model of the vehicle. Once we have these dimensions accurately and translate them to CAD environment, any modification can be tested easily. This is very useful for designing the end-production.

3.4.2 Automatic braking mechanism

An autonomous car should be able to do better than what human-beings act while driving. These actions have to be performed by electro-mechanical systems. The automatic braking mechanism is designed in order to actuate brake pedal of the vehicle according to the low level speed controller's output signals. The design has to be detachable so it can easily be converted into a conventional car when it is in manual mode. Final design is illustrated in Figure 3.9.

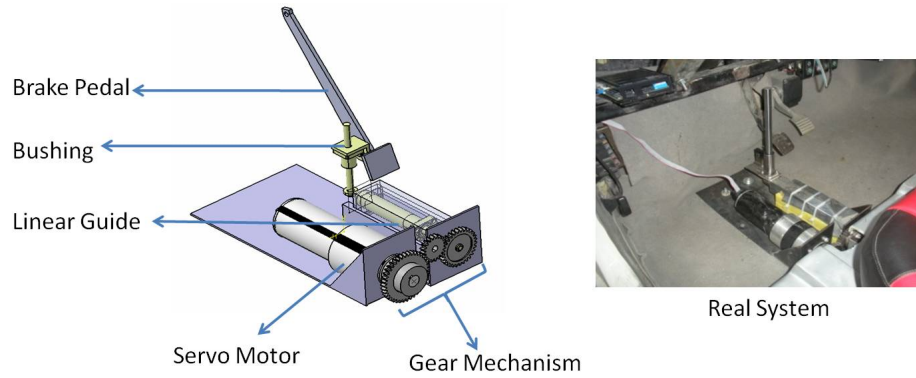


Figure 3.9: Automatic Brake System for Otonobil.

Linear guide converts the rotational torque into linear force and pulls the pedal using the bushing mechanism as illustrated in Figure 3.9.

3.4.3 Automatic steering mechanism

Similar to the automatic braking system, a UGV must steer itself according to the low level steering controller's output signals. Views of the steering system assembly in CAD environment and after modification are illustrated in Figure 3.10.

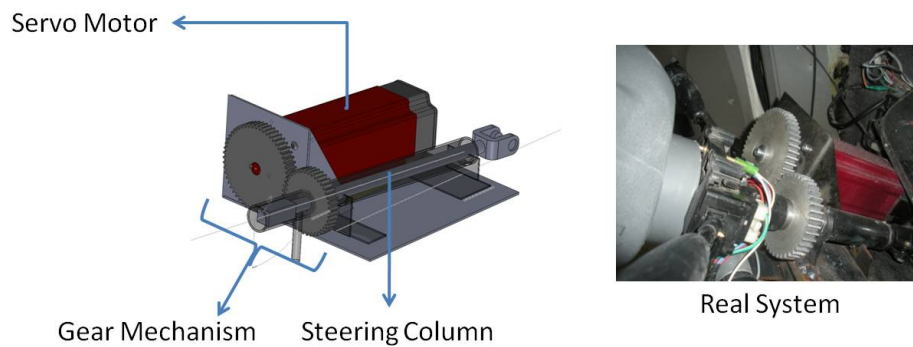


Figure 3.10: Automatic Steering System for Otonobil.

A gear mechanism is used for connecting the steering column and the servo motor as illustrated in Figure 3.10. The servo motor does not have a gearbox mechanism inside which lowers the inertia of the motor. The reason for using that kind of servo motor is about the manual mode of "Otonobil". Using this servo motor, it is possible to operate steering wheel by hand as in a conventional vehicle.

3.5 Communication and Interface Software

There are two types of communication structure for a UGV, one is for inside and the other is for outside of the vehicle. The sensors and computer system with their

communication are illustrated in Figure 3.11. CAN, Ethernet, RS232 protocols and analog signals are used for communication in Otonobil as shown in Figure 3.11.

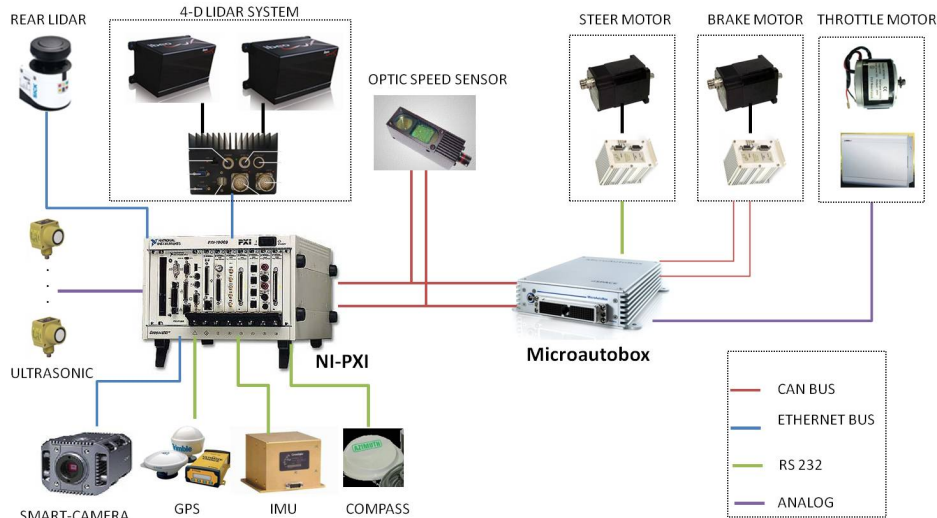


Figure 3.11: Sensor and Computer System and Their Communication in Otonobil.

There are three operation modes of Otonobil ; "classic mode", "remote control mode" and "autonomous mode" as it was explained previously. Switching between these modes can be done not only using the interface system as it is illustrated in Figure 3.6 but also from outside of the vehicle. Outside communication scheme of the Otonobil is illustrated in Figure 3.12.

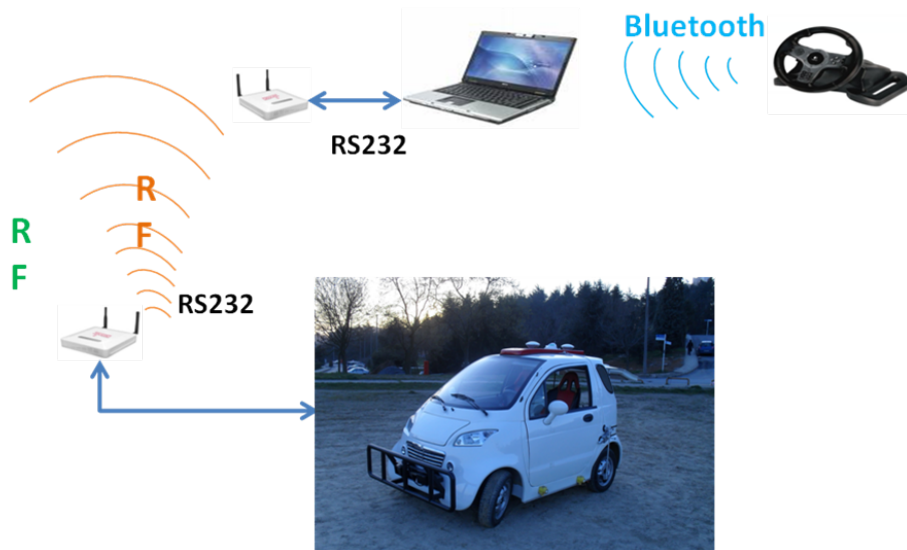


Figure 3.12: Outside Communication Scheme of Otonobil.

Bluetooth protocol is used between joystick and PC. PC sends/receives data to RS232 protocol via radio frequency (RF). There are 2 RF transceivers from UDEA company

[24] (UMD-B12P25) in the communication system, one is on PC side, the other is in the vehicle. RF transceiver electric circuit is illustrated in Figure 3.13.

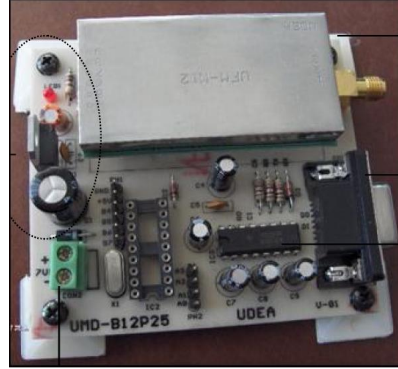


Figure 3.13: UMD-B12P25 RF Communication Module.

Two interface solutions are developed for this thesis for inside and outside communication. Interface software inside the vehicle runs in a classical PC and communicates with the MABX system. It is written using the "Control Desk" tool of MABX. Important signals can be analyzed and saved. On vehicle side, data from the sensors are gathered and processed by several algorithms running on National Instrument's (NI) computer. Then processed data is sent to MABX via CAN protocol. The screenshot of the interface software which is designed for inside of the vehicle is illustrated in Figure 3.14.

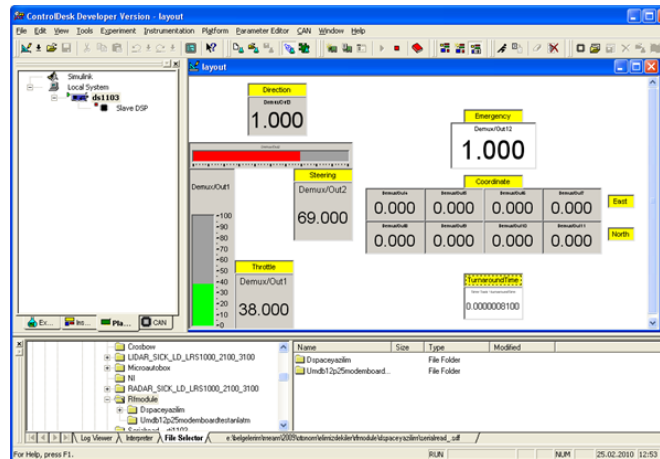


Figure 3.14: Interface Software for Inside of the Vehicle.

Otonobil has an interface software for the outside of the vehicle which sends and receives data from vehicle via serial port of the PC, using the RF transceiver module. This program is written using C++ in Visual Studio platform. A screenshot of this interface is illustrated in Fig.3.15.

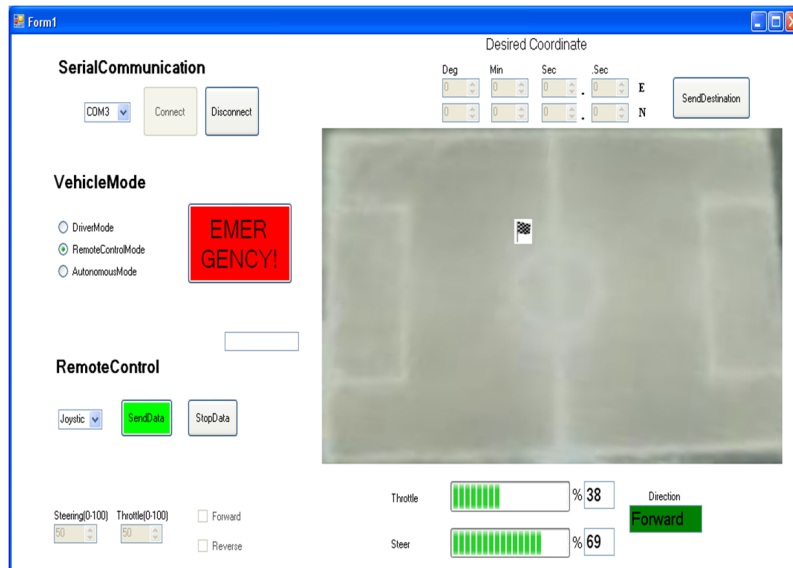


Figure 3.15: Interface Software for Outside of the Vehicle.

This interface manages the communication and makes the necessary settings for serial port. It also sends the operation mode of the vehicle which are the classical mode, the joystick mode and the autonomous mode. Interface sends the joystick data to the vehicle in “remote control mode”. The connected joystick is a standard game console from Logitech, which can be seen in Figure 3.12. Alternatively, acceleration and steering angle commands can be sent manually without a joystick in this mode.

In autonomous mode, GPS coordinates of the desired location are sent to the vehicle. Any coordinate can be sent to the vehicle as the goal coordinates to reach in this mode. On the right side of the Figure 3.15, the picture of the test field is shown and when the user clicks to anywhere on the picture, coordinate values are calculated automatically and sent to the vehicle.

An emergency button also sends emergency signal to the vehicle, which is interpreted as brake and stop on the vehicle side. Additionally, the interface takes the GPS position, vehicle speed and steering angle position values from the vehicle and shows them to the user.

3.6 Conclusions

Conversion procedure of a conventional automobile into a UGV is illustrated in this chapter. Conversion process is studied in 2 main parts, as electrical and mechanical

modifications. In literature, it is not easy to find a paper which gives enough information about autonomous vehicle conversion procedure. This chapter tries to fill the gap in this area.

4. A NOVEL OBSTACLE AVOIDANCE ALGORITHM: "FOLLOW THE GAP METHOD"

In this chapter, a novel obstacle avoidance method is designed and applied to the experimental autonomous ground vehicle system. The proposed method brings a new solution to the problem and has several advantages compared to previous methods [25]. This novel algorithm is easy to tune and it takes into consideration the field of view and the nonholonomic constraints of the robot. Moreover the method does not have a local minimum problem and results in safer trajectories because of its inherent properties in the definition of the algorithm. The proposed algorithm is tested in simulations and after the observation of successful results, experimental tests are performed using static and dynamic obstacle scenarios. The experimental test platform is "Otonobil" with Ackermann steering geometry which brings nonholonomic constraints is explained in Chapter 3 . Experimental results show that the task of obstacle avoidance can be achieved using the algorithm on the autonomous vehicle platform. The algorithm is very promising for application in mobile and industrial robotics where obstacle avoidance is a feature of the robotic system.

4.1 Introduction

Robot navigation refers to the robot's ability to safely move towards the goal using its knowledge and the sensorial information of the surrounding environment. Given a map and a goal location, path planning involves finding a geometric path from the actual location of the robot to the goal/target. This type of planning is referred to static path planning due to the fact that the map used in the algorithm is static, and not updated dynamically based on new information [26] [27]. There are many studies involved with static planning, such as, Probabilistic Roadmaps (PRM) [28], Rapidly Exploring Random Trees (RRT) [29] , Generalized-Sampling Based Methods [30], Visibility Graphs [31], Voronoi Diagrams [32] and cell decomposition methods [33]. Beside these, studies about finding an optimal solution for nonholonomic robots, can

be found in [34] [35]. The common ground of all these path planning methods is the necessity for a map of the whole workspace.

Obstacle avoidance is different from static path planning with its aim of avoiding unexpected obstacles along the robot's trajectory. In other terms, it shapes up the trajectory of the path planner as a dynamic path planning approach. Considering the needs of autonomous robot control, it is obvious that detecting and avoiding obstacles in real time is crucial for the performance. For this reason, many researchers have turned their attention to the obstacle avoidance problem developing interesting real-time approaches for avoiding unexpected static and dynamic obstacles.

Several methods have been proposed for obstacle avoidance starting from the 1980s till now. The earliest versions of these obstacle avoidance methods are bug algorithms [36]. These algorithms follow the easiest common sense approach of moving directly towards the goal, until an obstacle is found, in which case the obstacle is contoured until moving towards the goal is possible again. The trajectories of bug algorithms are sometimes very long and the robot is prone to move too close to obstacles.

Another common approach is the artificial potential field (APF) method [37]. In the APF approach, the obstacles to be avoided are represented by a repulsive artificial potential and the goal is represented by an attractive potential, so that a robot reaches the goal without colliding with obstacles. Main drawbacks of the APF method are summarized in [38] and local minima is the most dangerous problem of APF. This happens when all the vectors from obstacles and the goal point cancel each other out and make it impossible for the robot to reach the goal. There are a great number of studies focusing on this problem. The first method comes from definition of the potential function by specifying a function with no local minima like harmonic potential field approach [39]. However in this approach, the robot must know the map of the whole environment and this contradicts reactivity and local planning properties of obstacle avoidance. Other approaches for local minimum avoidance involve some practical ad-hoc solutions, such as those proposed in [40] [41] [42]. However, none of these approaches can offer an ultimate guarantee to avoid this.

The Virtual Force Field method (VFF) [43] uses a two-dimensional Cartesian histogram grid for obstacle representation. Each cell in the histogram grid holds a

certainty value that represents the confidence of the algorithm in the existence of an obstacle at that location. After that, APF is applied to the histogram grid, therefore the problems of the APF method still exists in the VFF method.

The Vector Field Histogram (VFH) [44] uses a two-dimensional Cartesian histogram grid like in VFF. After that, the histogram grid is reduced to a one dimensional polar histogram that is constructed around the robot's momentary location. In the second stage, the algorithm selects the most suitable sector from among all polar histogram sectors with a low polar obstacle density, and the steering of the robot is aligned with that direction. This method is very much goal oriented since it always selects the sector which is the same direction as the goal, but the selected sector can be the wrong one in some cases. This method also does not consider nonholonomic constraints of robots like the other methods mentioned above. More information about the concepts for dynamic obstacle avoidance can be found in [45].

In this chapter, a novel approach called the "Follow the Gap Method" (FGM) is presented as a novel obstacle avoidance algorithm. FGM ensures safety by directing the robot into the center of the maximum gap as much as possible while providing the reach of the goal point. FGM calculates a gap array around the robot, selects the appropriate gap, calculates the best heading vector through the gap by using specific geometric theorems and finally calculates the final angle considering the goal point. An important advantage of FGM over other methods is that it results in safer trajectories which will be shown in simulations. Moreover FGM accounts for the nonholonomic and the field of view constraints of the robot. Another important advantage is that it does not have a local minimum problem. FGM considers the nonholonomic constraints of the robot and field of view constraints coming from the sensor arrangement. Finally FGM is easy to tune with only one tuning parameter. Simulations and real tests are performed using the Ackerman steering ground vehicle platform. Successful results are achieved in simulations and experimental tests which are illustrated in following sections.

4.2 Problem Definition

Suppose that independent of the geometry of the robot and obstacles they are considered to be circular objects with minimum radius to include all physical

boundaries. Cartesian coordinate space is used for calculations. The location of the robot and its radius values are given by the tuple $(X_{rob}, Y_{rob}, r_{rob})$ and similarly the center location and radius of the obstacles are given by $(X_{obsn}, Y_{obsn}, r_{obsn})$ for n'th obstacle. The following assumptions are made to define the problem.

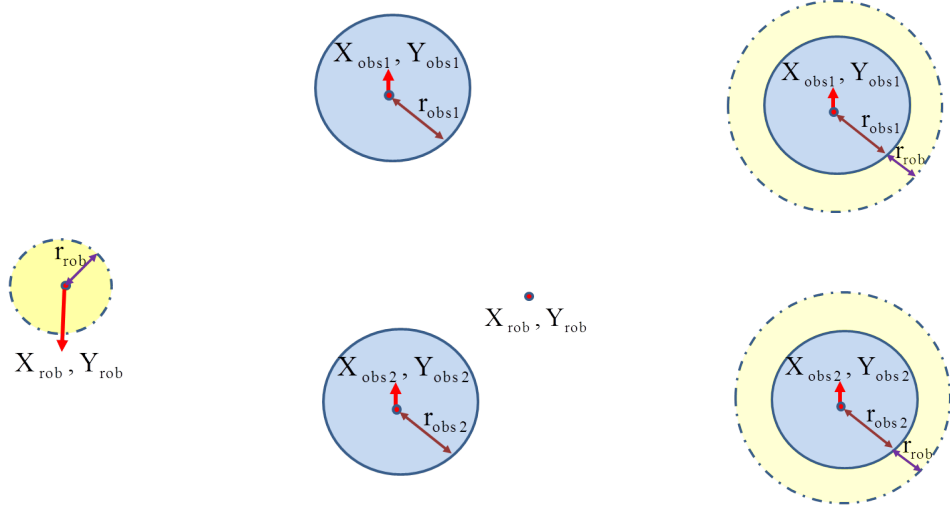
- The robot field of view is constrained by two rays with left Φ_{fov_l} and right Φ_{fov_r} angles and a distance constraint with d_{fov} . The robot does not have prior information about obstacles.
- The robot has nonholonomic constraint which is represented in a summarized form as a minimum turn radius r_{min} .
- All the coordinates/locations and object boundaries are measurable and constraint values Φ_{fov_l} , Φ_{fov_r} , d_{fov} , r_{min} are previously calculated according to the sensor arrangement and the geometry of the vehicle.

Using these assumptions, the aim of the obstacle avoidance algorithm is to find a heading reference purely reactive, in order to achieve the goal coordinates while avoiding obstacles with as large distance as possible, considering the measurement and nonholonomic constraints.

This chapter's theme is to find the desired heading vector for obstacle avoidance. Since no prior information is available to the robot, the nature of the algorithm should be reactive because, coordinates of any obstacle may change at any time and it can not be known previously. The algorithm must compute just the next action in every instant, based on the current context. In other words, any classical optimization algorithm like dynamic programming which calculates the reference heading values from goal to initial coordinates is not possible for real time applications due to the unknown sequencing of the obstacles during the journey. The problem is tried to be solved by using a new heuristic and fusing function between maximum gap and goal point.

4.2.1 Point robot approach

As it was given in the problem definition, it is assumed that the robot and obstacles are circular objects. In order to simplify the problem given previously, the radius of the robot is added to the obstacle radius as illustrated in Figure 4.1.



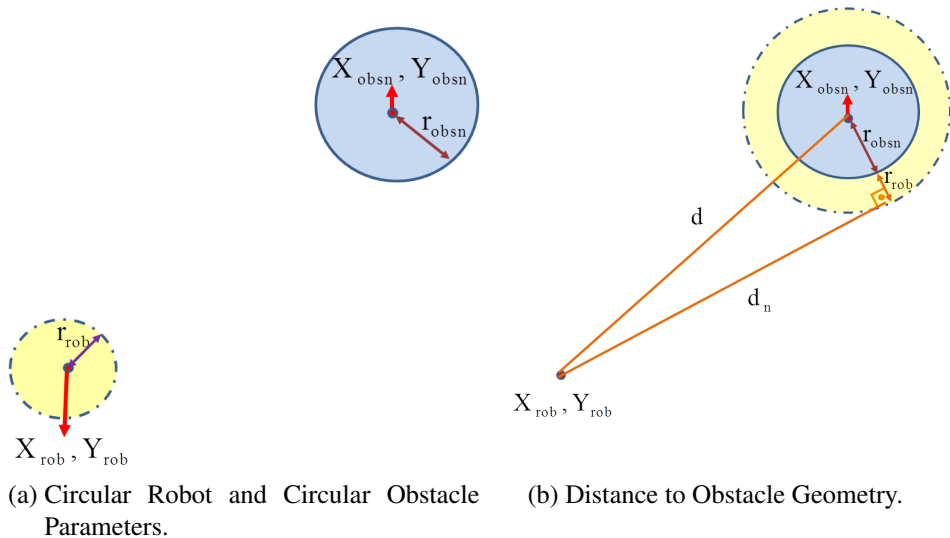
(a) Circular Robot with Circular Obstacles. (b) Point Robot with Enlarged Obstacles.

Figure 4.1: Obstacle Enlargement for Point Robot Representation.

The obstacle avoidance problem for a circular robot is now equivalent to obstacle avoidance for a point robot in Cartesian space. This guarantees a trajectory without collision if any gap is calculated. Otherwise, a collision risk exists because of the physical dimensions of the robot.

4.2.2 Calculation of "distance to obstacle"

Distance to obstacle boundary value will be used for heading angle calculations during the algorithm. For this reason, the formal definition of distance between the robot and the n 'th obstacle border (d_n) is given here. Figure 4.2 illustrates the parameters of the circular robot and the n 'th obstacle.



(a) Circular Robot and Circular Obstacle Parameters.

(b) Distance to Obstacle Geometry.

Figure 4.2: Distance to Obstacle Parameters and Geometry.

Using the distance to obstacle geometry in Figure 4.2b, d_n is found by using the Pythagorean theorem as illustrated in Equation 1.

$$\begin{aligned} d &= \sqrt{(X_{obsn} - X_{rob})^2 + (Y_{obsn} - Y_{rob})^2} \\ \Rightarrow d_n^2 + (r_{obsn} + r_{rob})^2 &= d^2 \\ \Rightarrow d_n &= \sqrt{(X_{obsn} - X_{rob})^2 + (Y_{obsn} - Y_{rob})^2 - (r_{obsn} + r_{rob})^2} \end{aligned} \quad (4.1)$$

In the rest of the chapter, the circular robot is shown as a point robot whereas each obstacle is enlarged with the robot's radius.

4.3 Follow the Gap Method

The follow the gap method is based on the construction of a gap array around the vehicle and calculation of the best heading angle for heading the robot into the center of the maximum gap around, while simultaneously considering the goal point. These two aims are considered simultaneously by using a fusing function. The algorithm can be divided into three main parts as illustrated in Figure 4.3.

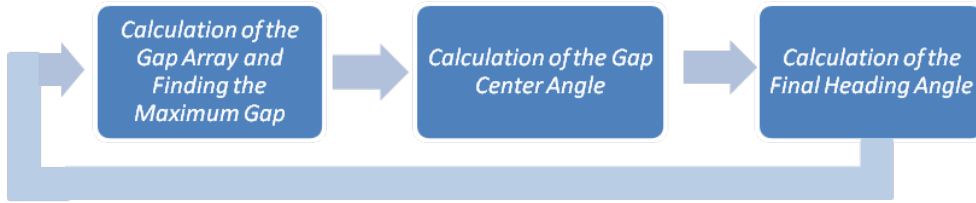


Figure 4.3: Steps of the Follow the Gap Method.

The steps shown in Figure 4.3 are explained in further detail in the rest of the chapter.

4.3.1 Calculation of the gap array and finding the maximum gap

After using the point robot approach, each obstacle around the robot is represented with two parameters. These are the border angle values of the obstacles. A sample of this representation is shown in Figure 4.4. (Φ_{obsi_l} and Φ_{obsi_r} denote the left and right border angle values of the i 'th obstacle respectively.)

In order to calculate the gap array, two more border values are needed in addition to the borders of obstacles. Figure 4.5 shows the evaluation of the gap borders. These border values are found by using the field of view of the robot and the movement constraints. Nonholonomic constraints are very common when writing the kinematic constraints

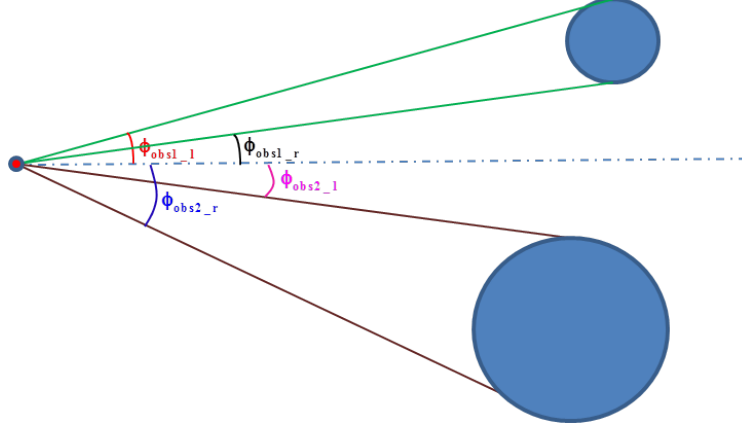


Figure 4.4: Obstacle Representation.

for certain kind of robots. Since the experimental platform is a nonholonomic ground vehicle, nonholonomic constraints are considered. Roughly speaking, a mechanical system is said to be nonholonomic if the vector space of the possible motion directions in a given configuration is restricted such that the restriction can not be converted into an algebraic relation between configuration variables [46]. The physical meaning of this constraint for the vehicle, which is used in simulations and experimental studies, can be observed in its inability to move sideways. Instead of directly moving sideways, the vehicle has to follow an arc to arrive at a lateral coordinate. There have to be both longitudinal and lateral displacements at the same time in a car-like vehicle. Figure 4.5 illustrates the field of view and the nonholonomic movement constraint which construct the border values for the vehicle. More information about nonholonomy can be found in [46] [47].

Figure 4.5a illustrates both the field of view and nonholonomic movement constraint separately. Circles represent the minimum radius (r_{min}) turn of the vehicle coming from the nonholonomic constraint and the dashed lines are for the field of view. Figure 4.5b illustrates the region in which the vehicle can view obstacles and reach any point in spite of its nonholonomic constraints. In order to understand which boundary is active for a boundary obstacle, decision rules are illustrated in Equation 4.2. The parameters of interest are given in Figure 4.6.

$$\begin{aligned} d_{nhol} < d &\Rightarrow \Phi_{lim} = \Phi_{nhol} \\ d_{nhol} \geq d &\Rightarrow \Phi_{lim} = \Phi_{fov} \end{aligned} \quad (4.2)$$

where;

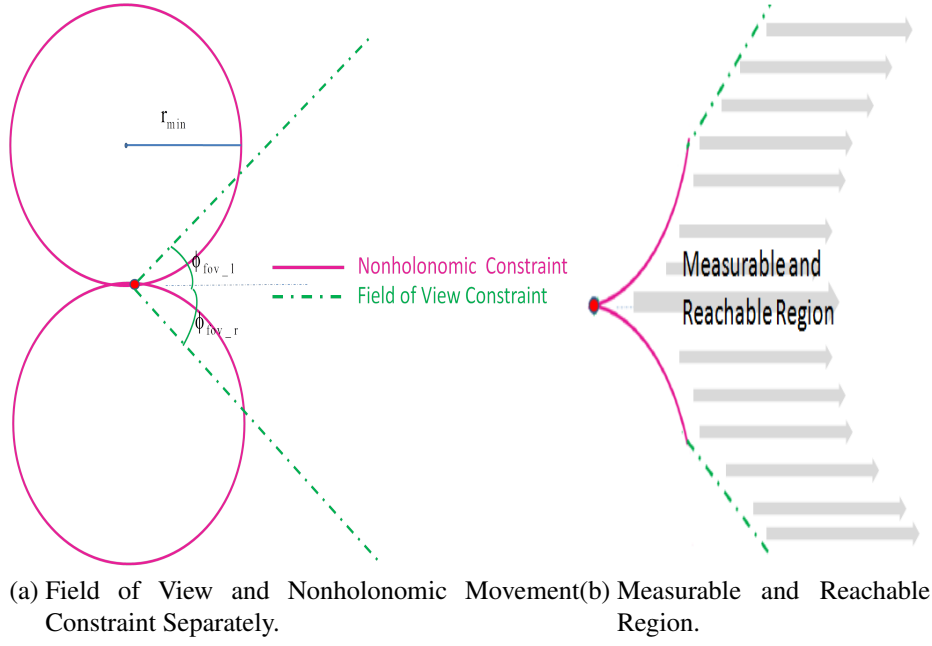


Figure 4.5: Gap Border Evaluation.

Φ_{lim} : Gap border angle (Φ_{liml} : For the left side of the vehicle. Φ_{limr} : For the right side of the vehicle.)

Φ_{nhol} : Border angle coming from nonholonomic constraint. (Φ_{nhol_l} : For the left side of the vehicle. Φ_{nhol_r} : For the right side of the vehicle.)

Φ_{fov} : Border angle coming from field of view. (Φ_{fov_l} : For the left side of the vehicle.

Φ_{fov_r} : For the right side of the vehicle.)

d_{nhol} : Nearest distance between nonholonomic constraint arc and obstacle border. (d_{nhol_l} : For the left side of the vehicle. d_{nhol_r} : For the right side of the vehicle.)

d_{fov} : Nearest distance between field of view line and obstacle border. (d_{fov_l} : For the left side of the vehicle. d_{fov_r} : For the right side of the vehicle.)

In Figure 4.6, d_{fov} , Φ_{fov} , Φ_{nhol} and d_{nhol} expressions are shown for both the left side and the right side of the vehicle. Obstacle 1 and obstacle 3 are boundary obstacles for the left and right side of the vehicle respectively. In an obstacle configuration as illustrated in Figure 4.6, using Equation 4.2, the gap border angle for the left side of the vehicle (Φ_{lim_l}) comes from the nonholonomic constraint (Φ_{nhol_r}) and the gap border angle for the right side of the vehicle (Φ_{lim_r}) comes from the field of view constraint (Φ_{fov_r}).

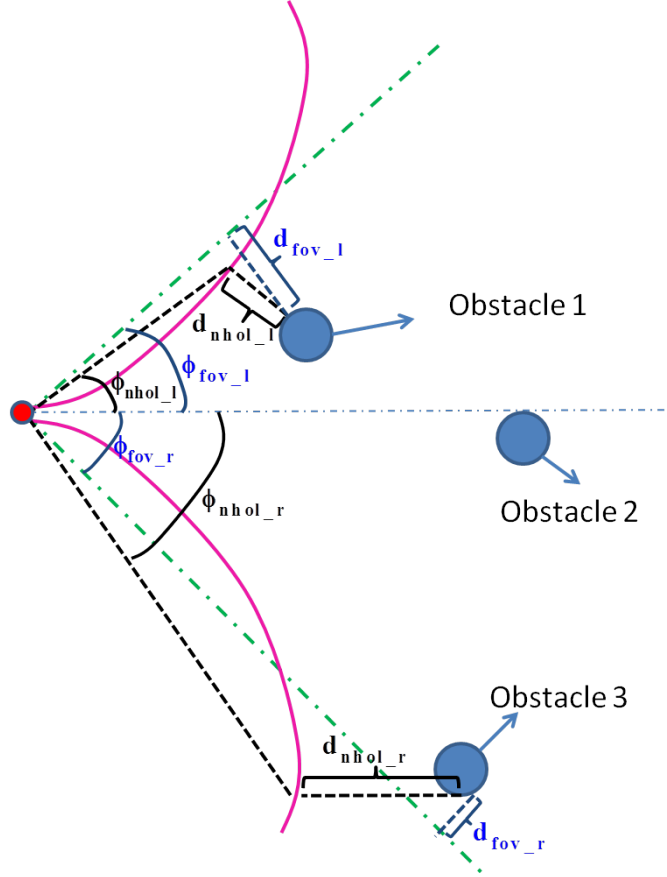


Figure 4.6: Gap Border Parameters.

After the representation of the gap border and obstacle borders, the gap array can be generated. There have to be $N+1$ gaps for N obstacles. $N+1$ elements of gap array are illustrated below in terms of the previous definitions.

$$Gap[N+1] = [(\Phi_{lim_l} - \Phi_{obs1_l})(\Phi_{obs1_r} - \Phi_{obs2_l}) \dots (\Phi_{obs(n-1)_r} - \Phi_{obs(n)_l})(\Phi_{obs(n)_r} - \Phi_{lim(n)_r})] \quad (4.3)$$

The maximum gap is the maximum of the gap array members and is selected for the second step of the algorithm. If more than one maximum gap with the same value exists, the algorithm selects the first calculated one.

4.3.2 Calculation of the gap center angle

After finding the maximum gap from the previous section, it is essential to find the gap center angle (Φ_{gap_c}) which ensures that the robot move through the center of the maximum gap. This is the angle of the median vector from robot to the line between the obstacles creating/causing the maximum gap. The gap center angle is shown in Figure 4.7.

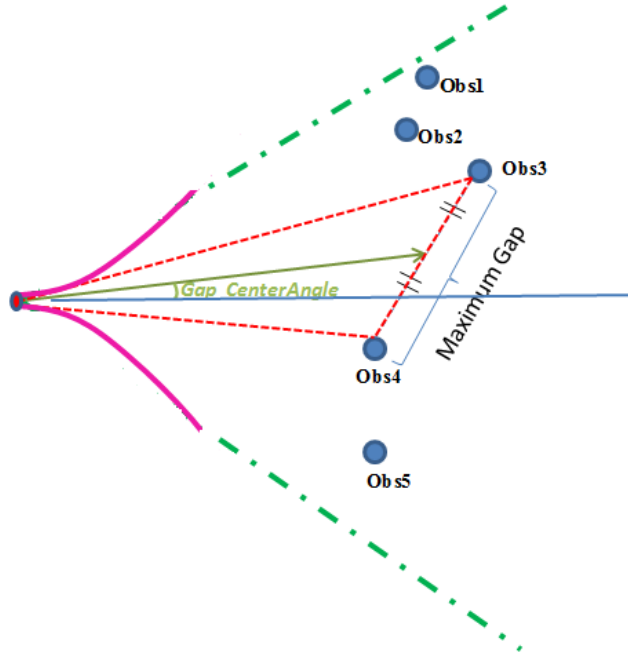


Figure 4.7: Gap Center Angle Representation.

In order to find the gap center angle, the Cosine and Apollonius theorems are used. The details of these theorems are given in Appendix.

It can be seen from Figure 4.7 that there is a triangle with dashed lines and the aim is to find the angle of median vector of this triangle. The triangle and the gap center angle are shown in detail in Figure 4.8.

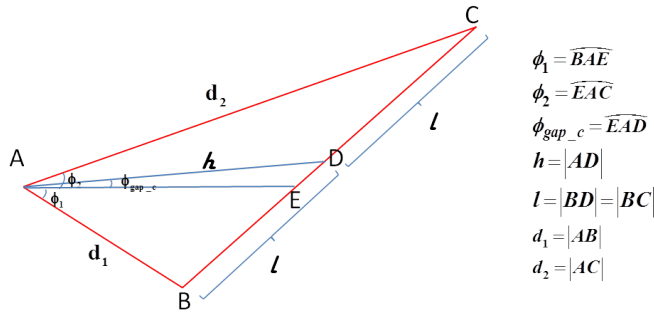


Figure 4.8: Gap Center Angle Parameterization.

The aim is to find the Φ_{gap_c} in terms of the measurable d_1, d_2, Φ_1, Φ_2 parameters which are shown in Figure 4.8. (d_1, d_2 are distances to obstacles from the maximum gap. Φ_1, Φ_2 are angles from obstacles of the maximum gap) Direction of the positive angle values are counterclockwise from AE line while the negative direction angle values are clockwise.

Firstly, the Cosine Rule is applied to the ABC triangle as it is illustrated in Equation 4.4.

$$(2l)^2 = d_1^2 + d_2^2 - 2d_1d_2 \cos(\Phi_1 + \Phi_2) \quad (4.4)$$

$$l^2 = \frac{d_1^2 + d_2^2 - 2d_1d_2 \cos(\Phi_1 + \Phi_2)}{4}$$

After that, the Apollonius theorem is applied to the ABC triangle.

$$d_1^2 + d_2^2 = 2l^2 + 2h^2 \quad (4.5)$$

Replacing l^2 with Equation 4.4;

$$h^2 = \frac{d_1^2 + d_2^2 + 2d_1d_2 \cos(\Phi_1 + \Phi_2)}{4} \quad (4.6)$$

Finally, the Cosine Rule is again applied to the ABD triangle as illustrated in Equation 4.7.

$$l^2 = d_1^2 + h^2 - 2d_1h \cos(\Phi_1 + \Phi_{gap_c}) \quad (4.7)$$

Replacing l^2 and h^2 with Equation 4.4 and Equation 4.6;

$$\frac{d_1^2 + d_2^2 - 2d_1d_2 \cos(\Phi_1 + \Phi_2)}{4} =$$

$$d_1^2 + \frac{d_1^2 + d_2^2 + 2d_1d_2 \cos(\Phi_1 + \Phi_2)}{4} - 2d_1 \frac{\sqrt{d_1^2 + d_2^2 + 2d_1d_2 \cos(\Phi_1 + \Phi_2)}}{2} \cos(\Phi_1 + \Phi_{gap_c})$$

$$\Rightarrow \Phi_{gap_c} = \arccos\left(\frac{d_1 + d_2 \cos(\Phi_1 + \Phi_2)}{\sqrt{d_1^2 + d_2^2 + 2d_1d_2 \cos(\Phi_1 + \Phi_2)}}\right) \quad (4.8)$$

The gap center angle (Φ_{gap_c}) is found in terms of the measurable d_1 , d_2 , Φ_1 , Φ_2 parameters in Equation 4.8.

The gap center angle could be found by calculating the average of Φ_1 and Φ_2 angles, ignoring the d_1 and d_2 distance values.

$$\Phi_{gap_c_basic} = \frac{\Phi_1 + \Phi_2}{2} \quad (4.9)$$

This variant of FGM, in which $\Phi_{gap_c_basic}$ is used instead of Φ_{gap_c} is called FGM-basic for comparisons in the simulations section. However, FGM-basic causes unsafe trajectories in some cases. The simulation results of such an obstacle configuration are illustrated in Figure 4.9.

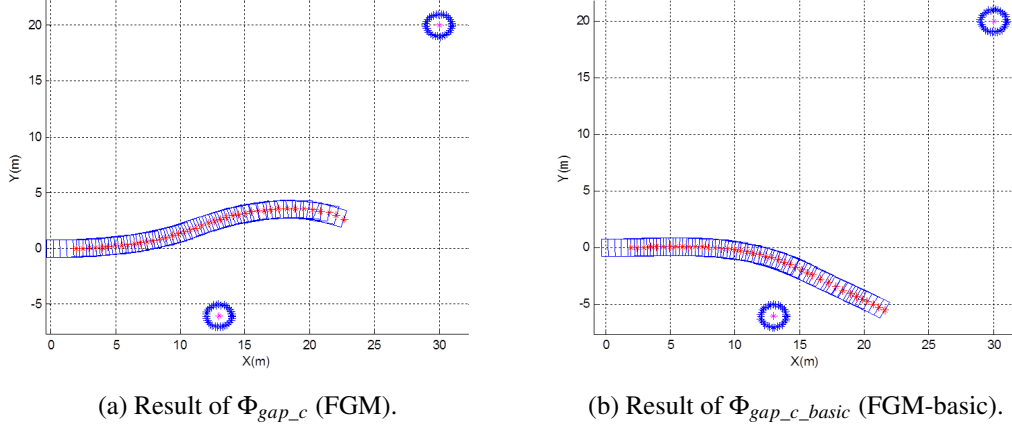


Figure 4.9: Comparison of Φ_{gap_c} and $\Phi_{gap_c_basic}$ Gap Angle Calculation.

Figure 4.9a shows the result of Φ_{gap_c} and Figure 4.9b shows the result of $\Phi_{gap_c_basic}$. There are two obstacles at the same coordinates for both of the simulations. It is shown that, Φ_{gap_c} has resulted in a safer trajectory. This is due to the calculations for finding the angle of the median vector from the robot to the line between the obstacles, as it was explained previously.

Here, with the term “safe”, maintaining a safe distance from the obstacle is meant. In the simulations section, a metric is defined for measuring this safety and for measuring the performances of different methods.

4.3.3 Calculation of the final heading angle

The last step of the algorithm is the calculation of the final heading angle. This final angle reference will be the reference heading vector of the robot for avoiding obstacles and arriving at the goal point. A sample configuration including obstacles and a goal point is illustrated in Figure 4.10.

The final angle is the combination of the gap center angle and goal angle. The combination structure depends on the minimum distance to obstacles around and weight coefficients. If the obstacles are near the robot, it should consider safety first;

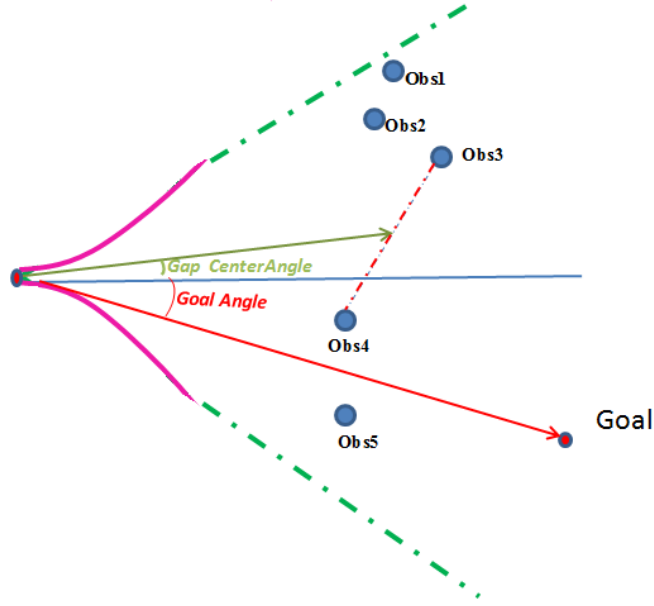


Figure 4.10: Obstacles with Goal Point.

i.e. the gap center angle, rather than the goal angle and vice versa. Fusing function which is illustrated in Equation 4.10 depicts the final angle calculation.

$$\Phi_{final} = \frac{\frac{\alpha}{d_{min}} \Phi_{gap_c} + \beta \Phi_{goal}}{\frac{\alpha}{d_{min}} + \beta} \quad \text{where } d_{min} = \min_{i=1:n}(d_n) \quad (4.10)$$

(Φ_{gap_c} : Gap Center Angle. Φ_{goal} : Goal Angle. α : Weight Coefficient for gap. β : Weight Coefficient for goal. n : Number of obstacles. d_n : Distance to n'th obstacle. d_{min} : Minimum of d_n distance values.)

The ratio of α and β determines the behaviour of the robot. For this reason, two parameters are not necessary for tuning. β is taken as “1” for simplicity and the α coefficient is used as a weight factor between the gap center angle and the goal angle. Equation 4.10 can be rewritten as:

$$\Phi_{final} = \frac{\frac{\alpha}{d_{min}} \Phi_{gap_c} + \Phi_{goal}}{\frac{\alpha}{d_{min}} + 1} \quad \text{where } d_{min} = \min_{i=1:n}(d_n) \quad (4.11)$$

This weighted average function is not only dependent on tuning parameter but also the minimum distance to the obstacle. Theoretically, the final angle approaches the gap angle when the distance to the obstacle approaches zero. The fusing function is

designed specifically to reflect this property. This can be seen from the Equation 4.12 by using the l'Hospital's rule.

$$\lim_{d_{min} \rightarrow 0} \Phi_{final} \cong \frac{\infty}{\infty} \Rightarrow \lim_{d_{min} \rightarrow 0} \Phi_{final} \cong \frac{\frac{\partial(\frac{\alpha}{d_{min}}\Phi_{gap_c} + \beta\Phi_{goal})}{\partial d_{min}}}{\frac{\partial(\frac{\alpha}{d_{min}} + \beta)}{\partial d_{min}}} = \frac{-\frac{\alpha\Phi_{gap_c}}{d_{min}^2}}{\frac{-\alpha}{d_{min}^2}} = \Phi_{gap_c} \quad (4.12)$$

Figure 4.11 shows how Φ_{final} changes with respect to α and minimum obstacle distance. For this figure, Φ_{gap_c} is $\pi/4$ and Φ_{final} is $-\pi/4$ which means the gap center and the goal are in different directions.

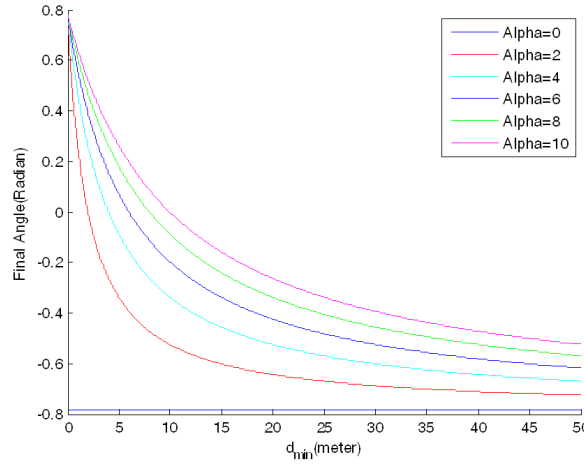


Figure 4.11: Final Angle with Respect to Minimum Distance and α Coefficient ($\Phi_{gap_c} = \pi/4$ radian and $\Phi_{goal} = -\pi/4$ radian).

According to the Figure 4.11, Φ_{final} converges to Φ_{gap_c} ($\pi/4$) for decreasing values of d_{min} (an obstacle is approaching to the robot) and Φ_{final} converges to Φ_{goal} ($-\pi/4$) for increasing values of d_{min} (an obstacle is moving far away). The α value determines how much the robot is goal oriented or gap oriented. For $\alpha = 0$, Φ_{final} is equal to $-\pi/4$ and increasing values of alpha brings Φ_{final} closer to $\Phi_{gap_c}(\pi/4)$ as is illustrated in Figure 4.11.

When compared to other obstacle avoidance methods, FGM has several advantages. FGM has only one tuning parameter, which is the α coefficient in Equation 4.11. So tuning the algorithm is very easy. FGM does not have the local minimum problem, as in APF and VFF. Different from previous avoidance algorithms, this new approach can consider the nonholonomic constraints of the robot and only feasible trajectories are generated. Another advantage is that the field of view of the robot is taken into account

and the robot is not forced to move toward unmeasured directions. Finally, FGM steers the robot to move towards the center of the maximum gap as much as possible. This results in safer trajectories compared with previous approaches, as will also be shown in simulations.

4.4 Simulations

Kinematic vehicle model which has been illustrated in Chapter 2 is used in simulations.

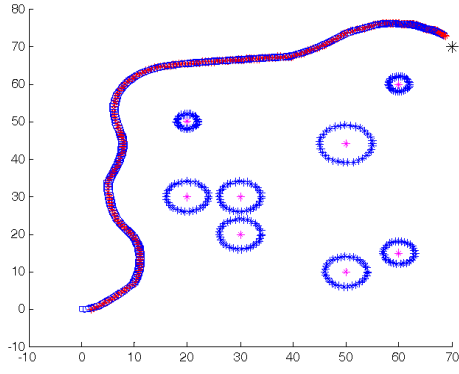
Every obstacle avoidance algorithm has tuning parameters for performance improvements. Since a fair benchmarking is a very challenging problem for different obstacle avoidance methods [48], comparisons have been done with the previously tuned APF algorithm [49], as APF is one of the most widely used obstacle avoidance algorithms in robotics area.

A series of Monte Carlo simulations are performed over random environments to compare the performance of the FGM, FGM-basic, APF and A* methods. FGM-basic is a variant of FGM, in which $\Phi_{gap_c_basic}$ is used instead of Φ_{gap_c} as explained in Section 4.3.2.

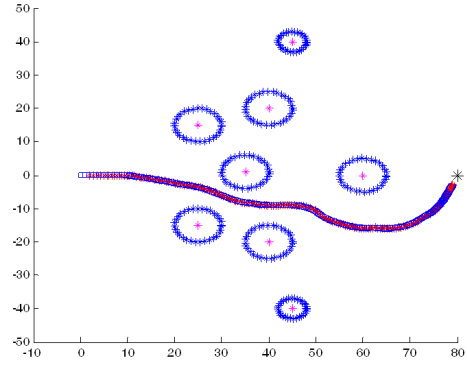
All the algorithms are coded in S-Function using C programming language. A stereo LIDAR (Light Detection and Ranging) sensor is mounted in front of the vehicle with total 150 degree field of view and 10m range as the experimental platform used in simulations. Circular obstacles with random radius and coordinate values are used in simulations. Even though the vehicle has a rectangular shape, an equivalent minimum radius circular robot is used for the calculations. This is done by taking the diagonal of the rectangle as the diameter of the circle. Tuning parameter alpha parameter is selected as 20. Simulation results of FGM for two different obstacle configurations are shown in Figure 4.12. Goal coordinates are [70-70] for Configuration-A and [80-0] for Configuration-B.

4.4.1 Artificial potential field theory

Before giving the comparisons in Section 4.4.2, APF method's equations are illustrated here. In the APF approach, the obstacles to be avoided are represented by a repulsive



(a) Configuration A.



(b) Configuration B.

Figure 4.12: Simulation Results of FGM for Different Obstacle Configurations.

artificial potential and the goal is represented by an attractive potential, so that a robot reaches the goal without colliding with obstacles as it is illustrated in Figure 4.13.

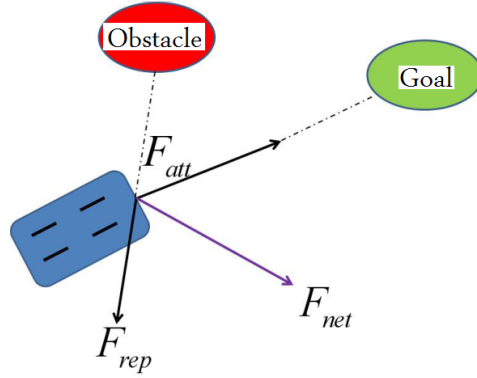


Figure 4.13: APF Forces.

In general, the robot is represented as a particle under the influence of a scalar potential field U [40], defined as follows.

$$U = U_{att} + U_{rep}, \quad (4.13)$$

where U_{att} and U_{rep} are the attractive and repulsive potentials respectively. The attraction influence tends to pull the robot towards the target position, while repulsion tends to push the robot away from the obstacles.

The vector field of artificial forces $F(q)$ is given by the gradient of U .

$$F_{net} = -F_{att} + F_{rep} = -\nabla U_{att} + \nabla U_{rep} \quad (4.14)$$

The most commonly used form of potential field functions proposed by Khatib [37] are defined as follows:

Repulsive Potential Field

$$U_{rep} = \begin{cases} \frac{1}{2}\eta(\frac{1}{d} - \frac{1}{d_0})^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases} \quad (4.15)$$

where $d = |q - q_o|$, q is the current position of the robot, q_o is the obstacle position, d_0 is the influence distance of the force and η is an adjustable constant.

Attraction Potential Field

$$U_{att} = \frac{1}{2}\xi d^2 \quad (4.16)$$

where $d = |q - q_a|$, q is the current position of the robot, q_a is the position of an attraction point and ξ is an adjustable constant.

The corresponding force vectors for the fields above can be found by calculating the gradients of these potentials. Following equations illustrates the attraction and repulsion forces.

$$F_{att}(q) = -\nabla U_{att} = \xi(q - q_a) \quad (4.17)$$

$$F_{rep}(q) = \nabla U_{rep} = \begin{cases} \frac{1}{2}\eta(\frac{1}{d} - \frac{1}{d_0})\frac{(q-q_0)}{d^3} & d \leq d_0 \\ 0 & d > d_0 \end{cases} \quad (4.18)$$

Direction of the total force vector F_{net} gives the desired heading angle which is calculated as illustrated in the following equation.

$$\Phi_{des} = \arctan\left(\frac{F_{netY}}{F_{netX}}\right) \quad (4.19)$$

Previously tuned parameters for the APF method is illustrated in Table 4.1.

Table 4.1: APF Parameters.

η	ξ	$d_0(m)$
500	0,2	20

These APF parameters are used for both simulations and experimental tests.

4.4.2 Comparisons

As it was mentioned before, FGM does not suffer from the local minimum problems except the dead-end scenarios. Point robot can always avoid obstacle and reach to the goal if there is a gap around itself. However in APF and APF based algorithms like VFF, robot can not calculate any heading reference due to the local minimum problem even there are gaps around itself. Figure 4.14 shows the trajectories of APF and FGM in two different local minimum scenarios.

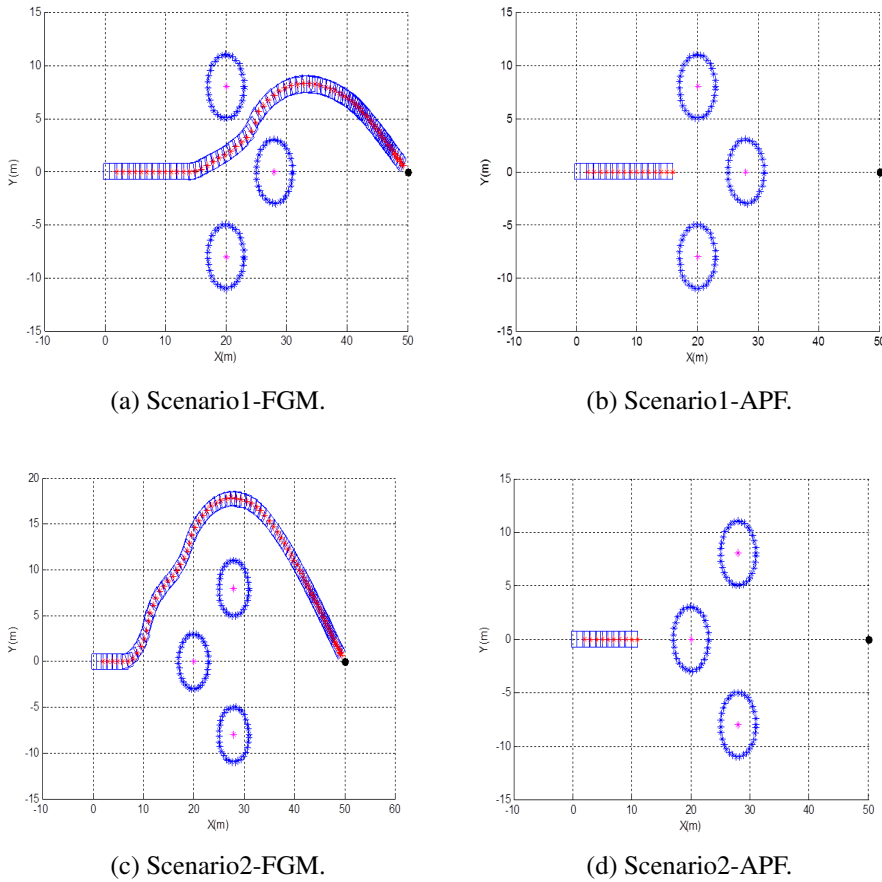


Figure 4.14: Trajectories of APF and FGM in Local Minimum Scenarios.

As it can be seen from the simulation results, FGM can reach the goal point while avoiding obstacles but in APF method, robot gets stuck because of the local minimum where all vectors from the obstacles and goal point zero each other. FGM selects the first calculated gap value if there are equal maximum gaps as mentioned in Section 4.3.1. This provides FGM to move if at least one gap exists. However, as a result of its local characteristic, a dead-end scenario of U-shaped obstacles is a problem for FGM as it is for APF. Detecting the dead-end scenario and avoiding it needs an upper level intelligence for such kind of local planners and can be solved with several

ad-hoc approaches which have been studied in the literature extensively. In [42] virtual obstacles and in [50] virtual goal points are added to the local map for trap conditions. Beside these, [40] uses multiple goal points and [43] uses the wall following technique for the same aim. One of these solutions can be used directly in FGM for dead-end scenarios.

The following metric for obstacle avoidance safety is defined for the comparison. This collision avoidance metric is based on the inverse of the distance-to-obstacle function. The same metric was also used in [51].

$$f(t) = \begin{cases} \frac{1}{d_{min}} - \frac{1}{d_0} & \text{for } d_{min} < d_0 \\ 0 & \text{for } d_{min} \geq d_0 \end{cases} \quad (4.20)$$

where d_{min} is the closest distance between the vehicle and obstacles and the given scalar d_0 denotes the distance to an obstacle that poses no danger for collision during execution.

The p'th norm of a function is defined as:

$$\|f\|_p = \left(\int_{t_0}^t |f(t)|^p dt \right)^{\frac{1}{p}} \quad (4.21)$$

In this chapter, the first norm (p=1) of the collision avoidance metric is calculated and taken as a performance criterion. This performance criterion measures the safety of the trajectory or in other words how far away from obstacles the trajectory is. The aim of FGM is obtaining the maximum distance to obstacles as much as possible reactively. This brings an additional path length to the shortest path. Figure 4.15 illustrates the trajectories of FGM, APF and A* shortest path algorithm for three different obstacle scenarios. A* uses a best-first search and finds the shortest path between starting and goal points [52]. A* is a global planning algorithm which means all the information about the map is given before it starts, in contrast to reactive algorithms like FGM and APF. In simulations, the A* algorithm uses 0,5 m grid size and does not consider any measurement or kinematic constraint.

Forty Monte Carlo simulations are performed for FGM, FGM- basic, APF and A* search methods and the results are illustrated in Figure 4.16. The collision avoidance norm value of the A* algorithm is not added in Figure 4.16 because the collision

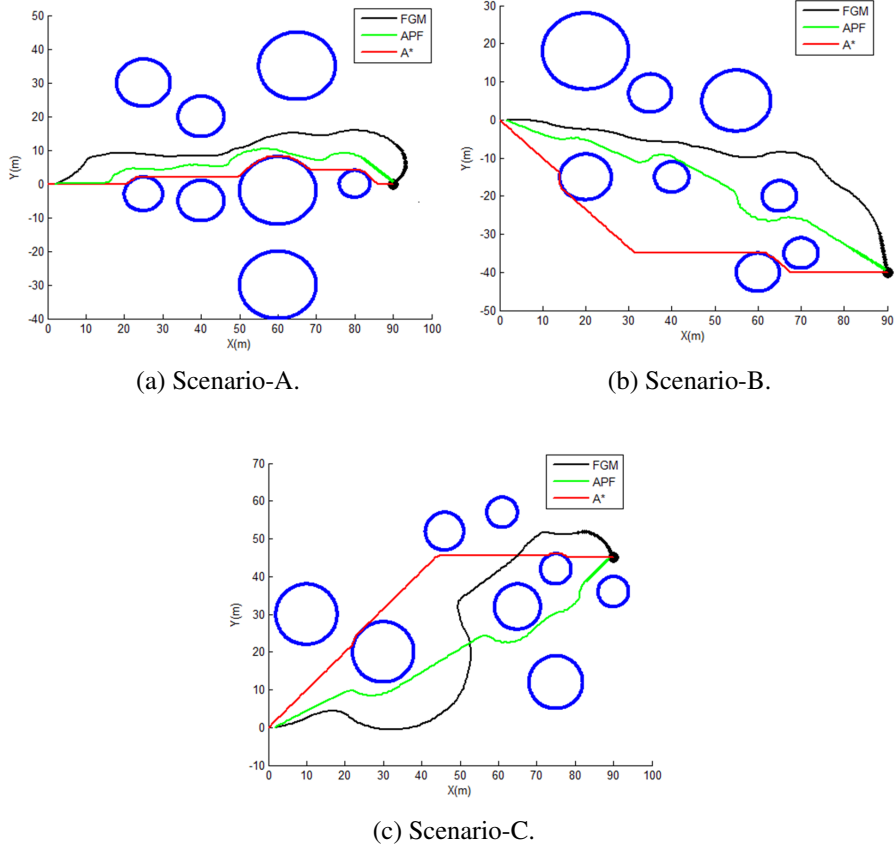


Figure 4.15: Comparison of FGM, APF and A* Shortest Path Trajectories.

avoidance norm approaches infinity as the distance to obstacle value approaches zero. This can be seen in Equation 16.

According to Figure 4.16, the collision avoidance norm of FGM is lower than both the APF and FGM-basic methods which means the trajectories are safer. Despite this, total traveled distance values are almost the same as it is illustrated in Figure 4.16. Average values of collision norm and distance traveled for this 40 Monte Carlo simulations are illustrated in Table 4.2.

Table 4.2: Average of Simulation Results ($d_0 = 25m$).

	Average Collision Norm	Average Distance Traveled(m)
FGM	0.186	65.82
FGM-basic	0.243	64.94
APF	0.31	63.57
A*	$\simeq \infty$	58.81

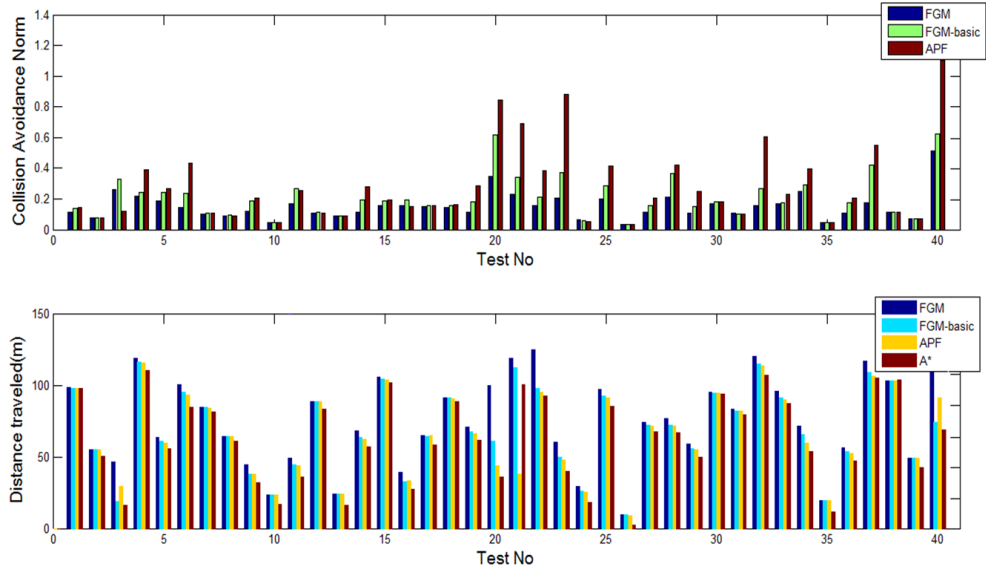


Figure 4.16: Collision Avoidance Norm and Traveled Distance Values for Follow the Gap, Follow the Gap-basic and Artificial Potential Fields Methods.

According to Table 4.2, FGM is % 23 safer than the FGM-basic and % 40 safer than the APF in terms of the norm of the defined metric while the total distance traveled values are almost the same.

4.5 Experimental Tests

After designing the algorithm and successful simulation results, the Follow the Gap method is implemented on the real autonomous vehicle, ‘Otonobil’. Otonobil has been explained in Chapter 3.

The obstacle avoidance algorithm based on the Follow the Gap Method is coded using C programming language. The real-time code runs in Microautobox hardware, which is listed in Table 3.2 and shown in Figure 3.11. The only tuning parameter, alpha, is selected as 20 in experimental tests as in the simulations. The field of view of the LIDAR is 150 degree and its measurement range is restricted to 10m. The first test configuration is composed of seven static obstacles with a goal point. The test field and the obstacle configuration are illustrated in Figure 4.17. Vehicle is driven with a 8km/h constant speed value in these tests. Speed planning and low level speed controller design subjects are explained in Chapter 5 and Chapter 6.

Results of the algorithm are shown in Figure 4.18. Figure 4.18a shows the trajectory of the vehicle with black dots. Blue dots are obstacle measurement results from the

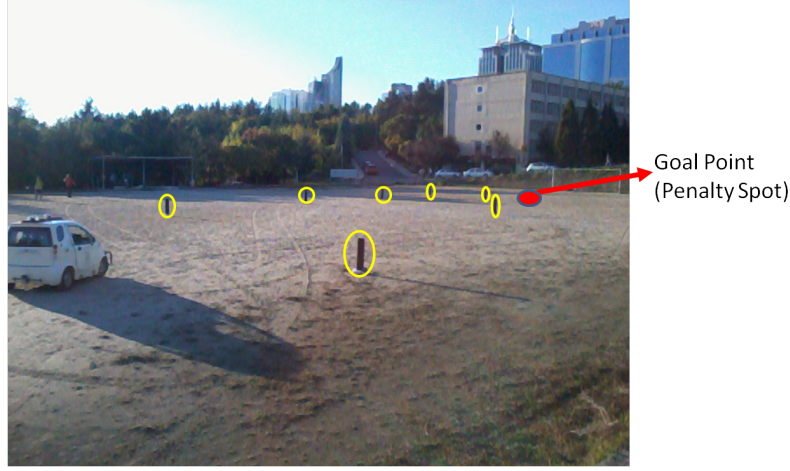
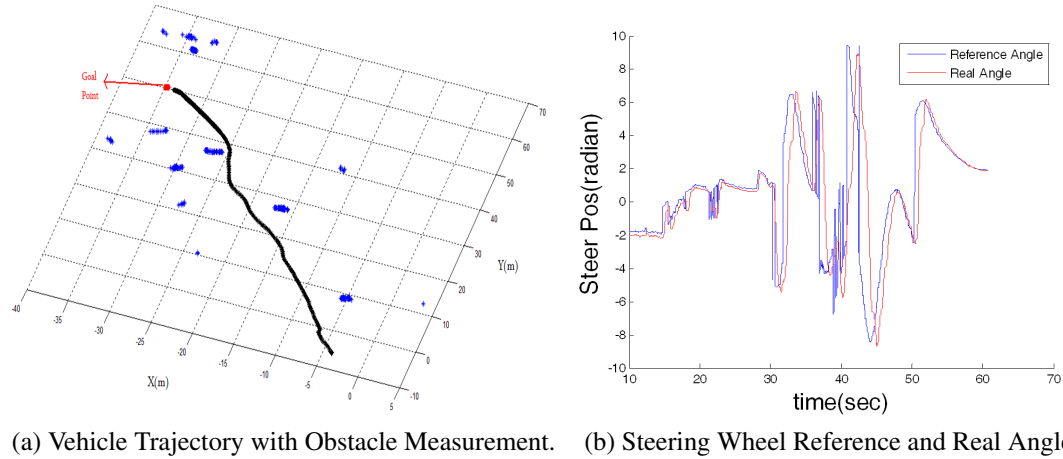


Figure 4.17: Test Field and First Obstacle Configuration.

LIDAR sensor. The goal point coordinates are $[-35, 50]$ which is shown with a red dot and starting point coordinates are $[-3, -3]$. As can be seen in Figure 4.18a, static perceived obstacles and even the goal point are a cluster of scattered points. The reason for the scatter is that the LIDAR and GPS receivers have some measurement error and noise. These two error sources directly affect the coordinate transformation calculations. Figure 4.18b shows the steering wheel reference angle and steering wheel real angle values during its journey. Steering motor's own controller [53] is used for tracking the desired steering angle.

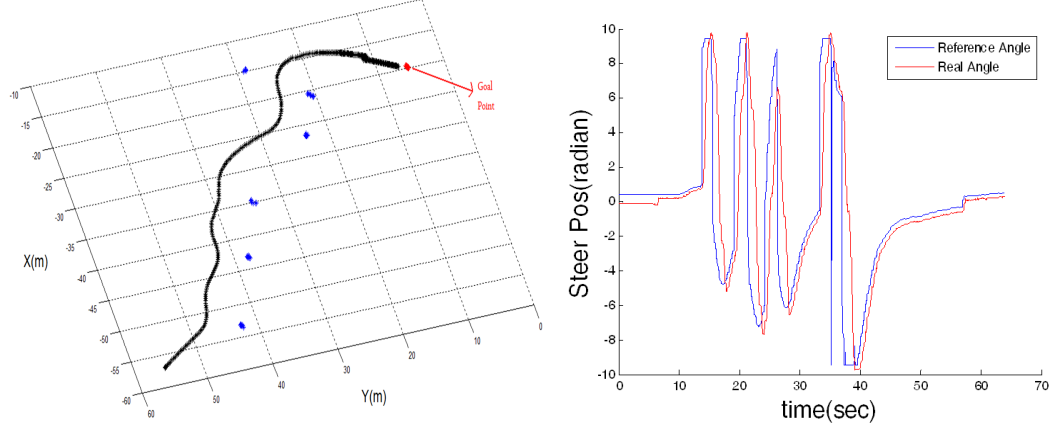


(a) Vehicle Trajectory with Obstacle Measurement. (b) Steering Wheel Reference and Real Angle.

Figure 4.18: Experimental Results of Test-1.

The results of the second test are illustrated in Figure 4.19. This time the obstacle configuration is different with the starting point coordinates $[-57, 58]$ and goal point coordinates $[-20, 5]$. According to Figure 4.19, the vehicle heads itself towards the goal point which means the final angle is equal to the goal angle, until it measures an obstacle. When it detects an obstacle, the appropriate heading vector is calculated by

the Follow the Gap Method and the vehicle avoids obstacles and arrives at the goal point. The FGM tries to maximize the distance to obstacle value while considering the goal point.



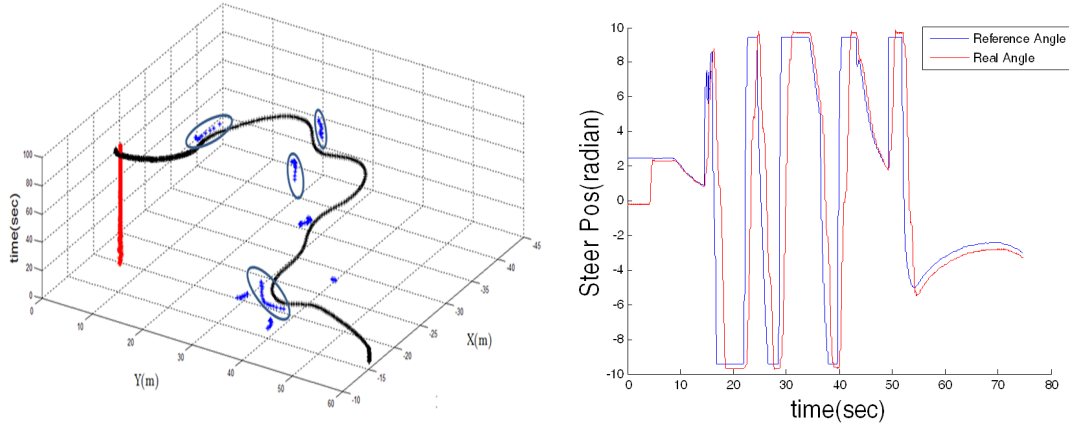
(a) Vehicle Trajectory with Obstacle Measurement. (b) Steering Wheel Reference and Real Angle.

Figure 4.19: Experimental Results of Test-2.

Finally, the Follow the Gap Method is tested in a dynamic obstacle scenario. Figure 4.20 shows the trajectory of the system in a dynamic obstacle case. In Figure 4.20a, for better analysis, the dynamic obstacle trajectory is illustrated in a 3-D plot where the third axis is for time. In this 3-D illustration, collision means the intersection not only for X-Y axis but also for X-Y and time axes. According to Figure 4.20, there is no collision and the vehicle avoids dynamic obstacles successfully. Dynamic obstacles are moving obstacles with unknown motion patterns, which are circled in Figure 4.20a. These moving obstacles are the people who are running in front of the vehicle in this scenario. A video demonstrating the vehicle performance within this scenario can be found at <http://www.youtube.com/watch?v=TohW9xokbaM>.

4.6 Conclusions

A new approach called the “Follow the Gap Method” is presented in this chapter. The proposed algorithm is proven to be successful in driving the nonholonomic autonomous ground vehicle from an arbitrary initial location to a goal location while avoiding collision with static and dynamic obstacles. The major contributions can be listed as follows.



(a) Vehicle Trajectory with Obstacle Measurement. (b) Steering Wheel Reference and Real Angle.

Figure 4.20: Experimental Results of Test-3.

- Gap center angle formulation enables the robot to head to the center of the maximum gap around which results in safe trajectories. Figure 4.16 shows this type of comparison between the Follow the Gap Method, Follow the Gap Method-basic and Artificial Potential Fields Method using Monte Carlo simulations.
- There is no local minimum problem. The APF method and VFF method have this serious problem as explained in the introduction.
- Nonholonomic constraint of the robot is taken into account and feasible trajectories are generated while the other methods do not have this property.
- The field of view of the robot is taken into account and the robot is not forced to move towards unperceived directions.
- Follow the Gap algorithm is easy to tune with only one tuning parameter, “alpha”.

In spite of successful results in both simulations and experimental tests, this new algorithm can be further extended to the dynamic model of the robot. This chapter illustrates the concept of the algorithm and it is seen that the kinematic vehicle model is sufficient for the proof of concept. However, at high speed values, a dynamic model and additional constraints from the dynamic vehicle model should be taken into account.

Additional simulation results using dynamic model of the vehicle with newly designed speed planning and speed control algorithms are illustrated in Chapter 7. Experimental

tests of FGM with new designed speed planning and control algorithms are also illustrated in Chapter 7 in static and dynamic obstacle scenarios.

5. A NEW FUZZY SPEED PLANNING METHOD FOR SAFE NAVIGATION

This chapter introduces a new speed planning strategy for autonomous navigation. Speed planning can be done considering lots of parameters using offline path planning strategies. However, in an obstacle avoidance scenario, which can be thought as a dynamic path planning, avoidance strategy must work fast. That is why previous strategies generally concentrate only on steering maneuvers for obstacle avoidance. This chapter concentrates on the speed planning part of the obstacle avoidance strategy. To this aim, a new speed planning approach using fuzzy logic [54] is developed for on-line speed planning with its purely reactive nature in this chapter. Methodic simulations are carried out to verify and demonstrate the effectiveness of the new method over previous methods. The maneuver strategy for obstacle avoidance is the artificial potential field method.

5.1 Introduction

Automatic vehicle speed determination is presently one of the hottest research topics in the intelligent transportation systems area. Cruise control (CC) and adaptive cruise control (ACC) systems have been very popular in the last few decades. In CC systems, the driver decides the reference speed and the vehicle automatically adjusts its speed to the predetermined value. In ACC systems, the vehicle automatically adjusts its velocity in order to maintain a proper distance between the preceding vehicle [55]. Both CC and ACC systems bring semi-autonomy to the ground vehicle technology.

Obstacle avoidance is a dynamic path planning strategy which is activated when an unexpected obstacle is met. Studies about obstacle avoidance for UGV are mostly about the steering maneuvers. However, desired speed determination is as important as the steering maneuver determination for avoiding obstacles. Figure 5.1 illustrates the role of the desired speed determination in an autonomous vehicle. Most of the obstacle avoidance algorithms ignore the desired speed determination part and separated solutions are developed for the speed planning strategy.

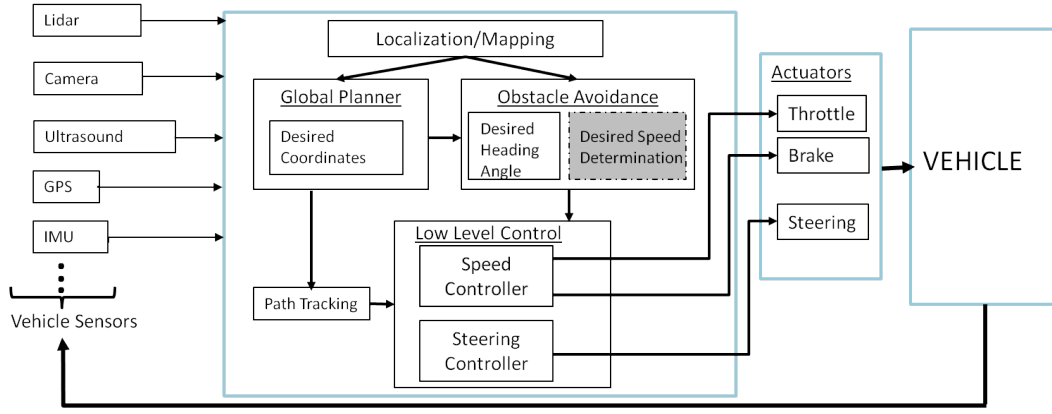


Figure 5.1: Role of Desired Speed Determination in a UGV.

Generally, speed reference is determined by using the information about road curvature and vehicle dynamics for ground vehicles [56] [57]. But the condition of the environmental factors, such as the obstacle density and minimum distance to obstacle values are very important for autonomous vehicles reference speed determination.

In [56], velocity reference is calculated from both the road curvature and the minimum obstacle distance values using fuzzy rules. In [58], desired velocity is planned using the minimum distance to obstacle and the angle to the goal position values using fuzzy logic. In [59], the criterion for vehicle speed determination is minimizing the lateral acceleration which is mostly related with vehicle dynamics. In [60], speed planning is performed using both vehicle dynamics and path properties. In [61], a speed planning algorithm is designed which uses the angle between each path segment related with the curvature of the planned trajectory.

In this chapter, a new type velocity planning method is developed using fuzzy logic. This fuzzy system consists of two cascade connected Mamdani-type Fuzzy Inference System for desired speed determination. This strategy considers both the danger level of the obstacles around and lateral dynamics of the vehicle. Different from other studies, the desired speed is found by considering the minimum distance between obstacle and vehicle, the angle of the nearest obstacle and the steering angle values together. The developed automatic speed determination strategy is used with the artificial potential field (APF) obstacle avoidance method [37] and compared with two speed planning methods that were used in APF, as a case study. APF theory has been given in Chapter 4.4.1. Another advantage of this new fuzzy system is its independent structure from the used obstacle avoidance method. It can be used with

any of the obstacle avoidance methods in literature. This new method is used with the new obstacle avoidance method FGM in Chapter 7.

5.2 Speed Planning Methods for APF

Originally, APF calculates the direction of the desired heading vector F_{net} for obstacle avoidance as it was illustrated in Figure 4.13. However, it does not contain any information about the speed reference of the robot. There are some studies about desired speed calculation for potential field method. Two of these methods are illustrated below which are used for the comparisons with the new designed strategy.

5.2.1 Method A

In [43], desired speed is calculated as:

$$V_{des} = \begin{cases} V_{max} & \text{for } |F_{rep}| = 0 \\ V_{max}(1 - |\cos \Theta|) & \text{for } |F_{rep}| > 0 \end{cases} \quad (5.1)$$

Where, Θ is the angle of heading vector. This method runs the robot at its maximum speed if there is not any obstacles around and the speed is reduced proportional to the angle of the obstacle ahead.

5.2.2 Method B

In [62], another speed determination strategy is given as illustrated in the following equation.

$$V_{des} = V_{input} e^{-\Sigma x_F} \quad (5.2)$$

Where x_F is the x-part of repulsive forces F_{rep} and V_{input} is the desired speed vector which comes from static path planner. Any obstacle through the motion direction of the vehicle reduces the desired speed of the static path planner.

In [40], desired speed is again the function of x_F like in [62]. But this time the relation is linear as shown in the following equation.

$$V_{des} = Cx_F \quad (5.3)$$

Where C is the velocity coefficient.

None of these methods consider both the distance and angle of the obstacle simultaneously. These two features are directly relational with the danger level of the obstacles. As a human, we plan our trajectory using both of these two information. If someone is near to us in the same direction, we have to reduce our speed and change our direction. If it is near but in another direction, this is less critical and there is no need to reduce the speed as much as the previous scenario. This new fuzzy method evaluates both the distance and the angle of the obstacle for speed planning.

Another advantage of this new speed planning method is its consideration of the vehicle dynamics at the same time. Avoidance algorithms can calculate aggressive steering references independent of the vehicle's speed and this might cause the loss of vehicle stability. This new method considers the steering angle of the vehicle besides the distance and angle of the nearest obstacle for speed planning.

5.3 Fuzzy Decision Making for Speed Planning

Design of the new speed planning algorithm using fuzzy logic rules is explained in the following three subsections.

5.3.1 Method overview

Determination of the desired speed of an autonomous robot directly affects the performance of the obstacle avoidance algorithm. Motivated by the fact that human performance is reliable in driving ground vehicle, fuzzy logic methods are used in order to mimick the experiences of the human driver without dealing with complicated mathematical models.

In this chapter, three parameters are used for the calculation of the desired speed with fuzzy logic. Two of them are for the danger level of the obstacle. These are the values of minimum distance to vehicle d_{min} and angle of the nearest obstacle ϕ_{near} . These two inputs are for the first fuzzy block. This block determines the risk level of the environment. The other parameter is the steering angle δ , which is used for stable vehicle dynamics in the second fuzzy block. Figure 5.2 illustrates the schematic of the cascaded fuzzy decision making approach.

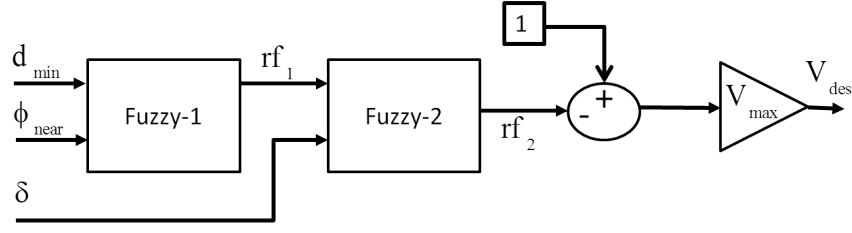


Figure 5.2: Cascade Fuzzy Connection for Speed Planning.

rf_1 and rf_2 are risk factor 1 and risk factor 2 values which vary between zero and one. Zero means no risk and one means full of risk. V_{max} is the maximum speed which can be thought as the gain coefficient and V_{des} is the desired speed value.

5.3.2 Design of Fuzzy-1 block

In human driving, we classify the objects around the vehicle in terms of risk level by analyzing its distance to vehicle and the angle to vehicle values. We use minimum distance to vehicle d_{min} and angle of the nearest obstacle ϕ_{near} values in order to calculate the risk factor of the obstacles around. If an obstacle is very near to the vehicle and stands in the same direction as the vehicle, it is very risky. If the obstacle is far away or in another direction, it is more negligible. Following meta-rules summarize the behavior of the Fuzzy-1 block.

- If the nearest obstacle is close by vehicle and its location is in the same direction with the vehicle, then it is a very risky situation.
- If the nearest obstacle is close by vehicle and its location is in different direction with the vehicle, then it is a little risky situation.
- If the nearest obstacle is far away from vehicle and its location is in the same direction with the vehicle, then it is a little risky situation.
- If the nearest obstacle is far away from vehicle and its location is in different direction with the vehicle, then it is a very little risky situation.

Using this knowledge, fuzzy rules are constructed for Fuzzy-1 block. Membership functions can be seen in Figure 5.3.

The fuzzy control rules are generated in the form of “if-then” structures. The proposed fuzzy control rule base for Fuzzy-1 block is illustrated in Table 5.1.

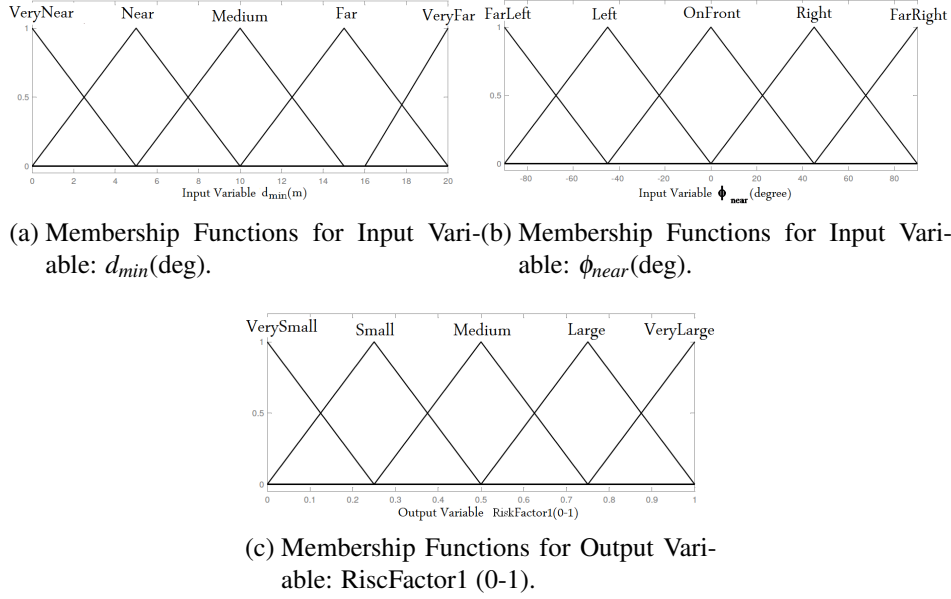


Figure 5.3: Membership Functions for Inputs and Output of Fuzzy-1 Block.

Table 5.1: Rule Base for Fuzzy-1 Block. (For Inputs-VN:Very Near, N:Near, M:Medium, F:Far, VF:Very Far, FL: Far Left, L: Left, OF: On Front, R:Right, FR: Far Right. For Output-VS: Very Small, S: Small, M: Medium, L: Large, VL: Very Large)

		Input1- d_{min}				
		VN	N	M	F	VF
Input2- ϕ_{near}	FL	L	M	S	VS	VS
	L	VL	L	M	S	VS
	OF	VL	VL	L	M	S
	R	VL	L	M	S	VS
	FR	L	M	S	VS	VS

Fuzzy-1 block uses five membership functions for d_{min} and five membership functions for ϕ_{near} which results with 25 rules. The fuzzy surface obtained based on the generated rules is illustrated in Figure 5.4 .

5.3.3 Design of Fuzzy-2 block

Vehicle dynamics is another point that has to be considered for speed planning. In this subsection, vehicle speed planning helps for yaw stability by considering the steering angle δ . Steering angle and rf_1 values are fed to the second fuzzy block in order to calculate the risk factor 2 rf_2 . A human driver should decrease, at least should not increase the vehicle speed if the steer angle is too large. If the environment is risky in terms of obstacles surrounding it, vehicle speed should be lower. Following meta-rules summarize the behavior of the Fuzzy-2 block.

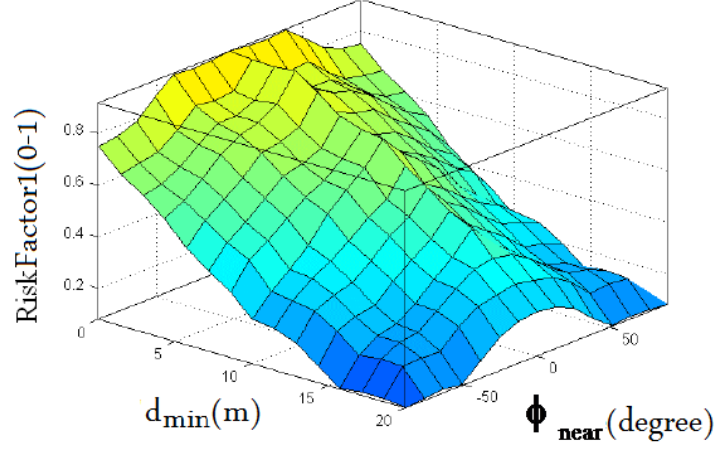


Figure 5.4: Fuzzy Surface for Fuzzy-1 Block.

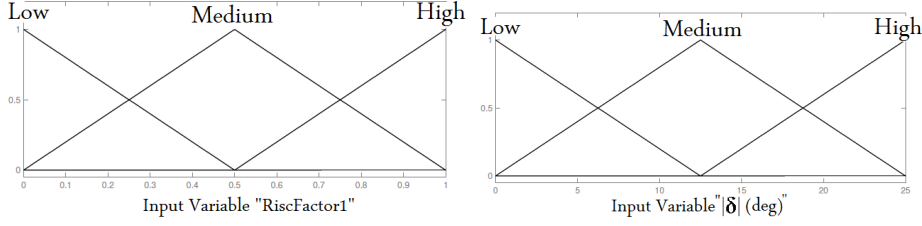
- If the risk level of the environment in terms of the obstacles surrounding the vehicle is highly risky and the reference steering angle value is high, then it is a very risky situation.
- If the risk level of the environment in terms of the obstacles surrounding the vehicle is highly risky and the reference steering angle value is low, then it is a risky situation.
- If the risk level of the environment in terms of the obstacles surrounding the vehicle is low and the reference steering angle value is high, then it is a risky situation.
- If the risk level of the environment in terms of the obstacles surrounding the vehicle is low and the reference steering angle value is low, then it is a very little risky situation.

Using this knowledge, fuzzy rules are constructed for the Fuzzy-2 block. Membership functions can be seen in Figure 5.5.

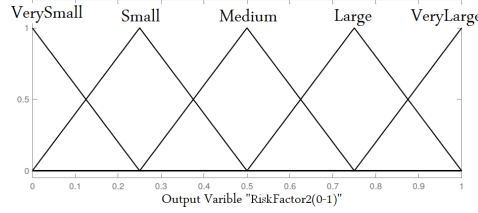
The proposed fuzzy control rule base for Fuzzy-2 block is illustrated in Table 5.2.

Fuzzy-2 block uses three membership functions for rf_1 and three membership functions for δ which results in 9 rules. The resulting fuzzy surface is illustrated in Figure 5.6.

By using 2 cascaded fuzzy blocks, the system is designed with 34 rules in total. If all the inputs were used in the same fuzzy block with same number of membership



(a) Membership Functions for Input Variable: RiskFactor1 (0-1). (b) Membership Functions for Input Variable: Absolute Steering Angle (deg).



(c) Membership Functions for Output Variable: RiskFactor2 (0-1).

Figure 5.5: Membership Functions for Inputs and Output of Fuzzy-2 Block.

Table 5.2: Rule Base for Fuzzy-2 Block. (For Inputs- *L*: Little, *M*: Medium, *H*: High. For Output- *VS*: Very Small, *S*: Small, *M*: Medium, *L*: Large, *VL*: Very Large)

		Input1- $r f_1$		
		L	M	H
Input2 $ \delta $	L	VS	S	L
	M	S	M	VL
	H	L	VL	VL

functions, 75 rules had to be determined in total and it would be harder to decide the rules for three inputs.

5.4 Simulations and Comparisons

Simulations are performed using nonlinear single track vehicle model with yaw dynamics in Matlab/Simulink® environment. Modeling equations and model parameters are given in Chapter 2.

The potential field method is coded using C programming language into the S-function block. A snapshot from simulation environment is illustrated in Figure 5.7.

The new approach for vehicle speed decision is compared with the previous methods in [43] and [62] which were explained in Equations 5.1 and 5.2. These methods are entitled as Method-A and Method-B respectively. V_{max} and V_{input} values are selected

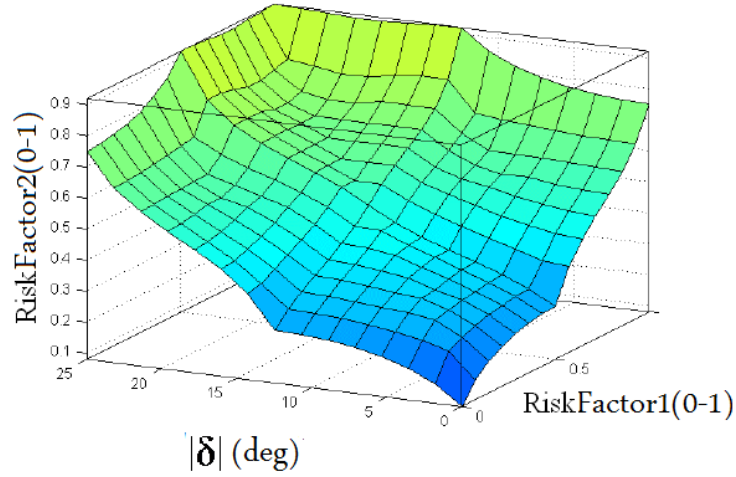


Figure 5.6: Fuzzy Surface for Fuzzy-2 Block.

as 36km/h in Equations 5.1 and 5.2. V_{max} value for the proposed fuzzy speed planning algorithm is also selected as 36km/h. All methods use the same PI low level speed controller for fair comparison. Designing the PI coefficients, which is not in the scope of this study, is performed by trial and error method. K_I is selected as 1000 and K_p is selected as 100 for each simulation. Low level speed controller scheme is illustrated in Figure 5.8.

Sample time of the simulation is 0.005ms. A sample simulation result which illustrates the vehicle trajectory for both Method-A, Method-B and the new fuzzy method are shown as stroboscopic plots in Figure 5.9. Eight circular obstacles with various diameters are scattered in simulation environment. Each method is tested with the same APF obstacle avoidance method and the same low level PI speed controller.

As can be seen in the simulation results, Method-A passes by obstacles very closely. In Method-B, the vehicle loses its yaw stability and spins. The new fuzzy method is safer than both of these two methods in terms of distance to obstacle and yaw rate values.

The following metric for obstacle avoidance and vehicle dynamics safety is defined for a numerical comparison. This collision avoidance metric is based on the inverse of the distance-to-obstacle function and yaw rate. A similar metric is used in [51].

$$f(t) = \begin{cases} \frac{1}{d_{min}} - \frac{1}{d_0} + 0.1|\dot{\psi}| & \text{for } d_{min} < d_0 \\ 0.1|\dot{\psi}| & \text{for } d_{min} \geq d_0 \end{cases} \quad (5.4)$$

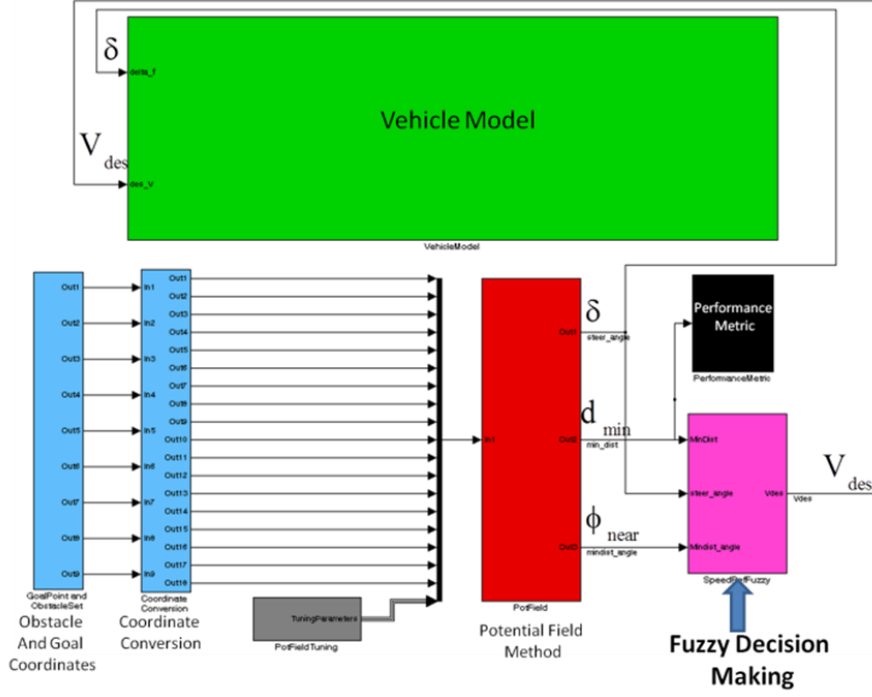


Figure 5.7: Simulation Snapshot from Simulation Environment.

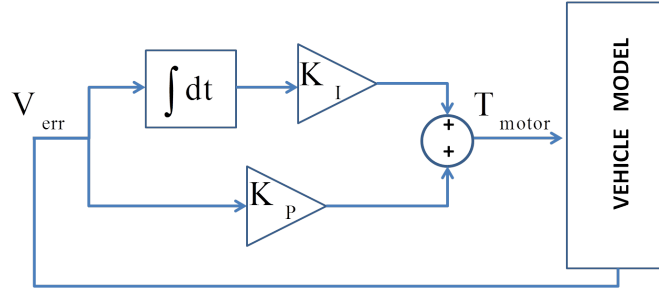


Figure 5.8: PI Controller for Low Level Speed Control.

Where d_{min} is the closest distance between the vehicle and obstacles and the given scalar d_0 , denotes the distance to obstacle that poses no danger for collision during execution.

P'th norm of a function is defined in the following equation.

$$\|f\|_p = \left(\int_{t_0}^t |f(t)|^p dt \right)^{\frac{1}{p}} \quad (5.5)$$

Second norm ($p = 2$) of the collision avoidance metric is calculated and taken as a performance criterion. This performance criterion measures the safety of the trajectory or in other words how much the trajectory is away from obstacles and the yaw stability. Twenty Monte Carlo simulations were performed for both Method-A, Method-B and

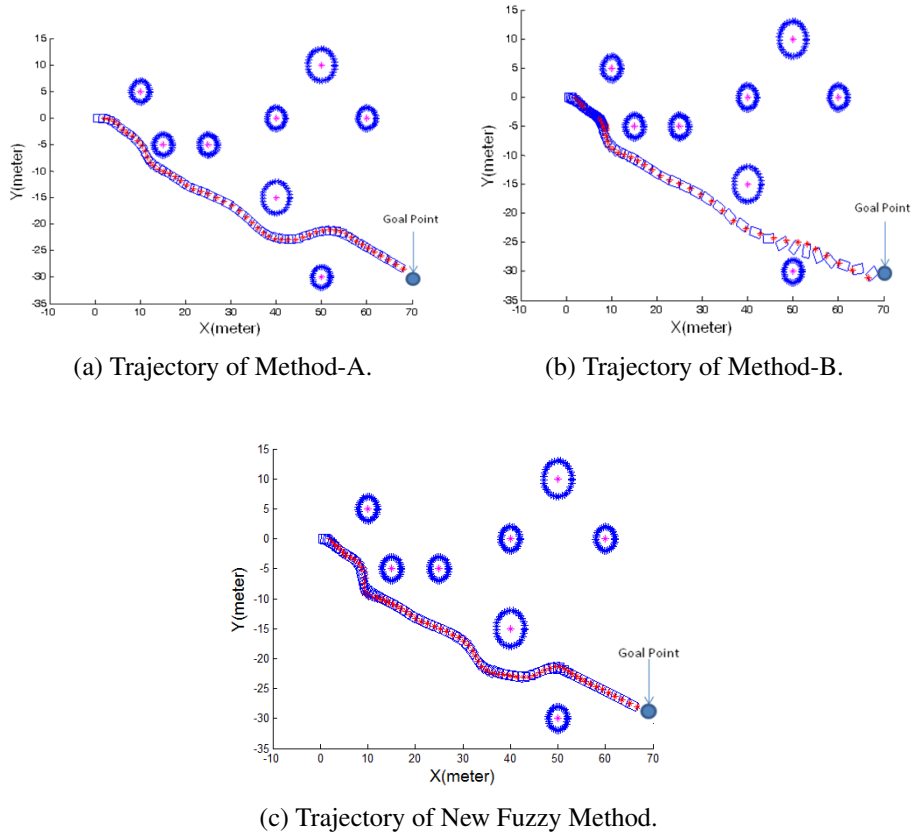


Figure 5.9: Trajectory Comparison of Three Algorithms.

new fuzzy method. Eight circular objects and goal coordinates are given to model randomly in these simulations. The results are illustrated in Figure 5.10.

According to the Figure 5.10, the new fuzzy logic approach for speed decision is generally better than the other two methods. Table 5.3 illustrates the average cost value of the twenty simulations performed.

Table 5.3: Average Cost Comparison.

	Method-A	Method-B	New Fuzzy Method
Average Cost	0.215	0.326	0.183

According to the Table 5.3, the proposed algorithm is %14.9 better than Method-A and %43.9 better than Method-B in terms of the defined cost function.

5.5 Conclusions

A new type velocity planning method using fuzzy logic is developed in this section. Different from the previous studies, this method consists of two cascaded Mamdani-type Fuzzy Inference Systems for desired speed determination. In this way,

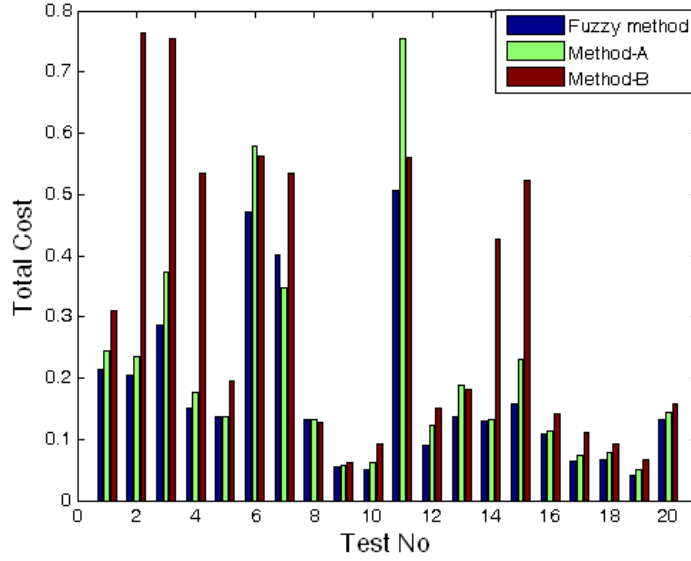


Figure 5.10: Total Cost Values for 20 Monte Carlo Simulations.

three parameters are considered; two of them are for obstacles around and one is for the vehicle stability. The cascaded structure results in a reduction of the number of rules, from 75 to 34. Twenty Monte Carlo simulations are performed for a comparison using dynamic vehicle model. A safety metric is designed for the comparison of the algorithm with previous methods. Simulation results show that this new fuzzy approach results in safer trajectories than the previous APF based speed planning methods. Since the proposed method does not depend on any specific function of the used obstacle avoidance method, which is artificial potential field method in our case, it can be used with all obstacle avoidance or planning methods. It is also possible to easily add extra parameters to the method with the same cascade fuzzy strategy.

6. A NEW FUZZY SPEED CONTROL STRATEGY CONSIDERING THE LATERAL VEHICLE DYNAMICS

This chapter introduces a new speed control strategy for autonomous/semi-autonomous navigation of ground vehicles. Different from the previous studies, steering angle is considered in addition to speed error and integral of the speed error [63]. Because of the highly nonlinear nature of the vehicle model, fuzzy logic strategy is used for controller design. Simulations are carried out to verify and demonstrate the effectiveness of the new method over the classical one which does not consider the steering angle. Both of these two methods have similar performances when steering angle is relatively low. However, in a more aggressive steering scenario, classical approach fails and vehicle loses its yaw stability while this new method still continues to track the speed with a stable yaw dynamics.

6.1 Introduction

Vehicle speed control is one of the most critical research topics of intelligent transportation area. The aim of the control is to achieve the predetermined speed value, using throttle and brake inputs. Vehicle speed control or in other words, longitudinal control is not only used in semi-autonomous applications like cruise control (CC) and adaptive cruise control (ACC) but also in full autonomous navigation.

The complexity and non-linearity of vehicle dynamics makes the development of speed controllers a difficult task. Different control strategies have been proposed in literature. In [64], fixed gain and gain scheduling PID controllers as well as the adaptive control method are proposed. Another PID controller is illustrated in [65]. Beside these, complex nonlinear control strategies [66] [67] [68] [69] have been implemented to solve the problem of longitudinal control. However, all of these methods require an accurate model of the system and exact system parameters to achieve satisfactory results. The nonlinearities and uncertainties in vehicle dynamics and both throttle and brake characteristics makes it difficult to be successful in real applications.

Fuzzy logic is very applicable for vehicle speed control due to its two main features. The first one is it does not require an exact mathematic model of the controlled system. This characteristic is very important when the controlled system is highly nonlinear and difficult to obtain an exact model. The other feature is that it allows us to mimic the behavior of a professional human driver and his experience. Inspired by these features, there are several studies for speed control of the vehicle using fuzzy logic.

In [70], a throttle and brake fuzzy control system was designed to perform the ACC+Stop&Go maneuvers. Two separated fuzzy controllers are designed to control the throttle and brake pedals. In order to avoid simultaneous actions of both the pedals, the values of the membership functions of the two controllers were well defined. In [71], a hierarchical control system is proposed, in which, an upper level controller determines the desired speed/acceleration for the controlled vehicle, while a throttle/brake fuzzy control system is proposed as the lower level controller. An additional switch logic algorithm is designed to avoid switching between throttle and brake actuators. In [72], throttle and brake references are generated from the same fuzzy controller. Different from other studies, the speed error and the integral of the speed error values are the inputs of the controller in order to eliminate the steady state error.

None of the previous studies consider the lateral vehicle dynamics in their vehicle speed control strategy since they are mostly concentrated on longitudinal dynamics of CC and ACC applications. But in a UGV, steering angle is determined autonomously and sharp maneuvers are possible in case of sudden obstacles. This makes the lateral dynamics very important for the speed control of the vehicle. Figure 6.1 illustrates the role of the speed control in a UGV.

In this chapter a new speed control method is developed using fuzzy logic. This fuzzy system consists of two cascade connected Mamdani-type Fuzzy Inference System for vehicle speed control. Different from other studies, this new method considers both the longitudinal and lateral vehicle dynamics. First fuzzy block works like classical speed controller and calculates the desired throttle/brake reference values. The second fuzzy block considers the steering angle command and adjusts the output of the first fuzzy controller using steering angle information.

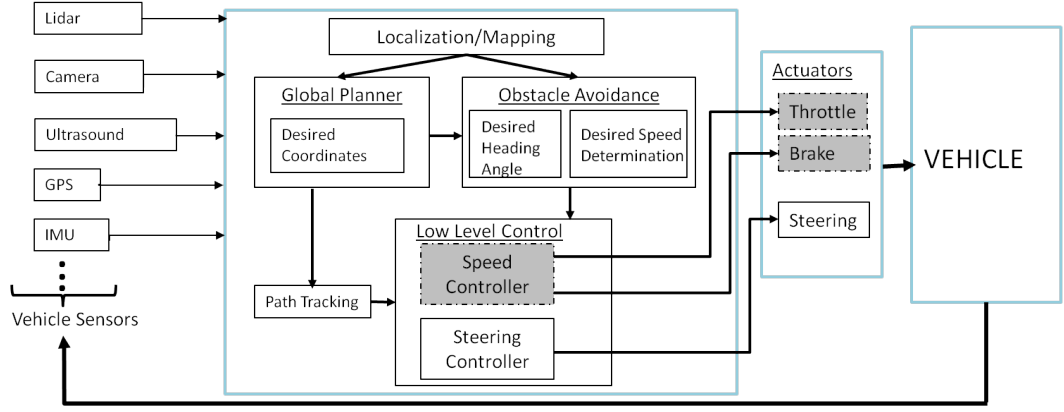


Figure 6.1: Role of Speed Control in a UGV.

6.2 Fuzzy Speed Controller

Motivated by the fact that human performance is reliable in driving ground vehicles, fuzzy logic methods are used in order to mimick the experiences of the human driver without dealing with complicated mathematical models. Different from previous studies, lateral dynamics for yaw stability is considered during the control process.

Three parameters are used for the calculation of the throttle and brake pedal outputs with fuzzy logic. These inputs are used in the control scheme with 2 cascaded Mamdani type fuzzy inference systems. Figure 6.2 illustrates this cascaded fuzzy decision making approach. Two of the inputs, which are speed error e_v and integral of speed error, are used for classical speed control. This structure eliminates the steady state speed error. k_1 and k_2 are the normalizing gain factors for the inputs of the Fuzzy-1 block. The output of the first fuzzy block is given to the second fuzzy block with absolute value of the steering angle command $|\delta|$, which is the third input for stable yaw dynamics. k_3 is the normalizing gain factor for the steering input. All inputs are guaranteed to be between zero and one using saturations.

6.2.1 Design of Fuzzy-1 block

This block is for the classical speed control of the vehicle. A similar version of this block can be found in [72]. In this block, rules are defined using e_v and integral of the e_v . Triangular membership functions are chosen for both inputs and the output of the fuzzy block. Five fuzzy sets for both the two inputs are defined as; NB, NM, Z, PM, and PB and also seven fuzzy sets for the output as; NB, NM, NL, Z, PL, PM and

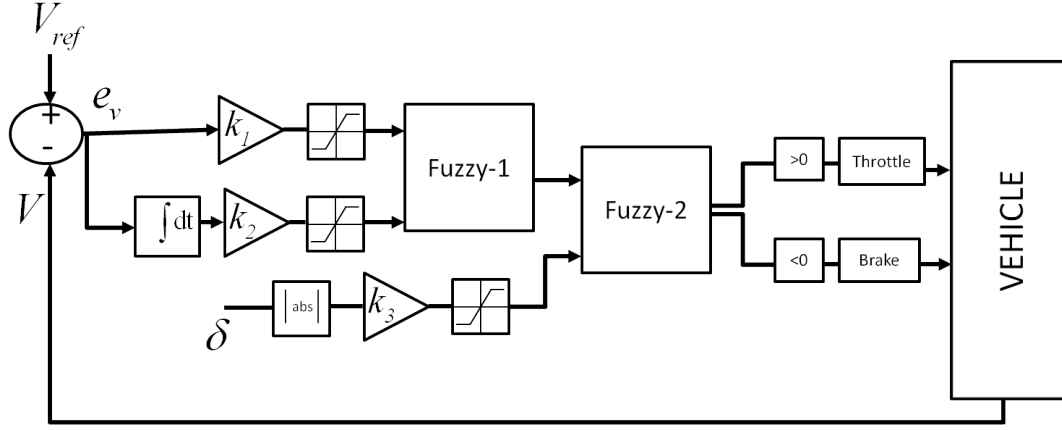


Figure 6.2: Cascade Fuzzy Connection for Speed Control.

PB . (**NB**: Negative Big. **NM**: Negative Medium. **NL**: Negative Little. **Z**: Zero. **PL**: Positive Little. **PM**: Positive Medium **PB**: Positive Big)

Fuzzy-1 block uses the error signal and the integral of the error signal in order to decide the desired acceleration value. Following meta-rules summarize the behavior of the Fuzzy-1 block.

- If the speed error is highly positive and the integral of the error value is highly positive, then the desired acceleration is highly positive.
- If the speed error is highly positive and the integral of the error value is highly negative, then the desired acceleration is zero.
- If the speed error is highly negative and the integral of the error value is highly positive, then the desired acceleration is negative.
- If the speed error is highly negative and the integral of the error value is highly negative, then the desired acceleration is highly negative.

Using this knowledge, fuzzy rules are constructed for Fuzzy-1 block. Membership functions can be seen in Figure 6.3.

The fuzzy control rules are generated in the form of “if-then” structure. The proposed fuzzy control rule base for Fuzzy-1 block is illustrated in Table 6.1.

Fuzzy-1 block uses five membership functions for the “ e_v ” and five membership functions for the “integral of e_v ” which results in 25 rules. The obtained fuzzy surface based on the generated rules is illustrated in Figure 6.4.

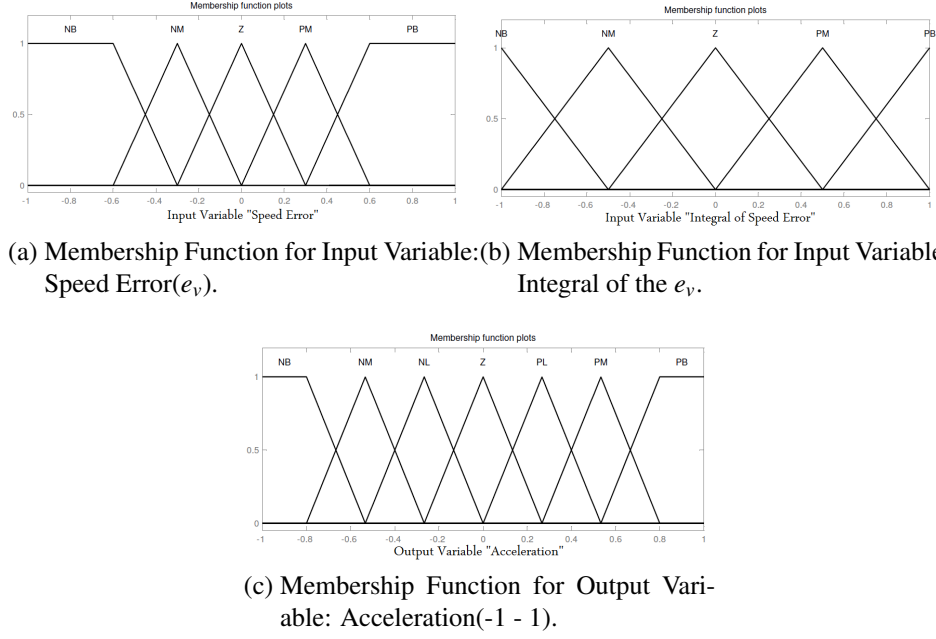


Figure 6.3: Membership Functions for Inputs and Output of Fuzzy-1 Block.

Table 6.1: Rule Base for Fuzzy-1 Block.

		Input1-Speed Error e_v				
		NB	NM	Z	PM	PB
Input2- $\int e_v$	NB	NB	NB	Z	Z	Z
	NM	NM	NM	Z	Z	PL
	Z	NM	NM	Z	PL	PM
	PM	NM	NL	PL	PM	PB
	PB	NM	NL	PM	PB	PB

6.2.2 Design of Fuzzy-2 block

The aim of this block is to provide the lateral stability of the vehicle while it follows a predetermined speed value. Pushing the brake pedal or throttle pedal during the turning period may cause skidding accidents depending on the steering angle, vehicle speed, friction factors and vehicle parameters. An experienced driver cares about pushing the pedals when he turns the steering wheel. Rules of Fuzzy-2 block are generated on this basis.

Fuzz-2 block takes the output of the Fuzzy-1 block which is a classical acceleration signal for speed control and fuses this data with $|\delta|$ value. The Fuzzy-2 block calculates the pedal reference signals between minus one and one. The output values between minus one and zero means a brake command and values between zero and one means a throttle command.

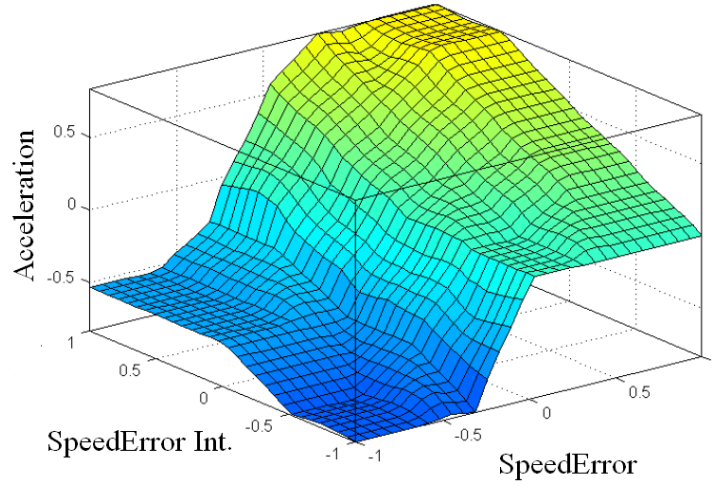


Figure 6.4: Fuzzy Surface for Fuzzy-1 Block.

Triangular membership functions are chosen for both inputs and the output of the fuzzy block. Seven fuzzy sets for “acceleration” input are defined as; NB, NM, NL, Z, PL, PM and PB. Four fuzzy sets for the absolute steering angle input are entitled as Z, L, M and B. Finally for the output “pedal signal”, seven fuzzy sets are defined as NB, NM, NL, Z, PL, PM and PB. (**NB**: Negative Big. **NM**: Negative Medium. **NL**: Negative Little. **Z**: Zero. **PL**: Positive Little. **PM**: Positive Medium **PB**: Positive Big **Z**: Zero. **L**: Little. **M**: Medium **B**: Big)

Following meta-rules summarize the behavior of the Fuzzy-2 block.

- If the desired acceleration is highly positive and the reference steering angle value is high, then the pedal signal is a little positive.
- If the desired acceleration is highly positive and the reference steering angle value is low, then the pedal signal is highly positive.
- If the desired acceleration is highly negative and the reference steering angle value is high, then the pedal signal is a little negative.
- If the desired acceleration is highly negative and the reference steering angle value is low, then the pedal signal is highly negative.

Using this knowledge, fuzzy rules are constructed for Fuzzy-2 block. Membership functions can be seen in Figure 6.5.

The proposed fuzzy control rule base for Fuzzy-2 block is illustrated in Table 6.2.

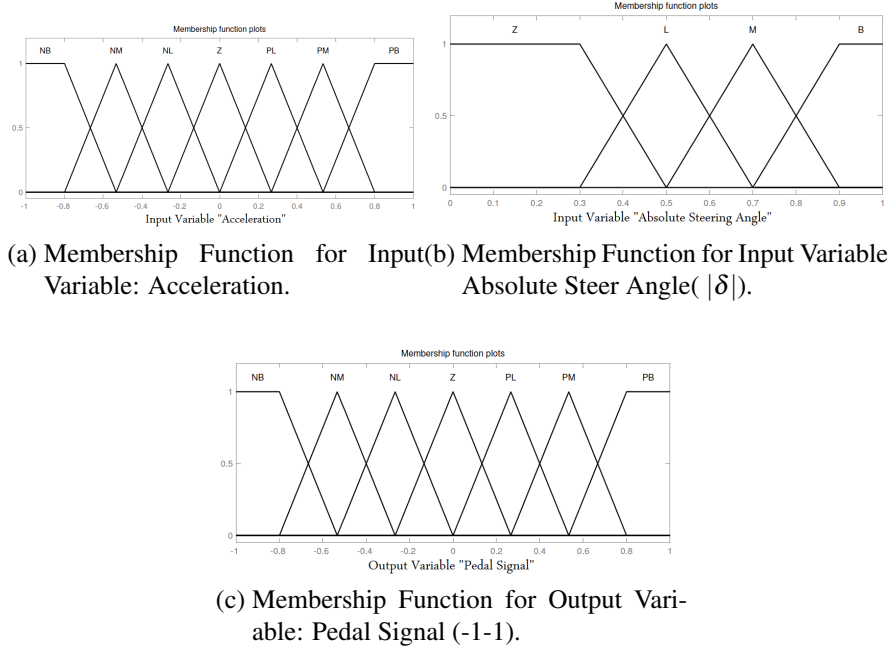


Figure 6.5: Membership Functions for Inputs and Output of Fuzzy-2 Block.

Table 6.2: Rule Base for Fuzzy-2 Block.

		Input1-Acceleration						
		NB	NM	NL	Z	PL	PM	PB
Input2-δ	Z	NB	NM	NL	Z	PL	PM	PB
	L	NM	NM	NL	Z	PL	PM	PM
	M	NL	NL	NL	Z	PL	PL	PL
	B	NL	NL	NL	Z	PL	PL	PL

The Fuzzy-2 block uses four membership functions for “steering angle” and seven membership functions for “acceleration” which results with 28 rules. The fuzzy surface obtained based on the generated rules is illustrated in Figure 6.6.

Five membership functions for the speed error, five membership functions for the integral of speed error and four membership functions for steering angle are defined for the whole fuzzy system. If all the inputs were used in the same fuzzy block, there had to be 100 rule combinations. Using the cascaded strategy, two separated fuzzy blocks reduce the total number from 100 to 53 which makes the design of the controller easier. Besides this, it is easier to design the rules for two inputs instead of three inputs.

6.3 Simulations and Comparison

Simulations are performed using nonlinear single track vehicle model with yaw dynamics in Matlab/Simulink® environment. Modeling equations and model

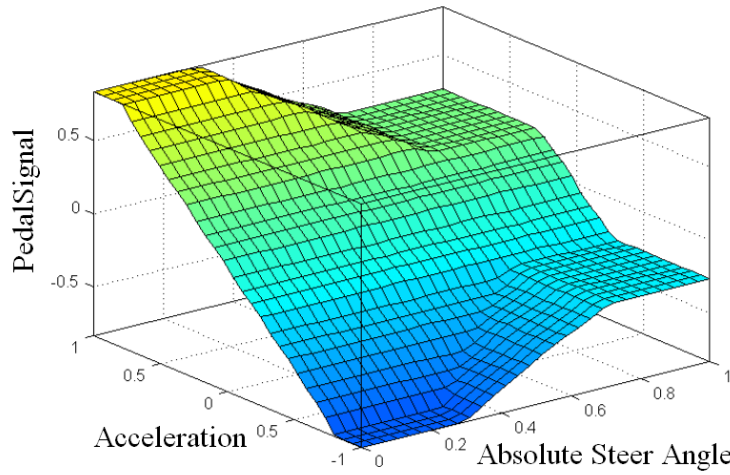


Figure 6.6: Fuzzy Surface for Fuzzy-2 Block.

parameters are given in Chapter 2. Values of the normalizing gain factors for three inputs are selected as illustrated in Table 6.3.

Table 6.3: Normalizing Gain Factors.

k_1	k_2	k_3
0.2	0.167	0.087

A randomly generated speed profile is given to the model for simulations. In order to analyze the effect of the Fuzzy-2 block to the controller, simulations are performed with only Fuzzy-1 controller at first, which is named as **Structure-1**. Then, the same speed profile is used with both Fuzzy-1 and Fuzzy-2 controller which is also named as **Structure-2**.

For a better comparison, two simulation scenarios are performed. For the first simulation, a steering angle profile is given to the vehicle in addition to the desired speed profile. This causes a very little difference in the performances of both structures. The reason of this little difference is the effect of the Fuzzy-2 block which reduces the acceleration rate in a steering situation. The benefit of the structure-2 can be seen clearly in the second simulation. The speed profile, pedal outputs and given steering angle profile for the first simulation are illustrated in Figure 6.7.

In the second simulation, the steering angle profile is more aggressive than the first scenario. Even though the similar steering angle value is used in the first simulation, this steering angle is given with higher speed values of the vehicle in the second one. According to the results, Structure-1 fails when steering input is given, the vehicle loses

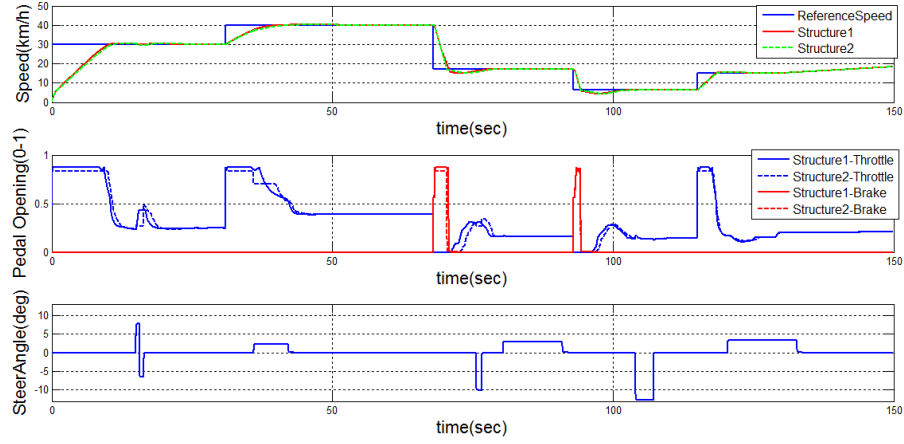


Figure 6.7: Vehicle Speed Tracking Comparison of Structure-1 and Structure-2 (Simulation 1).

its yaw stability and starts to skid. After unstable yaw dynamics and skidding, it is not possible to track the desired speed because it tries to give high throttle values in order to achieve the given speed. Structure-2 reduces the acceleration (pedal opening) when steering input is given and it tracks the desired speed with a stable yaw dynamics. The speed profile, pedal outputs and given steering angle profile for the second simulation are illustrated in Figure 6.8.

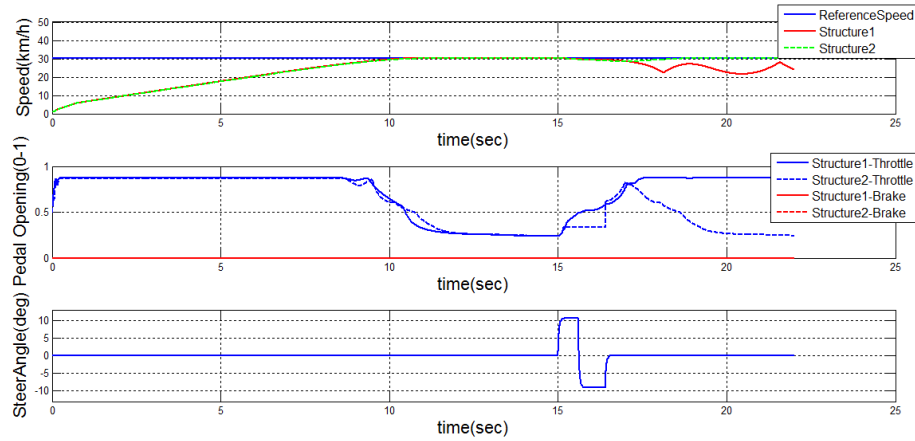


Figure 6.8: Vehicle Speed Tracking Comparison of Structure-1 and Structure-2 (Simulation 2).

The vehicle trajectories are illustrated in Figure 6.9 and Figure 6.10 for both of the control structures respectively, using the stroboscopic plot technique. The vehicle is represented as blue rectangles using its original dimension. The red points are the mid-points of the front axle for each time sample.

According to the Figure 6.9, vehicle turns left in the beginning of the steering command. After that, in the second phase of the steering command, which can be

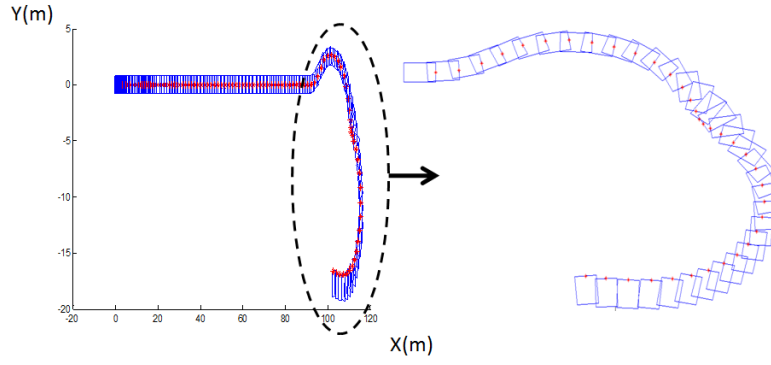


Figure 6.9: Unstable Vehicle Trajectory of Structure-1 (Simulation 2).

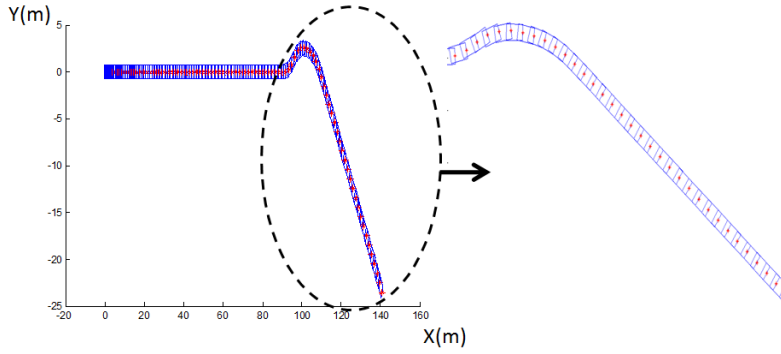


Figure 6.10: Unstable Vehicle Trajectory of Structure-1 (Simulation 2).

seen in Figure 6.8, vehicle tries to turn right but it starts skidding and loses its stability. Conversely, Figure 6.10 shows a stable vehicle trajectory in which vehicle turns left and then turns right properly according to the given steering angle command profile.

6.4 Experimental Results

Proposed method is experimentally tested in this part. Desired speed profile is given by the Control-Desk interface and steering input is given by the human driver. Figure 6.11 illustrates the experimental test results of the low level speed controller.

Desired speed, real speed, steering input, throttle output and brake output values are shown in Figure 6.11 According to the simulation results, it can be seen that vehicle can track the desired speed profile using both the throttle and brake inputs. It is also seen that, tracking capability increases when steering input is lower. Since the vehicle speed data is noisy as it is seen, output of the fuzzy blocks are noisy too. In order not be harmful to the vehicle motor driver, throttle output is given by using a low pass filter to the motor driver.

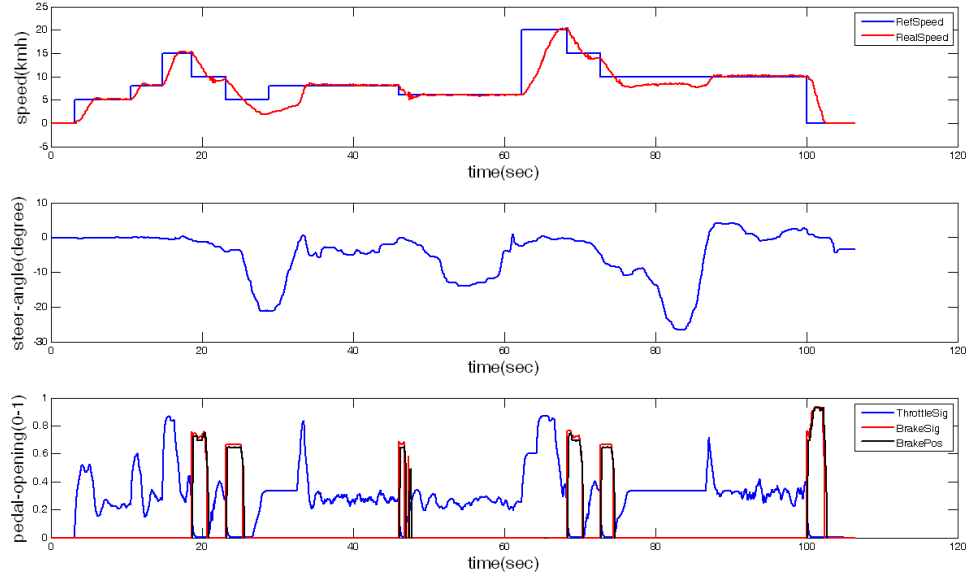


Figure 6.11: Experimental Results of Proposed Low Level Speed Controller.

6.5 Conclusions

A new type of speed control method using fuzzy logic is developed in this section. Different from the previous studies, the new method considers steering angle in addition to speed error and integral of the speed error. Because, steering angle value may cause skidding accidents when it is applied with aggressive throttle or brake pedal signals simultaneously.

Designed method consists of two cascaded Mamdani-type fuzzy inference systems for speed control. The cascaded structure results in a reduction of the number of rules, from 100 to 53. Dynamic vehicle model is used for the simulations. Simulations are performed in order to show the effectiveness of this new method over the classical approach.

According to the simulation results, both of these two methods have similar performances when steering angle and the vehicle speed values are relatively low. However, in a more aggressive steering scenario, classical approach fails and vehicle loses its yaw stability while the new method still continues to track the speed with a stable yaw dynamics.

Additional simulation and experimental results of this new speed control method are illustrated in Chapter 7. Chapter 7 illustrates the results of newly designed obstacle

avoidance (Follow the Gap), speed planning and low level speed control methods together.

7. SIMULATIONS AND EXPERIMENTAL RESULTS OF DESIGNED STRATEGIES TOGETHER

Three new methods have been designed for heading angle calculation, speed planning and low level speed control in order to safely avoid obstacles in Section 4, Section 5 and Section 6 respectively. Each method have been tested and comparisons have been performed separately till this section. All the new designed methods are combined and used together in simulations and experimental applications in this section.

7.1 Simulations

Dynamic vehicle model of designed autonomous vehicle is used for simulations. Modeling equations and used parameters for the model are illustrated in Section 2. Parameters for FGM, speed planning and low level speed control algorithms are used as the same values which are illustrated in Section 4, Section 5 and Section 6. Two random scenarios are simulated in this section. The first one is with static obstacles and the second one is with dynamic obstacles.

7.1.1 Simulation scenario 1 - static obstacles

Figure 7.1 illustrates the resulted trajectory of the first obstacle scenario. In this simulation, all obstacles are static and scattered randomly. Starting coordinate is [0,0] and goal coordinate is [70,-30].

Figure 7.1 shows that the desired heading angle which is calculated by the new designed obstacle avoidance algorithm FGM, which tries to maximize the distance to obstacle value as much as possible and guides the vehicle to the goal point. Reference speed, real speed, steering input and pedal input values for the first simulation are illustrated in Figure 7.2.

It can be seen from Figure 7.2 that, vehicle reduces its speed in a dangerous situation according to the speed planner which is designed in Section 5. Vehicle also reduces the

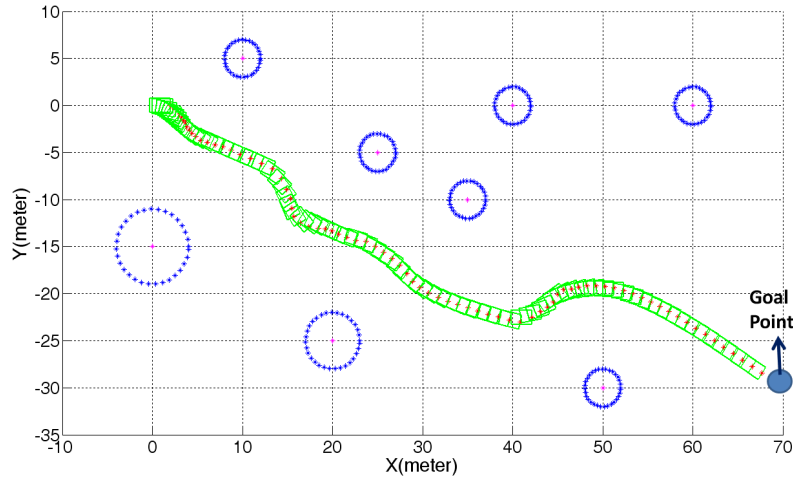


Figure 7.1: Vehicle Trajectory for Simulation Scenario-1.

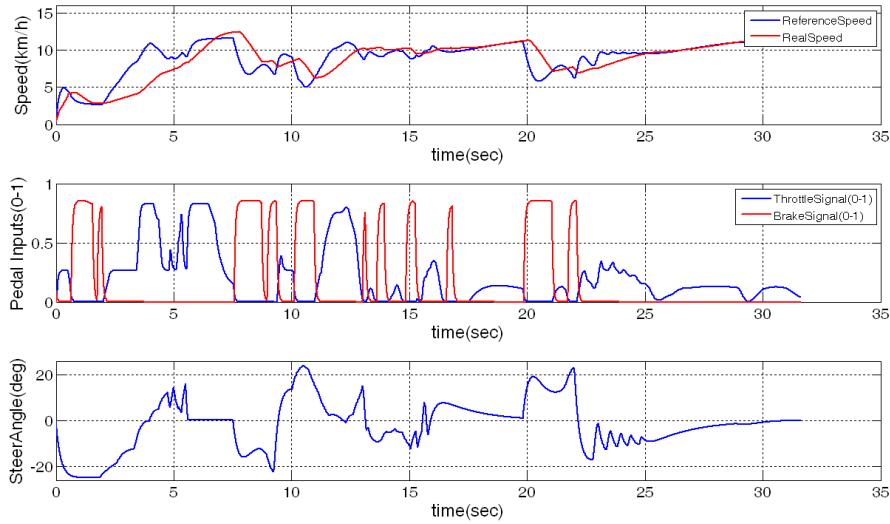


Figure 7.2: Reference Speed and Real Speed with Steering and Pedal Inputs for Simulation Scenario-1.

pedal inputs in high steering angle regions as it is designed in low level speed controller in Section 6.

7.1.2 Simulation Scenario 2 - dynamic obstacles

Figure 7.3 illustrates the resulted trajectory of the second obstacle scenario. In this simulation, five obstacles are static and scattered randomly while three obstacles are dynamic and moving linearly. Direction of the obstacles are shown with dashed arrows. Starting coordinate is $[0,0]$ and goal coordinate is $[70,40]$.

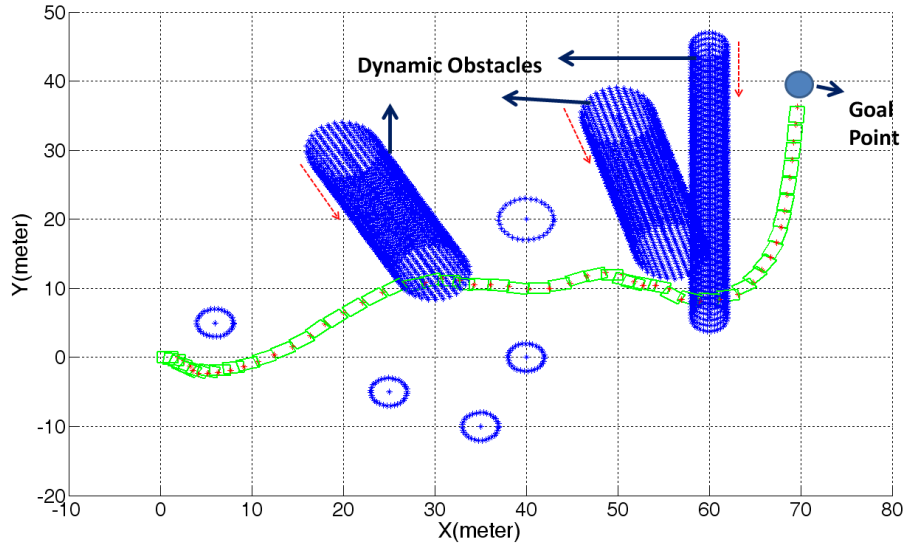


Figure 7.3: Vehicle Trajectory in 2D, for Simulation Scenario-2.

According to the Figure 7.3, it is hard to see if there is a collision with a 2D figure . For this reason time axis is added to same figure and vehicle trajectory is shown in Figure 7.4 in 3D. In this case, definition of the collision changes into the intersection of three axis. This means that vehicle and any obstacle are in same place at the same time. Figure 7.4 illustrates that there is no collision during the trajectory. Each cycle of the FGM is performed in 10 msec . Purely reactive nature of the FGM provides the vehicle to avoid dynamic obstacles easily.

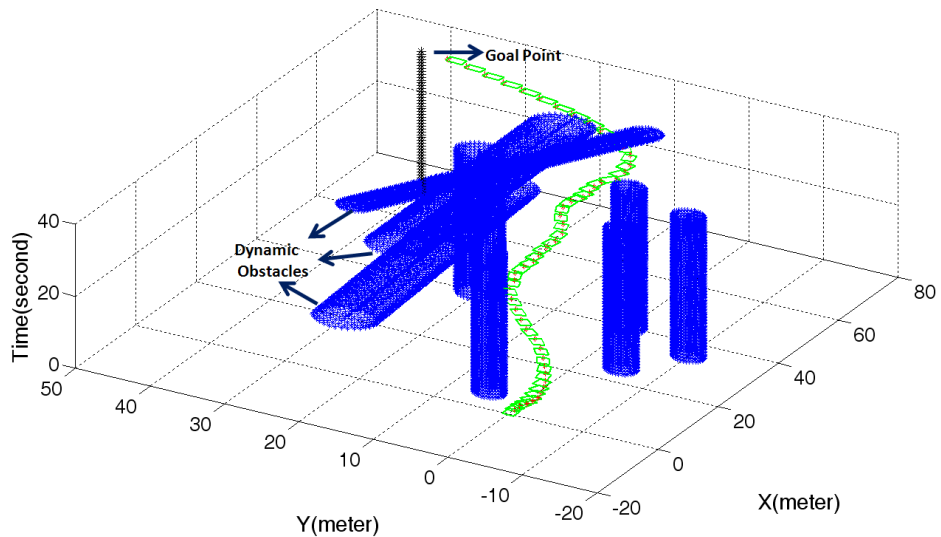


Figure 7.4: Vehicle Trajectory in 3D, with Additional Time Axis for Simulation Scenario-2.

Figure 7.4 shows that there is no collision since any of the obstacles are not in same coordinates with the vehicle at the same time. Results of the speed planning and low level speed controller algorithms are shown in Figure 7.5 for the dynamic obstacle scenario.

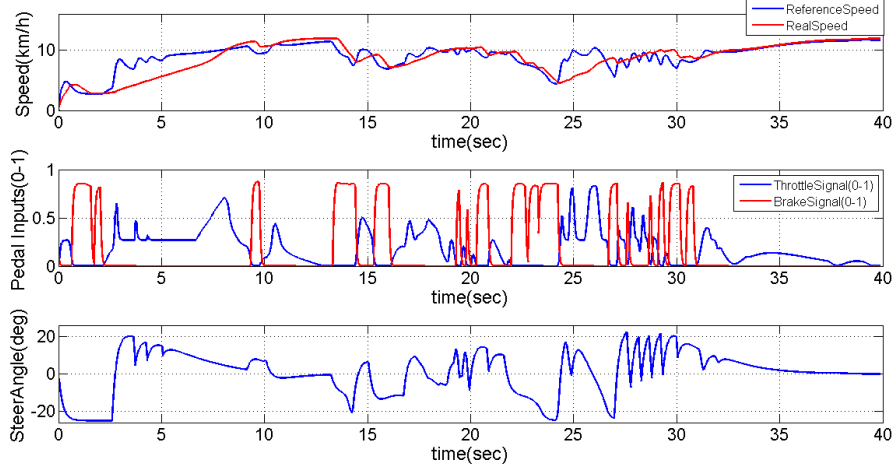


Figure 7.5: Reference Speed and Real Speed with Steering and Pedal Inputs for Simulation Scenario-2.

After the simulations, all designed methods are tested experimentally for both static and dynamic obstacle scenarios in next section.

7.2 Experimental Results

Experimental tests of the algorithms are performed using the designed UGV 'Otonobil', which is explained in detail in Section 3. Test place is the football stadium of Istanbul Technical University. Static obstacles are scattered randomly to the surface. Goal point is near the penalty kick point. All parameters of the algorithms are used as the same values with the simulations. Figure 7.6 illustrates the test field with static obstacles and goal region. Goal region is the circle around a goal point with 2 meters diameter. When the vehicle enters to this region, it assumes that it has achieved and presses brake pedal automatically.

Experimental tests are performed for two main scenarios. First one is with static obstacles and the second one is with dynamic obstacles as it was done in simulations. No prior information is given to the vehicle about obstacle configuration. The only information is the goal coordinate to be reached.

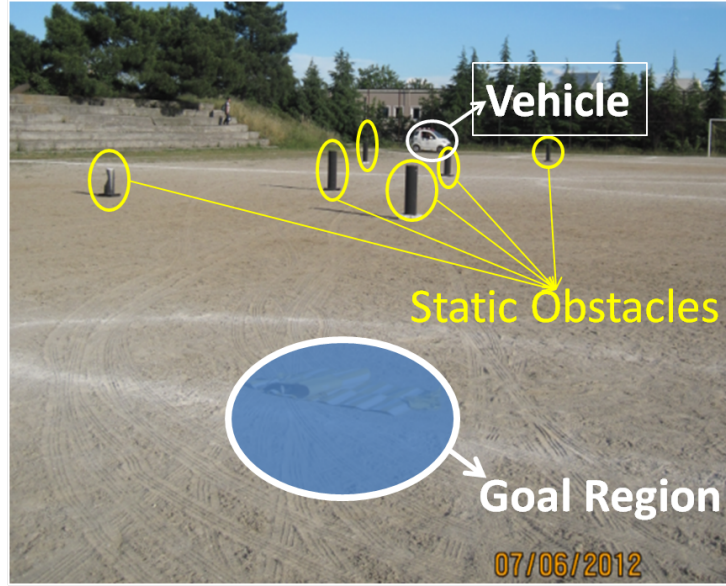


Figure 7.6: Test Field for Experiments.

7.2.1 Experimental test scenario 1 - static obstacles

Figure 7.7 illustrates the resulted trajectory of the first obstacle scenario. In this test, all obstacles are static and scattered randomly. Starting coordinate is $[-1, -2]$ and center of the goal coordinate is $[50.9, -29.4]$. Black dots show the center of gravity coordinate of the vehicle and the blue dots are obstacle borders which are measured by the LIDAR sensor. Red point illustrates the goal point. As it can be seen in Figure 7.7, static perceived obstacles and even the goal point are a cluster of scattered points. The reason for the scatter is that the LIDAR and GPS receivers have some measurement error and noise. These 2 error sources directly affect the coordinate transformation calculations and causes a little scatter in experimental figures.

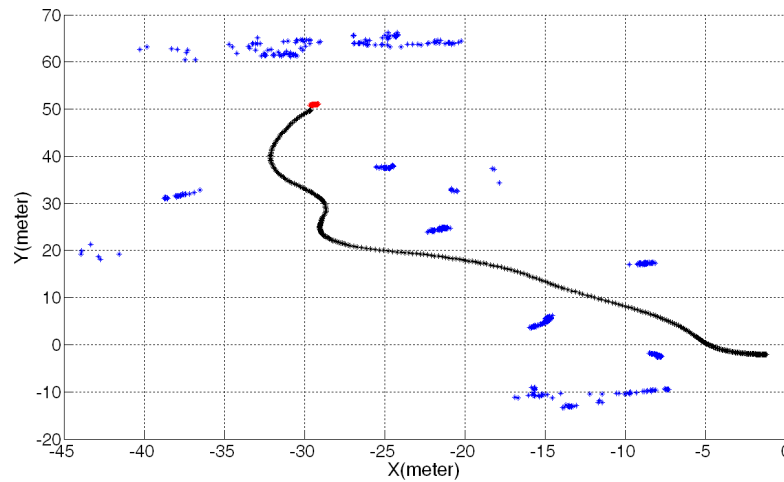


Figure 7.7: Vehicle Trajectory for Experimental Test Scenario-1.

Figure 7.7 shows that the desired heading angle which is calculated by the new designed obstacle avoidance algorithm FGM, tries to maximize the distance to obstacle value as much as possible and guides the vehicle to the goal point. Reference speed, real speed, steering input and pedal inputs for the first experimental test are illustrated in Figure 7.2.

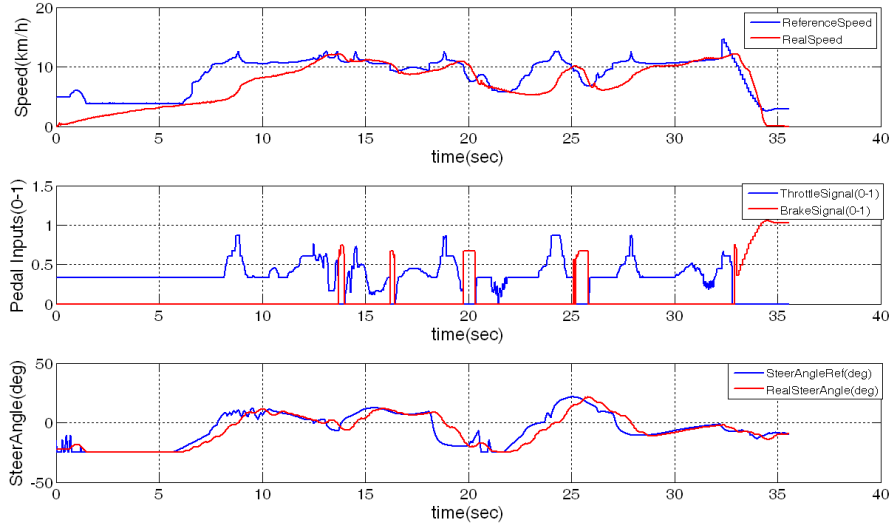


Figure 7.8: Reference Speed and Real Speed with Steering and Pedal Inputs for Experimental Test Scenario-1.

After the static obstacle scenario, final experimental tests are performed in dynamic obstacle scenario.

7.2.2 Experimental test scenario 2 - dynamic obstacles

Figure 7.9 illustrates the resulted trajectory of the second experimental obstacle scenario. Besides static obstacles on the ground, a dynamic obstacle who is me, runs towards the vehicle and stops in front of it several times. Starting and goal coordinates are same as the previous test.

According to the Figure 7.9, it seems that there are intersections between obstacles and the vehicle. This is the result of 2D illustration as it was mentioned in simulations section. For this reason, time axis is again added to Figure 7.9 and vehicle trajectory is shown in Figure 7.10 in 3D. Figure 7.10 illustrates that there is no collision during the trajectory. None of the obstacles are in the same place at the same time. Vehicle avoids both the static and dynamic obstacles successfully using the new FGM algorithm.

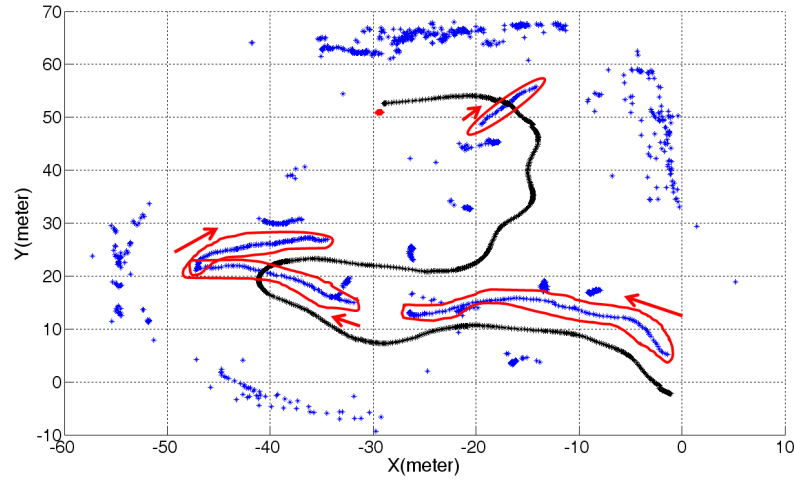


Figure 7.9: Vehicle Trajectory for Experimental Test Scenario-2.

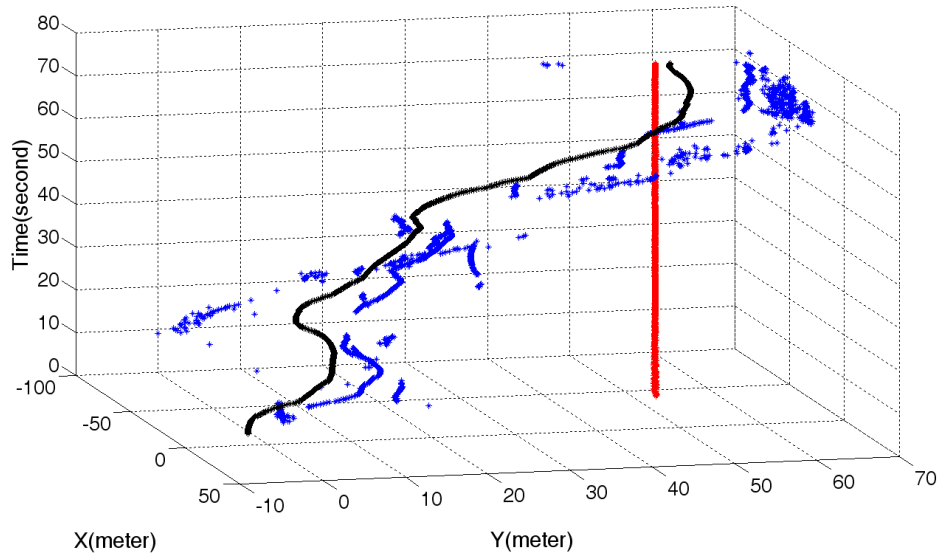


Figure 7.10: Vehicle Trajectory in 3D, with Additional Time Axis for Experimental Test Scenario-2.

Finally, results of the speed planning and low level speed controller algorithms are shown in Figure 7.11 for the dynamic obstacle scenario of experimental tests.

It can be seen from Figure 7.11 that, vehicle reduces its speed in a dangerous situation according to the speed planner which is designed in Section 5.

Vehicle also reduces the pedal inputs in high steering angle regions as it is designed for low level speed controller in Section 6.

Steering angle reference value is calculated based on the Follow the Gap Method which is illustrated in in Section 4.

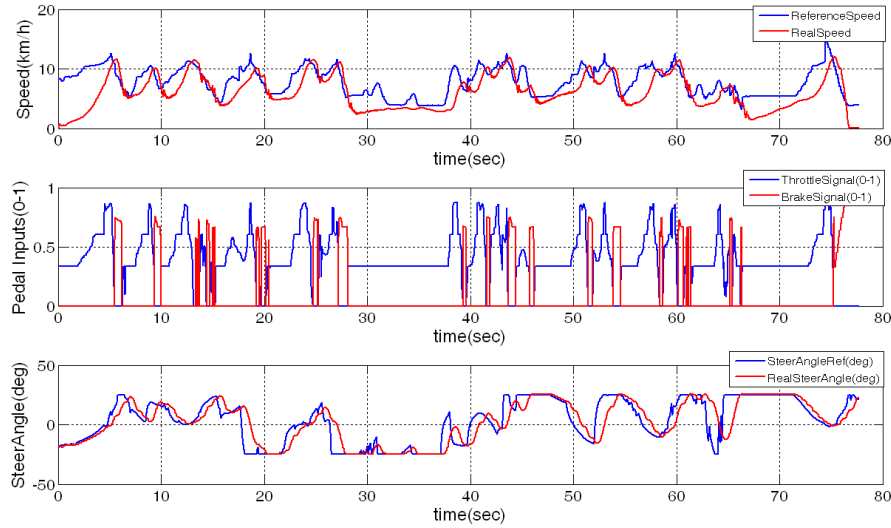


Figure 7.11: Reference Speed and Real Speed with Steering and Pedal Inputs for Experimental Test Scenario-2.

7.3 Conclusions

In this section, three new methods are simulated and tested together. Follow the Gap Method calculates the desired heading angle value and sends it to the automatic steering system for obstacle avoidance, considering the nonholonomic and field of view constraints. Similarly, vehicle speed is planned using the obstacle information around and the steering command for safe navigation. Finally, low level speed controller calculates the throttle and brake pedal signals in order to track the desired speed value, considering the lateral vehicle dynamics.

Simulations and experimental tests are performed for both static and dynamic obstacle scenarios. Single track dynamic vehicle model is used for the simulations and successful results are obtained for both of the dynamic and static obstacle scenarios. Vehicle trajectory and important graphs are shown in this chapter.

For experimental tests, no prior information is given to the vehicle about the environment as it is same for the simulations. Experimental test results show that, the unmanned ground vehicle avoids the static and dynamic obstacles successfully, adjusts its speed according to the obstacle distribution around and tracks the desired speed value considering the lateral vehicle dynamics. All of these operations are performed in real-time, using the computer system of Otonobil. The related graphs and vehicle trajectories are illustrated using the real time experimental data.

8. CONCLUSIONS

This thesis brings new solutions for three sub-problems of an autonomous ground vehicle. These sub-problems are calculation of the heading angle reference for obstacle avoidance, desired speed determination and low level speed control.

First of all the Follow the Gap Method calculates the desired heading angle value and sends it to the automatic steering system for obstacle avoidance considering the nonholonomic and field of view constraints. Similarly, vehicle speed is planned using the obstacle information around and the steering command for safe navigation. Finally, low level speed controller calculates the throttle and brake pedal signals in order to track the desired speed value, considering the lateral vehicle dynamics.

Obstacle avoidance ability is one of the most important subsystems of autonomous robots and this is not just calculation of the heading angle reference value. Nature of the obstacle algorithm should be reactive because, coordinates of any obstacle may change at any time and can not be known previously. This prevents the use of the classical optimization techniques in obstacle avoidance problems. The algorithm must compute just the next action in every instant, based on the current context. There are too many studies about this subject in literature.

The most important property of the designed FGM algorithm is maximizing the distance to obstacle value as much as possible. Comparison of the algorithm is done with the Artificial Potential Fields (APF) method and the A* shortest path algorithm. According to the fair simulation results, the FGM results in safer trajectories than the others. This safety is measured numerically using the designed safety metric. The FGM is free from the local minimum problem which can be seen in the APF and APF based algorithms such as the virtual force field (VFF) method. The FGM has only one tuning parameter α , which adjusts the ratio between obstacle avoidance and arriving at the goal point. The reactive nature of the algorithm provides successful results even in dynamic obstacle scenarios.

Determination of the desired speed value is very important too for obstacle avoidance, like the desired heading angle calculation. Most of the obstacle avoidance algorithms in literature calculate only the reference heading vector for obstacle avoidance. This can be enough in low speed values but in higher values, the dynamic properties of the robot become important. For this reason, speed decision is studied as a separated problem and a new speed planning strategy is developed. Two cascade connected fuzzy inference systems (FIS) are used for speed planning. First block calculates a risk factor by using the distance to obstacle and angle to obstacle values while the second FIS is used for stabilizing the yaw dynamics. A safety metric is designed for Monte Carlo simulations which shows that this new method is safer than the classical methods.

Low level speed controller also affects the avoidance performance. The aim of the low level speed controller is to calculate the throttle and brake pedal signals in order to track the desired speed. A new low level speed controller using fuzzy logic is designed in this thesis. Similar cascaded FIS structure is used here as in speed planning algorithm. Three input values for FIS are; speed error, integral of the speed error and the steering angle values. According to the simulation results, the new method results in safer yaw dynamics especially for the aggressive maneuver scenarios.

All designed algorithms are simulated and compared with the existing methods. Kinematic and dynamic vehicle modeling issues and the used vehicle parameters are explained in a separated section. Each algorithm is coded using C programming language into the S-functions using Matlab/Simulink environment.

Beside simulations, all designed algorithms are tested using experimental autonomous ground vehicle platform. The experimental platform is a full autonomous ground vehicle which is converted from a conventional electric vehicle during this thesis work. Conversion procedure is also explained in detail within this thesis.

Both the simulation and experimental results show that these new methods have significant advantages and results are very promising for future studies.

REFERENCES

- [1] **Lumand, H. and Reagan, J.A.** (1995). Interactive highway safety design model: Accident predictive module, *Public Roads Magazine*, **58**(3), pp.14–17.
- [2] **Thurner, T.** (1998). X-By-Wire: Safety Related Fault Tolerant Systems in Vehicles, Technical Report, X-by-Wire Consortium.
- [3] **Anderson, S.J., Peters, S.C., Iagnemma, K.D. and Pilutti, T.E.** (2009). A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios, *Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics*, SMC'09, IEEE Press, Piscataway, NJ, USA, pp.2032–2037.
- [4] **MIT, T.** (2007). DARPA Urban Challenge Team MIT, Technical Report, Team MIT.
- [5] **Brandt, T.** (2007). A Predictive Potential Field Concept for Shared Vehicle Guidance, *Ph.D. thesis*, Paderborn University.
- [6] **Damiani, S.** (2008). State-of-the-art of current Collision Avoidance Systems, Technical Report, Reposit Consorsium.
- [7] **Ehmanns, D. and Spannheimer, H.** (2004). Deliverable D2D Roadmap Development, Technical Report, ADASE-2 Consotium.
- [8] **Jansson, J. and Jochen Pohl, A.E.** (2007). Systems for Collision Avoidance in Road Vehicles, teknikrapor, IVSS Project Report.
- [9] **Johansson, J. and Gustafsson, F.** (2002). Decision Making for Collision Avoidance Systems, *SAE 2002 World Congress and Exhibition*.
- [10] **Giesbrecht, J.** (2004). Global Path Planning for Unmanned Ground Vehicles, Technical Report, Defence RandD Canada Suffield.
- [11] **Urmson, C.** (2007). Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge, Technical Report, Carnegie Mellon University.
- [12] **Url-1**, <<http://g-nius.co.il>>, date retrieved: 27.09.2012.
- [13] **Url-2**, <<http://www.irobot.com>>, date retrieved: 27.09.2012.
- [14] **Reinholtz, C.** (2007). DARPA Urban Challenge Technical Paper-Victor Tango, Technical Report, Virginia Tech University.

- [15] **Kammel, S. and etal** (2007). DARPA Urban Challenge Team AnnieWAY Technical System Description, Technical Report, University of Karlsruhe, TU Munich, Fraunhofer Gesellschaft, Universität der Bundeswehr Munich.
- [16] **Yigit, T.** (2004). Araç Dinamiği Modelleri Geliştirilmesi ve Savrulma Devrilme Engelleyici Kontrolde Kullanımları, *Ph.D. thesis*, Istanbul Technical University.
- [17] **Güvenç, L.,** (2005), Vehicle Control Systems-Lecture Notes, *Istanbul Technical University*.
- [18] **Sezer, V., Uygan, C.M.I., Hartavi, E.A., Güvenç, L., Acarman, T., Kılıç, V. and Yıldırım, M.** (2008). Maximizing Overall Efficiency Strategy (MOES) for Power Split Control of a Parallel Hybrid Electric Vehicle, *Simulation of Commercial Vehicles-SAE International 01-2682*.
- [19] **Rajamani, R.** (2011). *Vehicle Dynamics and Control*, Mechanical Engineering Series, Springer.
- [20] **Url-3,** <<http://www.bestarmotor.com>>, date retrieved: 27.09.2012.
- [21] **Genta, G.** (1997). *Motor Vehicle Dynamics: Modeling and Simulation*, Series on Advances in Mathematics for Applied Sciences, World Scientific.
- [22] **Sezer, V., Dikilitas, C., Ercan, Z., Heceoglu, H., Oner, A., Apak, A., Gokasan, M. and Mugan, A.** (2011). Conversion of a conventional electric automobile into an unmanned ground vehicle (UGV), *IEEE International Conference on Mechatronics (ICM-2011)*, pp.564 –569.
- [23] **Vlasic, L., Parent, M. and Harashima, F.** (2001). *Intelligent Vehicle Technologies: Theory and Applications*, Society of Automotive Engineers.
- [24] **Url-4,** <<http://www.udea.com.tr>>, date retrieved: 27.09.2012.
- [25] **Sezer, V. and Gökaşan, M.** (2012). A Novel Obstacle Avoidance Algorithm: ‘Follow the Gap Method’, *Elsevier-Robotics and Autonomous Systems*, **60**(9), pp.1123–1134.
- [26] **Fujimura, K.** (1991). *Motion planning in dynamic environments*, Computer science workbench, Springer-Verlag.
- [27] **Chakravarthy, A. and Ghose, D.** (1998). Obstacle avoidance in a dynamic environment: a collision cone approach, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **28**(5), pp.562 –574.
- [28] **Kavraki, L., Svestka, P., Latombe, J.C. and Overmars, M.** (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*, **12**(4), pp.566 –580.
- [29] **Lavalle, S.M.** (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning, Technical Report.

- [30] **Chakravorty, S. and Kumar, S.** (2011). Generalized Sampling-Based Motion Planners, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **41**(3), pp.855–866.
- [31] **Lozano-Pérez, T. and Wesley, M.A.** (1979). An algorithm for planning collision-free paths among polyhedral obstacles, *Communications of the ACM*, **22**(10), pp.560–570.
- [32] **Aurenhammer, F.** (1991). Voronoi diagrams, *ACM Computing Surveys*, **23**(3), pp.94.
- [33] **Siegwart, R. and Nourbakhsh, I.R.** (2004). *Introduction to Autonomous Mobile Robots*, MIT Press.
- [34] **Reeds, J.A. and Shepp, L.A.** (1990). Optimal paths for a car that goes both forwards and backwards, *Pacific Journal of Mathematics*, **145**(2), pp.367–393.
- [35] **Liang, T., Liu, J., Hung, G. and Chang, Y.** (2005). Practical and flexible path planning for car-like mobile robot using maximal-curvature cubic spiral, *Robotics and Autonomous Systems*, **52**(4), pp.312–335.
- [36] **Lumelsky, V.J. and Stepanov, A.A.** (1987). Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape, *Algorithmica*, **2**(1-4), pp.403–430.
- [37] **Khatib, O.** (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *The International Journal of Robotics Research*, **5**(1), pp.90–98.
- [38] **Koren, Y. and Borenstein, J.** (1991). Potential field methods and their inherent limitations for mobile robot navigation, *Proceedings 1991 IEEE International Conference on Robotics and Automation*, **2**(April), pp.1398–1404.
- [39] **Kim, J.O. and Khosla, P.K.** (1991). Real-time obstacle avoidance using harmonic potential functions, *Proceedings 1991 IEEE International Conference on Robotics and Automation*, **8**(3), pp.338–349.
- [40] **Castaneda, M.A.P., Savage, J., Hernandez, A. and Cosío, F.A.,** (2008), *Local Autonomous Robot Navigation Using Potential Fields*, inTech., isbn:978-953-7619-01-5.
- [41] **Shimoda, S., Kuroda, Y. and Iagnemma, K.** (2005). Potential Field Navigation of High Speed Unmanned Ground Vehicles on Uneven Terrain, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp.2828–2833.
- [42] **Ang, M.H. and Krishnan, H.** (2000). *Virtual obstacle concept for local-minimum-recovery in potential-field based navigation*, volume 2, pp.983–988.
- [43] **Borenstein, J. and Koren, Y.** (1989). Real-time obstacle avoidance for fast mobile robots, *IEEE Transactions on Systems, Man and Cybernetics*, **19**(5), pp.1179–1187.

- [44] **Borenstein, J. and Koren, Y.** (1991). The vector field histogram-fast obstacle avoidance for mobile robots, *IEEE Transactions on Robotics and Automation*, **7**(3), pp.278–288.
- [45] **Xu, F., Brussel, H.V., Nuttin, M. and Moreas, R.** (2003). Concepts for dynamic obstacle avoidance and their extended application in underground navigation, *Robotics and Autonomous Systems*, **42**, pp.1–15.
- [46] **Bloch, A.M.** (2003). Nonholonomic Mechanics and Control, *IEEE Transactions on Automatic Control*, **49**(5), pp.38–39.
- [47] **Kolmanovsky, I. and McClamroch, N.H.** (1995). Developments in Non-holonomic Control Problems, *IEEE Control Systems Magazine*, **15**(6), pp.20–36.
- [48] **Nowak, W., Zakharov, A. and Blumenthal, S.** (2010). Benchmarks for Mobile Manipulation and Robust Obstacle Avoidance and Navigation, *BRICS Deliverable D31*.
- [49] **Sezer, V., Çağrı Dikilitaş, Ercan, Z., Heceoglu, H., Boyraz, P. and Gökaşan, M.** (2011). Hardware and Software Structure of Unmanned Ground Vehicle Otonobil, *5th Biennial Workshop on DSP for In-Vehicle Systems*.
- [50] **Zou, X.Y. and Zhu, J.** (2003). Virtual local target method for avoiding local minimum in potential field based robot navigation., *Journal of Zhejiang University Science*, **4**(3), pp.264–269.
- [51] **Nair, S. and Kobilarov, M.** (2011). Collision avoidance norms in trajectory planning, *American Control Conference(ACC)*, pp.4667–4672.
- [52] **Hart, P.E., Nilsson, N.J. and Raphael, B.** (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *Ieee Transactions On Systems Science And Cybernetics*, **4**(2), pp.100–107.
- [53] **Url-5,** <<http://www.qcontrol.com/SP/OB/>>, date retrieved: 27.09.2012.
- [54] **Sezer, V., Ercan, Z., Heceoglu, H., Bogosyan, S. and Gokasan, M.** (2012). A New Fuzzy Speed Planning Method for Safe Navigation, *IEEE - International Conference on Vehicular Electronics and Safety*, pp.381–386.
- [55] **ISO-FDIS**, (2002)., Transport Information and Control Systems—Adaptive Cruise Control Systems—Performance Requirements and Test Procedures, *Int. Std. ISO/FDIS 15622*.
- [56] **Perez-D'Arpino, C., Medina-Melendez, W., Guzman, J., Fermin, L. and Fernandez-Lopez, G.** (2009). Fuzzy logic based speed planning for autonomous navigation under Velocity Field Control, *2009 IEEE International Conference on Mechatronics*, pp.1–6.
- [57] **Kim, D.** (2011). Control Strategy for High-Speed Autonomous Driving in Structured Road, *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* pp.186–191.

- [58] **Tzafestas S.G. and Zavlangas P.**, (2006). Industrial and Mobile Robot Collision-Free Motion Planning Using Fuzzy Logic Algorithms, *Industrial Robotics: Theory, Modelling and Control*, InTech, pp.301–334.
- [59] **Chen, Q. and Ozguner, U.** (2005). Real-time navigation avoidance for autonomous vehicles: A fuzzy obstacle and goal approach algorithm, *Acc Proceedings of the 2005 American Control Conference*, pp.2153–2158.
- [60] **Madow, A., Munoz, V., Fernandez, R. and Garcia-Cerezo, A.** (1997). Dynamic speed planning for safe navigation, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS '97.*, volume 1, pp.231 –237.
- [61] **Nagel, J., Trepagnier, P.G., Koutsougeras, C., Kinney, P.M. and Dooner, M.** (2006). The Culebra Algorithm for Path Planning and Obstacle Avoidance in Kat-5, *18th IEEE International Conference on Tools with Artificial Intelligence ICTAI '06*, pp.247 –253.
- [62] **Nourani-Vatani, N., Bosse, M., Roberts, J. and Dunbabin, M.**, (2006). Practical Path Planning and Obstacle Avoidance for Autonomous Mowing, Citeseer.
- [63] **Sezer, V., Ercan, Z., Heceoglu, H., Gokasan, M. and Bogosyan, S.** (2012). A New Fuzzy Speed Control Strategy Considering Lateral Vehicle Dynamics, *IEEE - Intelligent Transportation Systems Conference*, pp.19–24.
- [64] **Ioannou, P., Xu, Z., Eckert, S., Clemons, D. and Sieja, T.** (1993). Intelligent cruise control: theory and experiment, *Proceedings of the 32nd IEEE Conference on Decision and Control*, pp.1885 –1890.
- [65] **Hunt, K., Kalkkuhl, J., Fritz, H., Johansen, T. and Gottsche, T.** (1998). Experimental comparison of nonlinear control strategies for vehicle speed control, *Proceedings of the 1998 IEEE International Conference on Control Applications*, volume 2, pp.1006 –1010.
- [66] **Mammar, S. and Netto, M.** (2004). Integrated longitudinal and lateral control for vehicle low speed automation, *Proceedings of the 2004 IEEE International Conference on Control Applications*, volume 1, pp.350 – 355.
- [67] **Guo, L., Huawei, L. and Xu, W.** (2010). The integrated controller of the steering /braking stability for the vehicles, *International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, volume 3, pp.288 –291.
- [68] **Leith, D., Leithead, W. and Vilaplana, M.** (2005). Robust lateral controller for 4-wheel steer cars with actuator constraints, *44th IEEE Conference on Decision and Control and European Control Conference CDC-ECC '05*, pp.5101 – 5106.
- [69] **Kodagoda, K., Wijesoma, W. and Teoh, E.**, (1999). Robust Un-coupled Fuzzy Controller for Longitudinal and Lateral Control of an AGV, *Computational Intelligence*, volume 1625 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp.370–381.

- [70] **Naranjo, J., Gonzalez, C., Garcia, R. and de Pedro, T.** (2006). ACC+Stop go maneuvers with throttle and brake fuzzy control, *IEEE Transactions on Intelligent Transportation Systems*, 7(2), pp.213 – 225.
- [71] **Zhao, J. and El Kamel, A.** (2010). Coordinated throttle and brake fuzzy controller design for vehicle following, *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp.659 –664.
- [72] **Cabello, F., Acuna, A., Vallejos, P., Orchard, M. and del Solar, J.** (2011). Design and validation of a fuzzy longitudinal controller based on a vehicle dynamic simulator, *9th IEEE International Conference on Control and Automation (ICCA)*, pp.997 –1002.

APPENDICES

APPENDIX A.1 : Cosine Theorem

APPENDIX A.2 : Apollonius Theorem

APPENDIX A.1

Cosine Theorem(Law of Cosines):

Figure A.1 illustrates the notation for the Cosine Theorem.

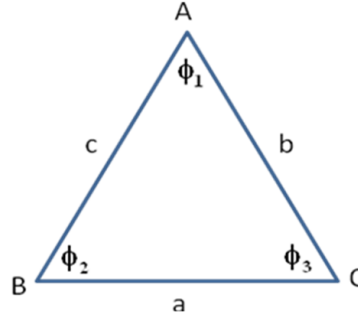


Figure A.1: Cosine Theorem Notation

According to the Figure A.1, the law of cosines says:

$$a^2 = b^2 + c^2 - 2bc \cos \phi_1 \quad (\text{A.1})$$

$$b^2 = a^2 + c^2 - 2ac \cos \phi_2 \quad (\text{A.2})$$

$$c^2 = a^2 + b^2 - 2ab \cos \phi_3 \quad (\text{A.3})$$

APPENDIX A.2

Apollonius Theorem:

Figure A.2 illustrates the notation for the Apollonius Theorem.

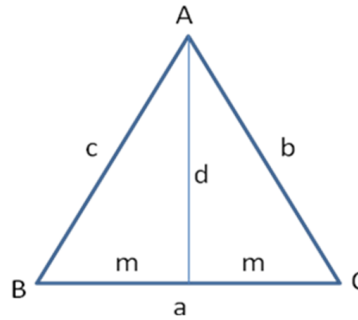


Figure A.2: Apollonius Theorem Notation

According to the Figure A.2, Apollonius Theorem says:

$$b^2 + c^2 = 2m^2 + 2d^2 \quad (\text{A.4})$$

CURRICULUM VITAE



Name Surname: Volkan Sezer

Place and Date of Birth: 16.10.1983

E-Mail: sezervolkan@gmail.com

B.Sc.: Yıldız Technical University - Electronics and Telecommunication Engineering

M.Sc.: Istanbul Technical University-Istanbul Technical University - Mechatronics Engineering

Professional Experience and Rewards:

2005-2008: Research Engineer-MEKAR (Mechatronics Research Center), Istanbul Technical University

2008-2009: Research Engineer-Tubitak (Turkish Scientific Technical Research Organization) MAM(Marmara Research Center)- Energy Institute

2009- :Researcher - MEAM(Mechatronics Education and Research Center), Istanbul Technical University

2009- : Research Assistant-Istanbul Technical University Mechanical Engineering Department

List of Publications and Patents:

▪ **Sezer V.**, Gokasan and M., Bogosyan S. (2011). A Novel ECMS And Combined Cost Map Approach For High Efficiency Series Hybrid Electric Vehicles *IEEE Transactions on Vehicular Technology*, **60(8)**, pp. 3557 – 3570.

▪ Uygan I.M.C., Hartavi A.E, **Sezer V.**, Acarman T. and Güvenc L. (2010). Propulsion System Design of a Hybrid Electric Vehicle *International Journal on Vehicle Design*, **52(1/2/3/4)**, pp. 96-118.

▪ Danis S. , Aytekin B., Dincmen E., **Sezer V.**, Ararat Ö., Öncü S., Güvenç A.B., Acarman T., Altuğ E. and Güvenç L. (2010). Framework for Development of Driver Adaptive Warning and Assistance Systems That Will Be Triggered By a Driver Inattention Monitor *International Journal on Vehicle Design*, **52(1/2/3/4)**, pp. 20-37.

▪ **Sezer V.**, Biliroğlu A.Ö., Demirci M., Uçarol H., Kural E., Özsu E., Kılıç V., Koçkan I., Yıldırım M., Hartavi A.E., Uygan I.M.C., Acarman T. and Güvenc L. (2012). Design and Implementation of a Series-Parallel Light Commercial Hybrid Electric Vehicle *The 24th International Battery, Hybrid and Fuel-cell Electric Vehicle Symposium and Exhibition (EVS24)*, May 13-16, Stavanger, Norway.

- Kural E., Ararat Ö, **Sezer V.**, Solak Y., Kütük O., Özsu E., Uçarol H. (2009). Development of a Hybrid Electric Mid-Sized Bus Concept for Urban Transport *SAE Powertrains, Fuels and Lubricants Meeting*, June 2009, Florence, Italy.
- **Sezer V.**, Uygan, İ.M.C., Hartavi, A.E., Acarman, T., Güvenç, L., Kılıç, and V., Yıldırım M. (2009). Maximum Overall Efficiency Optimization Strategy (MOES) for Power Split Control of a Series-Parallel Hybrid Electric Vehicle, *SAE Commercial Vehicles Conference*, October 2008, Illinois, USA.
- Uygan I.M.C., Hartavi A.E., Güvenc L., Acarman T., Kılıç V., Özelgin İ., Yıldırım M. and **Sezer V.** (2008). FOHEV II - The Ford Otosan Hybrid Electric Research Prototype Vehicle, *International Conference on Automotive Technologies, (ICAT'08)*, November 13-14, 2008 İstanbul, Turkey.
- Bahar M. D., Çimen A. M., Biliroğlu Ö. A., Elcik E, Solak Y., **Sezer V.**, Demirci M. and Uçarol H., (2008). Design and Prototyping of Hybrid Electric Vehicle Electronic Control Unit, *International Conference on Automotive Technologies, (ICAT'08)*, November 13-14, 2008 İstanbul, Turkey.
- Hartavi A.E., Uygan I.M.C.,**Sezer V.**, Acarman T. , Güvenc L, Kılıç V.and Yıldırım M. (2008). Electric Regenerative Power Assisted Brake Algorithm for a Front and Rear Wheel Drive Parallel Hybrid Electric Commercial Van, *SAE Commercial Vehicles Conference*, October 2008, Illinois, USA.
- Hartavi A.E., Uygan I.M.C.,**Sezer V.**, Acarman T. , Güvenc L, and Goktan A.G. (2008). Development of an Electric Power Assisted Brake Algorithm for Electrically Driven Land Vehicles, *9th International Symposium on Advanced Vehicle Control*, October 6-9 2008, Japan.
- Hartavi A.E., Uygan I.M.C.,**Sezer V.**, Acarman T. ,Kılıç V. and Yıldırım M. (2008). Signal Interfacing for Hybrid Electric Vehicular Electronics and an Implementation Study, *International Conference on Vehicle Electronic Systems, (ICVES 2008)*, September 22-24, 2008,USA.
- Uygan I.M.C., **Sezer V.**, Hartavi A.E., Acarman T. and Güvenc L. (2008). Powertrain Design for Hybrid Electric Vehicles, *4. Otomotiv Teknolojileri Kongresi*, June1-4, 2008, Turkey.
- Danis S. , Aytekin B., Dincmen E., Sezer V., Ararat Ö., Öncü S., Güvenç A.B., Acarman T., Altuğ E. and Güvenç L. (2008). Framework for Development of Driver Adaptive Warning and Assistance Systems That Will Be Triggered By a Driver Inattention Monitor, *4. Otomotiv Teknolojileri Kongresi*,June1-4, 2008, Turkey.
- Hartavi A.E., Uygan I.M.C., **Sezer V.**, Acarman T. and Güvenc L. (2008). Development of Hybrid Braking Algorithm for Electric/Hybrid Electric Vehicles, *The 3rd AUTOCOM International Workshop on Hybrid Electric Vehicle Modeling and Control*,June 12, 2007, Turkey.
- Hartavi A.E., Uygan I.M.C., **Sezer V.**, Acarman T. and Güvenc L. (2008). Propulsion System Design of a Hybrid Electric Vehicle, *The 3rd AUTOCOM International Workshop on Hybrid Electric Vehicle Modeling and Control*,June 12, 2007, Turkey.
- **Sezer V.** and Özyılmaz L. (2005). TCP/IP ile uzaktan kontrollü devre tasarımı, *Elektrik-Elektronik-Bilgisayar Mühendisliği 11. Ulusal Kongresi*,November 22-25, 2005 İstanbul, Turkey.

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Sezer V.** and Gökaşan M. (2012). A Novel Obstacle Avoidance Algorithm: "Follow the Gap Method", *Elsevier - Robotics and Autonomous Systems*, 60(9), pp. 1123-1134.
- **Sezer V.** and Gökaşan M. (2012). Low Level Speed Controller Application for Unmanned Ground Vehicles, *İTÜ Journal/Engineering*, Accepted.
- **Sezer V.**, Boyraz P., Ercan Z., Dikilitaş Ç., Heceoğlu H., Öner A., Öke G. and Gökaşan M. (2012). Unmanned Ground Vehicle Otonobil: Design, Perception, and Decision Algorithms, *Smart Mobile In-Vehicle Systems Next Generation Advancements, Springer (Book Chapter)*, Submitted.
- **Sezer V.**, Dikilitaş Ç., Ercan Z., Heceoğlu H., Öner A., Apak A., Gökaşan M. and Mugan A. (2012). Conversion of a Conventional Electric Automobile Into an Unmanned Ground Vehicle (UGV), *IEEE - International Conference on Mechatronics*, pp. 564-569, April 13-15, 2011, İstanbul, Turkey.
- **Sezer V.**, Dikilitaş Ç., Ercan Z., Heceoğlu H., Boyraz P. and Gökaşan M. (2011). Hardware and Software Structure of Unmanned Ground Vehicle Otonobil, *5th Biennial Workshop on DSP for In-Vehicle Systems*, September 4-7, 2011 Kiel, Germany.
- **Sezer V.** (2012). Unmanned Ground Vehicle 'Otonobil' with New Algorithms for Obstacle Avoidance, *Challenges and Opportunities for Green and Smart Vehicles - Pre-Conference Workshop in Conjunction with IEEE ICVES'12*, July 24 , 2012, İstanbul, Turkey.
- **Sezer V.**, Ercan Z., Heceoğlu H., Bogosyan S. and Gökaşan M. (2012). A New Fuzzy Speed Planning Method for Safe Navigation *IEEE - International Conference on Vehicular Electronics and Safety*, pp. 381-386, July 24-27, 2012 Istanbul, Turkey.
- **Sezer V.**, Ercan Z., Heceoğlu H., Gökaşan M. and Bogosyan S. (2012). A New Fuzzy Speed Control Strategy Considering Lateral Vehicle Dynamics *IEEE - Intelligent Transportation Systems Conference*, pp. 19-24, September 16-19, 2012 Anchorage, Alaska, USA.
- Ercan Z., **Sezer V.**, Heceoğlu H., Dikilitaş Ç, Gökaşan M. and Mugan A. and Bogosyan S. (2011). Multi-Sensor Data Fusion of DCM Based Orientation Estimation for Land Vehicles *IEEE - International Conference on Mechatronics*, pp. 672-677, , April 13-15, 2011, İstanbul, Turkey.
- Ercan Z., **Sezer V.**, Heceoğlu H., Boyraz P. and Gökaşan M. (2011). Multi-Sensor Data Acquisition, Processing and Logging Using Labview *5th Biennial Workshop on DSP for In-Vehicle Systems*, September 4-7, 2011 Kiel, Germany.