PATH PLANNING USING HEURISTIC ALGORITHM IN DYNAMIC ENVIRONMENT

DINAMIK ORTAMDA SEZGISEL BIR ALGORITMA KULLANARAK YOL PLANLAMA

Zahra ELMI

PROF. DR Mehmet Önder EFE

Supervisor

Submitted to

Graduated School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Doctor of Philosophy

in Computer Engineering.

2019

DEDICATION

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving family whose words of encouragement and push for tenacity ring in my ears. My brother Hadi and my sisters Soheila and Semira have never left my side and are very special for me. You are my best cheerleaders.

ÖZET

DINAMIK ORTAMDA SEZGISEL BIR ALGORITMA KULLANARAK YOL PLANLAMA

Zahra ELMI

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Mehmet Önder EFE

Kasım 2019, 153 sayfa

Mobil robotların ve otonom araçların navigasyonu bilgisayar ve kontrol bilimlerindeki önemli konulardan biridir. Yol planlama ve engellerden kaçınma, mobil robotlar ve otonom araçlar için sürüş zorlukları konularıdır. Robotikte temel konulardan biri yol planlamasıdır. Yol planlama problemi, karmaşıklığın aracın veya robotun serbestlik dereceleri ile arttığı bilinen bir NP zor problemidir. Yol planlamasının asıl amacı, bir robot veya araç için tehlikeli bir ortamda güvenli ve pürüzsüz bir yol bulmak ve böylece engellerle çarpışmadan robotun başlangıç noktasından varış noktasına hareket etmesini sağlamaktır. Bu tez, bir mobil robotun ve iki farklı yaklaşıma sahip otonom bir aracın navigasyonuna yönelik yol planlama konuları incelenmiştir. İlk olarak, çekirge algoritmasını kullanan yeni bir yol planlama yaklaşımı, mobil robotun dinamik ve bilinmeyen ortamlarda navigasyonunu sağlamıştır. Bu amaç için, iki farklı yaklaşım sunulmuştur. Engelleri tespit etmek için bir duyusal sistem kullanılmış aynı zamanda hızları bilinmyen statik ve dinamik engelleri tahmin etmek ve çarpışmaları önlemek için yeni bir yöntem geliştirilmiştir. Mobil robot, elde edilen bilgileri kullanmış ve çarpışma içermeyen, optimum ve güvenli bir yol bulabilmiştir. Bu tezde önerilen yaklaşım kalabalık ve karmaşık ortamlarda test edilmiştir. Simülasyon sonuçları, yaklaşımın tüm test ortamlarında başarılı olduğunu göstermektedir. Ayrıca, önerilen yaklaşım çeşitli sezgisel yöntemler ve hibrit yaklaşımlarla karşılaştırılmıştır. Karşılaştırma çalışmaları burada tanıtılan çalışma süresi, optimallik, kararlılık ve başarısızlık oranı açısından umut verici olduğunu öngörmektedir. İkincisi, otonom bir araç için Model Öngörülü Denetleyiciye dayalı yeni bir yol planlaması geliştirmiştir, şerit tutma, şerit değiştirme ve şerit birleştirme gibi manevralar hakkında otomatik olarak karar verilmiştir. Güvenliği sağlamak için ayrıca yol sınırı, engeller ve şeridin merkezi için üç farklı potansiyel alan fonksiyonu kullanılmıştır. Yol sınırlarının potansiyel fonksiyonu aracın yol sınırlarının dışına çıkmasını engeller, engel olarak adlandırdığımız araçların potansiyel fonksiyonu aracı engellerden veya çevresindeki araçlardan uzak tutar ayrıca şerit merkezile ilgili potansiyel fonksiyonu aracın merkez şeridin çizgisini takip etmesini sağlamaktadır. Farklı senaryolarda önerilen yol planlama denetleyicisi test edilmiştir. Elde edilen sonuçlar, bir yol planlama kontrolörü kullanarak aracın engellerle çarpışmayı önlediğini ve aracın uygun dinamikleri ile yolun düzenlemelerini izlediğini göstermektedir. Yol planlama kontrolörü, otonom aracın güvenliğini garanti etmektedir.

Anahtar Kelimeler: Yol Planlama, Çekirge Optimizasyonu Algoritması, Engelden Kaçınma, Bilinmeyen statik ve dinamik ortam, Model Tahmini Kontrol, Yapay Potansiyel Fonksiyonu.

ABSTRACT

PATH PLANNING USING HEURISTIC ALGORITHM IN DYNAMIC ENVIRONMENT

Zahra ELMI

Doctor of Philosophy, Department of Computer Engineering

Supervisor: Prof. Dr. Mehmet Önder EFE

November 2019, 153 pages

Navigation of mobile robots and autonomous vehicles is one of the important issues in computer and control sciences. Path planning and obstacle avoidance are current topics of navigational challenges for mobile robots and autonomous vehicles. One of the essential issues in robotics is path planning. The problem of path planning is a well-known NP-hard problem where the complexity increases with the degrees of freedom of the vehicle or robot. The main aim of path planning is to find a safe and smooth path in a dangerous environment for a mobile robot or vehicle so that the robot moves from the starting point to the destination point without colliding with obstacles. This thesis has investigated path planning issues for navigation of a mobile robot and an autonomous vehicle with two different approaches. First, a novel path planning approach using the grasshopper algorithm is presented for the navigation of a mobile robot in dynamic and unknown environments. For this purpose, two different approaches are presented. A sensory system is used to detect the obstacles and a new method was developed to predict

and avoid static and dynamic obstacles while the velocities of the obstacles are unknown. The mobile robot uses the obtained information and finds a collision-free, optimal and safe path. The proposed approach in this thesis was tested in crowded and complex environments. Simulation results demonstrate that the approach is successful in all test environments. Also, the proposed approach is compared with several heuristic methods and hybrid approaches. The comparison work stipulates that the approach introduced here is promising in terms of running time, optimality, stability and failure rate. Second, a new path planning based on Model Predictive Controller (MPC) for an autonomous vehicle is developed, which automatically decides about the mode of maneuvers such as lane-keeping, lane changing, and lane merging. For ensuring safety, we have additionally used three different potential field functions for road boundary, obstacles, and center of the lane. The potential fields of road boundaries keep the vehicle from going out of the road boundaries, the potential field surrounding vehicle keeps the vehicle away from obstacles or surrounding vehicles and also the potential field of lane centering leads to tracking of the centerline of the lane by the autonomous vehicle. The proposed path planning controller on the different scenarios have been tested. The obtained results represent by using a path planning controller the vehicle avoids collision with obstacles and observes the regulations of the road by appropriate dynamics of the vehicle. The path planning controller guarantees the safety of the autonomous vehicle.

Keywords: Path Planning, Grasshopper Optimization Algorithm, Obstacle Avoidance, Unknown static and dynamic environment, Model Predictive Control, Artificial Potential Field.

ACKNOWLEDGEMENTS

After a so great experience, I obviously have many people to thank...

At first, I would like to deeply thank Prof. Dr. Mehmet Önder Efe, my supervisor, for the continuous support of my Ph.D. study and related research, for his patience, motivation, and genuine concern and all he has done for me in these years. His scrupulousness and thoroughness have been the best teaching I could ask and receive. I am thankful to him because he has done my research experience as good as I guess it should be.

I am thankful to Asst. Prof. Murat Aydos, Prof. Dr. Coşku Kasnakoğlu, Assoc. Prof. Ahmet Murat Özbayoğlu, and Asst. Prof. Adnan Özsoy who were more than generous with their expertise and precious time.

I also appreciate Prof. Dr. Ahmad Akbari from Sahand University for his many suggestions and constant help during all these years. He has been an incredible support for research.

Last but not least, I would like to thank my family for showing great patience and understanding during the course of the project and supporting me spiritually throughout writing this thesis and my life in general.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURES

X_i	Position of the i^{th} grasshopper
S_i	Social interaction of the i^{th} grasshopper
G_i	Force of gravity on the i^{th} grasshopper
A_i	Wind advection
s	Strength of social forces
d_{ij}	Distance between the i^{th} grasshopper and the j^{th} grasshopper
f	Attraction intensity
l	Scale of attraction length
g	Gravitational constant
$\hat{e}_g$	Unit vector towards the center of the earth
u	Drift constant
$\hat{e}_w$	A unity vector in the wind direction
ub_d	Upper boundary an in the d^{th} dimension
lb_d	Lower boundary in the d^{th} dimension
$\hat{T}_d$	Target value
c_1 and c_2	Coefficients to shrink the comfort zone, repulsion zone, and attraction zone
c_{max}	Maximum value of c
c_{min}	Minimum value of c
n	Number of the current iteration
L	maximum number of iterations
D_{GT}	Minimum Euclidean distance between the target and i^{th} grasshopper

D_{GO}	Euclidean distance between the neighboring obstacle and position of the i^{th} grasshopper
D_{RO}	Distance between the robot and the nearest obstacle
m	Vehicle's mass
v_x	Longitudinal velocity
v_y	Lateral velocity
r	Yaw rate of the vehicle in its center of gravity
X	Longitudinal position
Y	Lateral position
θ	Heading angle of the vehicle
I_z	Momentum of inertia' vehicle around its vertical axis
F_{y_f}	Total lateral force of front tires
F_{y_r}	Total lateral force of rear tires
α_f	Sideslip angles of the front tire
α_r	Sideslip angles of the rear tire
δ	Angle of steering
c_f	Cornering stiffness values of the front tire
c_r	Cornering stiffness values of the rear tire
x	State vector
u_c	Input vector
A	State matrix
B	Input matrix
C	Input matrix
δ_{max}	Maximum steering angle
T_{max}	Maximum propelling torque

R_{eff}	Effective radius of the wheel
$F_{x_T} - max$	Maximum total longitudinal tire force
$F_{y_{f,r}} - max$	Maximum front or rear lateral tire force
μ	Friction coefficient of the tire
$F_{z_{f,r}}$	Front or rear vertical force of the tire
h	Height of the center of gravity of the vehicle from the ground
$F_{y_{f,r0}} - max$	Nominal maximum lateral front or rear tire force
$F_{z_{f,r0}}$	Nominal vertical front or rear tire force
l_f	Distance from vehicle's center of gravity to front axle
l_r	Distance from vehicle's center of gravity to rear axle
L_w	Lane width
N	Prediction horizon
N_c	Control horizon
N_{rc}	Repetition control horizon
y	Output matrix tracking
y_{des}	Desired lateral position
$v_{x_{des}}$	Desired speed
l_{des}	Index number of the desired lane from the right
ΔY_R	Lateral offset of the road compared to a straight road
Q	Tracking weighting matrix
R	Input weighting matrix
S	Input change weighting matrix
U	Potential field
v	Vehicle's lateral velocity

W	Vehicle's weight
W_0	Obstacle's added width
X	Vehicle's longitudinal position
X_0	Minimum safe longitudinal distance
X_1	Longitudinal position of left side of contact rectangle
X_2	Longitudinal position of right side of contact rectangle

GA	Genetic Algorithm
FL	Fuzzy Logic
NN	Neural Network
PSO	Particle Swarm Optimization
ACO	Ant colony algorithm
SA	Simulated Annealing
BA	Bee algorithm
FA	Firefly Algorithm
BFOA	Bacteria Forging Optimization algorithm
CSA	Cuckoo search algorithm
GOA	Grasshopper Optimization Algorithm
MPC	Model Predictive Control
VRBVG	Virtual rubber band visibility graph
EDD	Equi-Distance Diagram
FEM	Finite Element Method
APF	Artificial Potential Field
MRSs	Multi-Robot Systems
SNN	Spiking Neural Network

DPSO	Darwinian PSO
UCAV	Uninhabited Combat Air Vehicle
DABC	Directed Artificial Bee Colony
AI	Artificial Intelligence
GOA-GA	Grasshopper optimization algorithm- Genetic algorithm
RRT	Rapidly-exploring Random Tree
PF	Potential Field
AUV	Autonomous Underwater Vehicle
RRT*	Rapidly-exploring Random Tree star
IHA	Intermediate Heading Angle
CoG	Center of Gravity
ZOH	Zero-Order Hold
FOH	First-Order Hold
SQP	Sequential Quadratic Programming

1. INTRODUCTION

In recent decades, one important research issue is the Autonomous Mobile Robot. This has experienced a major evolution in planning, control, application, and other aspects, so autonomous mobile robots are useful for human activities. The autonomous robots are shown with different shapes such as mobile manipulators, two or three-wheel robots, Omni-directional robots, and car-like robots. Navigation is one of the main tasks of the autonomous mobile robot. The mobile robot is able to detect the configuration space and move from the start coordinate to the target coordinate autonomously, safely and without human intervention using an appropriate navigation system. This chapter consists of four sections; the first section of the chapter presents motivation and background from the proposed research. The second section discusses the main aim of the research and its scope in the scientific and engineering area. The originality and main idea of the research are discussed in the third section. The thesis layout is presented in the fourth section.

1.1. Motivation and Background

Mobile robots are applied in a variety of domains including manufacturing, military, medicine, space exploration, engineering, and farming. The use of high-performance robots is inevitable due to the limited human capability and the high accuracy of robot motions and the capability for operating in an environment detrimental to human health or avoiding human error in repetitive and boring activities. The technical and industrial applications of mobile robots are very important. Mobile robots can be used for military surveillance, inspection, underground mining, entertainment, factory automation, space exploration, and transportation tasks in terms of features such as reliability (such as surveillance that is consecutive and reliable execution of monotonous tasks), accuracy, accessibility (such as inspection of narrow spaces, hazardous environments or remote inaccessible sites for humans), or cost (transportation systems using autonomous mobile robots can be cheaper than standard track-bound systems).

Finding a safety path and having successful navigation in such environments is an important challenge for the robot mobile. Therefore, path planning strategy is a fundamental requirement to successfully navigate mobile robot from the start coordinate to the target coordinate with obstacle avoidance. Besides minimizing navigation time, communication delay, and energy consumption, the founded path should be optimal in terms of path length.

An autonomous mobile robot is an artificial intelligence machine that is able to perceive the environment information (e.g. the obstacle and target positions), plan an optimum path by avoiding static and moving obstacles, and quickly react to environmental conditions without human interventions. Path planning in an unidentified environment is an important problem for moving robot navigation. Nowadays, the real-time implementation of the mobile robot is constantly growing, so the mobile robot with obstacle avoidance mechanism is an important requirement of today. In term of the determination of its position in the reference frame and also planning towards the target, autonomous navigation of the mobile robot is a complicated process. The navigation approach consists of 4 steps as follows and represented in Figure 1-1.

- Perception
- Localization / Mapping
- Cognition / Planning
- Motion control

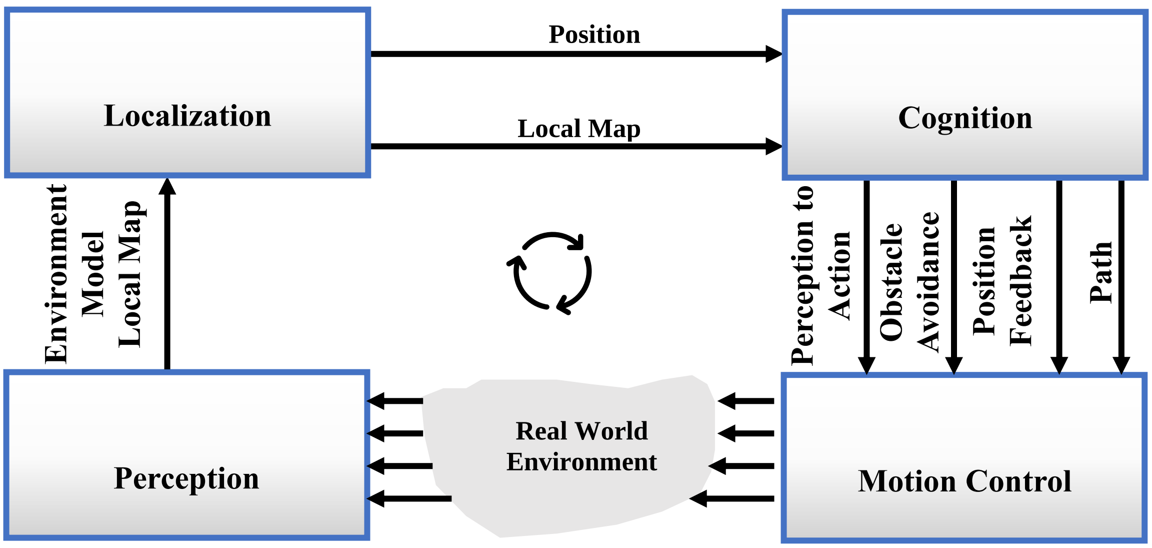


Fig. 1-1: The navigation process of mobile robot.

The initial information of the environment is provided by using sensors and the given information is utilized to build the surrounding map (perception). The obtained information by sensors is used to determine robot location in the environment (localization). After localization, the robot should plan the path from the start coordinate to the target coordinate (Cognition / Planning) and control the movement of the robot actuators (for motion control). By following the navigation process, the path planning approach for mobile robot navigation is developed which is able to find a desired optimal path without collision from the start point of the robot to the target point in the unknown uncertain environment.

When the environment does not contain obstacles, mobile robot navigation is not an important issue; but when there are different static and dynamic obstacles in the environmental map, it is a significant research issue for optimization. For known and unknown environments, many approaches to solving moving robot navigation are discussed. The path planning approaches are categorized as follows:

1. Global path planning (Off-line path planning approach),
2. Local path planning (On-line path planning approach).

In global path planning, the initial knowledge from the environmental map (e.g. position, shape, and obstacle size) is used for path planning while in local path planning, any initial knowledge from the environmental map is not necessary. Due to low computational cost, real-time execution, and the capability to handle the incertitude of the environmental map, the local path planning approach is widely used in comparison to global path planning. The conventional global path planning approaches such as Roadmap, Voronoi diagram, Sub-goal network, Cell decomposition, and Artificial potential field are not suitable for on-line implementations. Therefore, for on-line implementation of mobile robot navigation problem, the artificial intelligence approaches such as Genetic Algorithm (GA), Fuzzy Logic (FL), Neural Network (NN), Particle Swarm Optimization (PSO), Ant Colony Algorithm (ACO), Simulated Annealing (SA), Bee Algorithm (BA), Firefly Algorithm (FA), Bacteria Foraging Optimization Algorithm (BFOA), Cuckoo Search Algorithm (CSA), and combination of the above-mentioned algorithms have been utilized.

In this thesis, navigation strategies are designed and developed based on artificial intelligence algorithms for a mobile robot in an uncertain environment and hybrid of them. The main aim of this thesis is to plan an intelligent path planner to avoid the static and dynamic obstacles and reach to target in minimum travel time. To find the optimal path, the Grasshopper Optimization Algorithm (GOA), Genetic Algorithm, and the hybrid version of them are studied. This hybrid path planner has tested for different scenarios and has simulated to check feasibility in an uncertain environment. With regard to the length of the path and the duration of navigation, the developed hybrid path planner using GA, and GOA is useful in comparison to a single path planner.

In addition, a path planning based on Model Predictive Control (MPC) has been developed for the autonomous vehicle using artificial potential field. The potential field approach is based on attractive and repulsive functions; the attractive function causes a vehicle to move toward the target while the repulsive function prevents the vehicle from a collision with obstacles. In the target point, the target potential field has a minimum value, then it tries to attract the vehicle, the PF of obstacle has a maximum cost in the obstacle positions which repels the vehicle from the obstacle. The main objective of this thesis is to navigate the vehicle to the target point without any collision by tracking the objective function controller term The repulsive function as PF is therefore considered only.

1.2. Aims of Proposed Research

This thesis follows two important objectives; the first is to plan and develop a path planner based on an artificial intelligence approach to have optimal path planning with of static and dynamic obstacles in an unknown environment. Also, to have a suitability path planner and resolve the problem of the path planning, the grasshopper optimization algorithm has been combined by the genetic algorithm which has been tested and evaluated in different environments.

The second objective is path planning for the autonomous vehicle using MPC based on three artificial potential field functions in different environmental conditions. Using artificial potential field methods in the path planning problem of autonomous vehicles presents a very promising solution as a forward simulation of vehicle motion. The key challenge is how to incorporate vehicle dynamics into the route planning process to build feasible trajectories. In producing the desired vehicle behavior, the slope, height, and shape of the potential field play a major role.

The principal aims of the proposed research in the thesis are as follows:

1- Building path planning controller based on the grasshopper optimization algorithm for mobile robot navigation problem.

2- Developing a hybrid navigation controller based on the grasshopper optimization algorithm and genetic algorithm.

3- Analyzing the architecture, kinematic, and dynamic of the autonomous vehicle.

4- Designing and developing MPC based on the artificial potential field.

5- Simulating the proposed approaches to validate.

The mobile robot used for the first objective should have the characteristics as follows:

- It should understand its environment by the obtained information of sensors.
- It should move without slipping in its environment.
- It should have an appropriate mechanism to detect and avoid obstacles.
- It should intelligently update itself by using self-learning ability.
- It should not inflict any loss to the environment.

To path planning the mobile robot, three important behaviors are considered:

- Target seeking behavior: by this behavior, the mobile robot continuously searches the target until achieves it.
- Obstacle seeking behavior: when there is an obstacle on the mobile robot path, this behavior causes the mobile robot to cross the obstacles with a safe distance.
- Wall following behavior: by using this behavior, the mobile robot could come out from the trap like situation. During navigation, the mobile robot follows the obstacle walls.

1.3. Originality and main idea of the proposed research

First, a novel artificial intelligence path planner and a new hybrid path planner to path planning in an unidentified environment with static and moving obstacles are represented as the proposed research in this thesis. The popular approaches such as genetic algorithm and grasshopper optimization algorithm are combined to get the benefit over the other heuristic approaches. The grasshopper optimization algorithm and, a combination of it with genetic algorithm and fuzzy logic have not been yet utilized to mobile robot path planning with static and dynamic obstacles.

In following, an MPC based on artificial potential field function has offered to path planning of autonomous vehicles. Three different artificial potential field functions are used as the cost function of MPC for obstacle avoidance. The used vehicle model is a linear bicycle model that shows the vehicle dynamics for highway scenarios. By using three different artificial potential fields, the maneuvers of lane-keeping, lane changing, lane merging, and intersection crossing are performed. It is demonstrated that the proposed method potentially represents an effective and powerful solution to the navigation control of autonomous vehicles.

1.4. Layout of thesis

The sections of this thesis are organized as follows:

Chapter 1 offers a brief overview of the navigation of the mobile robot and the main objective of the proposed research.

Chapter 2 provides background and detailed survey of the various approaches of mobile robot navigation.

Chapter 3 presents a background of path planning based on MPC.

Chapter 4 presents the applications of the grasshopper optimization algorithm for mobile robot navigation. A new fitness function by using the grasshopper optimization algorithm has been derived for safety centric path planning and avoidance of static and dynamic obstacles. To improve accuracy and performance the proposed approach, the novel hybrid of GOA and GA approaches has presented. In the following, the simulation results of the new proposed controller and its proposed hybrid with GA have also discussed for path planning of mobile robot at the end of the chapter.

Chapter 5 develops a new motion planning based on the model predictive controller with three potential fields. A vehicle bicycle model with linear tire models has been presented. To keep the model valid by keeping the tire in its linear force region, tire constraints have been introduced. The potential field functions have been considered for obstacles and road lanes and centerline. For MPC problem, the objective functions consist of the vehicle model as its model, vehicle constraints as its constraints, and the potential fields. Finally, to evaluate the efficiency of the motion planning based on MPC, the simulation results of the proposed MPC for the different complicated scenarios is demonstrated.

Chapter 6 concludes the proposed research in this thesis and gives a future work of research.

2. BACKGROUND AND LITERATURE REVIEW

This chapter focuses on different proposed strategies in the navigation domain of the mobile robot. To understand and develop path planning strategies in different environmental conditions, several classical and reactive navigation approaches have investigated. At the end of this chapter, the abstract of the literature and previously used approaches have presented.

2.1. Introduction

The path planning approach for an autonomous mobile robot is a navigation task in an unknown uncertain environment that is performed without human interference. In this approach, the mobile robot navigates to reach to the target while it detects obstacles during the movement and avoids collision with them. In the path planning problem, the models of the mobile robot and its environment should be quantifiable. The model of a mobile robot includes the dimension of the robot, differential and kinematic equations, and control parameters on the robot movement. The environment model consists of the position of the robot (robots), obstacle, and representation of the map. The necessary requirement of navigation for any mobile robot is self-localization, path planning, map building, and collision avoidance, respectively. The localization of the robot represents the capability of a mobile robot to explore and detect its position and to orient itself in the reference frame.

Path planning is the development of localization in which it needs the specification of the current position of the mobile robot and the position of target location in the reference frame. Map building is a metric map that describes the position of the mobile robot into the reference frame. And finally, in collision avoidance, the mobile robot responds to its environment by using the sensors of obstacle detection. The mobile robot navigation system is performed as global and local (the main parts of the navigation system is shown in Figure 2-1). The global navigation is the capability to locate the robot position based on map conditions and the movement towards the desired destination position. While the local navigation is the capability to locate the position of the mobile robot based on static and dynamic obstacles in the environment without collision. In many works of literature to solve path planning problems, the conventional and reactive approaches have been considered.

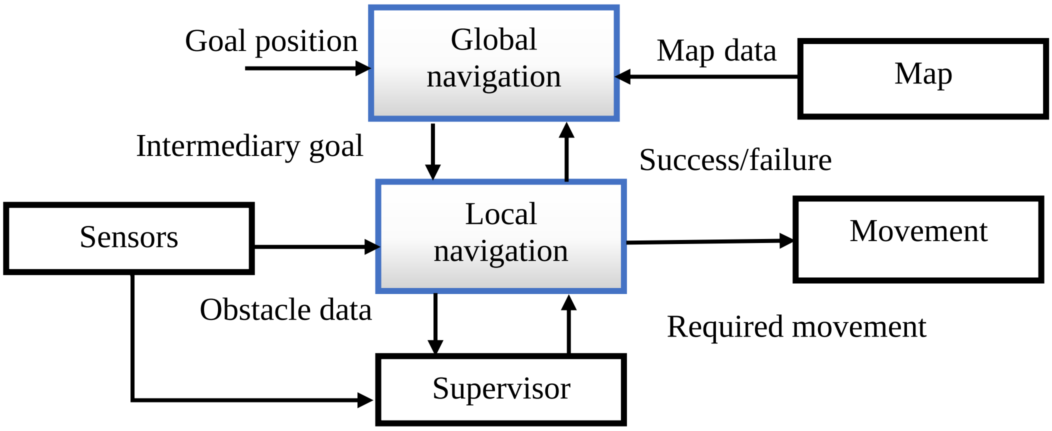


Fig. 2-1: Main parts of the navigation system.

The conventional methods are deterministic and they fail when there is a discontinuity in an objective. But the reactive methods have the capability to explore space on the global configuration until they present different solutions and search for the possible solution in the local region. In navigation problems, by using the path planning algorithm, the mobile robot plans and builds an appropriate path without collision then moves towards to target point by the built path. The path planning of the mobile robot is divided into a chain of functional units which is presented in Figure 2-2. After investigating many research papers in the field of robot path planning, many existing research works are identified and categorized for each technique.

2.2. The used navigation techniques for mobile robot

The research over the navigation of the mobile robot field leads to creating efficient navigation techniques to control and navigate the robot in industrial and household applications. In the last few decades, many studies on navigational approaches have been presented by various researchers to find an appropriate methodology for controlling the robot. The research work of this thesis is to plan and develop an effective path planning approach for one mobile robot by using a heuristic algorithm namely GOA and its hybrid with another heuristic algorithms in the static and dynamic environment. The different methods used to navigate the mobile robot are divided into two categories such as classic and reactive approaches which have been discussed below.

2.2.1. Classic approaches

Many classic approaches are utilized to work out navigation problems. The researches based on the classic approaches are presented below.

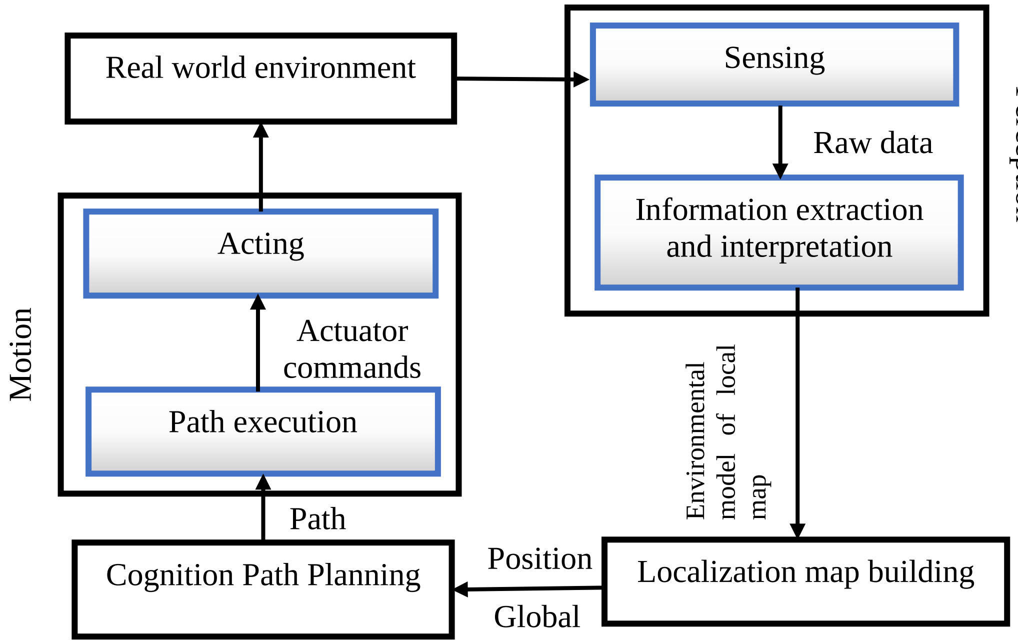


Fig. 2-2: The navigation diagram of mobile robot.

2.2.1.1. Roadmap approach

The roadmap approach is named as highway approach. Roadmap creates a path from one location to another and it is the connection between the free spaces that are shown as a set of the one-dimensional curve [1]. After constructing the roadmap, it is applied as a sequence of the homogenous trajectories that the planner will search among them to find an optimal solution. The nodes on graphs are usually waypoints which the robot requires them to have a successful movement. So, the roadmap method is used to find the shortest path from the start point of the robot to the target position. To develop a roadmap approach, Visibility and Voronoi graphs are utilized.

The visibility graph approach is first introduced by Hart and Nilson [2] in 1969 and is utilized to the Shakey robot presented by Lozano-Prez and Wesley [3] in 1979. The running time of the first algorithm is $O(n^3)$. In many applications like graphics and robotics, VG is used. The visibility graph approach links the start point with the nodes from the map to the target point and explores them for finding an optimal path. Figure 2-3 shows the visibility graph so that the dark blue regions and yellow line represent the obstacles and the found path from the start point to the target point, respectively. In other words, it can be said that VG is a set of polygonal configurations at an undirected graph where vertices of obstacles show the vertices of the graph and the lines between pairs of vertices show the edges. If there is an open line between two vertices, it does not intersect any obstacles.

This method can be handled for the environmental map with polygonal obstacles where the vertices of polygon show nodes and the edges are the connector between nodes [4]. In 1986, M. Sharir and A. Schorr [5] have provided several studies to reduce the complexity of the visibility graph as $O(n^2 log(n))$. In [6], the faster algorithms of visibility graph are presented. In [7], researchers used a visibility graph based on polygon aggregationThe principal aim of this method is to group small obstacles and combine polygons after the clustering. Furthermore, the algorithm [8] proposes separating C-spaces in many regions and then uses the graph of visibility to parallel assembling that component. The shortest path was calculated by merging each partial minimum distance path. In [9] also, the visibility graph was improved in order to overcome its drawbacks by sharing local information among multiple robots. The proposed Polygons generated by the algorithm from a series of interconnected segments and merge them if necessary. The outcome of the information sharing is compared by a visibility graph to the non-sharing information, and the algorithm proposed is better. The enhanced VG in a Virtual Rubber Band Visibility Graph (VRBVG) method to produce a VG assuming that C-spaces are unknown and placed outside the sonar coverage vehicle. They utilized torpedo-type under-actuated vehicles to navigate in an uncertain underwater condition. Despite all the efforts to improve the visibility graph, it is an offline realizable solution in a 2D environment. Also, according to proof of Canny [10], the path planning problem in 3D is NP-hard.

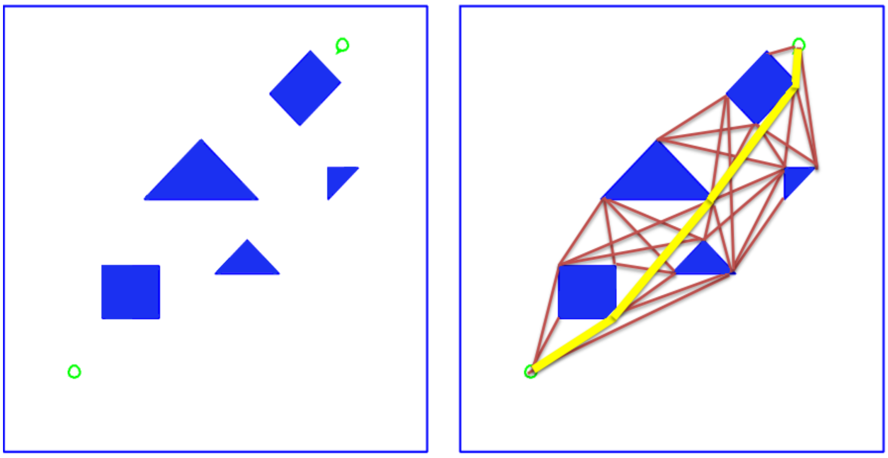


Fig. 2-3: Visibility graph.

For four decades, the Voronoi theory has been used. Descartes utilized diagrams like Voronoi in his Traité de la Lumiére published in 1644 to demonstrate the matter's situation in the solar system and its environment. Since 1970, Voronoi diagrams have been used and several different surveys on various algorithms, implementations, and generalizations of Voronoi diagrams have been introduced [11]. The Voronoi graph is another algorithm of the roadmap that is used to navigate the mobile robot [12]. This divides the area into sub-regions so that each edge is made from two neighboring points of obstacle boundaries using equidistant points. The Voronoi diagram can be constructed in just $O(n\log(n))$ time, where n shows the number of the vertices. Figure 2-4 shows the Voronoi graph. The application of the Voronoi graph to navigate the mobile robot is presented by [13] to keep the mobile robot away from the dynamic obstacle in the defined environment.

Several enhancements are presented for efficient path planning to enhance performance and fix disadvantages such as long loops and sharp turns in the Voronoi diagram. It is worth noting that this approach is not able to obtain the optimal path and its executive processes are complex. For the development of successful path planning among with Voronoi diagram, the different strategies such as skeleton maps are introduced by Yang et al. [14]. The hybrid of Voronoi and visibility diagrams is presented by [15] which has provided the optimality in the planner environment. In [16], a probability application for the roadmap method to comprehend and develop a strategy for path planning is introduced. This method, though, is not appropriate to achieve the optimal length of the path. To improve the proceeding of seeking the shortest path,[17] by using the lazy-in-collision-detection method, a little modification is presented in the probabilistic roadmap approach.

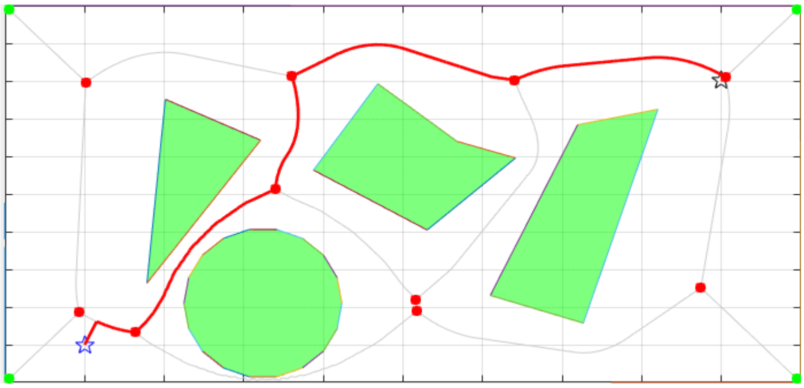


Fig. 2-4: Voronoi diagram.

MAPRM method [15] is a path planning algorithm driven by the Voronoi diagram method, gathers sampled configurations on the medial axis. Also, the Equi-Distance Diagram (EDD) [18] is another method based on the Voronoi roadmap which is defined by linking the maximum local value of a clearance function defined using distance functions. The major drawback of this method is that their roadmap is built offline and the environmental information has to be provided in advance. In [19], to follow an aircraft's kinematic limitation, improvements are done on the Voronoi diagram in three phases. First, the basic Voronoi diagram produces a preliminary diagram. Secondly, by smoothing the impractical corner of all routs from the start coordinate to the target coordinate, the preliminary produced diagram is enhanced. The cost of the edge of the modified Voronoi diagram is then modeled and calculated. At last, by using Dijkstra's algorithm, the optimal path is determined. The enhanced Voronoi diagram is much lower than the fundamental one. The other enhancement is performed on the basic Voronoi diagram in [20] with Delaunay triangulation. The proposed approach is evaluated by 25 different environments. The results show that the modified Voronoi diagram is less expensive in computational terms and it responds in a shorter period of time. However the algorithm generates a path that may not be the shortest, and it is one of the disadvantages of this algorithm. The improved Voronoi diagram is used in a dynamic environment over an unmanned air vehicle. The path is generated using the Voronoi diagram-based radar threat field To find the shortest distance, Dijkstra's algorithm is then applied. In [21] the images are collected and then grouped into a smaller group to reduce the computational time. By using a path planning approach based on the fuzzy interference mechanism, a smooth robot path is also developed.

2.2.1.2. Cell Decomposition

This approach is one of the popularly used approaches for mobile robot navigation. The approach of cell decomposition primarily seeks an obstacle-free cell and using these cells creates a finite graph [22]. This approach divides the configuration into cells and guarantees that each cell is discrete, non-overlapping, and there is not involved by any obstacle and uses connective graphs to travel from one cell to another cell until reaching the target [23]. During the traveling, cells with no obstacles are determined to reach the path planning approach from the start coordinate to the target coordinate. The cells with the obstacles existing on the trajectory divide into two new cells until a cell with no obstacles are generated and these cells have been added to the sequence and used to find the optimal path from the start coordinate to the target. The start and target locations are characterized by start and target cells in a cell decomposition approach. The sequence of cells with no obstacles, which connects these areas, indicates the needed trajectory [24]. There are several types of cell decomposition approaches:

1- Exact approach of cell decomposition.

2- Approximate approach of cell decomposition.

3- Adaptive approach of cell decomposition.

Figure 2-5 shows the exact cell decomposition [25], the cells have not the special shape and size but could be determined through environmental map, shape, and obstacle positions. This approach uses a regular grid in different ways. The first phase in exact cell decomposition is decomposing the free configuration into trapezoidal and triangular cells by plotting parallel lines from each vertex of each internal polygon in the space of configuration to the outer boundary. Next, in the connective graph, each cell is numbered and shown as a node. In the connective graph, the adjacent nodes are connected in the configuration space. An obtained trajectory in this graph is a channel in the free configuration defined by the striped cell sequence. This channel is then converted into a free trajectory by linking the start configuration to the target via the intersection of midpoints of the neighbor cells in the channel.

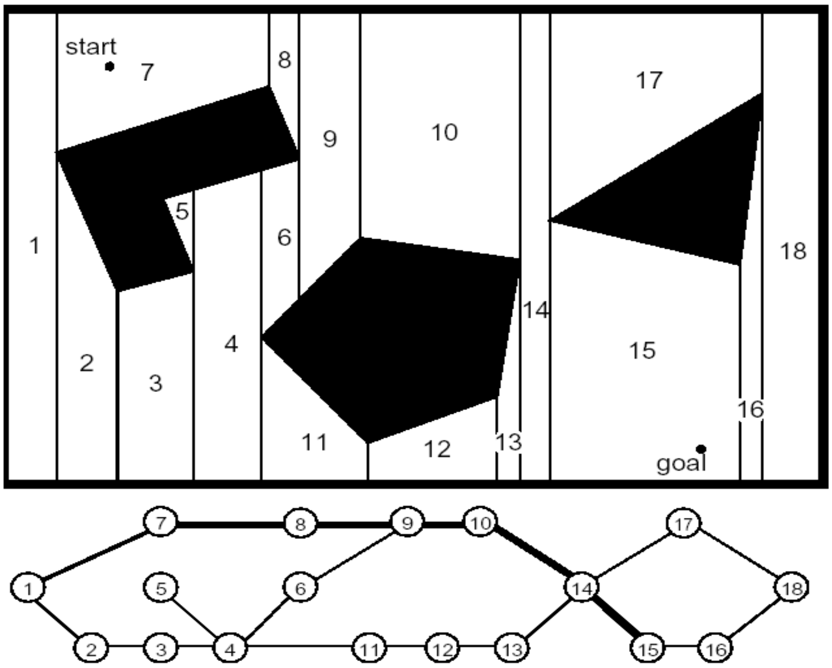


Fig. 2-5: Exact approach of cell decomposition.

An approximate approach of cell decomposition [26], a regular grid is located on the configuration space and all grid cells are predetermined in terms of shape and size to make it easy to use. This approach is named approximate cell decomposition; since physical objects, boundaries do not require to correspond with the cell boundaries predefined. Each physical object in the grid region is regarded as an obstacle, otherwise, it is a free configuration. To discover a trajectory, the center of each cell is regarded to be a node in the search graph. As observed in Figure 2-6, nodes may be either 4-connected or 8-connected. This indicates whether the traveling path of the robot has been considered diagonally between them or not.

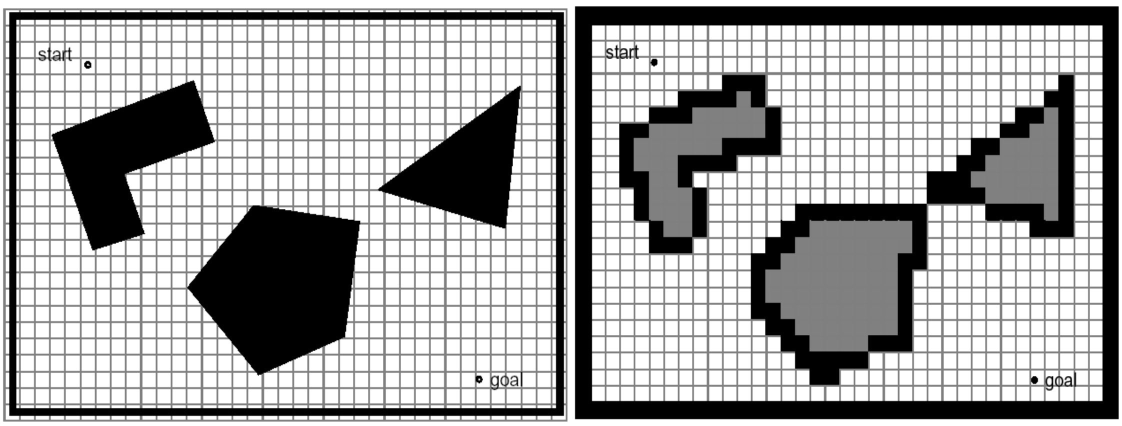


Fig. 2-6: Approximate approach of cell decomposition.

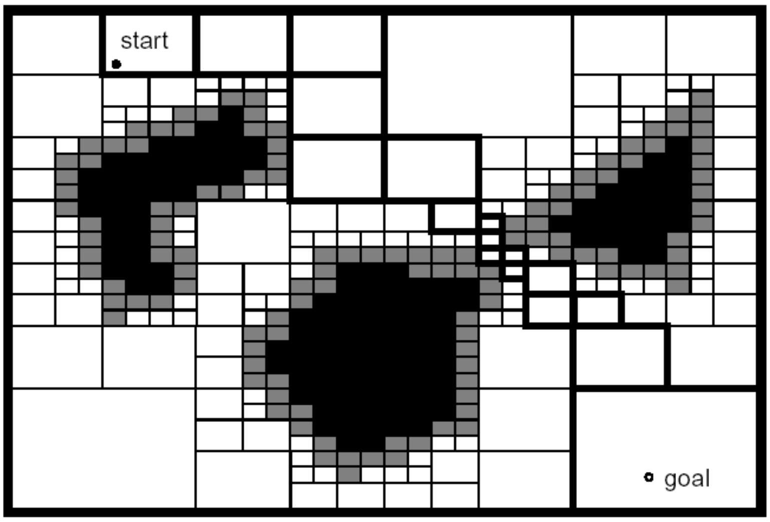


Fig. 2-7: Adaptive approach of cell decomposition.

The adaptive approach of cell decomposition comprehends the available data in free configuration and applies the fundamental concepts of preventing free configuration in regular cell decomposition. Samet [27] and Noborio [28] introduced quad tree-adaptive decomposition. This approach splits the configuration space into cells with large sizes, but if a configuration cell is partially occupied by the obstacle, it will split into four equal sub-cells. Then these sub-cells are split again until all cells are either full or empty. The obtained configuration includes cells of various sizes and concentrations; however, the borders of each cell are very closely aligned with the borders of the obstacle which shown in Figure 2-7. When the robot receives new knowledge from the environment, it updates its information according to new obstacles, thus adaptive approach of cell decomposition fails in complex and dynamic environments. Therefore, completely restoring the information structure of the map is necessary.

In order to overcome the drawback of the cell decomposition method and to improve the effectiveness of the approach by splitting the cell quarterly, an improvement is introduced in [22]. Next, each cell is evaluated to see whether any obstacles are present. The cell is then split in the quarter again. This approach is utilized to achieve an optimal trajectory. The exact approach of cell decomposition and comprehensive path planning are implemented in [29]. The phase for this algorithm started by dividing the mine area into an exact cell. Thus the covering path each cell is generated with the direction of coverage and the obtained paths of the graph are based on the graph of adjacency.

A moving trajectory is produced by including all graph trajectories with all the direction of coverage. Finally, the shortest moving trajectory has been calculated for all graph trajectories and the comprehensive trajectory has been produced. One of the disadvantages of cell decomposition is to have a non-optimal trajectory. By using L_1 norm, squared L_2, and L_∞ norm, three different equations are generated in work [30]. This approach is utilized three equations to describe the metrics as an alternative to the use of the midpoints of the cell in the basic cell decomposition. Cell decomposition was developed explicitly into configuration spaces in [31]. For tasks such as palletizing and handling, they introduced a path planning approach. For tasks such as palletizing and handling, they have introduced a path planning approach. It is generated cylindrical cell decomposition in the configuration space for the robot with six degrees of freedom to accelerate the time without requiring the addition of an obstacle into configuration spaces.

2.2.1.3. Potential field

The artificial potential field method for navigation of the mobile robot was proposed by Khatib [32] in 1986. Target and obstacles operate as charged surfaces, and the sum potentials generate an imaginary force on the mobile robot. This imaginary force attracts the mobile robot to the target point and repulses it from the existing obstacles in the environment. The robot uses the negative gradient as shown in Figure 2-8 to avoid collision with obstacles and achieve the target. A lot of researchers have utilized the discussed approach on the path planning of the mobile robot and avoiding obstacles. In [33] for mobile robot navigation, the potential field method is introduced by Garibotto et al. Kim et al. [34] have presented a novel strategy by using the potential field to avoid obstacles in an unknown environment. To avoid a minimum local problem, they have applied a harmonic function. Also, the solution to the local minimum problem has been proposed by Borenstein et al.[35]. In this approach, the dynamic characteristic of mobile robot navigation has been regarded. In order to avoid the obstacle collision, the evaluation and analysis of the potential field approach in a moving environment are carried out in [36].

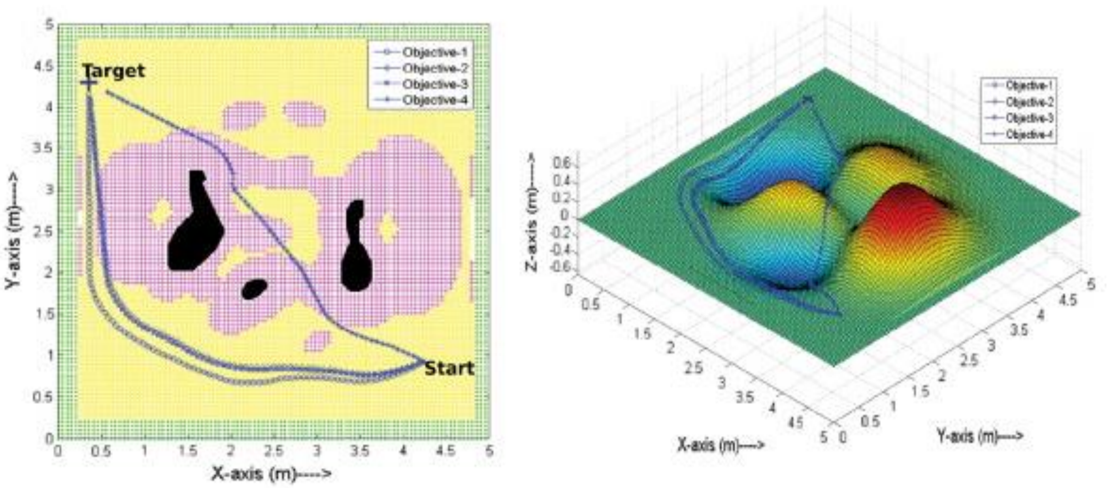


Fig. 2-8: Artificial potential field approach for navigation of the mobile robot.

By using the electrostatic laws [37], the new enhancement in the potential field method is made. The electrostatic design develops a potential function and generates a free-collision trajectory in real-time. It's not easy to avoid dynamic obstacles in real-time, therefore, [38] discusses a mechanism of speed control to realize obstacle location and speed to reach the target. In order to avoid local minima, and to get to global optimum, Shi et al. [39] present the superior potential function and superior repulsive potential function. In [40], by using potential field strategies such as oscillation and conflicts, the existing problems in the navigation of the mobile robot has solved. They also developed an enhanced version of the potential field in order to reduce oscillation and conflict when the target is near to the obstacle. Also, in the presence of obstacles, they provide rotational force to generate a better trajectory. In [41], the improved potential field approach to avoid the oscillations problem by Biswas et al. is presented. In this paper, the comparison is made between the two approaches i.e. traditional and Levenberg-Marquardt approaches; it has been claimed that the solution based on the Levenberg-Marquardt approach is better than the conventional approach. The proposed approach reduces the oscillation and a trajectory without collision is provided.

By using the potential field approach in [42, 36], the problem of multiple mobile robot navigation has overcome. The usage of the Finite Element Method (FEM) along with the potential field is reviewed in [43]. The approach presented converts the problem of navigation into the problem of electrostatics and then the FEM solves it. The ROBOPATH simulation tool has been utilized by Pradhan et al. [44] to evaluate the validity and applicability of the potential field approach. Also, this approach has been applied to the multi-robot navigation in different environmental conditions, and due to using a collaboration strategy without collision generates better solutions.

Using an energy-based approach known as an artificial potential field for Multi-Robot Systems (MRS), a path planning approach based on global offline is introduced. A path planning technique based on the developed artificial potential field is hosting on the basis of the potential field and is more efficient in finding the shortest path [45]. Another potential field approach used the cinematics of a six-wheel rover for movement on rough 3D terrain where the comparative value of the paths is derived from four different cost functions in terms of energy, traction force slip, and deviation from a straight line. Wide analyses and tests show that this technique is successful in finding trajectories [46].

2.2.2. Approaches of Computational Intelligence

Recently, approaches of computational intelligence have been recognized as common strategies for navigation of the mobile robot compared to conventional approaches. It also possesses a wide capability to deal with environmental uncertainty. In a known and unknown environment, there are many advanced techniques for navigation of the mobile robot. Parhi has categorized navigation strategies in [47] to perform the comparative study of advanced approaches such as the hybrid approach. Various computational intelligence approaches are mentioned below.

2.2.2.1. Genetic algorithm

A genetic algorithm is an evolutionary approach that was first introduced by Bremermann [48] in 1958 but was first used by Holland [49] in 1975 in the field of computer science. To optimize a specified function for minimum or maximum operation, this computational evolutionary algorithm is used. To determine and find the most feasible solution, the introduced methodology is based on the biological process of reproduction and the natural selection mechanism. The randomness existing in the GA is to be defined by one in the process of evolution to handle the randomization. The genetic algorithm is more effective and robust than the algorithm of random search without any additional information on the mentioned problem. GA's key characteristic makes it easier to find a solution where all other optimization strategies are unable to perform due to continuity and linearity deficiencies. One of the meta-heuristic algorithms is GA which is used to optimize the problem as a powerful and robust tool. It is more effective than the other approaches to optimization by finding global optimization and high parallelism.

In 1975, Holland [49] has introduced a principal definition of GA for the problem of optimization based on Darwin's theory of fittest survival. It has been then widely developed in the research areas of mobile robots. In a static environment with a polygon obstacle in [50], Shibata et al. have suggested a path planning strategy based on GA. Shing et al. [51] have introduced a path planning for the mobile robot in real-time with using the search strategy based on GA, in the same year. Xiao et al. [52] have presented an effective strategy of path planning based on GA in an unknown environment that satisfies the goals such as the optimum length of the path, smoothness of the path and avoiding obstacles. Kang et al. [53] have introduced a solution to the dead-end problem of navigation of the mobile robot. By using GA, they avoid appearing the robot in a complex-crowded environment. The algorithm starts online training to find a suitable chromosome when the robot trapped in the given circumstance and helped the robot to get out of this condition.

Shi et al. [54] have introduced a path planning approach with the dynamic obstacle in an unknown environment. Also, the genetic algorithm uses along with other methods to obtain a hybrid method to path planning. To path planning of the mobile robot, a hybrid navigation approach based on GA and fuzzy has been developed by Pratihar et al. [55]. In this approach, fuzzy logic is employed to avoid the collision and the genetic algorithm is used during navigation to optimize the path. The GA analysis for mobile robot navigation has been introduced by Hui et al. [56] which is combined with the fuzzy logic, neural network, and is compared to the potential field approach. The result shows that the fuzzy-GA and fuzzy-neural approaches have a better performance than the potential field in terms of optimal time.

The algorithm proposed is more robust and adaptive. Wang et al. [57] have used the GA with PSO as a useful combinational tool in the welding operation for robot path planning. The combination algorithm increases the particle variety and capability of a global search, as a result, it leads to generate a welding path without collision and improve the efficiency of welding. Kala [58] has introduced the consecutive application of GA for the problem of combinatorial optimization. A multi-robot path-planning algorithm has also implemented to locally search in a limited computational infrastructure.

Liu et al. [59] also have solved multi-objective optimization problems with GA. In this method, by implementing novel operators such as repair, deletion and cutting, the energy consumption and idle time are reduced. to navigate the mobile robot in big configuration space, Kuyucu et al. [60] have enhanced GA which the goal is to present the multi-objective task with the hybrid different mechanism such as seed-inspired by genetic transposition, a robust crossover, and optimization of multi-objectives.

On the other hand, Yang et al. [61] have presented a navigation strategy for the multi-mobile robot in a dynamic environmental map. In many research works, some improvements to traditional GA due to the slow rate of convergence, lack of cooperation between population, and local optimal have been introduced. Hong et al. [62] have developed the enhanced version of GA for global mobile robot path planning. In this research, the co-evaluation process is used for collaboration among the population of multiple mobile robots until the collision between robots is avoided and obtained a proper path during navigation.

The proposed algorithm is tested on the multi mobile robot with different static obstacles and it is claimed that the obtained result is better than basic GA. A novel type of GA for the problem of a traveling salesman by taking into account the dynamic target with respect to time has been introduced by Carlos et al. [63]. By using a simple prediction, the robot can reach the target and find a near-optimal solution. To gain effective path planning in a 2D environment, the adaptive GA by Karami et al. [64] is introduced. To prevent premature convergence, an adaptive selective operator has been replaced. This helps in keeping the population diversity in the obtained solution and improving the solution's quality.

2.2.2.2. Fuzzy logic

Zadeh [65] has introduced Fuzzy logic in 1965 that is almost utilized in major fields of research and engineering with a high degree of uncertainty, complexity, and non-linearity such as data classification, decision making, pattern recognition, and automatic control. The fuzzy logic theory is inspired by the significant ability of the human to inferring with knowledge based on perception. Fuzzy rules-based logic presents a systematic strategy for the linguistic rules which have obtained by reasoning and decision-making with unknown and ambiguous knowledge.

The problem of navigation based on fuzzy logic rules is divided into simple components that each component has consisted of a series of the rule of fuzzy logic to get well-defined goals set.

- This approach is utilized to model uncertain systems with uncomplete and unaccurate knowledge by using logical inferring in decision making.
- This approach generates a crisp result based on initial knowledge that is unclear, unknown, noisy and ambiguous.

A solution to the real-world problem and the knowledge-based on a make-decision process with mathematics theory development is introduced by Zadeh in 1965 [65]. The fuzzy logic-based mobile robot could exactly navigate in an unknown environment by using the collected data by the sensor and being able to make its decisions. The collected data by sensor plays an essential role to avoid collision with the obstacle and build a map of the environment. The accurate map of the environment is applied to navigate point-to-point, localization of the robot, diagnosis of landmark and planning of the trajectory.

In order to achieve successful navigation, a map modeling method is used to identify and learn the world by the facts. The behavior of the robot is developed using the fuzzy rules set that combine the numerical information obtained by sensors and the human experts' linguistic information [66]. The rules of if-then and [67] and engine of inference are the main elements of the controller based on fuzzy logic which models the behavior of mobile robots. To obstacle avoidance, a controller based on fuzzy is presented by Zavlangas et al. [68].

For the navigation of an omnidirectional mobile robot, a system based on Sugeno fuzzy along with the functions of triangular and trapezoidal is introduced. Due to requiring the knowledge of an expert and interference of humans, an impressive generation of fuzzy rules is the major problem of navigation. In [69], the automatic generation of the fuzzy rule using human and machine learning techniques is presented by Castellano et al. Some algorithms [70] such as a genetic algorithm and neural networks are used to automatically generate the fuzzy rules [71]. Also, fuzzy methods are successfully used in unstructured static [72] and dynamic [73] environment by preventing continuously loop building and backtracking.

A fuzzy approach for reactive mobile robot navigation is developed to solve trap-like situations and stuck in the loop by Motlagh et al. [74]. Park et al. [75] have presented two fuzzy controller approaches in the complex-crowded environment. The first approach in this paper uses the behaviors of avoiding the obstacle, searching target, and following the wall while the second approach navigates the mobile robot to keep its away from trap like circumstances. The narrow passage navigation problem is presented by Huq et al. [76].

A fuzzy controller which works with minimum rules to plan a path for the curved trajectory is developed by Moustris et al. [77]. The fuzzy logic approach has been applied in sensor-based hybrid navigation to enhance the perception of the new environment [78]. By using a modified fuzzy associative memory, the technique of sensor fusion to develop navigational principles to carry out practical tasks by Parasuraman in [79]. In the complex crowded environment, although navigation of the mobile robot needs a large space of input for matching the knowledge of the environment, in addition, it is required for optimizing the number of rules. While the proposed method has provided the multiple input space and has also reduced the number of rules. the subsumption approach for the wheeled mobile robot based on the fuzzy rule principle has been introduced by Flanagan et al. in [80] so that the initial environmental model is not needed for navigation. In this paper, the “sense-plan-act” rule is implemented that allows the robot to generate random behavior such as suddenly response.

To obstacle avoidance, they have also used the shadow appearing in the sensory region. Khatib et al. [81] and Lee et al. [82] have solved and developed a mobile robot navigation problem in a dynamic environment. The authors have introduced fuzzy as an approach based on data which is used to path planning with dynamic obstacles. To navigate multiple mobile robots in an uncertain crowded environment, a cooperative approach is presented by Hoy et al. in [83]. In order to have successful navigation, they have used the auxiliary controller based on fuzzy with limited abilities of sensing and communication for a static environment.

To enhance the mobile robot abilities for changeable conditions, a stereovision-based mechanism with fuzzy logic is introduced by Kang et al. [84] and Al-Mutib et al. [85]. In order to track the moving objects, the fuzzy controller based on Mamdani is planned for a wheeled mobile robot by Abadi et al. [86]. In the proposed method, to select the best parameter, they have used the PSO algorithm and fuzzy logic as a combination approach. Castillo et al. [87] have addressed the efficient performance of fuzzy logic in order to preserve the diversity in the ant colony algorithm and to prevent premature convergence.

Toloue et al. [88] have used the neural network to develop the application of type-2 fuzzy. This method is implemented as a combination approach for the parallel mobile robot in order to handle the ambiguity of a higher level. Authors have proposed an effective solution with low computing cost to control the location of the parallel robot based on 3-Prismatic-spherical-prismatic in comparison with conventional methods. This approach ignores the pruning phase of nodes and keeps important principles when appropriate. Rami et al. [89] have introduced a path planning strategy for the multi-mobile robot. By using a controller based on probabilistic fuzzy and the neural network, they activated motion coordination between robots. In the given methodology, the follower robot follows the leader robot's position. They have used 2 level controllers. To navigate the leader robot, A fuzzy system based on the first-order Sugeno is developed in order to achieve a high-level controller while a low-level controller is implemented for the follower robots. To control the heading angle and speed of the robots, a two-layer controller based on fuzzy logic has been utilized by Fu et al. [127] to reduce the uncertainty of the environment.

2.2.2.3. Neural network

The neural network is an intelligent system and mathematical tool that has been used in different domains such as computation, medicine, engineering, economics, and many others. The artificial neural network works somewhat like the brain’s human. The neural network consists of a number of basic and interconnected computational components called neurons, clustered into a structured graph topology and comprised of several successive layers that interconnected by a series of synaptic weight links. Synaptic weights are frequently associated with different numerical values so that allow the ANN to change and adapt its behavior on the basis of the problem is being addressed. The architecture of the neural network is illustrated in Figure 2-9.

The first artificial nerve presented in 1943 by Warren McCulloch (Physiologist) and Walter Pitts (Logician) [91], but the available technology didn't allow us to have more progress. The neural network was applied to create a network to compute the logical tasks and pattern recognition until the 50th century. However, the neural networks were introduced as a wide challenge due to the advent of faster computers and the exploration of novel skills of neural network and learning algorithms in 1980.

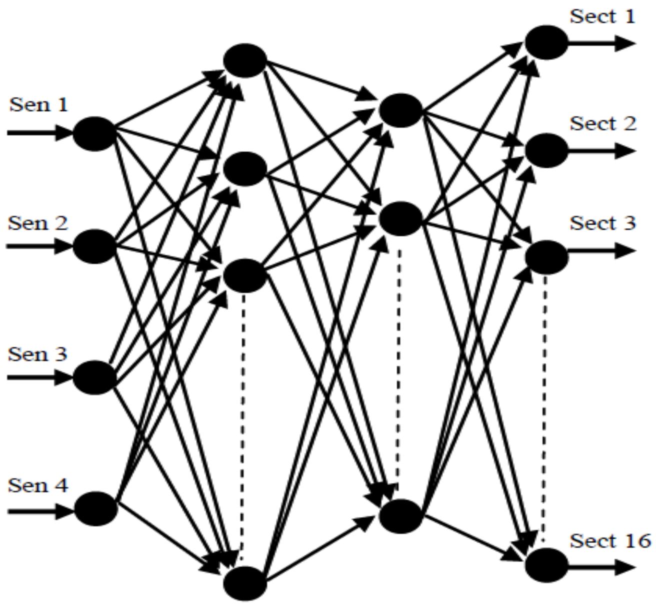


Fig. 2-9: The architecture of the neural network [99].

The input layers connect to hidden layers for specific operations in the neural network through a framework of weighted connections. To produce the appropriate response, the hidden layers link to the output layer. Features including the capability of generalization, enormous parallelism, distributed displaying, the capability to learn, neural network fault tolerance is also provided for mobile robot navigation. Janglova [92] has introduced the use of the neural network in an uncertain environment for wheeled mobile robot navigation. To generate a free-collision path, two mechanisms based on the neural network are used. The first mechanism tries to find free space by using the sensing data from the environment and the second mechanism finds a safe path without any collision.

The mobile robot navigation based on cellular neural networks in a semi-structured environment is presented by Siemiatkowska [93]. In this network, two hidden layers are used. The signal sent by the map cells and neurons as the target location is considered as the input of the first layer and to reach real neurons, the current location is enabled. The shortest path for the robot from the start point to the target is identified by the second layer. The third generation of neural networks namely Spiking Neural Network (SNN) for obstacle avoidance is introduced by Wang et al. [94]. This approach is used to solve the nonlinear classification and high dimension clustering problem. The automatic learning to avoid human intervention in the navigation process is introduced by Qiao et al. [95].

To achieve the navigation mission during the training without human intervention, the characteristic of the proposed approach is which according to the environmental complexity the neural network sets the insertion and deletion operations of new hidden layers. The use of a neural network is provided for the technique of Fast Simultaneous Localization and Mapping (Fast SLAM) by Li et al. [96]. This approach removes the accumulation of errors generated by the wrong model of odometry and the incorrect linearization of the nonlinear function of SLAM. The use of neural network-based fast SLAM has improved the mobile robot navigation which finds a free collision path in an uncertain environment. Yong-Kyun et al. [97] have provided a neural network model with a potential field method that is applied for achieving cooperative and competitive coordination for control-based behavior. The proposed neural network-based approach categorizes the environment based on sixteen prototypes of the topological map that characterize the navigation of the local environment. While the potential field approach chooses the correct behavior. By using this hybrid method, the mobile robot keeps itself away from trap-like situations.

Also, a combination approach based on neural network and fuzzy is introduced to navigate the mobile robot in the complex-crowded environment along with static obstacles [98]. A new combination approach of neural network and fuzzy logic is proposed to navigate the mobile robot in [99]. The neural network approach obtains the optimal number of the rules of activation to decrease computation time in real-life implementations. To navigate the mobile robot, the Sonar-based neural network application has been introduced by Pal et al. [100]. Using ultrasonic sensors, Medina–Santiago [101] have offered a control system based on a neural network for navigation of the mobile robot in real-time. With the use of GPS and compass sensors, Capi et al. [102] have introduced real-time navigation in urban areas using the neural network. To enhance the efficiency of the neural network, Syed et al. [103] have changed the fundamental neural network to type GAPCNN which achieved the parameter of convergence for the mobile robot in the static and dynamic environments.

The proposed approach has been updated by implementing directional auto wave control and increased activation neurons based on the adaptive threshold technique, and the robot must also consider the overall environmental dynamics. Hendzel [104] has suggested the neural network-based path planning approach which is used the given network to avoid collision with the obstacle and combinational actions such as "following the wall," "searching the target", and "attaining the center of the collision-free space". The self-learning approach based on neural network is offered by Markoski et al. [105]. Quinonez et al. [106] have proposed a pattern recognition approach as a mechanism for navigation of the mobile robot in an uncertain environmental map.

2.2.2.4. Particle swarm optimization

The particle swarm optimization (PSO) is a meta-heuristic approach which is inspired by the animals' social behavior such as fish schooling and bird flocking. It has been developed in 1995 by Eberhart and Kennedy [107] and is an optimization method for solving the various science and engineering problems. While PSO mimics the animals ' social behavior that needs no leader to achieve the goal position. When the bird’s flock moves to seek the food source, they don't need a leader and move with one of the members who are close to the food source. Therefore, through suitable communication with the population's members, the birds flock obtains their needed solution.

The PSO method includes a group of particles which each particle is a possible solution. Also, this algorithm is widely applied in the navigation of the mobile robot. Tang X. et al. [108] have shown problem navigation of the mobile robot by considering mapping and locating in an unknown environment. The proposed approach has used the multi-agent particle filter that minimizes the computations and keeps more stable convergence characteristics. Xuan et al [109] are utilized the PSO algorithm along with the Mesh Adaptive Direct Search algorithm that finds an accurate path and keeps the robot away from the local optima. This hybrid method generates efficient results than GA and Extended Kalman Filter.

To represent the dynamic and time-dependent constraints of the navigation of mobile robot, Atyabi et al. [110] have suggested a development of fundamental PSO named Area Extended PSO. The proposed approach is used in the seeking of rescuing survivors and disarming the bomb. In the complex environment, Tang et al. [111] have proposed cooperative path planning using PSO to handle multi-mobile robot navigation. Couceiro et al.[112] have introduced several improvements to navigate the multi-mobile robot in real-life applications.

They changed the PSO and Darwinian PSO (DPSO) system in order to avoid obstacles and communicate with each other. The proposed algorithm is tested on 12 physical robots and the obtained results are near to global optimal. By using a multi-category classifier, Chen et al. [113] have provided capability based on learning for the unknown environment with the control strategy of the human experts. For this purpose, PSO is utilized in a short time to achieve higher accuracy. Compared to conventional grid search, the proposed algorithm has higher classification effectiveness without the convergence of prematureness.

Das et al. [114] have proposed a hybrid approach for effective path planning. In this approach, the implementation of PSO and the enhanced gravitational search algorithm is illustrated as a combination technique to determine the optimal path planning in the complex and crowded environment for the multi-mobile robot. By implementing the co-evolutionary methodology which updates IGSA acceleration and PSO velocity, the combination approach makes the equilibrium between exploitation and exploration. He et al. [115] have suggested the use of particle swarm optimization for the underwater robot. The PSO-UFastSLAM approach is defined in order to obtain a better approximation precision and to minimize the particles to achieve better results.

A PSO based online CCPP algorithm [116] is presented in [117] which is based on the high-resolution grid and successfully finds a smooth path with minimal cost for coverage. Pessin et al. [118] have compared PSO and ACO for garbage collection and recycling problem in a multi-robot scenario. PSO performed well in the respective scenario due to its quick convergence than ACO. Another PSO based partial search algorithm is proposed in [119], shows better results in terms of time efficiency than classic PSO and GA in solving TSP. PSO is also used with other EAs for faster convergence. Two PSO variants, Fuzzy Ant Supervised by PSO (Fuzzy-AS-PSO) and Simplified Ant Supervised by PSO (S-AS-PSO) are proposed in [120] to solve the ACO parameter adjustment problem. The proposed algorithms showed good performance in solving the TSP problem than existing methods.

2.2.2.5. Ant colony algorithm

The algorithm of the ant colony is an intelligent algorithm has developed in 1992 by Marco Dorigo [121]. This algorithm is an approach based on the population applied to solve the hybrid problems of optimization. It is inspired by the ant's behavior and its capability to seek the shortest trajectory from its nest to the source of food. It has been used in different science and engineering fields such as the problem of job-shop scheduling problem of vehicle navigation, the problem of assignment problem of setting and more. Due to its high capability to solve the real-time problem, ACO is applied to handle the navigation problem of the mobile robot to obstacle avoidance and successful path planning.

Guan-Zheng et al. [122] have introduced the implementation of ACO for real-time path planning of the mobile robot. ACO adoption increases convergence rate, diversity in solution, computing performance, and dynamic convergence compared to other algorithms such as GA. Using ACO, Liu et al. [123] have implemented multi-robot navigation. The obstacle avoidance strategies are provided in this approach for different robot systems in the static environment. To enhance the selective approach, they used the specialized method. When the ant reaches the corner of deadlock, the penalty function will be performed to modify the trail intensity to avoid the robot's path deadlock. A combination approach for multi-robot navigation has been introduced by Castillo et al. [124]. In the proposed algorithm, the combination effect of ACO and fuzzy logic has been considered. The fuzzy logic approach plays a significant role in the control of diversity in the ACO.

The main objective of the suggested approach is to avoid fully convergence by the dynamic differences of a specific parameter. Purian et al. [125] have offered the implementation of the ACO approach to navigate the mobile robot in an unidentified dynamic environmental map. The authors have utilized the ACO to select and optimize the rules of fuzzy.

In order to solve planning problems with multi-goal and obstacles, an algorithm is introduced in [126]. In the mentioned paper, ant colony optimization is associated with a point-to-point path planning algorithm based on sampling. For performance evaluation, a numerical comparison with two present algorithms based on sampling has been presented. A mechanism to solve the SLAM-Problem by implementing an optimization algorithm based on the ACO is proposed in [127]. A data structure like the tree is produced, where the trajectory from the root to the leaf shows a metric map approximation. The optimal path is achieved by the use of the ACO.

In [128], X. Chen et al. have suggested a two-stage framework of ACO capable of overcoming the main problem of imbalance between premature convergence and the optimum trajectory. An efficient ACO based-path planning strategy is implemented in [129]. The robot is taken into account as a point that locates in an exact cell in a discrete display of the environmental map. The path length and the complexity of navigating must be considered in the cost function evaluated by a fuzzy logic mechanism. The methodology has the flexible ability to change in the environmental map so it can design a global trajectory planning for the mobile robot with moving obstacles.

Hoff et al. [130] have proposed two robot foraging algorithms based on ACO that make it possible for the coordination between robots. This technique uses direct interaction between robots instead of using environmental markers. They assume which robots have restricted the abilities of sensing and interaction and there is no clear global positioning. ACO has also been improved for the localization of multiple source odors [131].

The suggested algorithm has improved by ACO consists of three steps: local search, global search, and pheromone upgrade. The first two are additional search phases in comparing with the original ACO and enhance the robot search efficiency. In target tracking the ACO methodology has been utilized to the multi-dimensional assignment problem. A new information association methodology based on an enhanced ACO technique is implemented in [132]. ACODA models have considered a measurement as an ant, a track as a city and the problem of information association as the food position of ants inspired with how ACO is implemented to the problem of the traveling salesman [133].

2.2.2.6. Simulated Annealing

Simulated annealing is a simple and effective method of meta-heuristic and random search that is utilized in large search spaces for solving optimization problems. When the search space is discrete it is most widely employed. This algorithm has been introduced by Kirkpatrick, Gelette, and Vecchi in 1983 as a probabilistic approach. Cerny had used it to determine the global minimum of an objective that could have multiple local minima. In this method, a comparison between the cooling and freezing of metal into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a general system has used; This provides a fundamental of an optimization technique for the hybrid and other problems. It is often used for nonlinear problems. SA technique is similar to a bouncing ball that can jump on mountains from the valley to the valley known as the global maximization problem.

Samuel et al. [134] have introduced a new model based on a few simulated annealing to optimize global continuous variables. This paper has used methods which are composed of several simulated annealing processes with coupled acceptance probabilities. A new simulated annealing algorithm has been introduced by Erken et al. [135] to optimize the size of a PV / wind integrated combined energy system with battery storage. The main aim of the objective of this paper is to minimize the total cost of the combinational energy system. The variables of decision and the wind turbine rotor swept the area and the capacity of the battery is shown in the PV scale. The simulated annealing was used by Kuo[136] to find vehicle navigation with the lowest fuel consumption. The proposed method calculates the total fuel consumption for the problem of the vehicle's time-dependent navigation which velocity and driving time depending on driving time.

Miao et al. [137] have presented an enhanced simulated annealing approach used for the path planning of the mobile robot in dynamic environmental maps with static and moving obstacles. The proposed approach consists of two additional mathematical operators namely switching and deleting operators. The proposed method has been compared with the standard simulated annealing. The results indicate that the performance of the mobile robot is enhanced in terms of path length and processing time.

Behnck et. al [138] have suggested a novel approach based on simulated annealing to path planning of Unmanned Aerial Vehicles. This approach is solved by an m-TSP problem. The obtained results are demonstrated that the proposed method is able to calculate minimal trajectories for two UAVs that performing on an area and try to match the types of points of interest and UAVs specialties. Ganeshmurthy et al [139] have used a heuristic method that searches the feasible path. The proposed algorithm has been combined with simulated annealing approach for dynamic path planning of the mobile robot.

The resultant path has enhanced the runtime and offline path planning with the hybrid approach. Turker [140] has used the simulated annealing technique to achieve an almost optimal path in a limited 2D radar environment. In addition, a simplistic hazard avoidance approach was introduced and used in the solution generated by SA to keep away from standard circular radar attacks. The results indicate which the suggested approach is provided with suitable solutions in a reasonable time period, based on its ability to avoid local minima using Metropolis approval principle. By using threat avoidance, the best solution is obtained and the resulting path is simple and free- threat.

2.2.2.7. Bee colony algorithm

Bees Algorithm (BA) [141] is a meta-heuristic algorithm that models the bee colonies' foraging behavior to find the closest food source. The artificial bee colony initially presented by Dervis Karaboga in 2005, it has inspired honey bees ' intelligent behavior to overcome multi-dimensional and multi-modal optimization challenges. This methodology is an optimization mechanism that has provided a search procedure based on the population. The main goal for bees is to explore food source locations with high nectar. In the artificial bee colony approach (ABC), the bee colony comprises of three categories of bees: employed bees, onlookers and scouts. Bees of employed identify the source of food within the range of the source of food in their memory. They then share their knowledge with the onlooker bees in the hive, and the onlooker bees choose one of the sources of food in the range of the sources of food selected by themselves. The employed bee, which its source has been left, becomes a scout and begins to search for a new source of food at random.

Tereshko has considered the bee colony method as a dynamic system to receive data and knowledge from the environment and to adjust its behavior on the basis of the data collected [142]. To overcome the complex traffic and transport problems, a system based on bee swarm intelligence is suggested by Teodorovic in [143]. He has used the bee colony optimization to solve deterministic hybrid problems, and also uncertain hybrid problems [144]. Drias et al. have introduced a novel method based on bees swarm optimization and modified its characteristics based on the total weighted satisfiability problem [145].

Benatchba et al. have also presented a metaheuristic approach to overcome a 3-sat problem based on the bee propagation process [146]. Wedde et al. have introduced a new BeeHive navigation algorithm inspired by interaction and evaluation strategies and processes of the honey bee. In this proposed approach, bee agents move via network areas named foraging zones. Bee agents navigate via network areas named foraging zones. On their path, the network state information is updated based on local navigation tables [147].

A new artificial bee colony-based chaotic strategy to navigation problem of the Uninhabited Combat Air Vehicle (UCAV) has been suggested in [148]. Using the ergodicity and irregularity of the chaotic variable enables the basic ABC to left the local optimum and accelerates the procedure of seeking the optimal parameters. The result demonstrates which the proposed approach is practical in the dynamic environment and it is easy to incorporate pop-up attacks. The ABC algorithm has been used in [149] to reduce the path and explore the optimal solution in the problem of the traveling salesman. The optimal distance based on ABC Algorithm is shorter and error-free. In a 2-D workspace [150], a mobile robot global trajectory planning strategy is introduced based on Directed Artificial Bee Colony (DABC). A method to navigate the bees towards the target is proposed in this paper by evaluating the fitness enhancement and saving the suitable path to navigate more bees in that path.

A path planning of the mobile robot is provided in [151] relying on ABC optimization. In order to minimize the path length for all robots, the robot calculated the mobile robot's movement path from the starting position to the target on the environment. To obtain optimally of the next position of all the robots, a local path planning schema based on ABC optimization is developed.

2.2.2.8. Firefly Algorithm

Firefly approach is presented in 2008 by Yang [152] which is inspired by the fireflies' brightness behavior. It is known as a meta-heuristic approach. The objective is the random states and general classification as the firefly's trial and error that occurs stochastically in natural conditions. Firefly is a winged beetle of the Lampyridae family. It is often known as lightning bugs because of its capability to generate the light. Through an oxidation process of Luciferin with Luciferase enzymes, light generates and this operation occurs rapidly. This cycle of light generation is recognized as bioluminescence and this light is used by firefly to shine without losing thermal energy. Fireflies are utilized the mentioned light to select the mate, share a specific message and even scare other animals wanting to eat fireflies.

The FA is recently used as an optimization technique and its use is expanding across almost every field of science and technology. Most researchers use the capability to self-planning, self-adaptation and self-organize FA for different optimization problems such as detection of robot faults [153], dispatched problem of economic emissions [154], optimization of performance and redundancy [155], optimization problem of mixed structural variables [156], the problem of cooperative networking [157], the problem of hybrid optimization [158], the problem of presentation learning [159], the problem of the dynamic environment [160] and so on.

To enhance the efficiency of the firefly algorithm, some researchers have used the Gaussian distribution function for improving convergence speed [161], some have also upgraded FA for avoiding the sudden movement of the firefly algorithm when there is no brighter firefly [162]. This algorithm has the ability to search an optimal solution in science and engineering fields, therefore, it is considered as an effective algorithm [163]. In [164], a new solution based on FA is presented to solve the multi-objective problem by considering nonlinear constraints. In addition, Baykasoglu et al. [165] have applied FA to provide a solution to the complex environments' real-life problem.

The proposed algorithm is tested to solve multi-dimensional knapsack problems in static and dynamic environments. The result obtained demonstrates the efficiency of the proposed approach is better than the genetic algorithm and differential evolution. It is used as a hybrid approach due to the efficiency of the firefly algorithm. A hybrid approach to solve the classification problem is introduced by Alweshah et al. [166]. The firefly algorithm in this paper is hybrid with a simulated annealing algorithm, FA with Levy flight (LFA), and a hybrid of FA and SA, and Levy flight (LSFA). Such combinations have been implemented with the objective of achieving an optimal balance between exploration and exploitation in acquiring relatively close-optimal weights for the probabilistic neural network, optimizing classification accuracy and obtaining a quick convergence rate for classification problems.

By using FA, Maheshwar et al. [167] have enhanced the genetic algorithm's performance. This approach participates in a generation process of the chromosome population, it obtains global optimal as a primary state and keeps the system away from the local minimum. To solve the continuous optimization, the hybrid of the firefly algorithm and PSO is presented by Zouache et al. [168]. Baykasoğlu et al. [169] have improved the problem of a nonlinear approach. The authors have applied adaptive FA to optimize the problems of mechanical design. A chaotic map with FA has been used to present a solution to a nonlinear problem. This algorithm has been adapted to navigate the mobile robot [170].

Hidalgo-Paniagua et al. [171] have introduced a new FA-based algorithm for navigation of the mobile robot with a static obstacle. By this new algorithm, they have met the goals such as length of the path, smoothness of path and safety of the path. The FA is used for a single robot in the environment to determine the shortest free collision path [172]. Also, FA is an underwater mobile robot being tested by Sutantyon et al. [173]. They have implemented a scheduling technique for agent robots for avoiding interference and jamming in the maritime condition. Sutantyo et al. [174] have presented real underwater navigation based on a levy flight-firefly approach that has partial knowledge of the environment.

The new firefly-based path planning approach has been introduced in [175] which enhanced solution quality and speed of convergence by adjusting the randomization and light absorption variables. Additionally, a new updated FA version has been presented for solving the problem of path planning for uninhabited combat air vehicles [176]. A correction strategy has been applied to share knowledge between top fireflies during the light intensity update process [177]. By using the scheme of the multi-swarm population [178], FA could be utilized to respond quickly to the change in a dynamic environment. By considering the FA control variables, it is faster to adapt to the changing environment. In [178], the firefly algorithm is hybridized with learning automata to adjust the control variables for dynamic environments.

2.3. Discussion

A comprehensive literature survey of mobile robot navigation is provided in this chapter. The review is based on classical and Artificial Intelligence (AI) based approaches. It is obvious that artificial intelligence approaches have been preferred to classical ones. The high computational ability and applicability of AI is the reason for the selection of AI approaches for mobile robot navigation. Nowadays, most of the researches are based on AI approach while a few percentages of researches is performed by classic methods. The application of AI approaches is increasing in the science and engineering fields day by day. Figure 2-10 shows the percentage of using AI strategies for navigation of mobile robots in literature. However, some of AI approaches have little applications on mobile robot navigation.

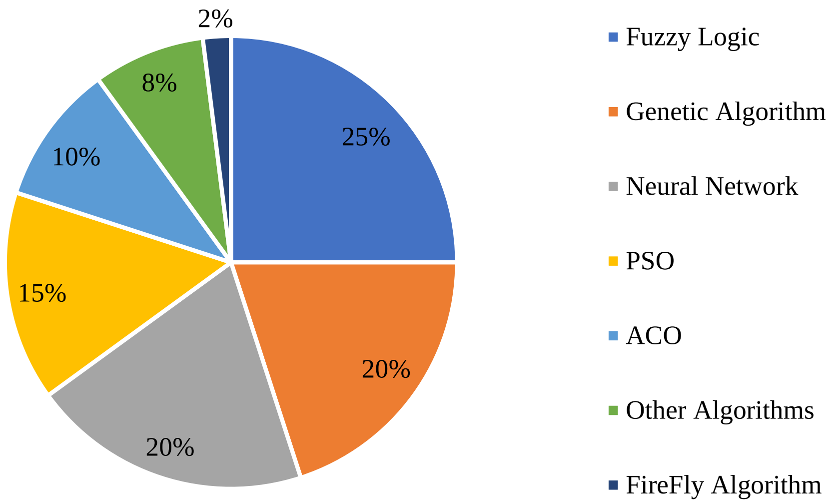


Fig. 2-10: Percentage of the investigated manuscript using AI approaches to navigate the mobile robot.

2.4. Summary

A detailed literature review of mobile robot navigation has been provided in this chapter based on classical and artificial intelligence-based approaches. The summary of this chapter is as follows:

- The mobile robot navigation has been investigated based on classic and artificial intelligence strategies in static or dynamic environments.
- The navigation based on AIs is preferred to the classic approaches.
- The researches based on AI for mobile robot navigation is inspired by nature which is used in the complex-crowded and uncertain environments.

This thesis presents the new navigational approaches such as Grasshopper optimization algorithm (GOA) and the hybrid of GOA- Genetic algorithm (GOA-GA) to path planning in both of static and dynamic environmental maps.

3. BACKGROUND AND LITERATURE REVIEW OF AUTONOMOUS VEHICLES

This chapter provides a more comprehensive background and literature survey on autonomous vehicles. The outline is as follows. The first section, the brief introduction of the autonomous vehicle is presented. The second section, the architecture of the autonomous vehicle is studied. In the next section, the techniques of motion planning are described. Forth section presents MPC path planning along with the description of the vehicle models and obstacle models. The final section is a summary of this chapter.

3.1. Introduction

The most driving accident occurs due to driver errors [179]. The autonomous vehicle is developed in hopes of reducing the number of accident and by deleting the main cause of accidents, the driver. However, there are situations for the autonomous vehicle that the crash is imminent and inevitable. In such a situation, it is expected that the autonomous vehicle responds properly. The collision with a different obstacle have different costs in terms of injury and entered damage. Therefore, designers must consider these costs and plan a maneuver that avoided the obstacle collision in the autonomous vehicle. The second objective of this thesis is to design an autonomous vehicle platform that detects the obstacle and path and plans a free-collision maneuver with low costs.

3.1.1. Architecture of autonomous vehicle

Autonomous vehicle systems have different architectures. Figures 3-1 and 3-2 demonstrate the general architecture of the autonomous vehicle and a comprehensive perceive of the various architectures in the complex system of the autonomous vehicle and also how they relate to each other. On the other hand, the whole autonomous vehicle system consists of three different modules, each of which completes its own work. There are:

1- Sensing: To provide the environmental data which can be utilized to plan;

2- Planning: To produce reliable and practical trajectories by using information obtained from the sensing module; and

3- Control: To work the vehicle and follow the path determined by the planning module.

A simple architecture is presented for decision stage of an autonomous vehicle in [180] which is the combination of decision and control stages. In this paper, the vehicle receives data from sensors such as GPS/INS sets, LIDARs, vision cameras, and radars. Also, receiving data from infrastructures and other vehicles via communication sets is possible. The perception stage provides knowledge about the structure of the road, the regulations of road [181], obstacles [182], and vehicle states [183] based on the data obtained by sensors. The decision and control stages decide about the vehicle path. The vehicle path generated based on information obtained by perception stage and it causes to generate the actuation commands.

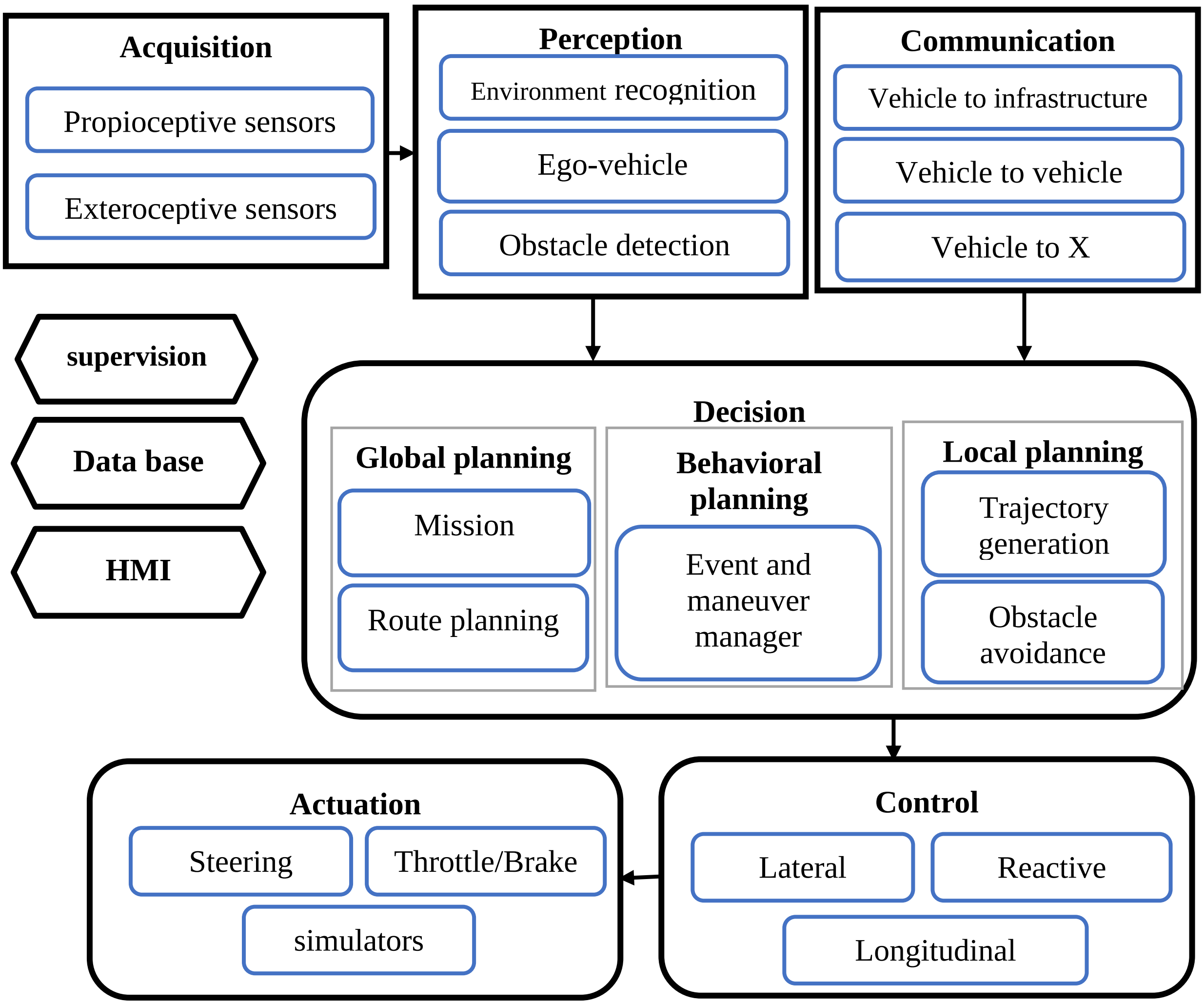


Fig. 3-1: A general architecture for autonomous vehicles [185].

The decision stage consists of the module of global path planning, the module of behavior planning, and module of path planning. The global path planning finds an appropriate path from the current status to the target status. The behavior planning determines the proper driving behavior such as stopping in a dangerous situation, changing lane, and following the route [186]. The local planning module plans the vehicle trajectory such that the vehicle carries out the planned behavior while avoids the obstacle collision.

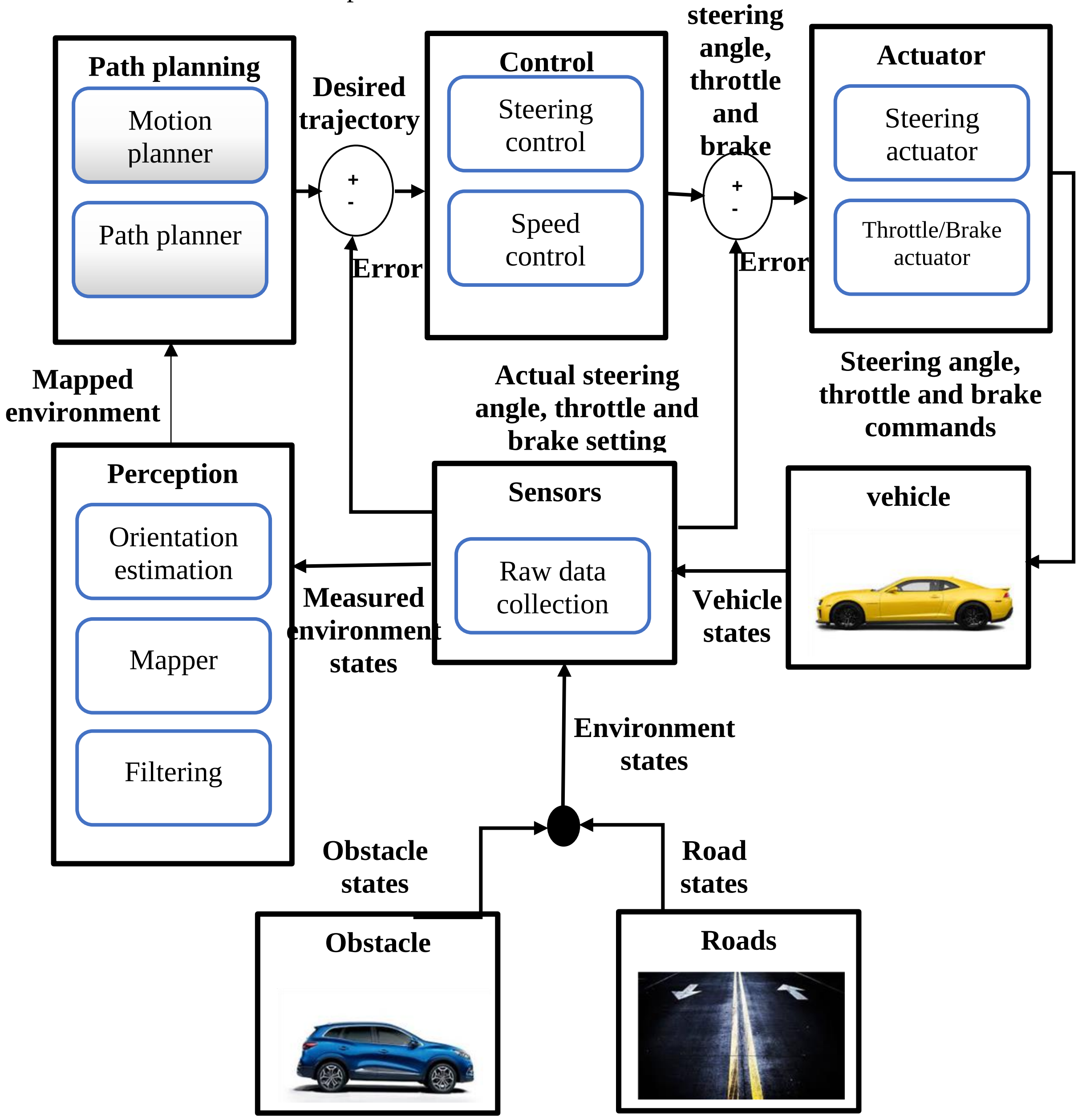


Fig. 3-2: An autonomous vehicle system's dynamic architecture.

Finally, the control module or path tracking module generates the actuator commands for the vehicle to follow the planned path while keeping the stability of the vehicle. The task of obstacle avoidance is performing in path planning module. Therefore, this module should plan the trajectory while minimizes the collision cost.

In this dissertation, a motion planning module is developed which is able to path planning with the lowest cost function. The module of path tracking considers the dynamic of the vehicle in order to preserve the vehicle's stability and enhance the vehicle's tracking performance [187]. If the dynamic of the vehicle and its constraints had not been included in the generating procedure of the path, the planned path would not be feasible to track the vehicle [188]. Therefore, a path planning technique should consider the vehicle dynamics and its constraints so that the generated trajectory be trackable by the vehicle. In addition, if the vehicle dynamics and its constraints are considered in motion planning approach, the technique of motion planning could use in the total capacity of the vehicle to reduce the crash cost.

If the task of the motion planning module covers the task of the path tracking, both modules of motion planning and path tracking can be combined. When the motion planning module considers the vehicle dynamics and its constraints in the process of trajectory generation and generates the actuator inputs as its outputs, the motion planning module can be used instead of both modules of motion planning and path tracking. In this thesis, the path planning module is developed.

3.2. Motion Planning Techniques

The module of motion planning should be able to plan a proper trajectory according to the structure and regulation of road, the configuration of obstacles, the dynamics of obstacles and the current states of the vehicle. As already mentioned, the module should consider the dynamics and constraints of the vehicle to find a feasible trajectory in the situation of an imminent crash. Furthermore, if the module generates the vehicle input, it can be used instead of both module of motion planning and path tracking.

Figure 3-3 demonstrates the autonomous vehicle's used motion planning strategies during the time. For autonomous road vehicles, motion planning strategies are defined that can be categorized as interpolating curve planners, graph search planners, sample-based planners, and optimization planners.

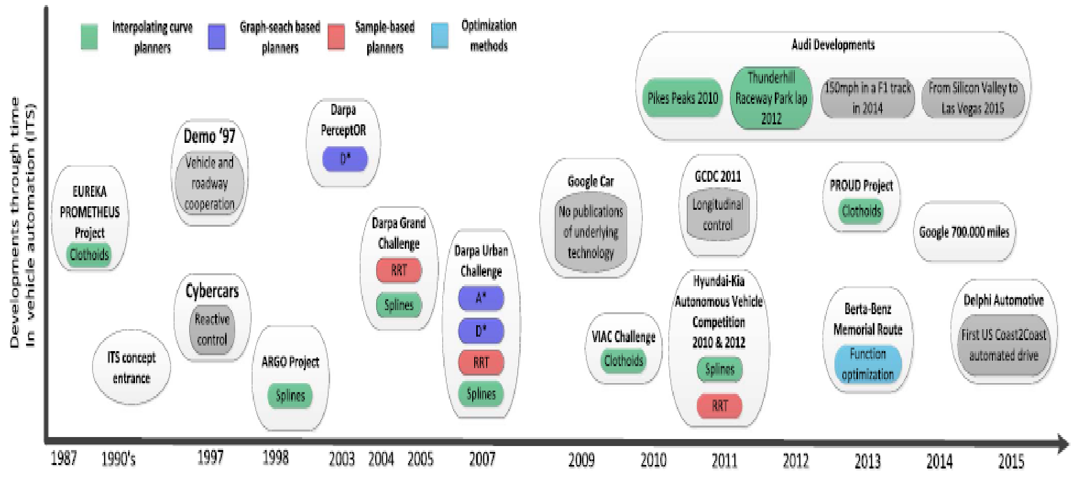


Fig. 3-3: The path planning techniques during the time and their associated motion planning techniques [185].

The planners of curve interpolation such as lines and circles [189], Clothoid [190], Polynomial [191], Spline and Bezier curves [192-193] are widely used for online path planning. These planners are similar to methods based on graph search and have low computational cost because the behavior of curve is defined by a few control points or parameters. However, the optimality of the obtained trajectory is not guaranteed and the vehicle's dynamic constraints are not considered during the planning process and are additionally needed a smoothing process for the obtained path. In [194], a new method is presented by Clothoid curve to reduce the length and curvature change of path. In this approach, two points are considered on the plane and the proposed algorithm generates a closed-form solution to connect two Clothoid sets for the position of a waypoint.

This approach reduces sudden changes in curvature and side-slip by the vehicle and enhances the movement's efficiency. To generate a trajectory in [195], the authors used the polynomial parameterization that represents kinematic constraints and moving obstacles. Besides, the velocity of the vehicle is planned using this parameterization. To figure out the optimal solution, guideline obtained by a Bezier curve is introduced. The result of simulation has shown that the proposed approach is better than the traditional one. In [196], the authors used a combination of RRT* and Spline techniques to generate a smooth path. The proposed bidirectional Spline-RRT* algorithm is based on the cubic curve and satisfies direction constraints for both start and target positions. This algorithm is not similar to other path planning algorithms and the obtained result for the vehicle is sub-optimal feasible.

The planners of graph-search generate a grid map and find out a trajectory from the start coordinate to the target coordinate through the grids. One of the simplest heuristic algorithms is the Dijkstra algorithm that is based on graph search and is able to find a minimum path between two different nodes by discretizing the environment [197]. The other algorithm is A* [198] and similar algorithms like D* [199] which is similar to the Dijkstra algorithm but it uses two cost functions to move from start position to target.

These algorithms are applied only on the environment with static obstacles. These algorithms guaranteed efficiency and optimality of obtaining path but the planned path depends on the resolution of the graph highly [200]. Moreover, considering the dynamic constraints of the vehicle is difficult during the planning process. Since the computational time of the planner is increased by increasing the problem dimension, this approach cannot be used in real-time if the dynamics states of the vehicle are considered.

The planners based on sampling sample the search state space and search the connection between samples. Rapidly-exploring Random Tree (RRT) [201] and similar planner such as RRT* [202] are the main planners of this category. In this approach, the planner can quickly find a trajectory between the initial and final states while is considered the vehicle dynamics. In this approach, finding an optimal path can take longer.

The main idea is to formulate motion planning strategy as an optimization problem that that takes into account the vehicle's desired efficiency and constraints. The optimization planners optimize an objective function of the vehicle states according to the constraints of states and input. The main technique of optimization planner is the Model Predictive Controller (MPC) technique. The vehicle dynamics is utilized as its models in this technique and the vehicle behavior in the optimization process of the objective function is predicted. It can optimize the performance of the vehicle by limiting the vehicle states and inputs and finding an optimal and free-collision path for the vehicle in a uniform manner. The MPC technique can be solved in a short time.

The Potential Field (PF) technique is another technique for motion planning which isn't usually utilized for autonomous road vehicles. The potential field approach is based on attractive and repulsive factors that allow the vehicle to travel towards the destination due to the attractive factor while the repulsive effect prevents from colliding the vehicle with obstacles. The target potential field has a minimum value at the target location when it attracts the vehicle, while the PF of obstacle has a maximum cose at the obstacle locations that repulses the vehicle from the obstacle. Wang et al. [203] used the exponential PFs for the lanes, hyperbolic PFs for the static obstacles and products of exponential and hyperbolic functions as PFs for moving obstacles. In research of Wolf et al. [204], different PFs has been presented to optimize the vehicle performance such as the quadratic PFs for lanes, hyperbolic PFs for the boundaries of road, and the exponential PFs for the vehicle.

In this dissertation, three different potential field functions have been used to avoid the collision. Also, the MPC technique is utilized as the technique of motion planning since this technique can systematically consider the future behavior of the vehicle along with the vehicle constraints, actuators, and obstacles. In addition, it can perform motion planning in a timely manner. It generates the inputs of actuators and the motion planning module by using this technique which can act as the both of motion planning and path tracking modules in a uniform manner. Here, the potential field is used as a method for obstacle avoidance.

3.3. MPC Motion planning

The research works of motion planning based on the MPC are reviewed in this section. The nature of motion planning based on MPC for an autonomous vehicle is non-convex. The vehicle lateral and longitudinal movements are used as the MPC model. due to nonlinear dynamic equations and nonlinear behavior of tires, it is nonlinear generally. The obstacle avoidance is performed either by adding potential functions in the cost function, which are non-convex or adding the constraints of obstacles which have non-convex nature. In some papers, the researchers consider the nonlinear nature of the problem and try to solve the nonlinear MPC. But according to the high computational cost of a nonlinear MPC, its use in the real world is difficult.

Therefore, some researches try to simplify the nonlinear MPC problem to the quadratic MPC problem which can be solved in the shortest time and be appropriate for real-time applications. The quadratic MPC consists of a quadratic cost function, a linear model, and linear constraints. The researchers work on finding a linear model for the vehicle by using linear constraints to predict the behavior of the vehicle as close as possible to the nonlinear model of vehicle. They also investigate a set of constraints of linear obstacle which construct a convex region including the most useful section of the non-convex obstacle-free region. In this thesis, a motion planning problem based on quadratic MPC is developed for real-time applications. In following, the vehicle models and obstacle avoidance methods used in the literature of motion planning based on MPC are investigated.

3.3.1. Vehicle model

The used vehicle model can be categorized for the motion planning of autonomous vehicle as point mass vehicle models, kinematic vehicle models, and dynamic vehicle models. The point mass vehicle models are the linear models which model the vehicle as a particle with the vehicle mass with ability to move in longitudinal and lateral directions. This model does not consider the tire model and geometry of the vehicle; therefore, it can make large tracking error. So, some state constraints can be added to create a feasible generated path. Lateral and longitudinal accelerations can be limited by the acceleration corresponding to the maximum tire force capacity [205]. In addition, the side slip angle of the vehicle cannot be large for a vehicle in a non-drifting maneuver and can be limited [205, 206]. A point mass model cannot adequately predict the vehicle's behavior even with these constraints.

Kinematic models are nonlinear models which model the vehicle based on its geometries. It does not consider the model of tire. Hence, to consider the comfort of the passenger and to avoid brake, constraints can be added on the lateral acceleration that limits the vehicle to the normal value driving [207]. Dynamics models of the vehicle consider the tire model on their model. Carvalho et al. [207] and Zhang et al. [208] have compared the open loop behaviors of vehicle kinematic models with the vehicle dynamic model. The results show that both models have the same performance in low velocities but when the maneuver has the steering angle larger than 1.5° [208], the dynamics model in higher velocities especially in velocities higher than 15m/s, performs noticeably better. Therefore, if an autonomous vehicle is considered to perform the high-speed maneuvers with large lateral accelerations, the dynamics vehicle models are preferred to the kinematic vehicle models to apply in MPC models.

Dynamics vehicle models are resulted based on Newton's second law by considering models of the tire as the forces of maneuver. The dynamics equations of the vehicle are nonlinear regardless of the model of the tire and can be said that the tire behavior is the main source of nonlinearity in the behavior of the vehicle. Tires have limited capacity and become saturated. The nonlinear models of the tire such as Pacejka and Brush tire models are used to model the nonlinearity in the model of tire. The linear tire models model the behavior of the tire in the linear area. When the linear tire model is valid, it usually accompanies a constraint on the side slip angle to limit the model in the linear region of the tire force.

Frasch et al. [209] have introduced a 4-wheel dynamic model of the vehicle along with lateral and longitudinal, yaw motion in the center of gravity of vehicle-based on 4 forces of the tire. In this research, the Pacejka model of the tire has been used and also the dynamics of the wheel has been considered in the model. To generate an accurate vehicle model, the load transfer from the lateral and longitudinal acceleration have been considered in the model of tire. The wheel dynamics increase the number of vehicle states by four but it has a very small effect on the accuracy of the vehicle model [210]. Gao et al. [205] have presented 4-wheel dynamics model similar to [209] without wheel dynamics. In this research, the load transfer effect is not taken into account in the model.

A four-wheel vehicle dynamics model can be simplified to a bicycle model regardless of wheel dynamics. The tires of each axle are modeled as one tire in bicycle model. In essence, this model is nonlinear. A bicycle model with a Pacejka tire model is presented by Yoon et al. in [210]. Because large tire sideslip angles are not favorable, the front and rear tire sideslip angles have been limited. A linear tire model for the vehicle is suggested by Park et al. [211] and Zhang et al. [251], while the motion equations of the vehicle are nonlinear. To keep the tire in its linear force region and keep the vehicle model valid, the sideslip angles of the tire has been limited.

In [185], a novel method is presented to predict and avoid the collision of static and moving obstacles in an uncertain environment. They used a decision-making process to estimate the velocity of obstacles by using the vehicle's sensory system information. Therefore, the vehicle is able to find a proper path reach to the target safely and without any collision. The result illustrates the efficient algorithm for complex and dynamic environments. In [212], an uncontrollable divergence metric is presented. A mechanism to switch between multiple predictive controllers is developed by using this metric to reduce return time of controller and maintain predictive accuracy.

A nonlinear MPC is offered in [213] for an autonomous underwater vehicle (AUV). An optimization model of receding horizon along with a Spline template solves the problem of path planning. A combination of the obtained result from path planning and MPC is used for tracking control. To determine the maneuvers mode for autonomous vehicles in dynamic environments, a path planning method with MPC is proposed [214]. To decide maneuvers of changing lane and keeping lane, convex relaxation method is used. For ensuring the safety vehicle, a collision avoidance method is developed. Also, for having comfortable and natural maneuver, the lane-associated potential field is presented.

3.3.1.1. Linear Bicycle Model

As already mentioned, the dynamics models of vehicle are nonlinear, therefore, the used MPC in these models also are nonlinear. Since the bicycle model considers the vehicle dynamics, it can be applied in the quadratic MPC to handle the high-speed maneuvers with large lateral accelerations [211, 215]. A nonlinear bicycle model with a Pacejka tire model and longitudinal load transfer is developed by Yi et al. [216]. The proposed model is linearized around the operating point. Also, the total vehicle acceleration has been limited to stay in the friction circle. The circle is approximated by half-spaces so that the quadratic constraint is approximated by linear constraints to be applied in a quadratic MPC.

Turri et al. [217] have proposed a four-wheel vehicle model for the vehicle's lateral motion with a Pacejka tire model where the vehicle's longitudinal motion is known. Also, the load transfer has been considered in the given model. Then, the model of the vehicle has been linearized. The front and rear tire models have been calculated as functions of the total longitudinal force and have been linearized. Also, the load transfer and the effects of combined slip have been considered.

To model the lateral motion of the vehicle, a nonlinear bicycle model with a Pacejka tire model has been presented by Gao et al. in [218]. A linear tire model for the rear tire force is provided and then the tire sideslip angle is limited. The tire cornering stiffness and the maximum tire sideslip angle is obtained which has the best approximation of the tire behavior and can generate a lateral force close to the maximum lateral force. For the front tire, the tire force in the motion equations has been used and the steering angle by the inverse Pacejka model has been derived. also, to keep the tires in their linear force regions the tire sideslip has been limited.

To lateral motion of a race vehicle, a nonlinear bicycle model with a brush model of the tire is presented in [219]. In [218], a tire inverse model for the front tire and the nominal path curvature as the road curvature has been introduced. So, instead of using a linear tire model for the rear tire, the brush model has been linearized around the nominal sideslip angle corresponding to the nominal path curvature. This is used for race vehicles working in large path curvatures at high speeds, which need large tire sideslip angles to track the path. To track path in the road vehicles, small tire sideslip angles are needed and the result of tire mode would be similar to a linear tire model. The research also limits the front lateral force and applies a stability envelope for the rear tire instead of constraining the tire sideslip angles. The envelope limits the yaw rate to its maximum steady state value corresponding to the road tire-road friction and the rear sideslip angles to the limits corresponding to the tire linear force region.

For lateral motion of a race vehicle in [220], Funke et al. have presented a nonlinear bicycle model with a brush tire model. In both front and rear tires, the brush tire model was linearized around the nominal sideslip angles corresponding to the nominal path curvature. A stability envelope similar to [219] has been presented. The combined slip effect has been assumed for the front lateral tire force limitation. The longitudinal force commanded by the driver is constant and the lateral force has been limited to the remaining tire capacity. In [219] and [220] the linearized tire models are similar to a linear tire model. In addition, the tire sideslip angle constraints keep the tires in its linear force range, so they keep the linear tire model valid. in [221], a vehicle-bicycle model has presented with linear tire models and the rear and front sideslip angles constrained.

In this thesis, a bicycle model with linear tire models is presented to model the vehicle's longitudinal and lateral movements for MPC-based motion planning. Similar to [218], the cornering stiffness values and maximum sideslip angles are calculated. Instead of limiting the front and rear sideslip angles, the lateral forces are limited in combined tire slip constraints which the generated constraints cover the sideslip angle constraints. The combined slip is important in the large tire forces. If the total capacity of tire uses for the longitudinal force, the tire has not lateral force capacity. Also, load transfer is important in large accelerations such as harsh braking. In this thesis, the effects of load transfer and combined slip are considered as a constraint.

The longitudinal load transfer is applied in the calculation of the normal tire forces similar to [216]. Funke et al. [220] and Turri et al. [217] model the lateral vehicle motion and used the combined slip as a constraint on the lateral force. In this thesis, the longitudinal motion is also modeled, and the combined tire slip is used as a constraint on the longitudinal force and the lateral force. If load transfer is not considered, the constraint will be an ellipse. In this thesis, to derive the tire constraint, load transfer equations are applied on the ellipse constraint equation. The obtained result is nonlinear and cannot be used in a quadratic MPC. However, the constraint constructs a closed convex space, which can be approximated by half-spaces similar to [216].

3.3.2. Obstacle Model

Obstacle avoidance is a significant task of the motion planning system. One way for this approach is generating a repulsive force which holds the vehicle away from the obstacle. This suggested approach is carried out by inserting a repulsive PF to the cost function of the optimization problems. Abbas et al. [222] and Gao et al. [205] are used hyperbolic potential functions that are based on the distance from the obstacle, and also Park et al. [227] and Yoon et al. [210] are applied a parallax potential field in the MPC cost function. The generated results are non-linear and non-convex which are solved as a nonlinear optimization problem.

The other way for the task of avoiding an obstacle is to keep or put the vehicle in the non-obstacle region. The non-obstacle region's nature is non-convex, and the non-convex constraints can produce this region. Liu et al. [228] generated a safe region in the detection region of LIDAR. This region that is cut off by obstacles is the region of semi-circle detection. The vehicle is restricted by the circle around each obstacle which this approach has been proposed by Gotte et al. [229]. Gao et al. [218] have constrained the vehicle by using the ellipse around each ellipsoidal obstacle. In order to keep the vehicle in the non-obstacle region, Liao et al. [230] have considered the obstacles as rectangles and have applied combined integer restrictions.

Frasch et al. [209] have assumed obstacles as rectangles while have used nonlinear constraints for generating a non-obstacle region. Qian et al. [231] have generated a quadratic nonconvex constraint for an obstacle on the side of the lane. The constraint keeps the vehicle away from the obstacle's the portion of the lane. When there is an obstacle in the lane middle, the vehicle behind the obstacle with a convex linear constraint has been kept.

The problems based on MPC with nonconvex limitations are nonlinear which include the high computational costs. In order to avoid obstacles, some researchers have controlled the lateral motion of the vehicle and have assumed that they knew the longitudinal motion before the obstacle was avoided. In this approach, the non-obstacle region has been gridded based on longitudinal motion to predict time steps. They have restricted the lateral position of the vehicle to an accessible convex lateral space at the corresponding grid for each prediction time step [220, 221]. This approach is useful for the conditions that the lateral movement is designed by the module for motion planning, such as a system of driving assistance that the driver controls the longitudinal motion. Meanwhile, the lateral and longitudinal motions are not simultaneously planned and cannot plan the maneuvers such as stopping behind an obstacle.

A problem based on MPC is solved several times for different values of brake by Erlien et al. [219]. In this approach, the longitudinal motion has been planned as the lateral motion at the same time. However, this method considers only specific values of brake and also increases the computation time. For each obstacle, a linear constraint is described in some papers. For each obstacle in front of the vehicle, Nilsson et al. [206] have provided a linear limitation with a fixed slope. The constraint line is determined which keeps longitudinal and lateral safety distances from the obstacle. Also, a similar constraint has generated an obstacle behind the vehicle. If the longitudinal safe distance is small, the constraint causes difficulties for a vehicle passing an obstacle on its side, and if it is large, a section of the space behind the obstacle cannot be used for a maneuver of stopping behind the obstacle.

The similar constraints are used in [232], a horizontal constraint for the situation which obstacle is on a different lane from the vehicle and it is close to the vehicle in the longitudinal direction is considered. This variation cannot solve the mentioned problem for the obstacles on the same line of the vehicle. Based on the signed distances of vehicle and obstacle, Carvalho et al. [233] have proposed a new constraint. The signed distance of two objects is their minimum distance if they are not in contact. This constraint limits the signed distance to be non-negative, which is non-convex. The paper has linearized the constraint around the states that have predicted. This approach has produced a linear constraint with slopes, which is based on the vehicle's relative position and obstacle and has addressed the problems of the constraint by [206]. When there is an obstacle in front of the vehicle, this method is generated a constraint that is limited the vehicle to stop behind the obstacle and does not allow a swerving maneuver. If the obstacle is ahead of the vehicle without lateral distance, it is considered to be in the front of the vehicle.

In this thesis, the different potential filed functions are presented in the proposed cost function to avoid the obstacle collision. The proposed cost function is a nonlinear optimization problem and is solved as a convex quadratic approximation for MPC.

3.4. Summary

In this chapter, the background of motion planning based on MPC is presented. The main objective of the thesis is to develop an autonomous vehicle platform that generates a proper and free-collision trajectory based on Model Predictive Controller (MPC). In an autonomous vehicle, the motion planning module is module which plans the vehicle trajectory and performs the task of obstacle avoidance. The summary of this chapter is as follows:

- The architecture of autonomous vehicle is investigated.
- The different techniques of motion planning are introduced for the autonomous vehicles.
- The motion planning problem based on MPC is studied.

4. GRASSHOPPER OPTIMIZATION ALGORITHM FOR MOBILE ROBOT NAVIGATION

The heuristic approach is one of the most common automatic approaches for navigating in the science and engineering field which is inspired by the meta-heuristic algorithm. In this chapter, the application of meta-heuristic inspired by the nature algorithm especially the Grasshopper Optimization Algorithm (GOA) can be utilized for navigating the mobile robot in the unidentified and uncertain environmental map. The novel approach based on GOA is presented to develop new mobile robot navigation in the existence of static and dynamic obstacles.

4.1. Introduction

The nature-inspired meta-heuristic algorithm plays a significant role in planning and development of the navigation approach of a mobile robot. Due to their ability to search the global platform, the meta-heuristic algorithm is used to navigate the mobile robot to give up the diverse solution and find a suitable solution in the local region. Many heuristic algorithms are applied to navigating the mobile robot which includes the Genetic approach, Ant Colony approach, Particle Swarm Optimization, Artificial Bee Colony approach, Cuckoo Search approach, Bat approach, invasive weed optimization, Shuffled Frog Leaping approach, etc. In this dissertation, the grasshopper optimization algorithm is proposed with static and moving obstacles to navigate the mobile robot in the certain and uncertain environment. The proposed algorithm has been implemented to perform optimal path planning along with avoiding obstacles.

4.2. Overview of Grasshopper optimization approach

The Grasshopper Optimization approach (GOA) is inspired by the foraging behavior of grasshoppers [234]. Grasshopper is actually an insect considered to be a pest. These creatures can be seen individually in nature, but they are considered to be one of the largest swarms. Grasshopper's life cycle consists of two important phases: larval and adult. The major feature of the swarm in the larval phase is moving slowly or movement with small steps of grasshoppers. On the other hand, one of the essential characteristics of the swarm in the process of adulthood is greater and unpredictable movement. In both phases, the source/food search process is separated into two sections, namely exploration and exploitation.

In the exploration, grasshoppers are inclined to move quicly, while in the exploitation phase they encouraged to move locally. These two functions and the target searching are performed simultaneously by the grasshoppers. GOA is selected for the following important characteristics:

1. To have exploration and exploitation ability,
2. Don't trap in the local minimum,
3. Rapidly Convergence towards a global solution,
4. And doent have any influence on complexity of the problem.

The following mathematical model is used to simulate grasshopper's swarming behaviour,

$$X_i = S_i + G_i + A_i \tag{4-1}$$

where X_i presents the position of the i^{th} grasshopper, S_i shows the social interaction of the i^{th} grasshopper, G_i defines the force of gravity on the i^{th} grasshopper and A_i is the advection of wind.

$$S_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})\hat{d}_{ij} \tag{4-2}$$

where s shows the strength of social forces, d_{ij} is the distance calculated from the i^{th} grasshopper to the j^{th} grasshopper by $d_{ij} = |x_j - x_i|$. A unit vector between the i^{th} grasshopper and the j^{th} grasshopper is calculated as $\hat{d}_{ij} := \frac{x_j - x_i}{d_{ij}}$. The social forces identified by function s are given as follows.

$$s(r) = fe^{\frac{-r}{l}} - e^{-r} \tag{4-3}$$

where f is the intensity of the attraction and l is the scale of attraction length. The G component of the model in (4-1) is calculated as follows

$$G_i = -g\hat{e}_g \tag{4-4}$$

where g is the gravitational constant and where$\hat{e}_g$ the unit vector to the center of the earth. In addition, the A element is obtained as follows in (4-1).

$$A_i = u\hat{e}_w \tag{4-5}$$

where u is a constant of drift and $\hat{e}_w$ a vector of unity in the direction of wind. After replacing S, G and A in (4-1), the model will be.

$$X_i = \sum_{j=1.j\neq i}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \tag{4-6}$$

Here N shows the number of grasshoppers. In the optimization algorithm, (4-6) is not used because as it prevents the optimization algorithm from exploring and exploiting near solution search space. In fact, this nymph grasshopper model is created for the swarm grasshopper that is placed in free space. In addition, this mathematical model is not used for solving the optimization problem, because the grasshoppers quickly reach to comfort zone and it is not possible for the swarm to converge to a specific point. The changed grasshopper location version is used to change the position of the grasshopper is as follows:

$$X_i^d = c_1\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c_2 \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|)\frac{x_j - x_i}{d_{ij}}\right) + \hat{T}_d \tag{4-7}$$

where ub_d is an upper boundary in the d^{th} dimension, lb_d is a lower boundary in the d^{th} dimension, and $\hat{T}_d$ shows the value of target i.e. the best solution. c_1 and c_2 are coefficients to reduce the zone of comfort , repulsion, and attraction. In (4-7) the element of gravity is not applied and the direction of the wind is always assumed to be in the direction of the target direction. The target point (global best), its current position and the position of all other grasshoppers can be used to calculate the next position of the grasshopper. This means that GOA needs all agent searches to participate in determining every grasshopper's next location.

As already mentioned, the first section of (4-7) considers the current position of grasshopper according to all other grasshoppers. While the second section is utilized in order to reduce the agent movement around the target. This means that the second section takes into account the exploration and exploitation of the whole swarm around the target. Specifically, c_1 parameter can reduce the grasshopper movement near the target i.e. it balances the exploration and exploitation of whole swarm near the target. Also, c_2 parameter reduces the comfort, attraction and repulsion zones between grasshoppers i.e. c_2 reduces space linearly to guide grasshoppers for finding the optimal solution in search space. The adaptive c_1 parameter is able to reduce forces of attraction and repulsion in proportion to the number of iterations.

As the iterations continue, c_2 parameter reduces the convergence of the search around the target. In terms of balancing the exploration and exploitation, c_1 is reduced in proportion to increase the number of iterations. This method allows GOA to effectively exploit and then perform later stages of the optimization. Similarly by increasing the number of iterations, the c_2 value is reduced to minimize the comfort zone. Both parameters (c_1 and c_2) are considered as a single parameter and it is calculated as follows:

$$c = c_{max} - n\frac{c_{max} - c_{min}}{L} \tag{4-8}$$

where c_{max} and c_{min} are the maximum and minimum amount of c. n shows the number of the current iteration and L represents the maximum amount of iterations. The pseudo code for GOA is illustrated in Figure 4-1.

Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, d= number of dimensions
Generate initial population of n grasshoppers' $\mathbf{x}$i= (i=1, 2, …., n)
Calculate the fitness of every grasshopper
T = The best search engine
while criteria for stopping are not met **do**
    Update c_1 with Eq. (4-8)
    Update c_2 with Eq. (4-8)
    for every grasshopper *gh* in population **do**
        Standardize the distances between grasshoppers in [1,4]
        Update the location of the *gh* by Eq. (4-7)
        If it is needed, update bounds of *gh*
    end for
    If a better solution is found, update T
end while
Output the *T*.

Fig. 4-1: Pseudo code of GOA.

4.3. Objective Function Formulation using GOA

The main objective of this chapter is to acquire an effective path planning approach for the mobile robot in the existance of static and moving obstacles. The effort has been proposed to plan and develop the optimal approach of mobile robot path planning by using grasshopper optimization algorithm in terms of length of time and path. Path planning approach for the mobile robot is to find optimal factors to satisfy a set of specific requirements based on the objective function including detection and avoidance of obstacle, handling the trap-like situation, avoidance of randomly walking and generation of the optimal path. When the mobile robot is moving in the environmental map, it obtains the data of the given environment by the attached sensors of the robot.

It assists to find the location o the robot in an uncertain and unknown environmental map. The used sensors provide the information about shape, size, and positions of obstacles and the robot move toward the target without any collision by using the sensory information.The main aim of this thesis is to provide and create an optimal and accurate path planning approach for a mobile robot by a grasshopper optimization approach.

First, the optimization problem of path planning is converted to a minimization problem and then an objective function based on the position of the target and the obstacle is defined. Finally, the grasshopper optimization algorithm is used for solving this optimizating problem. The best global value of grasshopper is selected at each iteration and the robot moves towards these positions during the execution process. The robot frequently updates the information based on the sensory information, and according to this information, the objective function of the optimization changes. If there are not any obstacles on the movement path of the robot, the robot will be able to find the target point directly without using GOA.

Otherwise, the robot uses the obstacle avoidance mechanism to navigate in the unknown environmental map. The path planning problem here is solved for four different scenarios, namely, the path planning among static obstacles and moving obstacles, moving target and the combination of static and moving obstacles. The key aim of the proposed navigational approach based on GOA are as follows:

1. To plan and develop the effective path planning algorithm to to prevent observed obstacles in the path.

2. To prevent the robot's spontaneous motion in its setting as per the optimum time.
3. To create the uniqueness in the simulation result.
4. To achieve better performance in compration to the other navigational controller.

4.3.1. Obstacles

The environment consists of n obstacles, i.e. $O_1, O_2, \ldots, O_n$ and their position coordinates are represented as (xO_1, yO_1), $(xO_2, yO_2), \ldots, (xO_n, yO_n)$. The obstacles have circular and rectangular shapes. In this paper, the obstacles can be static and dynamic. If an obstacle is stationary, its velocity is zero; otherwise, its velocity is (v_x, v_y) along x and y axes. The velocity of an obstacle is set randomly and is equal to or less than the velocity of the robot. The velocity and location vectors of obstacles (their speed and orientation) are unknown for the robot. The robot is supposed to detect the obstacles and move on the arbitrary path.

When the position and velocity of the obstacles are unknowns for robot, the robot must be fitted with detectors and distance sensors to obtain the necessary data on the surrounding area. The robot is equipped with range sensors that provide 360° proximity information with radius R. When the robot moves to a new location in the configuration space, the distance to the surrounding obstacles is determined first. by reading its proximity sensor and then saves the results in a matrix that includes the positions of the obstacles. The velocity information is inferred from the consecutive position measurements.

4.3.2. Robot sensing

As already mentioned When an obstacle reaches the robot's range the distance between the obstacle and the robot is defined and the direction of the obstacle is projected [235]. It is suggested that an obstacle with (r_x, r_y) position is entered into the robot range at time t, and the position of the obstacle at t and t+1 times are equal to (x0, y0) and (xT, yT), respectively, where T is the sampling period. If the obstacle's location does not shift, the obstacle will be static otherwise it will be dynamic. Then the direction of the obstacle moving is evaluated and the robot will choose the next move based on the obstacle's potential velocity vector and GOA. If the next trajectory of the obstacle intersects the robot path, the robot will be away from its original path. Also, If the distance from the robot to the obstacle increases, the obstacle is dynamic and it is moving away from the robot; Otherwise the obstacle travels in the direction of the robot. The sensor includes four circles with specific radiuses R_1, R_2, R_3, R_4.

The largest circle is the maximum range and the smallest circle is the minimum safe distance from the robot to the obstacle. There are two other intermediate circles between the smallest and largest circles. The used sensor range for this paper is demonstrated in Figure 4-2(a) and is formulated as below:

$circle_i = check\ around\ (r_{x_i}.r_{y_i}.R_i)$ where i=1, 2, 3, 4 and R=0.2, 0.3, 0.4, 0.5 (4-9)

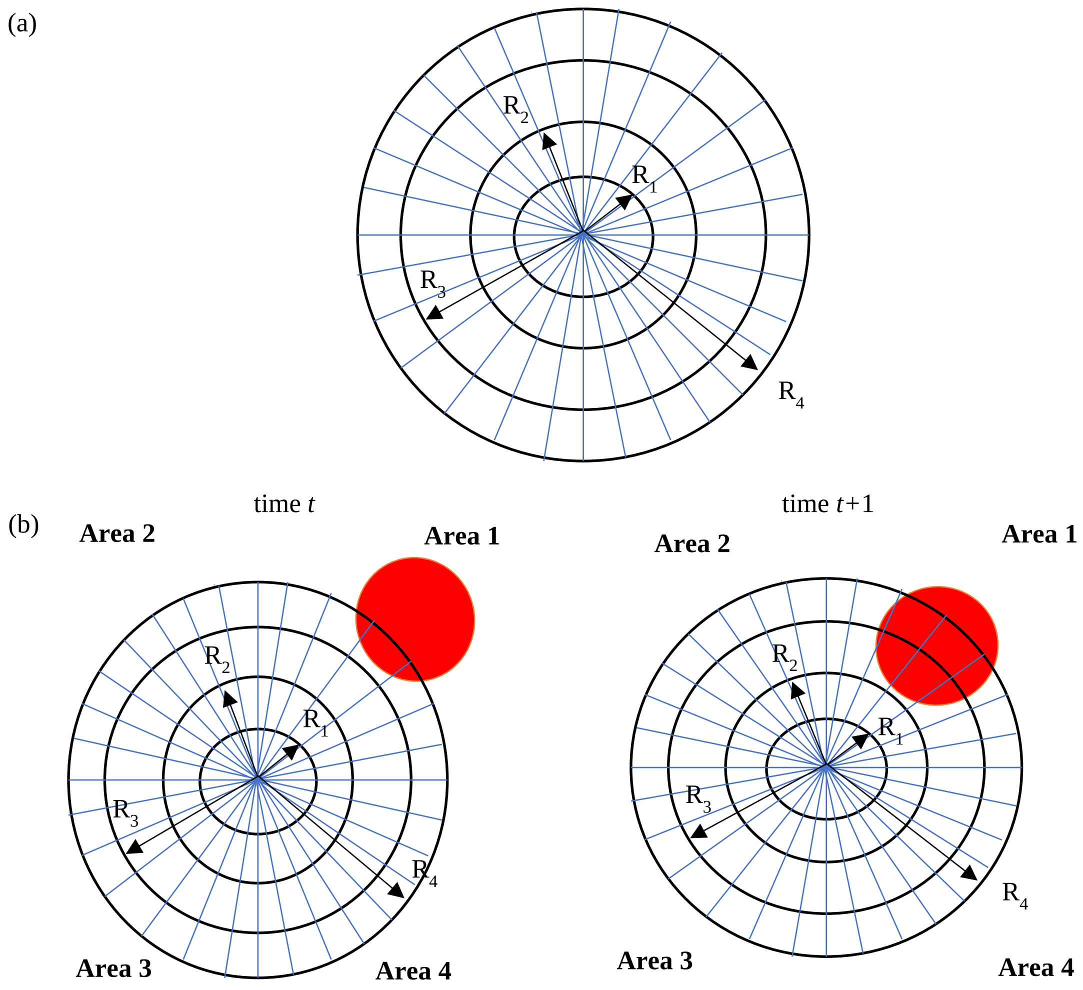


Fig. 4-2: (a) The range sensor, (b) Detection of moving obstacle at t and t+1 times.

As seen from Figure 4-2(b), these circles are seprated into four regions to evaluate the orientation of moving obstacle. The first region is between 0 and 90°, the second region is between 90° and 180°, the third region is between -180° and -90°, the second region is between −90 °and 0 that is shown in Figure 4-2(b). The robot should save the point of instruction between the obstacle location and these circles and identify which regions have these points of intersection (IP). To estimate the velocity vector of each obstacle, the positions are first saved in two consecutive iterations. Then to estimate the next trajectory of the moving obstacle, the reading sensor saves 0 value for free path and 1 value for the regions inside the obstacle.

The used equation for determining the intersection points is:

$$IP\ of\ circle_i = check\ in\ Polygon(No.of\ polygon.\ r_{x_i}.r_{y_i}.x_{polygon}.y_{polygon}) \quad (4\text{-}10)$$

If the prediction indicate that the number of intersection points does not change, then the obstacle is static; otherwise, it is dynamic. Measuring the number of intersection points will determine the direction of the moving obstacle. If the number is increased, then the direction of the motor obstacle is toward the robot; if not, the obstacle moves away from the robot.

As can be observed from Figure 4-2(b), If the obstacle's direction is toward the first region, then the robot keeps away from this region. For each quarter the number of intersection points is formulated as follows:

$$NO.IP\ in\ quarter_i = \sum_{i=1}^{4}(IP\ of\ circles\ in\ quarter_i) \quad (4\text{-}11)$$

4.3.3. Target-seeking behavior

Here, the grasshopper is chosen from the group of random grasshoppers that has the minimum distance from the target and the maximum distance from the obstacles. This is a continuously searching process for the grasshopper until it completely finds and reaches the target. The Euclidean distance between the target and grasshopper is defined by (4-12),

$$D_{GT} = \sqrt{(x_T - x_{G_i})^2 + (y_T - y_{G_i})^2} \quad (4\text{-}12)$$

where D_{GT} is the minimum Euclidean distance from the target to ith grasshopper. (x_T, y_T) represents the coordinate of the target position. ($x_{G_i}.y_{G_i}$) is the coordinate of i^{th} grasshopper position.

4.3.4. Obstacle-seeking behavior

The navigation problem is a difficult task for a moving robot. For effective navigation in an unknown environmental map, the robot needs a mechanism of obstacle avoidance. When an obstacle is detected by means of a sensor, the grasshopper algorithm generates a random number of grasshoppers in near the obstacle and the grasshopper with the best value of the objective function is selected. This grasshopper has selected in a way that its distance from the nearest obstacles is maximum. The robot is in the position of a newly selected grasshopper and starts the procedure to search for the next grasshopper with the best value until it obtains a safe and optimal path. The best grasshopper is chosen using Euclidean distance from a grasshopper to the nearest obstacle that is represented as follows.

$$D_{GO} = \sqrt{(x_O - x_{G_i})^2 + (y_O - y_{G_i})^2} \tag{4-13}$$

where D_{GO} is the Euclidean distance from the neighboring obstacle to position of the i^{th} grasshopper. (x_O, y_O) shows the coordinate of the target position. Likewise, in a complex environmental map, the selection of the neighboring obstacles is an important task to generate the optimal path. Therefore, the distance of the neighboring obstacles is calculated by (4-14).

$$D_{RO} = \sqrt{(x_{O_n} - x_R)^2 + (y_{O_n} - y_R)^2} \tag{4-14}$$

Here, D_{RO} is the distance from the robot to the nearest obstacle, (x_{On}, y_{On}) coordinate shows the position of the nearest obstacle and (x_R, y_R) is the coordinate of the robot position.

The objective function of GOA for the optimization of path planning is based on target-seeking and obstacle-seeking behaviors and this is formulated as in (4-15).

$$G_i = \lambda_1 . \frac{1}{\min O_n \epsilon O_s \|D_{GO}\|} + \lambda_2 . \|D_{GT}\| \tag{4-15}$$

When an obstacle enters into the active range of a sensor in the O_s environment, its numbers are determined by the reading sensor of the robot. In (4-15).When the grasshoppers (Gi) move away from the obstacle, the value of $\min O_n \epsilon O_s \|D_{GO}\|$ will be massive and when the grasshoppers become close to the target, the value of $\|D_{GT}\|$ will decrease. Therefore, the objective function of GOA is a minimization type of an optimization problem that is utilized to find the optimal path for the mobile robot in the

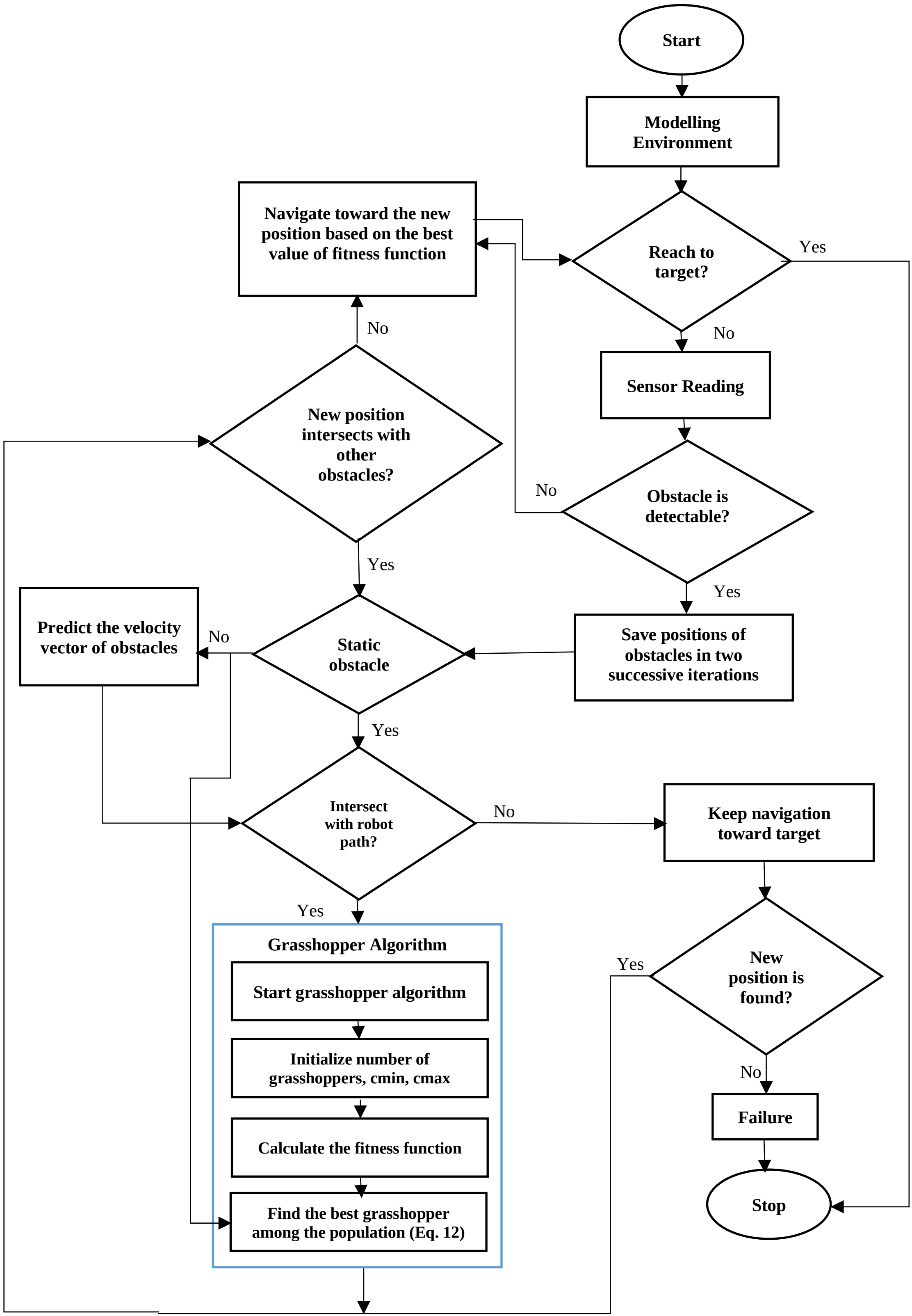


Fig. 4-3: The flowchart of proposed navigation algorithm in unknown dynamic environment.

Unknown environmental map. λ_1 and λ_2 are the controller parameters that are used for the safety of the path and the maximum and minimum of length of path in navigation. When λ_1 has the maximum value, the robot is able to move safely and without collision and when the λ_2 value is maximum, the path length is minimum.

A suitable set of these parameters, however, results in a useful objective function for planning the robot path. The path planning method for a mobile robot is posited by combining the above-mentioned components. After modeling the environment, the location of the robot, the target, and the obstacles are initialized. Then, the behavior of target seeking starts to find the target point so that the robot moves from its starting position to the target.

As most other path planning algorithms, the proposed algorithm starts by checking whether the target is reachable or not. If the target can be reached the robot moves straight to the target and the algorithm is complete. Otherwise, the proposed algorithm utilizes the behavior of obstacle seeking to predict the trajectories and positions of the obstacle(s). In this behavior based on the range sensor, the robot senses its surrounding environmental map whether any obstacles exist. If an obstacle is identified by the sensor, the next step is to determine whether it is static or dynamic. But if the obstacle is dynamic, the robot first predicts the next velocity vector of moving obstacle and then determines the appropriate orientation based on GOA. If the obstacle is static, then the grasshopper optimization algorithm is activated and it generates the population of grasshoppers randomly near to obstacle. The robot selects the appropriate grasshopper among the population based on (4-15) to reach the target.

If the robot finds a new location, it's the robot's next position. When the robot approaches the target, the proposed algorithm calculates arrival time and length of the path. But if the robot could not find a new position, then there is not a promising solution for the path planning problem and the algorithm fails. The proposed algorithm flowchart is shown in Figure 4-3.

4.4. Simulation Analysis

The suggested algorithm is simulated in this section and tested in different environments. There are both static and moving obstacles in the environments studied. In these environments, the position (static and dynamic) and velocity of the (dynamic) obstacles are different. The starting position of the mobile robot is a hexagon and the target point is a square. The proposed algorithm is performed 50 times in each environment. The characteristics of the environmental map are described in this section. The simulations are conducted in MATLAB 2017a environment using a 2.40-GHz Intel Core 7 Duo Processor. The simulation results of the environment with a static obstacle is shown in Figure 4-4. The used parameters of the simulation are listed in Table 4-1. It is obvious that the suggested algorithm provides a safe and short path along with acceptable arrival time at the first environment.

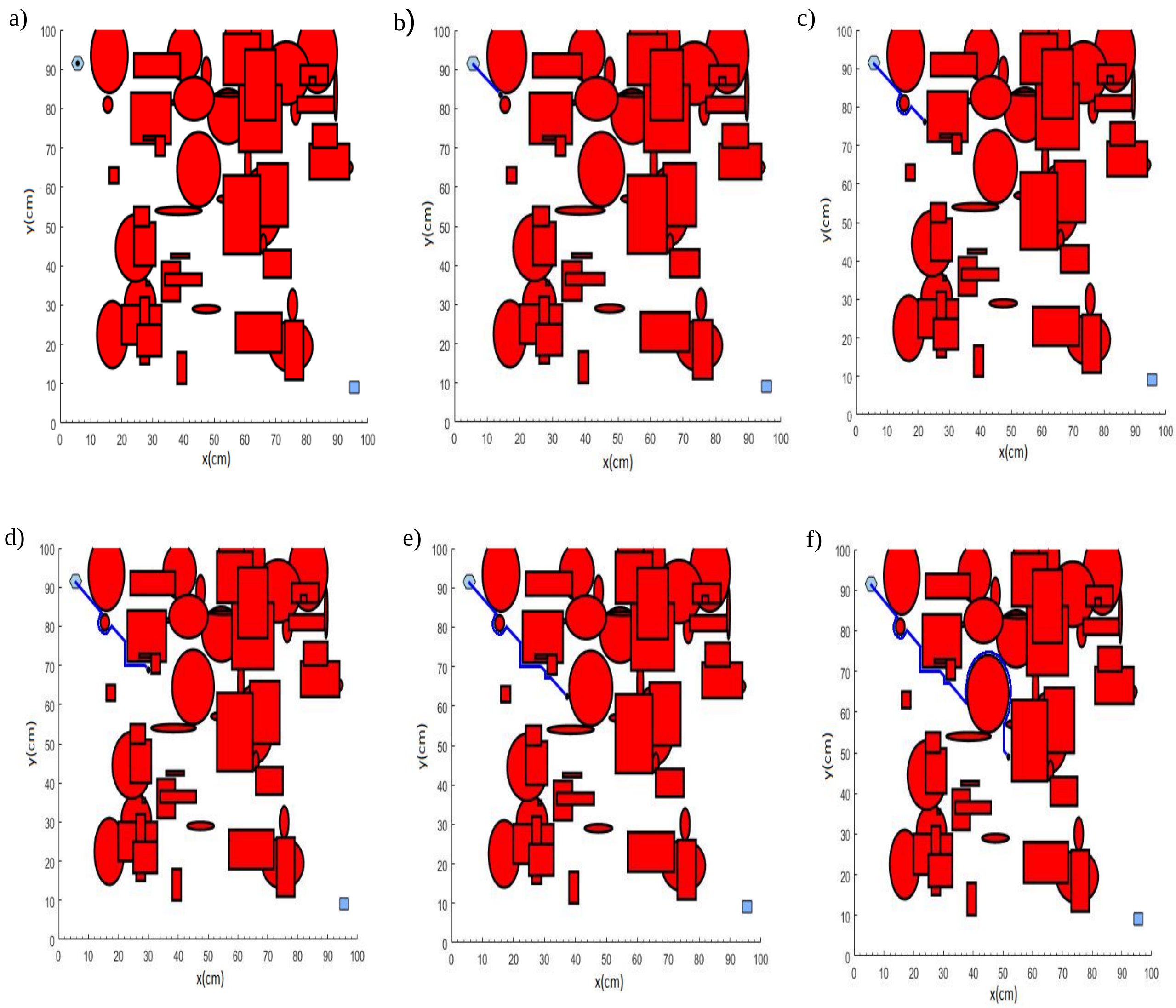
At the first scenario, the robot moves from the start coordinate to the target coordinate. There are 30 static obstacles in different positions randomly. First, the robot navigates toward the target point according to the optimal path between starting and target positions. The environment is crowded and the robot has to constantly check the environment after each step to make sure whether path safety is provided or not. If the sensor does not detect any obstacle on the robot path, it will continue its path towards the target and will record 0 in the sensor matrix. However, if it detects an obstacle (as seen in Figure 4-4b), it will try to avoid the collision with them by recording 1 in the sensor matrix and running GOA.

Table 4-1: Parameters of settings for simulation.

Parameters	**Values**
Swarm Size (N)	100
Maximum Value (cMax)	1
Minimum Value (cMin)	0.00001
Fitting parameter (λ_1)	0.1-1

Fitting parameter (λ_2)	0.01-0.0001

Then the robot distinguishes whether the detected obstacle is static or dynamic. The robot uses four range sensor circles to determine intersection points for two-time intervals among obstacles and sensor circles. The robot initially navigates toward the target point but it confronts a circle obstacle on its path. The robot starts the grasshopper algorithm to avoid the collision.



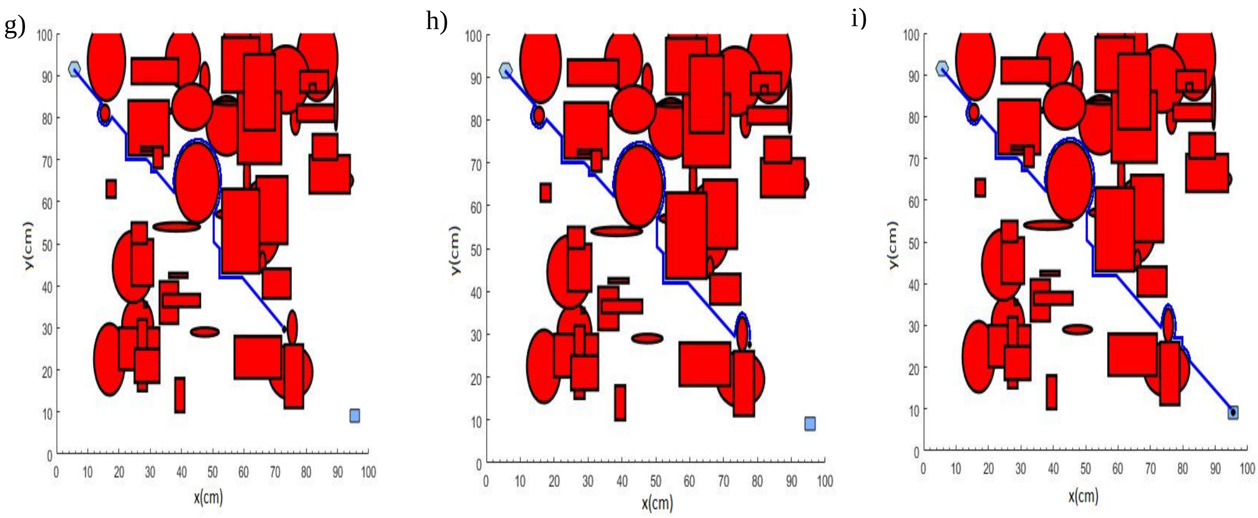


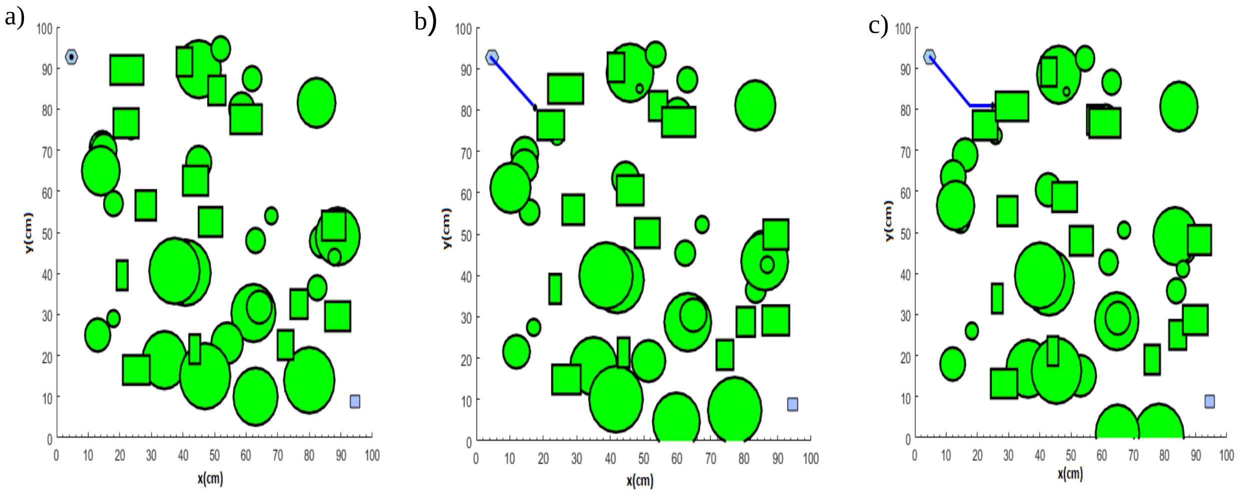
Fig. 4-4: Simulation results of the proposed algorithm in the environmental map with static obstacles.

First, the grasshopper algorithm generates the random number of grasshoppers nearby the obstacle and the grasshopper with the greatest target's objective is selected among the group of grasshoppers. The selected grasshopper has the maximum safe distance from the closest obstacle. Then, the robot successfully passes from the near obstacles. It returns to its original direction and path after passing the obstacle to meet the destination on the optimal path (Figure 4-4c). The robot follows the optimal path until it detects another obstacle on its path. As mentioned above, the robot passes from two obstacles by using GOA and then returns to its optimal path (Figures 4-4d and 4-4e). In the following, the robot confronts with a circular obstacle and passes from it by applying GOA.

To return to its original path, the robot is forced to cross the narrow path (Figure 4-4f). After passing the narrow path and returning to the original path, the robot reaches a rectangle obstacle and avoids collision with it by GOA again (Figure 4-4g). To reach the target, the robot continues its path until it faces a circular obstacle; and it starts the grasshopper optimization algorithm to avoid collision (Figure 4-4h). Again, the robot tries to return to its optimal path but there are two obstacles on its path. After passing them, the robot reaches the target point successfully (Figure 4-4i). The arrival time to the target is 40.15s. The path length based on Euclidean distance is 121.79cm, and the minimum traveled path length by the robot is 140.79cm. This difference is because of the crowded environmental map; therefore, the robot must cautiously pass from the near obstacles.

In the second scenario, there are 45 dynamic obstacles scattered to different positions randomly. The objects are now moving. The velocities of the obstacles are unknown which are measured by the sensors. Also, we assume that the obstacles move slower than the robot so that the robot can handle the dynamic environment. The robot navigates toward the target and moves on its optimal path (Figure 4-5b). When the robot reaches the rectangular obstacle, it moves in (+x) and (+y) directions; at the same time, the second rectangle obstacle arrives (Figure 4-5c). Since the velocity of the first rectangular object is smaller, the robot waits to pass the second rectangle and moves in (-y) direction (Figure 4-5d). Then, the robot reaches the circular obstacle; after passing it, the robot returns to the original path again (Figure 4-5e). It follows the optimal path until it confronts with the rectangular obstacle (Figure 4-5f).

Because the velocity of this rectangle is less than the velocity of the robot, it passes from obstacle by GOA and continues its path until the robot faces to another obstacle and again the robot navigates in (+x) and (+y) directions (Figure 4-5g).



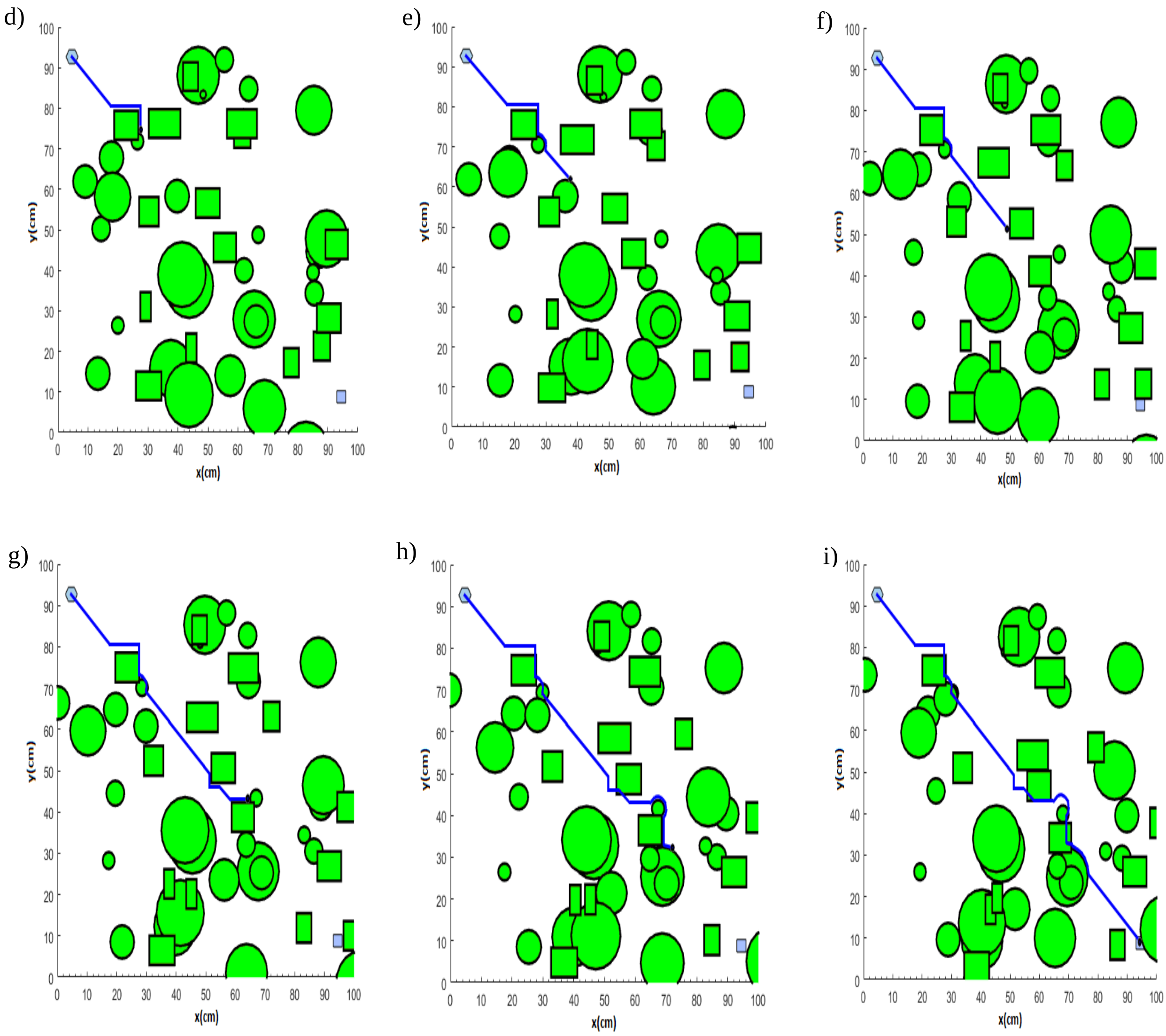
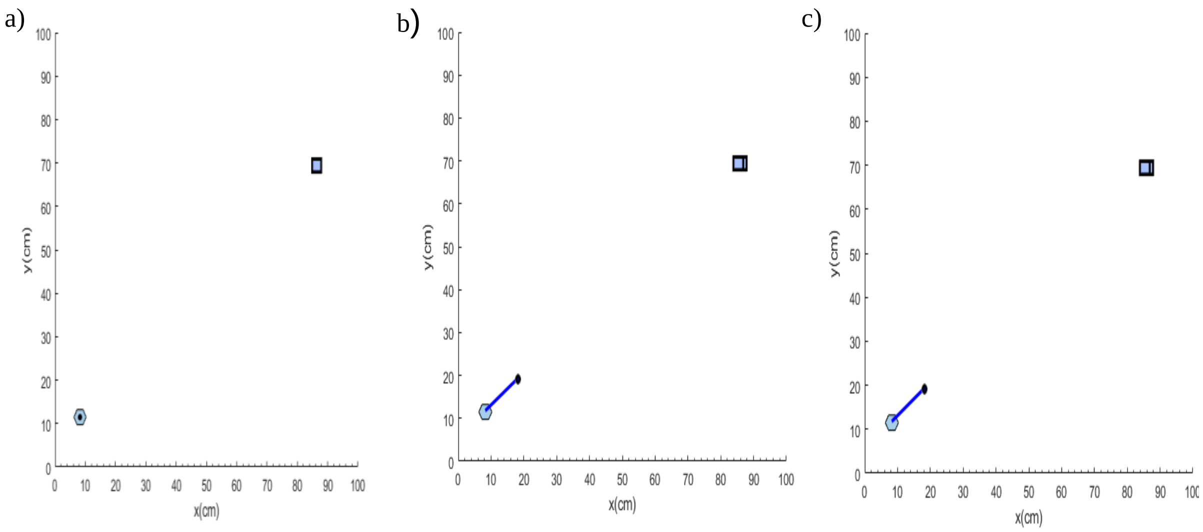


Fig. 4-5: Simulation results of the proposed approach in the environmental map with dynamic obstacles.



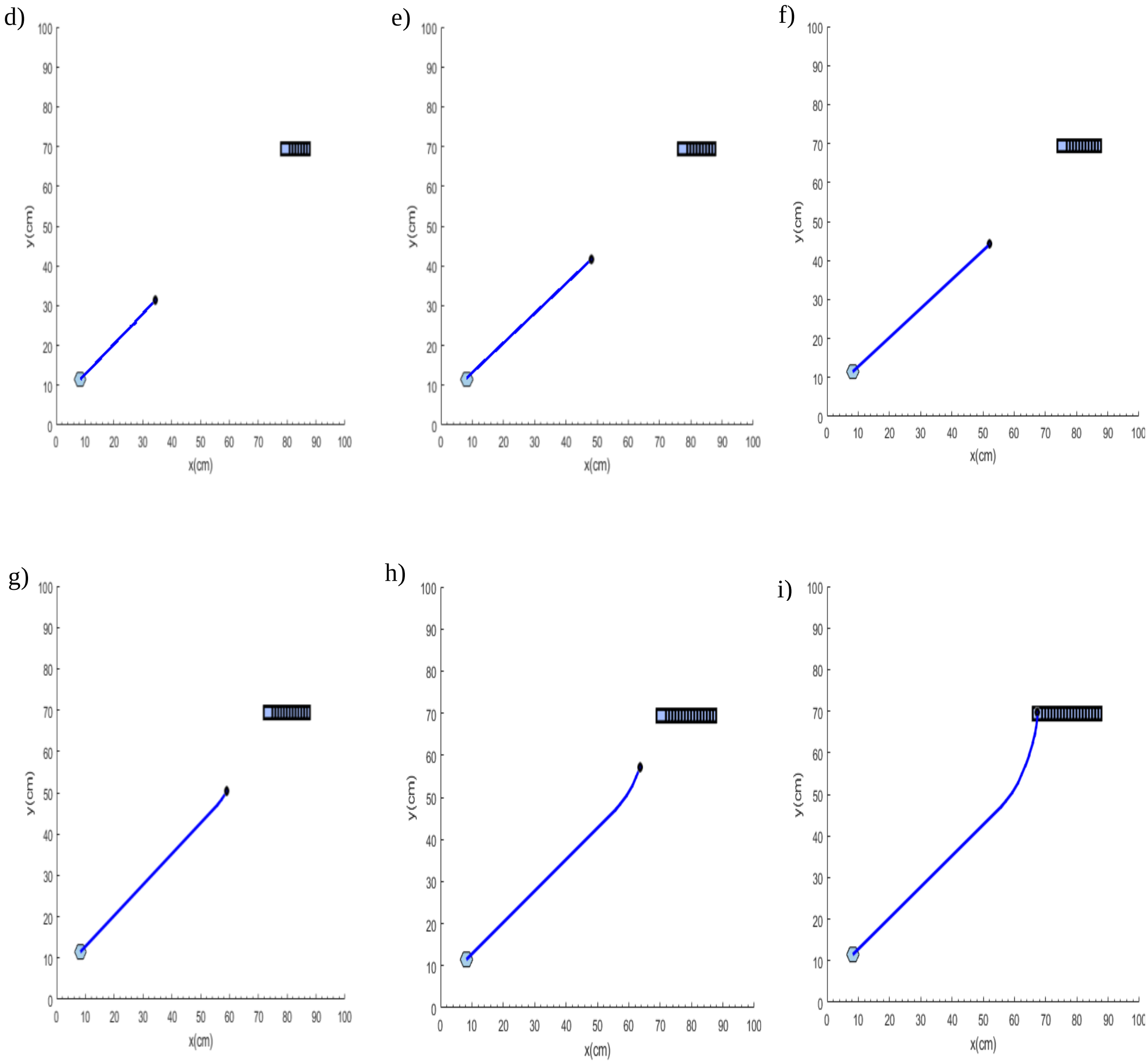


Fig. 4-6: Simulation results of the proposed algorithm in presence of dynamic target without obstacle.

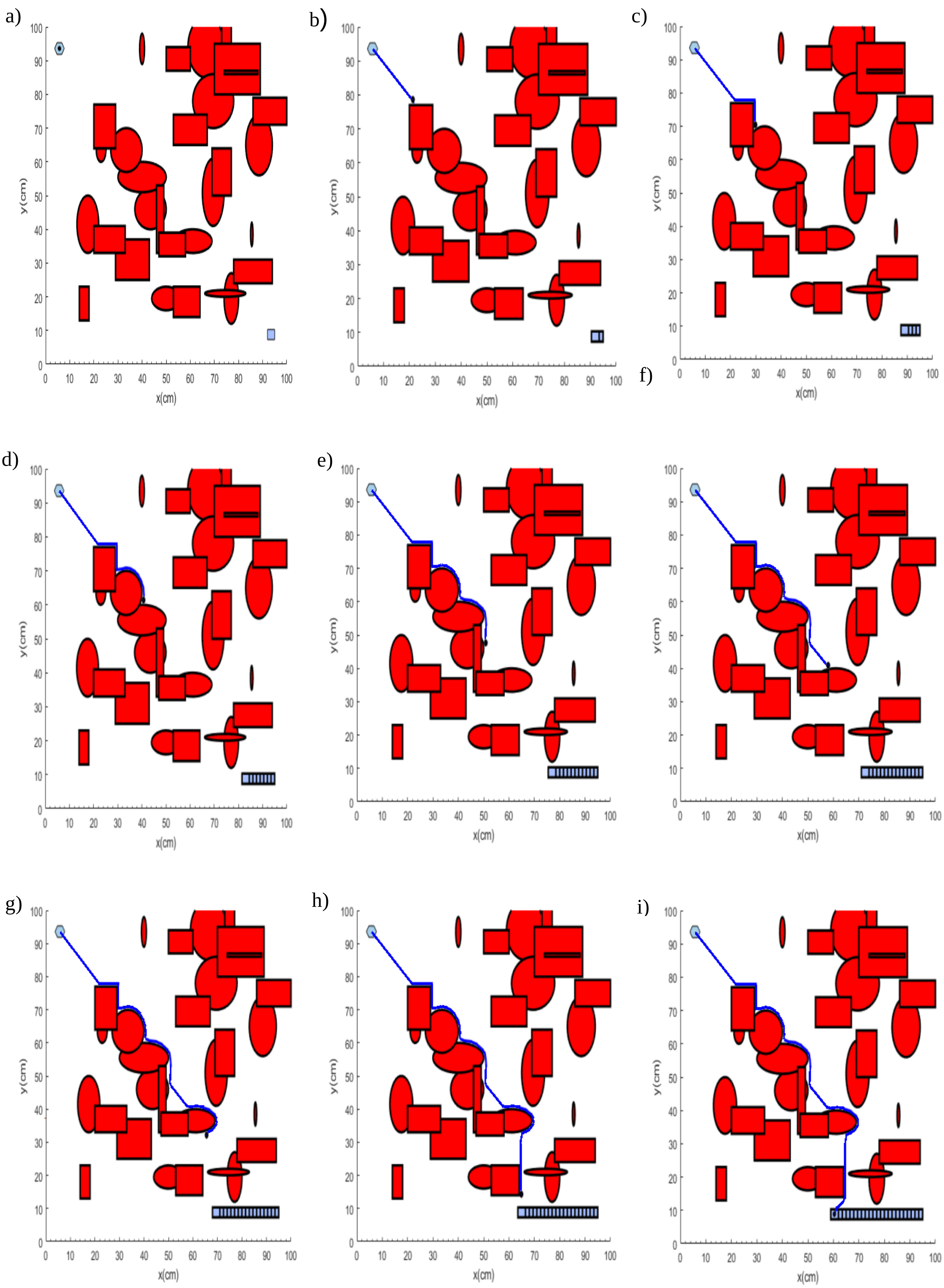


Fig. 4-7: Simulation results of the proposed algorithm in presence of dynamic target among with obstacle.

Then, the robot detects the circular obstacle and tries to prevent the collision. When the robot returns to the original path, it faces rectangular and circular obstacles (Figure 4-5h). After passing the circular obstacle, the robot can achieve the target because the path is free (Figure 4-5i). The arrival time is 38s nearly. The path length based on Euclidean distance is 122.82cm, and the minimum traveled path length by the robot is 135.33cm. The proposed algorithm is also tested for a dynamic target. The robot follows a dynamic target and reaches it successfully.

All of the movement steps are shown in Figures 4-6a to 4-6i. The arrival time is 20.1s in this scenario. The path length based on Euclidean distance is 95.13cm, and the minimum path length traveled by the robot is 80.93cm. In the fourth scenario, the proposed algorithm is tested in the existance of a dynamic target and 30 static obstacles. The robot initially navigates toward the dynamic target.

The robot moves on the original path and also follows a dynamic target until it faces the circular obstacle (Figure 4-7b). As mentioned already, the robot passes from the obstacle and it continues its path on the optimal path (Figure 4-7c). Because the rectangle obstacle is on the original path, the robot utilizes GOA to pass the given obstacle (Figure 4-7d).

To return to the original path, the robot should pass from the narrow path (Figure 4-7e and 7f). The robot encounters the circular obstacle in the path (Figure 4-7g). After passing it, the robot arrives at two rectangular obstacles and then it starts GOA from avoiding (Figure 4-7h). Finally, the robot follows a dynamic target and reaches it (Figure 4-7i). The arrival time of this scenario is 40.8s. Also, the path length based on Euclidean distance is 121.99cm, and the minimum path length traveled by the robot is 126.53cm.

In the fifth scenario, there is a combining of static and moving obstacles. The number of the static obstacles is 20 which are shown with the rectangle and curved corners and the number of dynamic obstacles are 15 that are shown with rectangles having sharp corners and circles. First, the robot navigates towards the target (Figure 4-8b). There are obstacles that are static and dynamic. When the robot senses the dynamic obstacle, a dynamic obstacle's velocity vector is recorded to predict the obstacle's trajectory for two-time intervals. The robot then concluded on the best next move based on GOA (Figure 4-8c). The dynamic obstacle is in the +x direction and the robot decides to move toward the -x-direction to prevent the dynamic obstacle (Figure 4-8d).

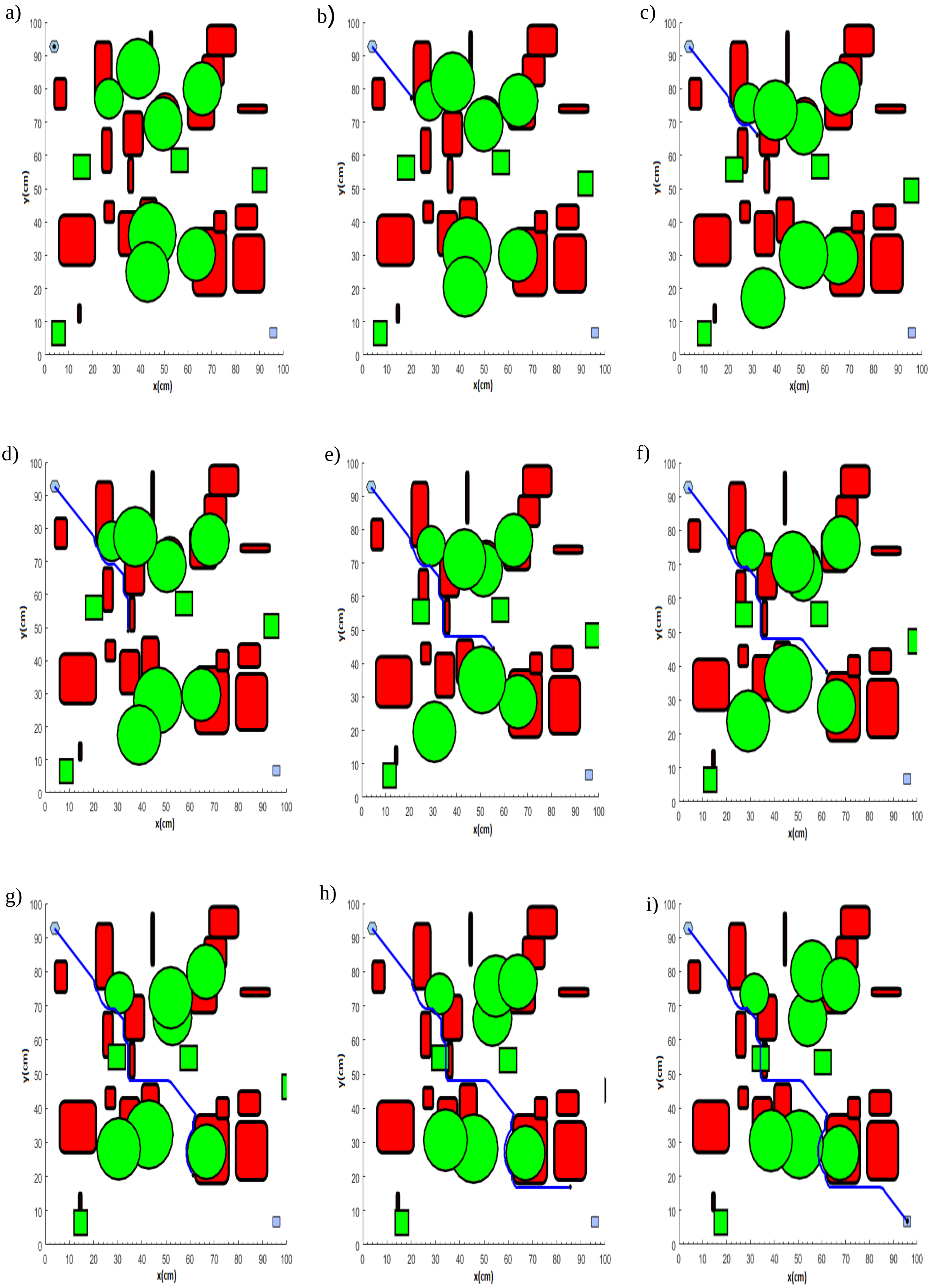


Fig. 4-8: Simulation results of unknown dynamic environmental map among with static and dynamic obstacle.

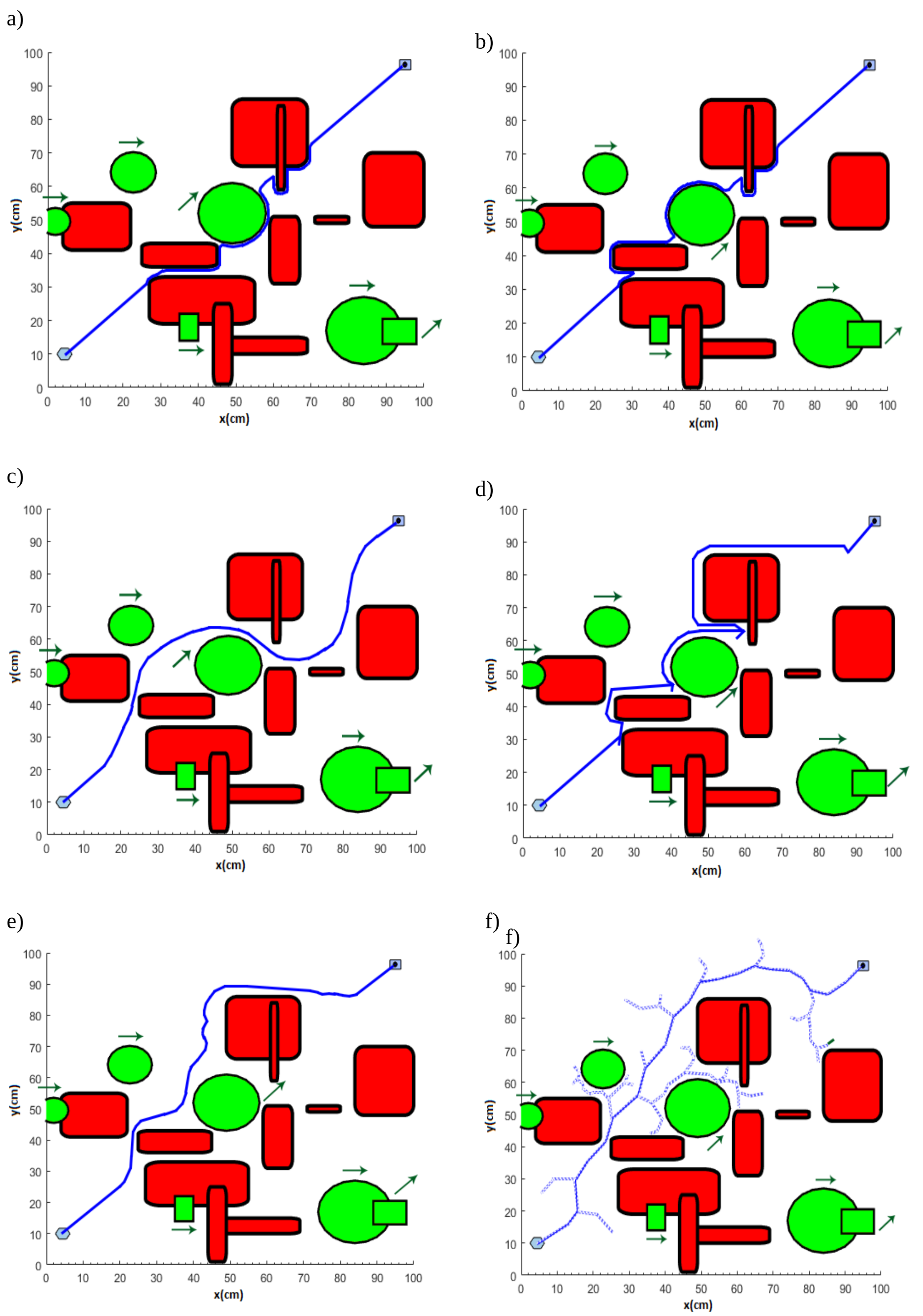


Fig. 4-9: The performance comparison results a) GOA b) PSO c) GA d) D* e) Neuro-Fuzzy f) RRT*.

After preventing the moving obstacle, the robot's navigation continues to reach to the target until it detects two static obstacles (Figure 4-8e). The robot returns to its initial location after crossing the obstacles and navigates to the target (Figure 4-8f). On the original path, there is a static and dynamic obstacle that moves circularly (Figure 4-8g). The robot navigates towards the optimal path after passing the obstacles (Figure 4-8h). Finally, the robot reaches the target (Figure 4-8i). The arrival time of this scenario is 37s. Also, the path length based on Euclidean distance is 125.74cm, and the minimum path length traveled by the robot is 140.30cm.

The proposed algorithm is successfully implemented in all scenarios. Furthermore, the results showed that the suggested approach has important features including low running time, high optimality, high stability. For all test problems, the failure rate is zero. Also, the path length and arrival time of the proposed controller for all of the scenarios are presented in Table 4-2.

Table 4-2: The path length and arrival time of proposed controller for all of scenarios.

Scenario	**Number of obstacles**	**Actual path length (cm)**	**Path length traveled by the robot (cm)**	**Arrival time (Sec)**
Static obstacles	30	121.79	149.79	40.15
Dynamic obstacles	45	122.82	135.33	38
Dynamic target	0	95.13	80.93	20.1
Dynamic target with static obstacle	30	121.99	126.53	33.8
Combination of static and dynamic obstacle	35	125.74	140.3	37

For example, as seen in Table 4-2, for static obstacle scenario, the number of obstacles, actual path length, path length traveled by the robot and arrival time are equal to 30, 121.79, 149.79 and 42.5, respectively. The performance of the proposed algorithm is checked and assessed with the several known methods such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), D*, Neuro-Fuzzy, and Rapidly-exploring Random Tree star (RRT*). The comparison results are illustrated in Figure 4-9. There are 10 static and 5 dynamic obstacles in the test environmental map that the movement direction of dynamic obstacles is shown by arrows. The actual distance from the start to target positions is 122.96cm. The efficiency of our suggested method is in comparison to the performance of the methods mentioned above With regard to safety, the length of the path, computational time and complexity. A comparison of the path lengths and computation times are presented in Table 4-3.

Table 4-3: The path length, computational time of heuristic methods for path planning.

Heuristic algorithm	**Path length (cm)**	**Arrival time (Sec)**
Grasshopper algorithm	144.34	32.02
PSO	161	68.2
GA	144.5	94.31
D*	198.08	132.8
Nero-Fuzzy	145.60	122.6
RRT*	145.96	105.5

All of the heuristic methods are able to find a safe path between the start coordinate and target coordinate in the given environment. But some of the used methods have drawbacks such as the poor quality of the resulting path, the high operating times of the planner and the inability to effectively solve complicated problems. It is clear from Table 4-3 that the path length obtained by the proposed controller is shorter than other heuristic controllers and also computation time of the proposed approach is shorter. Therefore, the proposed method can safely find a safe and collision-free path in crowded and dynamically changing environments. The advantages of using the grasshopper algorithm for optimization problems are handling both linear and nonlinear problems, high convergence speed, low computational cost. As the computational cost is low, it can be used in real-time applications.

4.5. Hybrid Controller of GOA and GA for navigation

This section presents a novel controller based on the Grasshopper optimization algorithm and Genetic algorithm to navigate the moving robot. The architecture of the suggested GOA-GA controller is shown in Figure 4-10 that calculates the desired angle of direction for optimal planning of the path in the robot environmental map. This controller consists of two parts. The inputs of the first part of the GOA-GA controller are the start coordinate of the mobile robot, target coordinate, the related information of distance and orientation of the obstacle which ha... e sensory information

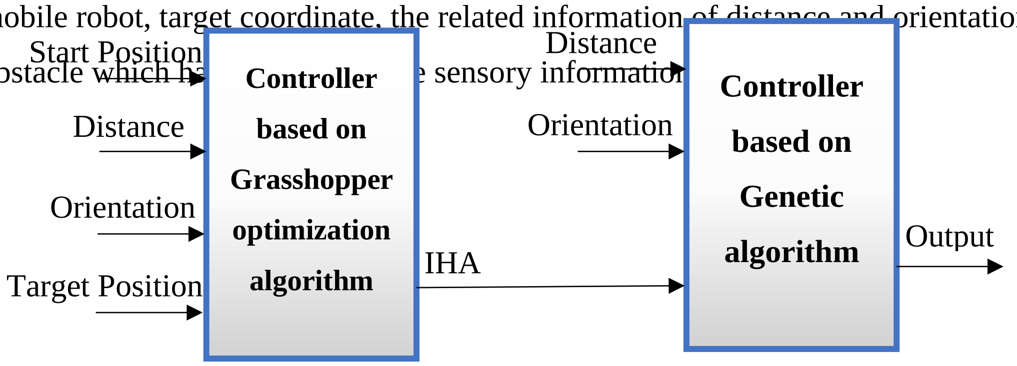
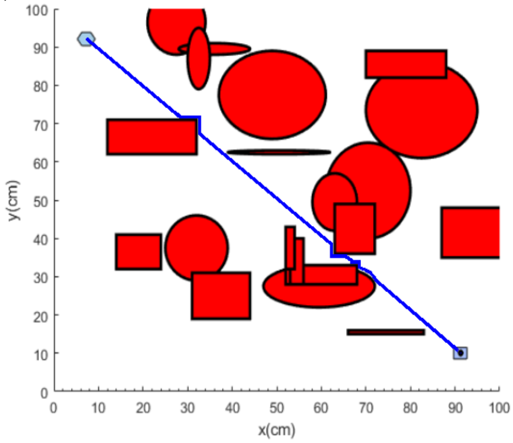


Fig. 4-10: Hybrid GOA-GA controller for navigating the mobile robot.

The function of this controller is for the intermediate heading angle (IHA). In addition to the IHA, the information of distance and orientation of the obstacle is the input of the second controller. The second controller is to be trained by using this information to obtain the final output of the proposed controller for navigating the mobile robot.

The simulation analysis of the hybrid controller based on GOA and GA and the capability of the suggested controller is also investigated in the existence of the static and moving obstacles. Figure 4-11 illustrates the results of mobile robot navigation in the presence of static and moving obstacles.

a)



b)

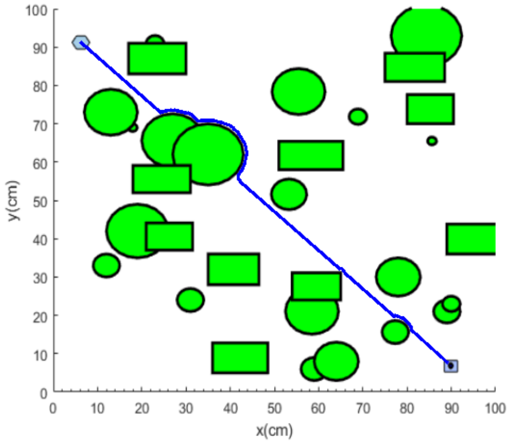


Fig. 4-11: The mobile robot navigation in the presence of a) the static and b) dynamic obstacles.

The arrival time of this scenario with 20 static obstacles is 28.4s. Also, the path length based on Euclidean distance is 117.4cm, and the minimum path length traveled by the robot is 121.84cm which is shown in Figure 4-11(a). The arrival time of this scenario with 20 dynamic obstacles is 32.3s. Also, the path length based on Euclidean distance is 118.93cm, and the minimum path length traveled by the robot is 123.18cm which is shown in Figure 4-11(b). The proposed hybrid controller has presented the efficiency of the navigation approach in the static and dynamic environmental map while preventing the obstacle.

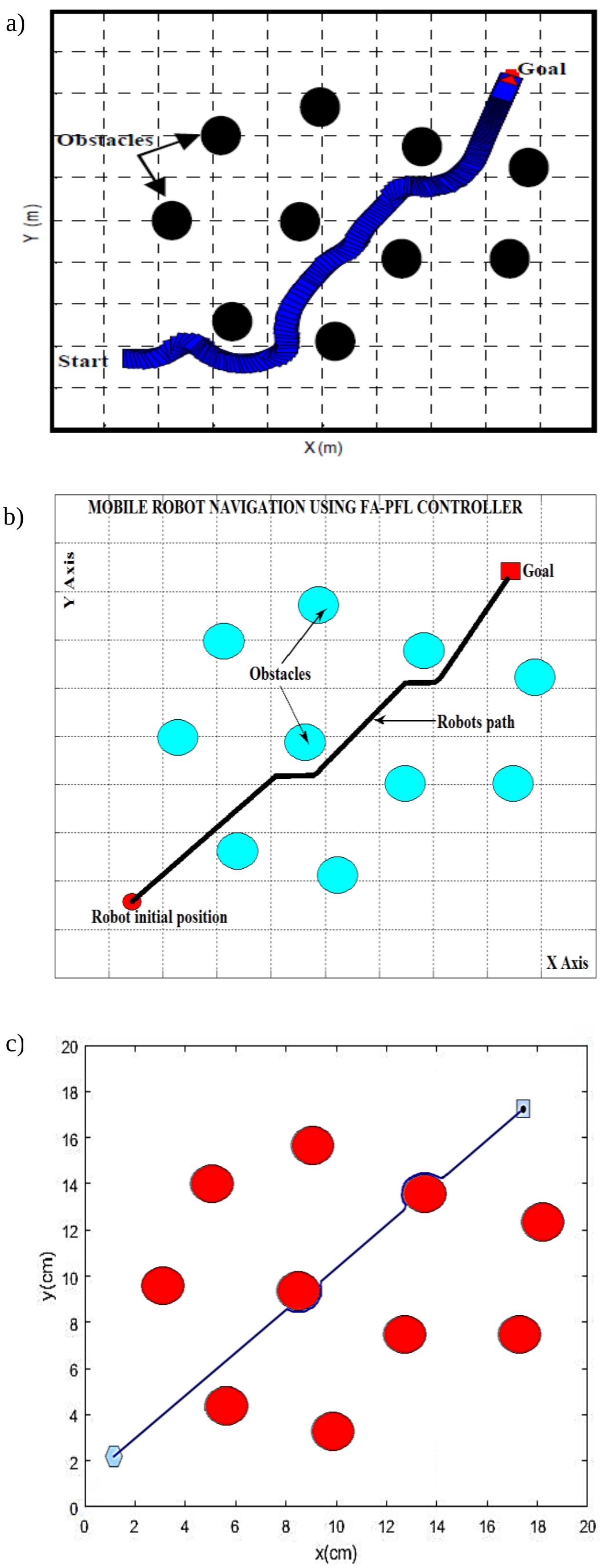
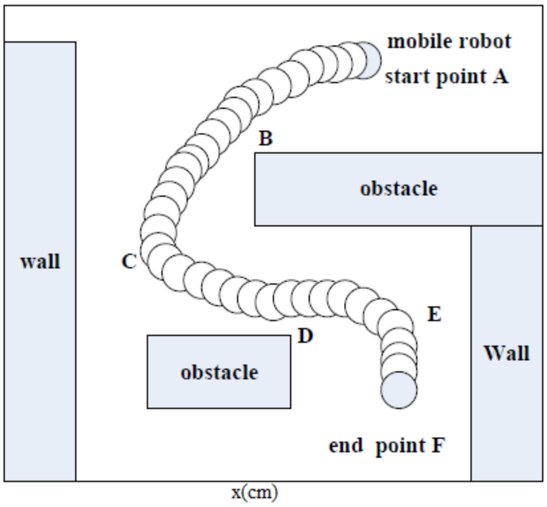
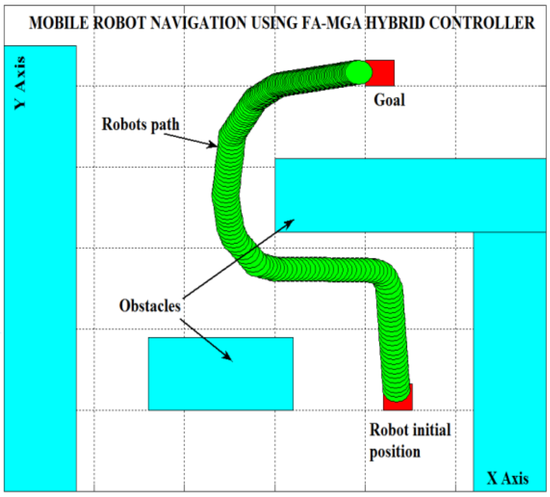


Fig. 4-12: Navigation using a) Neuro-fuzzy [223], b) the hybrid of Firefly Algorithm-Probability Fuzzy Logic [224], and c) the hybrid of proposed GOA-GA approach.

a)


mobile robot
start point A
B
obstacle
wall
C
E
D
Wall
obstacle
end point F
y(cm)
x(cm)

b)


MOBILE ROBOT NAVIGATION USING FA-MGA HYBRID CONTROLLER
Y Axis
Goal
Robots path
Obstacles
Robot initial
position
X Axis

c)

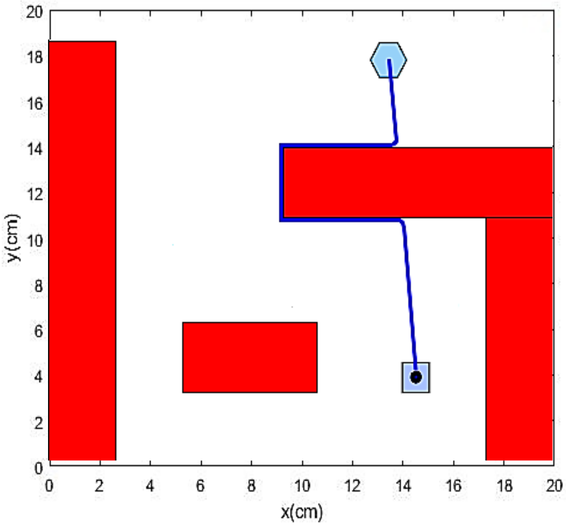

y(cm)
x(cm)
0
2
4
6
8
10
12
14
16
18
20

Fig. 4-13: Navigation using a) Fuzzy-Neural controller [225], b) the hybrid of Firefly Algorithm- Matrix based Genetic Algorithm [224], and c) the hybrid of proposed GOA-GA approach.

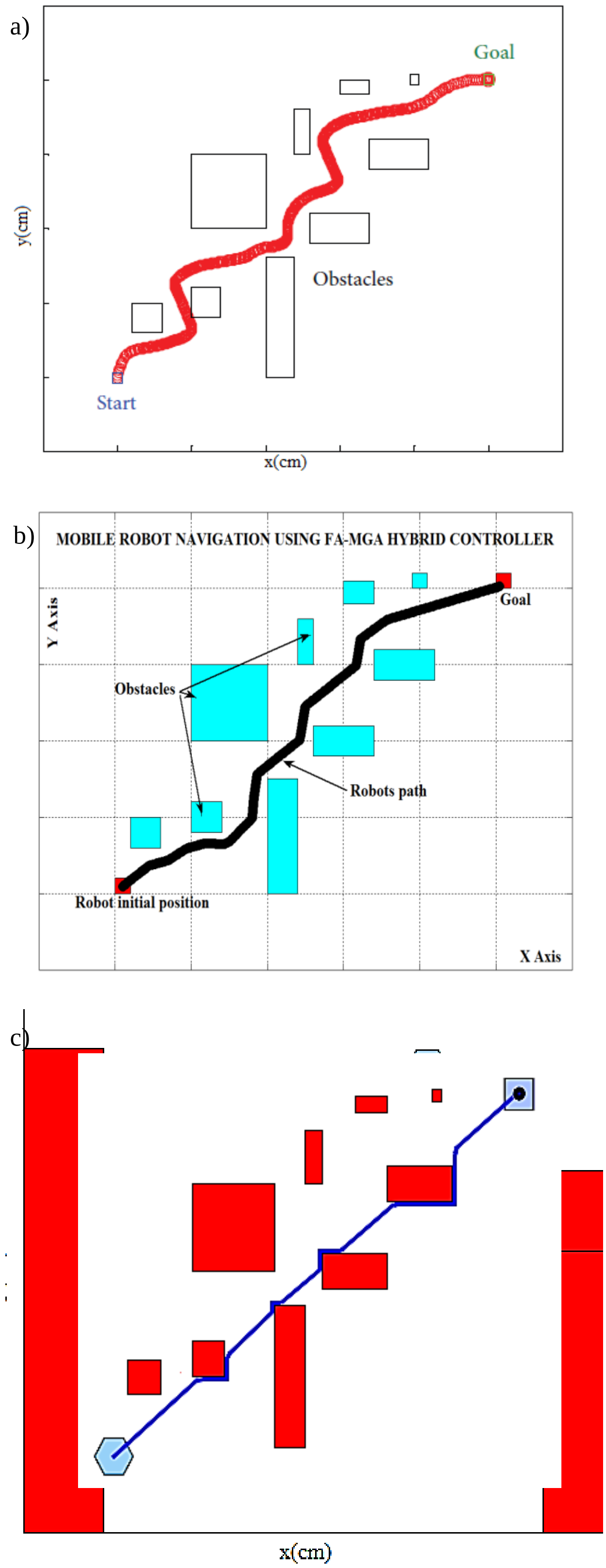


Fig. 4-14: Navigation using a) Fuzzy controller [226], b) the hybrid of Firefly Algorithm-Matrix based Genetic Algorithm [224], and c) the hybrid of proposed GOA-GA approach.

To present the sufficient performance, the compared result of the proposed controller for three different scenarios have shown in Figures 4-12, 4-13, and 4-14. Table 4-4 presents a comparison in terms of the length of path between the proposed GOA-GA controller with other AI controllers.

Table 4-4: Comparison of simulation result regarding path length.

Scenario No.	**Simulated path length in (a)**	**Simulated path length in (b)**	**Simulated path length in (c)**	**Percentage of path length Saved by the proposed GOA-GA**
Scenario-1 (Figure 4-12)	10.1	9	8.3	7.77
Scenario-2 (Figure 4-13)	12	10.1	9.6	4.96
Scenario-3 (Figure 4-14)	10.3	9.3	8.78	5.7

4.6. Summary

This chapter presents rigorous research of the grasshopper optimization algorithm to navigate moving robot in the existance of static and moving obstacles. The main achievements and results of the suggested approach is observed as follows:

- The proposed approach based on the GOA controller finds a grasshopper with the best value of the objective function at low computing cost in a minimum amount of time and considers it as the robot's new position in the environmental map. This feature provides an efficient fitness function for navigating mobile robots in the problem of path planning.

- The suggested solution succeeds in preventing obstacles (static and dynamic) and also effectively achieving the target.
- In a static and dynamic environmental map, the proposed controller is effective in handling the problem of a single mobile robot. By comparing the tabulated result with GOA-GA, it is clear that the controller proposed gives the optimum path length and navigation time. Compared to other AI controllers such as the Firefly Algorithm hybrid- Probability Fuzzy Logic and Firefly Algorithm hybrid- Matrix-based Genetic Algorithm, it saves the length of the path up to 7.77%, 4.96%, and 5.70% respectively.

As shown from the obtained results it is clear that the suggested approach based on the GOA controller and its hybrid controller with other AI approaches can be successfully used in an unpredictable environmental map to the problem of path planning.

5. MOTION PLANNING BASED ON MPC AND POTENTIAL FIELD

Today, autonomous robots can be utilized in every aspect of our daily life. Autonomous robots without human intervention are able to move into the environment and to perform their tasks safely and have a wide range of applications. In this chapter, correlations of the motion planning module with other autonomous vehicle modules have been presented. Then, the motion equations of a bicycle model are introduced. These equations are linearized and then discretized to utilize in the MPC problem. Also, the constraints of tire are presented and linearized to use in the MPC problem. Then, the potential fields are introduced for the road structure and different kinds of obstacles to produce the proper potential field. Next, the MPC-based problem is built on the model of the vehicle, the constraints, and the potential field. Motion planning based on MPC is evaluated with Carsim simulator in different and complicated scenarios, and the results are presented and discussed.

5.1. Introduction

A motion planning based on MPC has been developed in this chapter, which is utilized the potential field functions to avoid an obstacle. The presented motion planning based on MPC is quadratic. Therefore, its objective function should be quadratic and its model and its constraints should be linear. For modeling the vehicle's behavior in MPC, a linear bicycle-vehicle model is used. The proposed model utilizes the linear model of tire, and the constraints of tire are included in the MPC to keep the model validity in different environments. The tires have limited capacities which consider the constraints by including tire combined slip limitations and the effect of longitudinal load transfer on the limitations. Since the model of vehicle considers its limitations through the constraints, by using these constraints, MPC can generate the feasible trajectories on the constraints of tire force. Also, the constraints keep the tire in linear lateral force region to keep valid the linear models of tire and the model of vehicle.

5.2. Architecture of Autonomous Vehicle

The correlations of the motion planning module with other modules of an autonomous vehicle are introduced in this section. Figure 5-1 has demonstrated correlations through the architecture of an autonomous vehicle. The module of motion planning plans the vehicle's path while avoids the collision of the obstacle fulfills the road rules, follows the eligible commands, and supports the passenger's comfort.The motion planning module receives information about the obstacles, road, vehicle, and desired commands from the other modules.

The information of an obstacle consists of each obstacle's location, velocity, size, and classification. The information of a road consists of the road structure, number of lanes, and width of the lane. The information of the vehicle is the position of the vehicle, heading angle, lateral and longitudinal velocities, yaw rate, and normal forces of the tire. The information of obstacle, road, and vehicle have provided for the motion planning module from the perception module [227] and the estimation module [228]. In addition, the eligible commands are the desirable lane and velocity created by the behavior planning module.

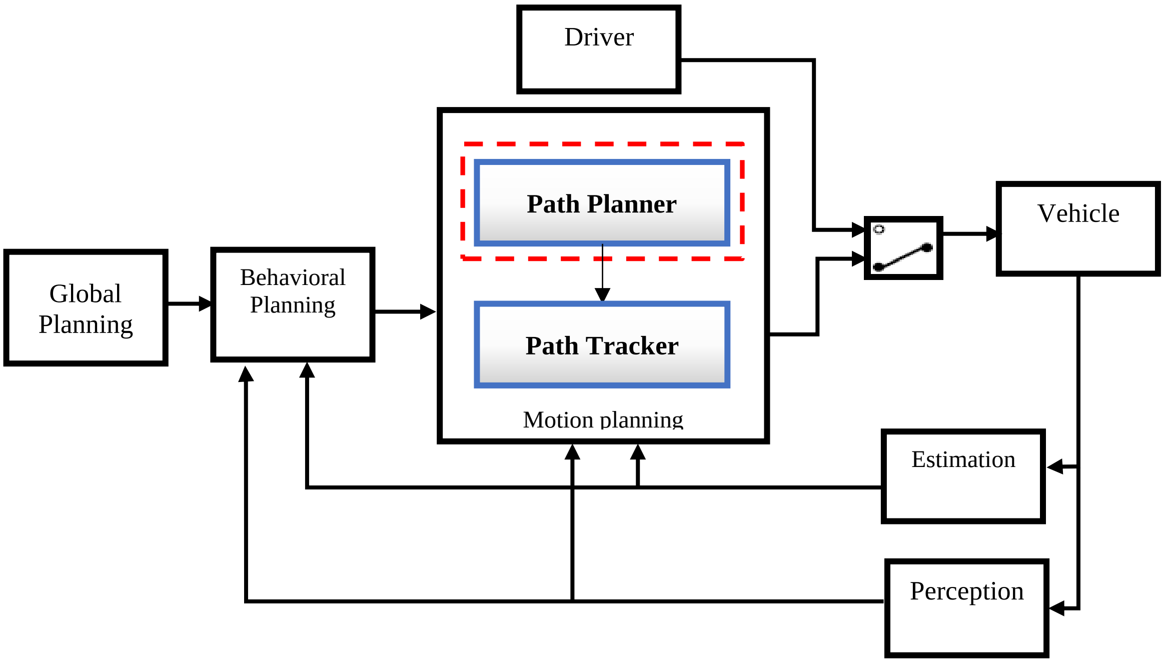


Fig. 5-1: The motion planning for the autonomous vehicle.

Each obstacle moves with as the same lateral and longitudinal velocities as current velocities to predict its position in MPC. The risk due to uncertain behaviors of the obstacle and errors in the state's estimation of the vehicle and obstacle has been considered in generating the safe distance of the potential functions. The running step time for MPC is 50*ms*, and any change in the obstacle behavior is rapidly considered in the motion planning of the vehicle.

The module of motion planning computes driving instructions that include the angle of the front steering and full instructions of longitudinal force.

5.3. The models of the vehicle

Both kinematic and dynamic vehicle models are provided with their hypothesis and constraints in this section.

5.3.1. Kinematic model

Kinematic is a branch of classic mechanic that explains the motion of points, bodies, and group of objects without considering forces which effects on the motion. The motion equations described by a kinematic model present the geometry relationships that control the system. For this reason, it is often known as motion geometry. A kinematic problem consists of the geometry description and deceleration of initial conditions of values relating to the position, acceleration, and speed of system points. The kinematic model of the vehicle is investigated in Figure 5-2.

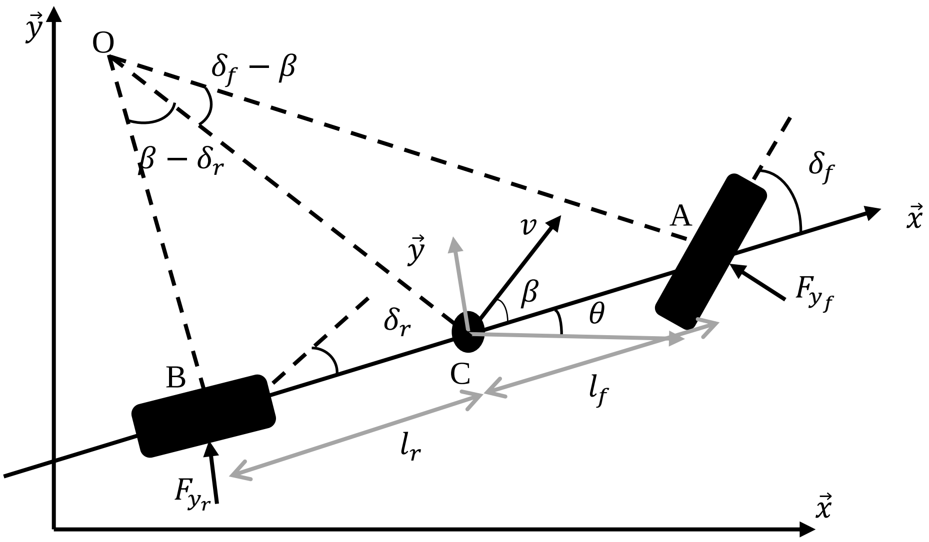


Fig. 5-2: The kinematic model of bicycle vehicle.

The above figure illustrates a bicycle model in which one center tire in *A* and *B* points is respectively displayed for two front wheels and two rear wheels. The front and rear wheels steering angle is respectively indicated by δ_f and δ_r. To model the vehicle, the angle of the front steering is only considered and the angle of the rear steering (δ_r) is set to zero.

The center of gravity of the vehicle (CoG) is point *C* in the figure. The distances between the given point and points *A* and *B* are respectively shown by l_f and l_r,. The sum of these two terms is the vehicle's wheelbase *L* as is shown below:

$$L = l_f + l_r \tag{5-1}$$

The vehicle has the planar motion; therefore, three necessary coordinates describe the vehicle motion: *X*, *Y*, and θ. The *X* and *Y* are inertial coordinates of the vehicle's center of gravity position, also θ is yaw angle that shows the orientation of the vehicle. The *v* vector is the speed at the center of the vehicle's gravity. This vector renders a β angle with the vehicle's longitudinal axis. It's called the angle of the slip.

The *O* point is the vehicle's immediate center of rotation and is determined by the *AO* and *BO* lines intersection. Both two lines are drawn a perpendicular to the two-wheel orientation. The length of the *OC* is the radius of the trajectory of the vehicle *R* and it is perpendicular to the vector *v*.

By applying the sine rule to triangles *OCA* and *OCB*, the equations can be defined as follows (the δ_r is equal to zero):

$$\frac{\sin(\delta_f-\beta)}{l_f} = \frac{\sin(\frac{\pi}{2}-\delta_f)}{R} \tag{5-2}$$

$$\frac{\sin(\beta)}{l_r} = \frac{1}{R} \tag{5-3}$$

By multiplying Eq. (5-2) with $\frac{l_f}{\cos(\delta_f)}$, the given equation becomes:

$$\tan(\delta_f)\cos(\beta) - \sin(\beta) = \frac{l_f}{R} \tag{5-4}$$

Also, by multiplying Eq. (5-3) with l_r, the equation (5-3) can be rewritten as:

$$\sin(\beta) = \frac{l_r}{R} \tag{5-5}$$

The equations (5-4) and (5-5) have added, the obtained equation is as below:

$$\tan(\delta_f)\cos(\beta) = \frac{l_f+l_r}{R} \tag{5-6}$$

This formula writes the radius R of the vehicle trajectory as a function of the front steering angle δ_f , the slip angle β, and l_f. If the R radius value changes slowly for reason low velocity, vehicles' yaw rate $\dot{\theta}$ is equal to the angular velocity ω that is defined as follows:

$$\omega = \frac{v}{R} \tag{5-7}$$

Hence, the yaw rate $\dot{\theta}$ can be written as below:

$$\dot{\theta} = \frac{v}{R} \tag{5-8}$$

By using Eq. (5-6), the equation (5-8) can be re-written as:

$$\dot{\theta} = \frac{v cos(\beta)}{l_f + l_r} \tan(\delta_f) \tag{5-9}$$

After all these hypothesizes, the overall equations of the kinematic model can be defined as follows:

$$\dot{X} = v cos(\theta + \beta) \tag{5-10}$$

$$\dot{Y} = v sin(\theta + \beta) \tag{5-11}$$

$$\dot{\theta} = \frac{v cos(\beta)}{l_f + l_r} \tan(\delta_f) \tag{5-12}$$

5.3.2. Dynamic model

When the velocity and the steering angle of the vehicle are low, the kinematic model provides satisfactory results; but when the speeds of the vehicle increase and the curvatures of the trajectory change during time, it can not be assumed that each wheel's velocity vector is parallel to the symmetry plane of the wheel. Thus, the vehicle dynamic model is developed instead of using a kinematic model.

The model of the bicycle is utilized to model the dynamics of the vehicle. As mentioned above, in this model, the vehicle's two front wheels are mounted on a single wheel in the center of the front axle and the two rear wheels are located in the center of the rear axle. Meanwhile, the model of the vehicle is shown in Figure 5-2. The bicycle model's motion equations are as follows:

$$m(\dot{v}_x - v_y r) = F_{x_T} \tag{5-13}$$

$$m(\dot{v}_y + v_x r) = F_{y_f} + F_{y_r} \tag{5-14}$$

$$I_z \dot{r} = l_f F_{y_f} - l_r F_{y_r} \tag{5-15}$$

$$\dot{\theta} = r \tag{5-16}$$

$$\dot{X} = v_x cos\theta - v_y sin\theta \tag{5-17}$$

$$\dot{Y} = v_y cos\theta + v_x sin\theta \tag{5-18}$$

$$I_z \dot{r} = l_f F_{y_f} + l_r F_{y_r} \tag{5-19}$$

where m is the mass of the vehicle. v_x, v_y, and r are the vehicle's longitudinal velocity, lateral velocity, and yaw rate at its center of gravity, respectively. X and Y are the vehicle's longitudinal and lateral positions and θ is the vehicle's heading angle. I_z is the vehicle momentum of inertia around its vertical axis. F_{x_T} is the total longitudinal force of tires. Also, F_{y_f} and F_{y_r} are the total lateral forces of the front and rear tires. When the vehicle has a front steering system, a linear version of the tire is used for the lateral forces of the tire [218].

$$F_{y_f} = c_f \alpha_f = c_f (\delta - \frac{v_y + l_f r}{v_x}) \tag{5-20}$$

$$F_{y_r} = c_r \alpha_r = c_r (- \frac{v_y + l_r r}{v_x}) \tag{5-21}$$

where α_f and α_r are the sideslip angles of the front and rear tires and δ is the angle of steering. In the model, c_f and c_r show the front and rear tire cornering stiffness values. Based on [218], these values are determined.

The linear dynamics of the vehicle can be derived as follows by linearizing equations (5-13) -(5-21):

$$x^{k+1} = A^k x^k + B^k u_c^k + C^k \qquad k=0 \dots n\text{-}1 \tag{5-22}$$

$$x = [X \;\; v_x \;\; Y \;\; v_y \;\; \theta \;\; r]^T \tag{5-23}$$

$$u_c = [F_{x_T} \quad \delta]^T \tag{5-24}$$

where x is the vector of the state, u_c is the vector of the input. A is the matrix of the state, B and C are the matrices of the input. Using the linear model as a basis for MPC, discretizing the model is essential in individual time steps along the horizon of prediction, therefore, obtained an affine time-varying model. The zero-order holding (ZOH) approximation at each point has often discretized continuous dynamic models. For the short time stages, a ZOH assumption works well because the continuous inputs assumed constant value and accurately reflected the vehicle's actual input. In order to discretize the model at *long* time steps, a first-order hold (FOH) approximation is utilized that is proposed by Brown et al. [236].

This method allows a more detailed interpolation of inputs and a better estimation of future state distribution. By using a matrix exponential approach [237] with a ZOH, the proposed vehicle model has been discretized for the first ten short time steps and FOH has also been used for the next 20 long time. The vehicle's discrete model is as follows:

$$x^{k+1} = \begin{cases} A^k x^k + B^k u_c^k + C^k & k = 0 \dots 9 \\ A^k x^k + B_1^k u_c^k + B_2^k u_c^{k+1} + C^k & k = 10 \dots 29 \end{cases} \tag{5-25}$$

where A^k, B_1^k, B_2^k, and C^k matrices of linearization that vary in time along the horizon of prediction.

5.3.2.1. Constraints on the vehicle

A vehicle has limits on the actuator's capabilities and the tire's force capabilities. The actuator's capacities are known as constraints:

$$|\delta| \leq \delta_{max} \tag{5-26}$$

$$F_x \leq \frac{T_{max}}{R_{eff}} \tag{5-27}$$

where δ_{max} is maximum steering angle, T_{max} is the maximum propelling torque, and R_{eff} is the efficient wheel radius. The vehicle's propulsion system contains four in-wheel motors in this dissertation, therefore, T_{max} is the overall sum of the maximum motor torques. Moreover, the wheel dynamics are neglected in Eq. (5-27). Limitations are introduced to the change in the steering angle:

$$|\Delta\delta| \leq \Delta\delta_{max}$$
(5-28)

where $\Delta\delta$ is the steering angle change in one phase, and $\Delta\delta_{max}$ shows its capacity.

To have an accurate prediction, the Equation (5-29) limitation should be considered in the prediction of the model predictive controller because the tire longitudinal and lateral force can not exceed the friction ellipse.

$$\left(\frac{F_{x_T}}{F_{x_{T-max}}}\right)^2 + \left(\frac{F_{y_{f,r}}}{F_{y_{f,r-max}}}\right)^2 \leq \mu^2 \tag{5-29}$$

where $F_{x_{T-max}}$ is the maximum longitudinal force of the tire, $F_{y_{f,r-max}}$ is the maximum front or rear lateral force of the tire, and μ is the friction coefficient of the tire. To remain within the linear region, the mentioned constraints also restrict the tires' lateral forces. The load transfer relies on the maximum forces in the constraint equations of (5-29). Because the dynamics of the bicycle vehicle takes into account the tires' maximum forces on the same wheel axis, the load lateral transfer is neglected. The effects of load longitudinal transfer on the vertical forces of the front and rear tires are:

$$F_{z_f} = \frac{Wl_r - F_{x_T}h}{l_f + l_r} \tag{5-30}$$

$$F_{z_r} = \frac{Wl_f - F_{x_T}h}{l_f + l_r} \tag{5-31}$$

where $F_{z_{f,r}}$ is the vertical force of the front or rear tire, h is the height of the center of gravity of the vehicle from the ground, W is the weight of the vehicle. The maximum total longitudinal force of the tire is not affected by load transfer. However, the transfer of longitudinal load impacts the total lateral force of the equation (5-29). It is assumed that the lateral force capacity of the tire modifies linearly with regard to the vertical force of the tire:

$$F_{y_{f,r-max}} = F_{y_{f,r0-max}} \frac{F_{z_{f,r}}}{F_{z_{f,r0}}} \tag{5-32}$$

where $F_{y_{f,r0-max}}$ and $F_{z_{f,r0}}$ are the nominal maximum lateral force of the front or rear tire and the nominal vertical force of the front or rear tire, which the nominal force is the force without load transfer.

The effect of load longitudinal transfer is added in the tire' force ellipse constraints by applying (5-32) and (5-31) on (5-29):

$$\left(\frac{F_{x_T}}{F_{x_{T-max}}}\right)^2 + \left(\frac{F_{y_f}}{F_{y_{f0-max}}} * \frac{Wl_r}{Wl_r - F_{x_T}h}\right)^2 \leq \mu^2 \tag{5-33}$$

$$\left(\frac{F_{x_T}}{F_{x_{T-max}}}\right)^2 + \left(\frac{F_{y_r}}{F_{y_{r0-max}}} * \frac{Wl_r}{Wl_f - F_{x_T}h}\right)^2 \leq \mu^2 \tag{5-34}$$

Additionally, the vehicle's speed should not exceed the limit on maximum speed. It is taken into account as a constraint:

$$u \leq u_{max} \tag{5-35}$$

where u_{max} shows the maximum allowed speed of the vehicle.

5.4. Artificial Potential Field Function (APF)

As already mentioned, the potential field method is built on attractive and repulsive factors that allow a vehicle to travel towards the destination due to the attractive factor while the repulsive function prevents from colliding the vehicle with obstacles. The target potential field has a minimum value in the target point, then the vehicle is attracted, the obstacle Potential Function (PF) has a maximum value in the obstacle positions and repels the vehicle from the obstacle [238]. The main objective of this thesis is to navigate the vehicle to the target point without any collision by tracking the objective function controller. Therefore, only the repulsive function is considered as PF. Potential field function is developed based on three metrics: the obstacle (U_O), the boundaries of the road (U_R) and the center of the lane (U_C). By reflecting the predicted surrounding environment, the sum of potential field functions is derived at each time of prediction. Obstacle vehicles are estimated by a model of constant velocity and information of these vehicles is also taken into consideration in real-time. The generally potential field is the sum of the PFs:

$$U_{tot} = \lambda_r U_R + \lambda_o U_O + \lambda_C U_C \tag{5-36}$$

where λ_r, λ_o and λ_C are weights PF for road, obstacle, and center of the lane, respectively. To model road laws and obstacles, using other functions is also possible.

5.4.1. Lane Marker PF

The potential field of the lane marker is used to avoid the vehicle from leaving the main road and driving too close to the borders to increase the risk of an accident. Therefore, in road boundaries, the lane marker should have a maximum value. In addition, the slope to achieve this maximum value is maximum and provides maximum value to restore power. In the meantime, to avoid changing lanes, a maximum value at the driving lane position is developed. The vehicle is, therefore, making an attempt to retain its present lane in order to avoid the costs involved. For this reason, the PF in the center of the lane is zero and locally symmetric, which is the preferred position when the vehicle does not face traffic or obstacles. The vehicle can overcome this resistance when changing the lane is necessary. Therefore, we use a 1D Gaussian function that approaches the left or right road boundary for a higher potential value. The proposed lane marker PF (U_R) is as follows:

$$U_R = A_r \exp\left(-\frac{(Y_h - Y_r)^2}{2\sigma_{rb}^2}\right) + A_r \exp\left(-\frac{(Y_h - Y_l)^2}{2\sigma_{rb}^2}\right) \tag{5-37}$$

where A_r is the maximum value of the road boundary potential field. Y_h is the lateral vehicle position from CoG in local road frame and Y_r, Y_l are lateral positions in right and left of the straight road center, respectively. σ_{rb} is the parameter for the road boundary potential field. It is assumed that the autonomous vehicle is driving on the highway and that the geometric shape of the road boundary is considered to be a first-order polynomial function. Figure 5-3(a) shows the 3D plot of the potential field of road boundary on the straight road with $Y_r=-3.8$, $Y_l=3.8$, $A_r=4$, and $\sigma_{rb}=1$.

5.4.2. Obstacle Potential Field

Obstacle PF (U_O) structure is more complex and important than road PF structure. The lane change maneuver is performed when the obstacle or the vehicle hits the ego vehicle according to the obstacle PF. This is based on the highway-driving structure and protocol. The vehicle could also switch to the left side to overtake the slower vehicles that preceded it. To accomplish this, obstacle PF is modeled as a function of the obstacle vehicle's measured location, the relative and absolute speed of ego and obstacle vehicles, and the curvature of the lane. Through usable sensor measurements from the obstacle, the location of the PF of obstacle is obtained. The longitudinal and lateral distances between the obstacle and the ego vehicle provided by x_O and y_O are the information obtained and do not include the obstacle vehicle's heading angle.

The obstacle vehicle's shape is taken into account rectangular as it gives a better estimate of an obstacle's outline. In addition, the continuous functions are required to represent the obstacle value such as hyperbolic function in order to avoid slope discontinuities in PF. This function is used as the distance that generates the desired potential field between the ego and obstacle vehicles. When the distance between the ego vehicle and the obstacle is too small the function's change rate increases significantly thus its value tends to infinite, preventing ego vehicle collision with the obstacle. The used repulsive potential function for an obstacle vehicle is as follows:

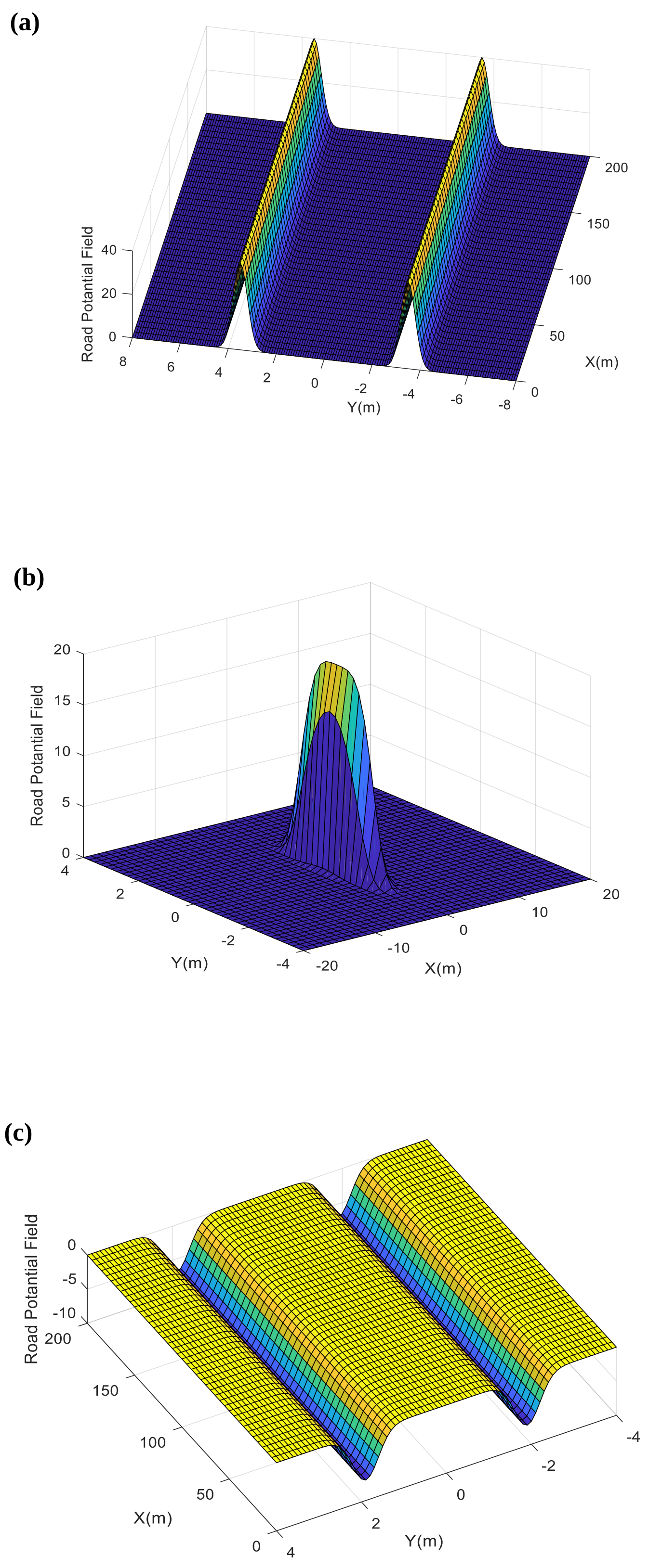


Fig. 5-3: The potential field function (a) road boundary (b) obstacle or surrounding vehicle (c) lane centering.

$$U_o = A_{obs}\exp\left(-\left(\frac{(x-x_{obs})^2}{2\sigma_x^2}+\frac{(y-y_{obs})^2}{2\sigma_y^2}\right)^c\right) \tag{5-38}$$

where A_{obs} shows the maximum potential field value of the obstacle. (x, y) is the vehicle's current position and (x_{obs}, y_{obs}) signifies the ego vehicle's closest point from the obstacle. σ_x and σ_y are the integration coefficient of the potential field of obstacle that determines the spread of the horizontal impact of the potential field. In (5-38), c is a coefficient for adjusting the form of the peak of obstacle's potential field. When the ego and obstacle are approaching in each direction, the difference of velocity between is their velocity otherwise, it is set to zero. The obstacle potential field can be placed in $(x_{obs}, y_{obs}) = (0,0)$, also $\sigma_x = \sigma_y = 1$ and $A_{obs}=20$ are considered and the obstacle potential field is shown in Figure 5-3(b).

5.4.3. Potential Field of Lane Centering

If there is not an imminent collision or the potential field of the nearby vehicle is greater than the defined threshold, and the autonomous vehicle should follow the centerline of the lane, therefore using a potential field function for the centerline of the lane (U_C) is also presented

$$. \ U_C = -A_C exp\left(-\frac{(y-y_C)^2}{2\sigma_C^2}\right) \tag{5-39}$$

where A_C is the maximum potential field value for the centerline. y_C is the lateral distance to the center of the lst lane on the specified coordinate of the road and σ_C is a parameter for the lane centerline potential field. A first-order polynomial function acquires the geometric shape of the potential field. Figure 5-3(c) shows the potential area of lane centering with $A_C=7$, $y_{Cr}=-1.9$, $y_{Cl}=1.9$, and $\sigma_C=1$. In near the centerline, the values of the potential field are negative since the centering corresponds to reward in autonomous driving.

5.4.4. MPC Framework

The MPC approach is related to combinating of optimal and adaptive control schemes. The method uses a controller-based model involved in the model's estimated state optimization step to generate the optimal control input. The MPC can adapt to changing circumstances, which is why it is similar to the adaptive controller. It manages input and output limitations at each control interval to resolve the problem of optimization.

According to these characteristics, MPC is an acceptable candidate for path planning and tracking based on the potential field.

A model predictive controller can be introduced based on the dynamic model of vehicle, potential field, and road regulations. An optimization problem of conflicting demands can be identified by using these objectives. The model predictive controller predicts the ego vehicle's response on a horizon called the prediction horizon (N) and optimizes vehicle response, evading obstacles, road regulation and command following based on this value. The preferred lane and speed are predetermined. Consequently, the desired lateral position (the center of the desired lane) and longitudinal velocity are the system outputs to be tracked:

$$y = [Y \quad v_x]^T \tag{5-40}$$

$$y_{des} = [Y_{des} \quad v_{x_{des}}]^T \tag{5-41}$$

$$Y_{des} = \left(l_{des} - \frac{1}{2}\right) L_w + \Delta Y_R \tag{5-42}$$

where y is the output matrix tracking, y_{des} is the desired lateral position, $v_{x_{des}}$ is the desired speed, l_{des} is the index number of the desired lane from the right, L_w is the lane width and ΔY_R is the lateral offset from the straight road. For path planning, the nonlinear problem of optimization can be formulated as follows:

$$\min_{u_c,\,\varepsilon} \sum_{i=1}^{N} \|y(k+i|k) - Qy_{des}(k+i|k)\|^2 + \|u_c(k+i-1|k) - Ru_c(k+i-2|k)\|^2 + \|u_c(k+i-1|k)\|^2 + U_R(k+i|k) + U_O(k+i|k) + \|\varepsilon_k\|^2 \tag{5-42}$$

$$s.t. \quad x(k+i|k) = A_d x(k+i-1|k) + B_d u_c(k+i-1|k) + C_d \tag{5-43}$$

$$y(k+i|k) = D_d x(k+i-1|k) + E_d u_c(k+i-1|k) \tag{5-44}$$

$$v_{x_{min}} < v_x < v_{x_{max}} \tag{5-45}$$

$$\left(\frac{F_{x_T}}{F_{x_T-max}}\right)^2 + \left(\frac{F_{y_{f,r}}}{F_{y_{f,r}-max}}\right)^2 < 1 \tag{5-46}$$

$$\varepsilon_k \geq 0 \tag{5-47}$$

$$\varepsilon_{k+1} = \varepsilon_k \qquad k \neq c_1 N_{rs} + 1, \qquad c_1 = 1,2,\dots, N/N_{rs} \tag{5-26}$$

$$u_{c-min} < u_c(k+i-1|k) < u_{c-max} \tag{5-48}$$

$$\Delta u_{c-min} < u_c(k+i-1|k) - u_c(k+i-2|k) < \Delta u_{c-max} \tag{5-49}$$

$$u_c(k+i|k) = u_c(k+i-1|k), \quad k > N_c, \quad k \neq c_2 N_{rc} + N_c,$$

$$c_2 = 1,2, \dots, (N - N_c)/N_{rc} \tag{5-50}$$

where $(k+i|k)$ index shows the potential values of $k+i$ at future time $k+i$ which is estimated at current time k. ε_k is the slack variables vector at k time. The objective function comprises of the quadratic term of tracking, changes in inputs, potential field functions, and slack variables. This parameter permits some violation and defines a penalty term for the purpose that can be used to penalize the violation. Q and R are the tuning matrices of the controller. The predicted states are obtained by (5-43). A_d, B_d and C_d are discrete matrices of state and input obtained through discretion (5-9). The tracking output is calculated by (5-44) and D_d, E_d are output and feedforward matrices. The speed and octagon approximation limitations are applied as soft constraints represented in equations (5-45) and (5-46), respectively.

These constraints are considered due to certain road regulations concerning the range of minimum and maximum speed and the longitudinal and lateral forces of the tire that can not exceed the ellipse of friction. By reducing the number of slack variables and control inputs in (5-47) and (5-50), the cost of the computation can be decreased. The vector of slack variable changes in every N_{rs} prediction steps, and after the first N_c prediction steps, the inputs of control change in every N_{rc} steps. The control inputs and their modifications are limited in (5-48) and (5-49) to meet the actuator limitations, where $u_{c\text{-}min}$ and $u_{c\text{-}max}$ are the matrices of lower and upper bounds of control input, and Δu_{c-min} and Δu_{c-max} are the matrices of lower and upper bound of the control inputs changes.

The given potential field is a non-convex and non-linear function; therefore the problem of optimization is non-convex and non-linear and its solution is costly. Thus, to reduce the computational time, the problem is converted into a quadratic and convex problem. For this purpose, PFs are first approximated by convex functions. Then, through the second-order Taylor series, the resulting convex function is approximated by a quadratic function. The obtained function is a closed convex quadratic approximation of the original function around the nominal point. The obtained gradient is equal to the gradient of the original function and also the Hessian matrix of approximated function is the closest positive definite matrix to the Hessian matrix of the original function in terms of Frobenius norm. Although the quadric approximation of the PFs increases the calculation time, the added time is negligible compared to the time needed for solving a nonlinear optimization problem [239].

Using these PFs, the problem of optimal control is a convex quadratic optimization problem. This problem is similar to a nonlinear problem solved by Sequential Quadratic Programming (SQP) in one sequence. An upper bound for the optimization error of each sequence of SQP is derived by Bogges et al. [240], where this error is the difference between the sequence result and local minimum of the nonlinear problem in the proximity of the initial value of the problem. According to this upper bound, if the initial value of the problem is closer to a minimum, the optimization error will be minimized and the vehicle's expected location will be equal to the vehicle's position at the minimum stage. Moreover, in the case of the Hessian matrix, the closer calculated ones of PFs to their values at the minimum will reduce the optimization error. So, in the neighborhood of the initial value of the problem, a PF with a smaller convex quadratic approximation error and a smaller variation of the Hessian matrix result in a smaller optimization error.

5.5. Simulation and Results

5.5.1. Scenarios

Path planning and control design are the most challenging issues in the realm of autonomous vehicles. Path planning in dynamic or structured settings such as roads involves global and local path planning in which, local path planning, international path planning are used. Urban path planning is a slow and deliberative method that is used to reach the target on long-distance paths. While local path planning is a faster process and is used for short-distance paths and deals with tasks such as vehicle stability, obstacle avoidance, comfortable and safety. This planner is more reactive and runs in real-time.

Driving on constructed roads can be simplified into two basic vehicle maneuvers, namely holding the lane and adjusting the lane. The main objective of lane management is to track the vehicle and remain in its current location by changing its direction and distance to the middle of the lane continuously. Lane changing is the most common maneuver in which vehicle changes its current lane to overtaking, obstacle avoidance and road departure. The maneuver may be different depending on the road, given the lane and obstacles on the road.

There are many maneuvers that arise in reality. By observing these maneuvers, safety and road regulations, the performance of path planning systems can be evaluated. If the path of the vehicle is safe, the vehicle can keep its lane. Otherwise, a lane changing needs to be planned and executed. This changing lane happens at the end of the road or when it faces another obstacle in its own path. At the end of the road, if there is no obstacle or another vehicle on the intended lane, changing lane is accomplished. Otherwise, the vehicle should reduce its speed and even might stop before the lane ends. When the vehicle faces an obstacle on its path, the vehicle must predict the path of the obstacle. If there is enough lateral distance, the vehicle could pass the obstacle, on the other hand, the vehicle is changing lane to overtake. Otherwise, the vehicle should stop behind the obstacle or cross it. These are the many maneuver samples that are taking place on the road.

Many scenarios are provided below to evaluate the performance of the autonomous vehicle:

1. Keeping Lane on the straight and curved road.
2. Maintaining a reasonable distance from the vehicle in front of the ego vehicle.
3. Changing lane with moving obstacle on the curved road.
4. Avoidance of collision with a static obstacle on the lane.
5. Merging into a highway while there are vehicles on the right lane.
6. Approaching a vehicle to the ego vehicle from the side.

5.5.2. Simulation

In this section, the performance of the proposed MPC is evaluated on the autonomous vehicle according to road regulation, obstacle avoidance, and maneuverability. The MPC formulation is solved by the fmincon solver Sequential Quadratic Programming (SQP) via the YALMIP toolbox in MATLAB/Simulink. The parameters of the dry road controller are presented in Table 5-1. The speed of the vehicle is 80 km/h and the controller sampling period is 50ms.

First scenario is related to a path planned in a normal highway that is shown in Figure 5-4(a). The road geometry is approximated using a 4^{th} order polynomial based on offline lane marking and mapped waypoints. It is one-way road with two lanes. The ego vehicle is represented by a dashed rectangle and is moving on the lane 1, the surrounding vehicle is exhibited as empty rectangle and is moving on the other lane. Each of rectangles demonstrates the position of ego and surrounding vehicles.

The key objective of this scenario is to demonstrate the ability of the lane to stay on the straight road. The sequence of rectangles represents the future path of the ego and the surrounding vehicle. The desired speed of the ego vehicle is higher than that of the surrounding vehicle in the other lane. The longitudinal speed is shown in figure 5-4(b). The steering angle and is represented in figure 5-4(c).

Table 5-1: The controller parameters.

Parameter	Value	Unit	Parameter	Value	Unit
$\boldsymbol{m}$	1625	*kg*	$\boldsymbol{F_{y_r-max}}$	10600	*N*
$\boldsymbol{I_z}$	2865.6	$kg.m^2$	$\boldsymbol{N}$	20	-
$\boldsymbol{l_f}$	1.108	*m*	$\boldsymbol{N_c}$	5	-
$\boldsymbol{l_r}$	1.502	*m*	$\boldsymbol{N_{rc}}$	5	-
$\boldsymbol{C_{\alpha_f}}$	98389	*Nm/rad*	$\boldsymbol{N_{rs}}$	10	-
$\boldsymbol{C_{\alpha_r}}$	198142	*Nm/rad*	$\boldsymbol{u_{min}}$	-[24800 0.2]	-
$\boldsymbol{L_w}$	3.5	*m*	$\boldsymbol{u_{max}}$	[13000 0.2]	-
$\boldsymbol{\lambda_r}$	1	-	$\Delta\boldsymbol{u_{min}}$	-[1600 0.02]	-
$\boldsymbol{\lambda_0}$	25	-	$\Delta\boldsymbol{u_{max}}$	[1600 0.02]	-
$\boldsymbol{\lambda_C}$	1	-	$\boldsymbol{Q}$	[0.2 0.01]	-
$\boldsymbol{F_{x_T-max}}$	24800	*N*	$\boldsymbol{R}$	[5e-8 500]	-
$\boldsymbol{F_{y_f-max}}$	10400	*N*	$\boldsymbol{S}$	[2e-9 100]	-

When the road is also curved, which is demonstrated in Figure 5-5(a), the ego vehicle tries to stay in its lane. In these two scenarios, the surrounding vehicle is moving to the side of the ego vehicle. Due to the use of the potential field, the surrounding vehicle keeps the ego vehicle away from the other lane and the obstacle vehicle. The longitudinal force and vehicle speed for this scenario are illustrated in Figure 5-5(b). The steering angle and lateral acceleration are shown in Figure 5-5(c).

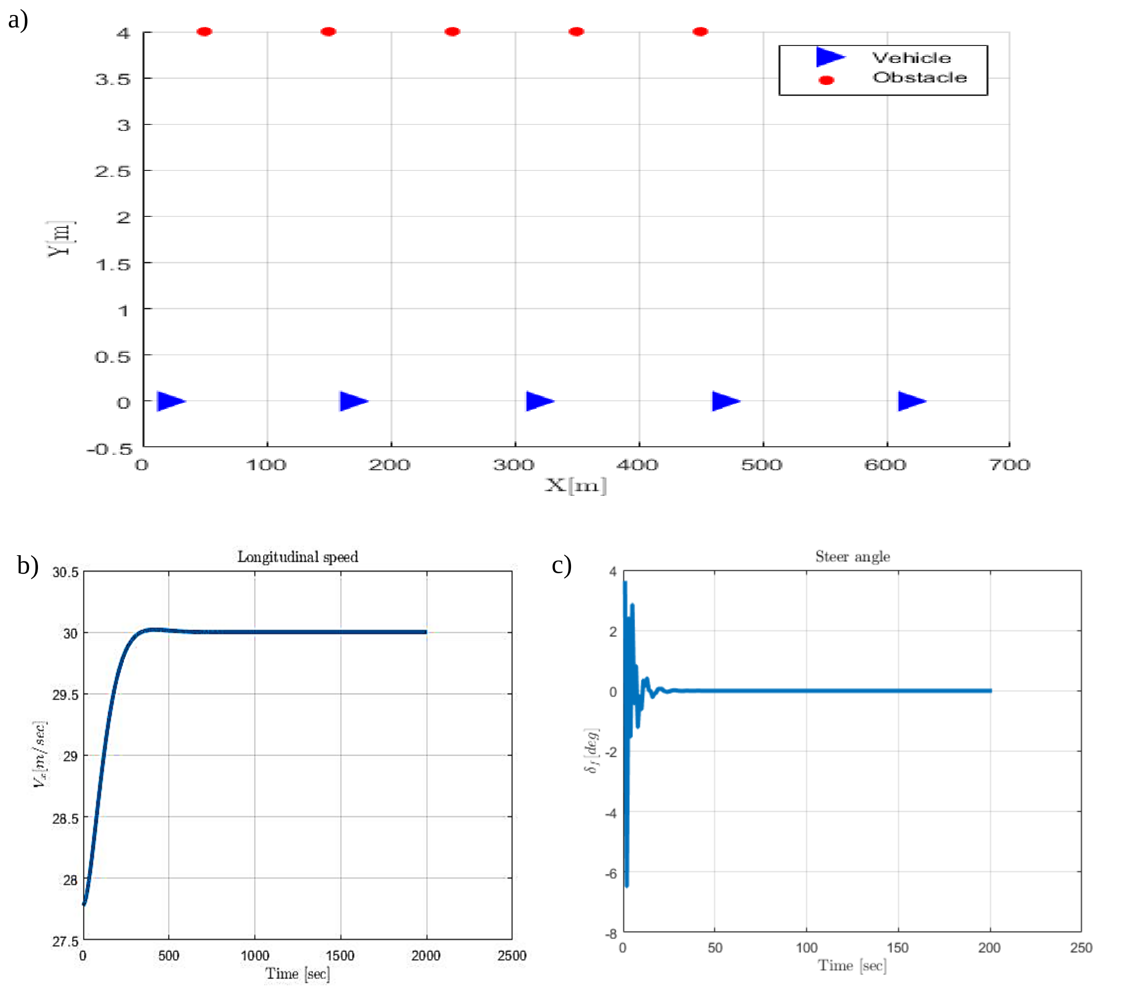


Fig. 5-4: (a) The path planning in straight road with keeping lane, (b) longitudinal speed, (c) steering angle for first scenario.

In the second scenario, the ego vehicle starts on lane 1 and it is commanded that should keep its lane. There is a moving obstacle vehicle in lane 2 with the same longitudinal position and speed as the ego vehicle. Carelessly, it changes its lane from the center of lane 2 to lane 1 at a constant lateral velocity within a specific time interval. The main objective of this scenario is to maintain sufficient distance from the vehicle in front of the ego vehicle to prevent any collision. The result of the simulation is shown in Figure 5-6(a).

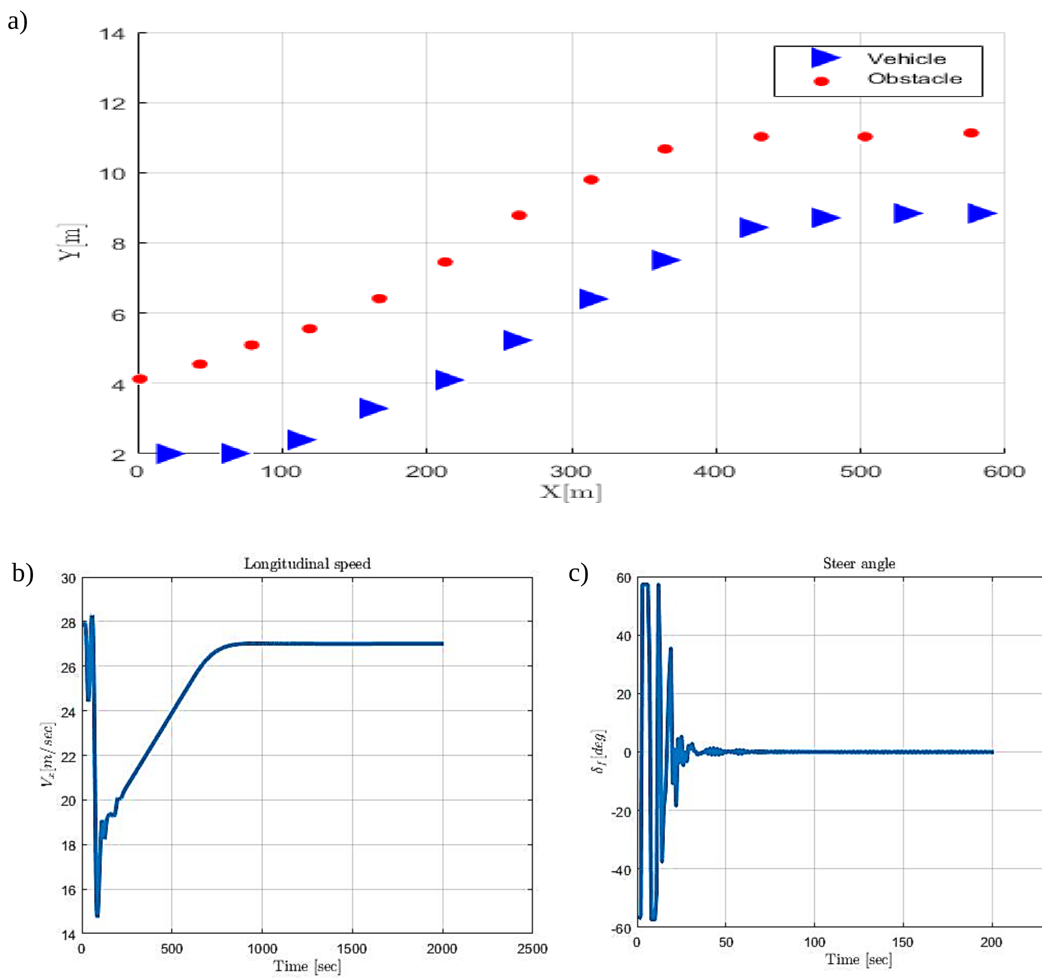


Fig. 5-5: (a) The path planning in curved road with keeping lane, (b) longitudinal speed, (c) steering angle for scenario with curved road.

Because of the potential obstacle field, the ego vehicle reduces its speed to provide sufficient space for the surrounding vehicle and then moves to the right to keep its lateral distance from the obstacle and prevents the collision. The road potential field guides the vehicle towards the middle of the lane and holds the vehicle in the expected lane by the potential field of lane centering. The moving obstacle is on the middle lane marker and the vehicle has made a longitudinal distance around 10 m to have a safe distance with the obstacle. The vehicle returns to the center of the lane due to the potential fields of the right road and center of the lane, and after making ample longitudinal distance from the obstacle.

a)

b)

c)

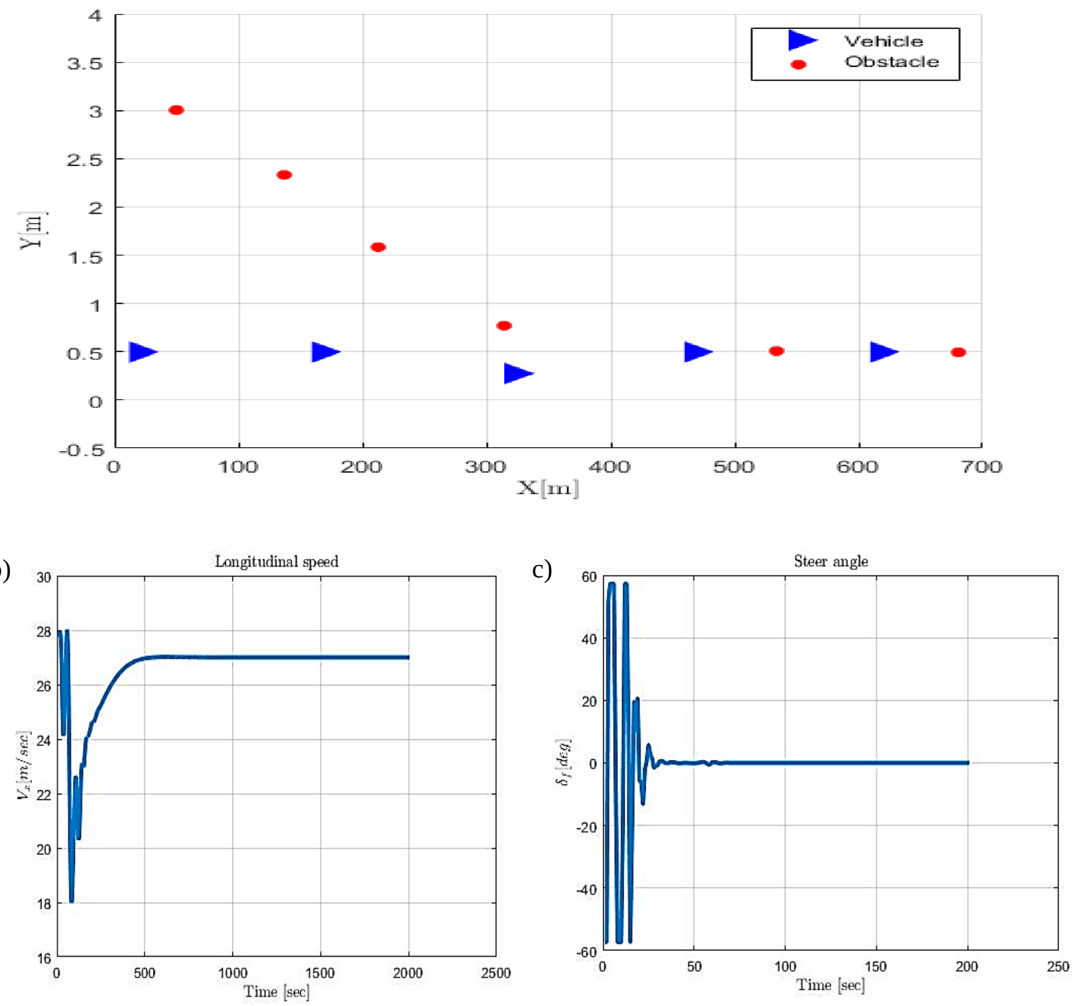


Fig. 5-6: (a) The path planning in straight road with changing abruptly lane of obstacle, (b) longitudinal speed, (c) steering angle for third scenario.

In the third scenario, while the obstacle on lane 2 is moving, the vehicle changes its lane. The result of the simulation is illustrated in Figure 5-7(a). Because there is a moving obstacle on the vehicle's side, the ego vehicle can not change its lane immediately; and the specified obstacle potential field holds the ego away from lane 2. The ego vehicle reduces its speed, then enters the middle lane and waits to pass the obstacle from its side. When there is enough distance between the obstacles in its front and behind, the vehicle moves to lane 2 by adjusting its speed while keeps its distance between obstacles. The changes in speed and lateral movements of the vehicle are based on the intended potential field functions which keep the vehicle away from the obstacles and road and also keep on the center of the lane.

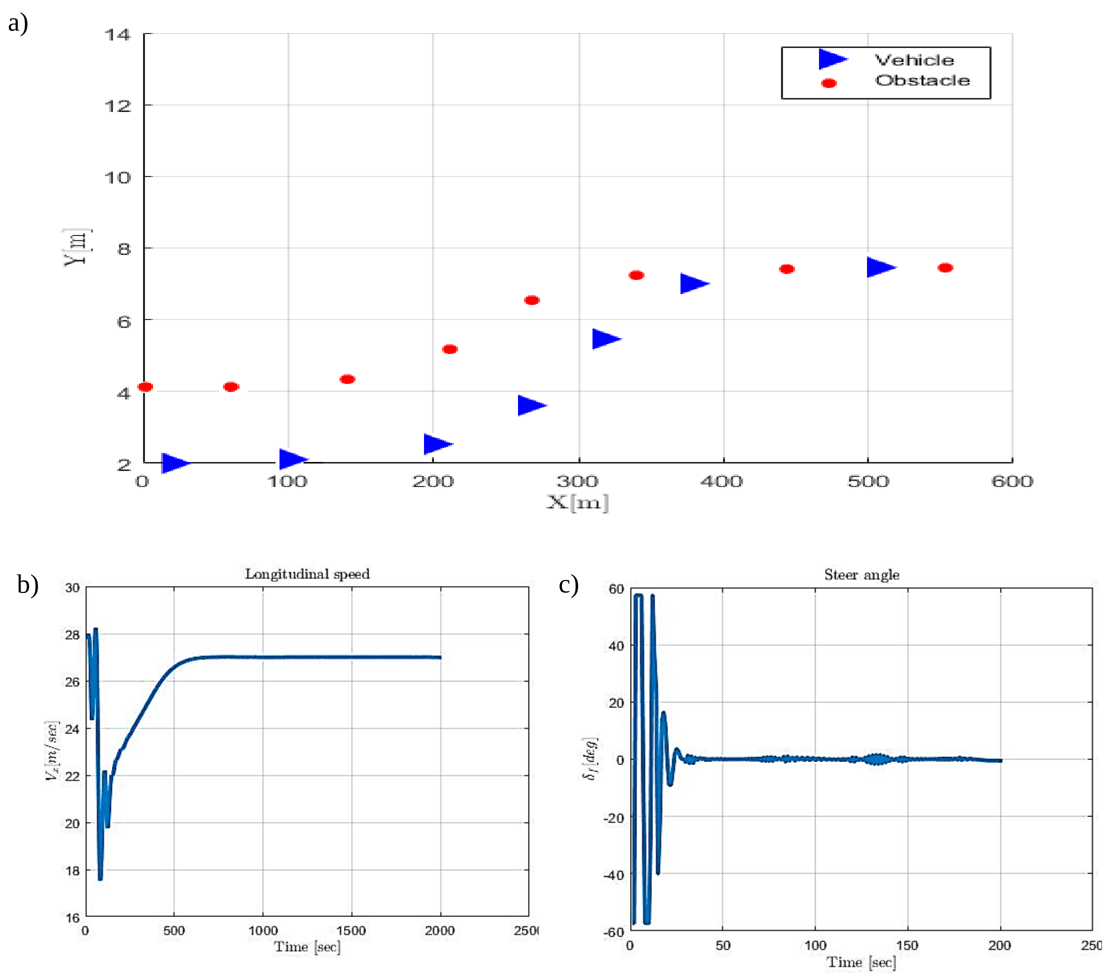


Fig. 5-7: (a) Changing lane the vehicle in curved road, (b) longitudinal speed, (c) steering angle and lateral acceleration for forth scenario.

For nonlinear and quadratic problems, this scenario is simulated. In this scenario, the vehicle is on lane 1 and should keep its lane using potential fields. On the lane, there is a static obstacle that is located in right boundary of road. It is a square obstacle with 0.5m length. Since there is sufficient lateral space between the vehicle and the obstacle, the vehicle can pass without collision from the side of the obstacle. The quadratic motion planning system performs similar to the behavior of the nonlinear motion planning system. The difference between the results of the simulation is more similar to the obstacle. at this location, the vehicle tries to keep away from the obstacle, to reduce its speed and to approach the lane centering while the potential field of lane centering leads the vehicle towards the middle of the lane and keeps the vehicle in the lane. The error of a quadratic convex function in approximating the PF near the obstacle becomes more apparent. On the other hand, the average calculation time of the nonlinear problem for a time step of this simulation is 15:80s while that of the quadratic problem is 0:008s. Since the step time is 0:05s, the quadratic problem can be solved in real-time. The other scenarios are simulated for the quadratic problem.

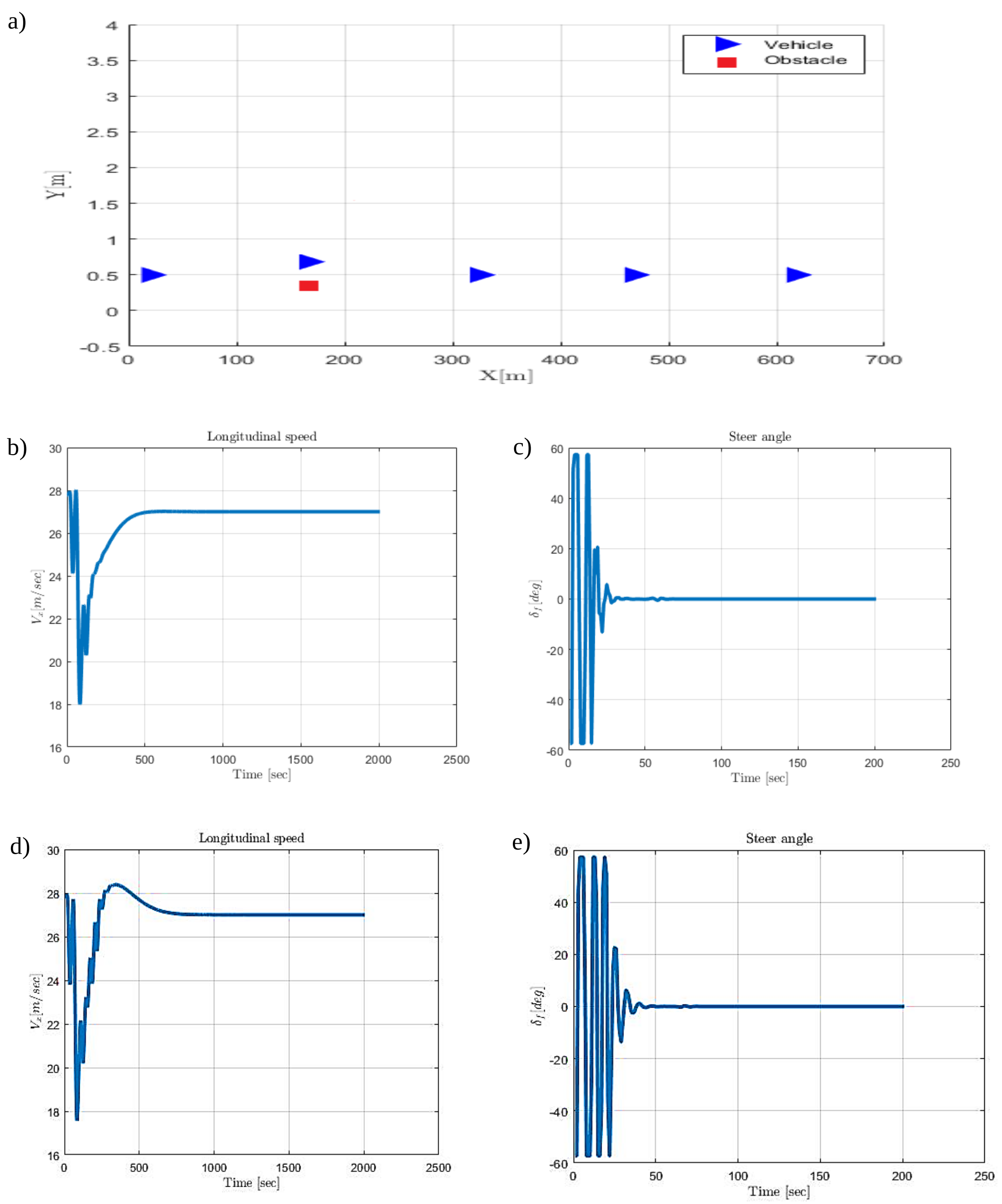
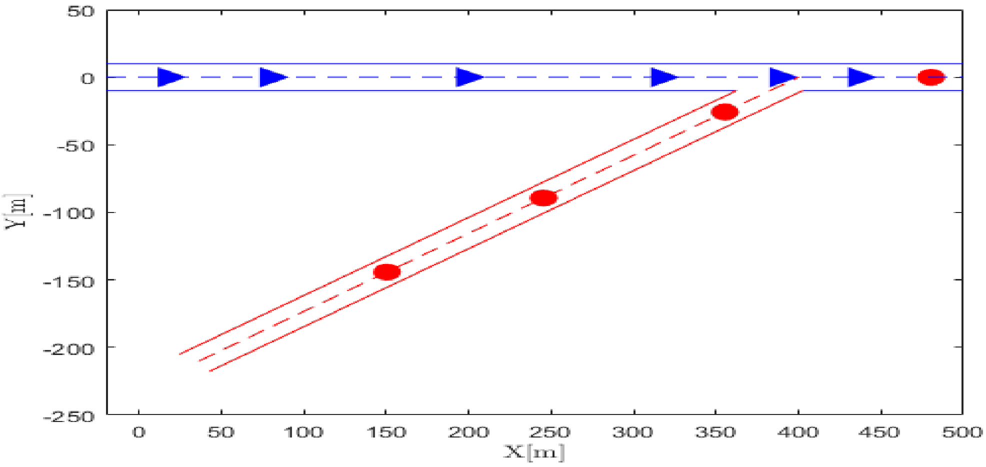


Fig. 5-8: (a) Avoidance of collision with a static obstacle by keeping lane, (b) longitudinal speed (quadratic problem), (c) steering angle (quadratic problem), (d) longitudinal speed (nonlinear problem), (e) steering angle (nonlinear problem) .

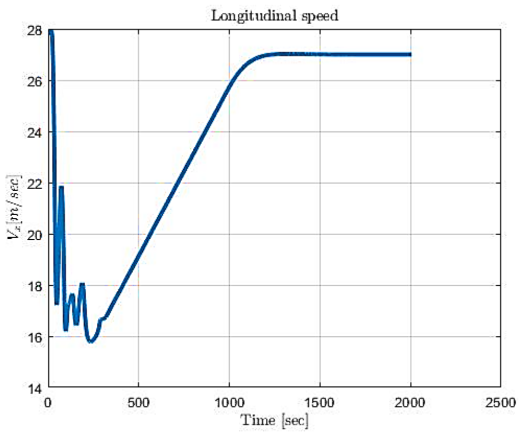
The simulation result for this scenario is shown in Figure 5-8(a). Also, the potential field functions are applied on the scenario. The road potential field leads the vehicle to move the right side, as well as the potential obstacle field, guides the vehicle to the left side of the lane. As mentioned, there is enough distance between the vehicle and static obstacle then the vehicle moves to the left of the lane for passing the obstacle while keeps its lane. After passing the obstacle, the vehicle returns to the intended lane and the changes of speed are not noticeably in this scenario.

This scenario is related to the merge highway that is shown in Figure 5-9(a) which illustrates the path planning in a ramp merging scenario and two obstacle vehicles in the merging lane. In order to generate a safe longitudinal path, the vehicle should fit itself between two adjacent obstacles or surrounding vehicles. The vehicle is on the straight lane in this scenario and the obstacle vehicles are on the other lanes and are going to merge into lane 1. First, the vehicle increases its speed to catch up with a gap, but by considering obstacles in its horizon it decelerates and keeps a safe distance from the obstacles in the front. Then the vehicle merges successfully into the lane.

a)



b)



c)

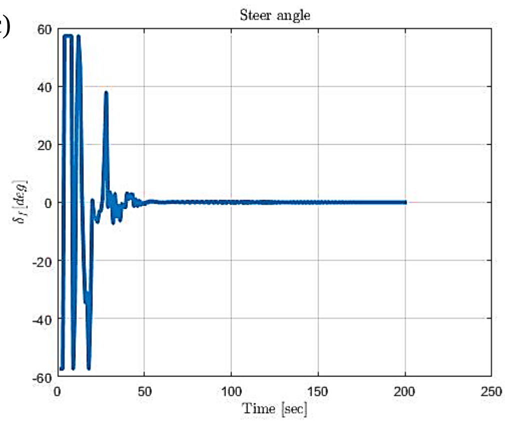


Fig. 5-9: (a) Merging into a highway, (b) longitudinal force and vehicle speed, (c) steering angle and lateral acceleration for scenario 6.

Figure 5-10 shows a merging maneuver along with an obstacle which is moving on lane 2 and the ego is moving on lane 1 that is ending at around 200 m. It should change its lane from lane 1 to the lane 2 while avoiding a possible collision to the obstacles. Due to the lack of lateral distance between the vehicle and obstacle, the vehicle cannot immediately and safely merge between them. In this scenario, the potential field used for obstacle keeps the vehicle out of lane 2 when the obstacle is passing from another lane. Additionally, the potential field used for a static obstacle adds to the end of the road to avoid its passage. Therefore, the vehicle reduces its speed and sometimes stops before reaching the ending of the road and after passing the obstacle, the vehicle changes safely its lane and then the road and lane centering potential fields keep the vehicle for going out from the road.

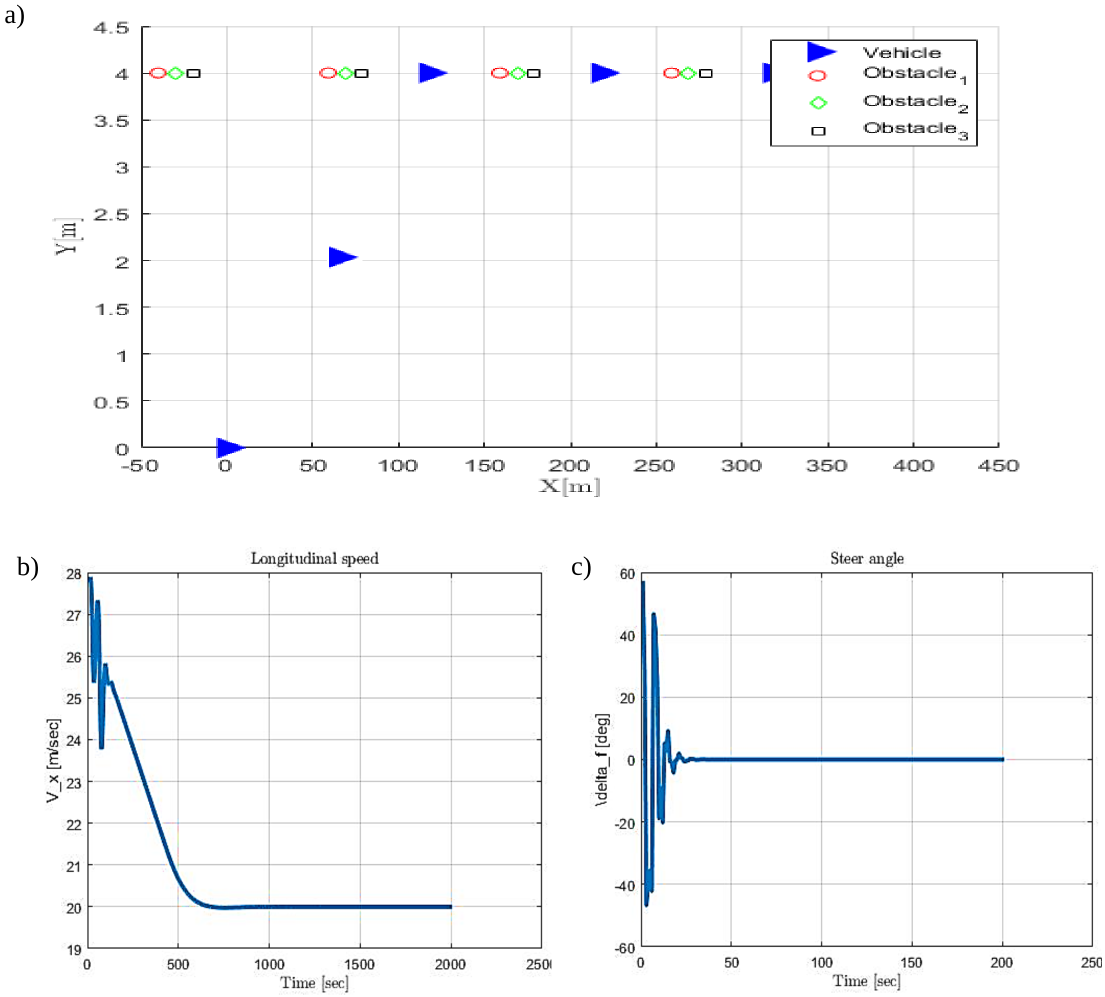


Fig. 5-10: Merging into a highway while the lane 1 is ending at around 200m. (b) longitudinal speed, (c) steering angle.

The results of the quadratic path planning approach are, as can be seen, very similar to those of the nonlinear path planning approach. The only difference is near the end of the lane. In this position, large deceleration makes an error in the position of predicted longitudinal vehicle. The performance of the quadratic problem is better than. The quadratic problem is feasible in real-time. The proposed path planning approach synthesizes different behaviors for different obstacles and this is a major contribution of the current work.

5.6. Summary

This chapter presents a motion planning based on MPC which utilizes different potential field functions to obstacle avoidance. The appropriate potential fields have been presented for different obstacles and road structures based on their characteristics. These potential fields have been used for observing road regulations and avoiding obstacle collision. For motion planning based on MPC, a dynamics model as the vehicle model along with actuator and tire constraints has been introduced. The proposed model predicts the behavior of the vehicle appropriately and generates feasible maneuvers. The summary of this chapter is as follows:

- MPC is a nonlinear problem. The used vehicle model for MPC is a linear bicycle model.
- To reduce the computational time, the MPC problem is approximated by a quadratic convex problem. Therefore, the quadratic MPC problem's performance and computational time are appropriate.
- To evaluate the performance of motion planning based on MPC, the complicated scenarios have been proposed. Performing the appropriate maneuvers in the complicated scenarios show the capability of the proposed motion planning based on MPC. When the vehicle receives the lane change command from the behavior planning module, if it is safe to do, the vehicle changes its lane. when there is enough space between two vehicles, merging the vehicle in between them is safe. If the current lane ends and the lane changes are not safe, the vehicle will reduce its speed or even stop before the lane ends and if it is safe to do, change its lane. Furthermore, if another vehicle approaches the vehicle carelessly from the side while remaining on the lane, the vehicle makes space for it as much as possible.

6. CONCLUSION AND FUTURE WORK

The research presented in this dissertation is designing, planning and testing the proposed approaches to path planning in different static and moving environments. This chapter presents a summary of the key contributions of the proposed approaches and further investigations.

6.1. Conclusion

The main objective of this thesis is to create and prototype two controllers for the mobile robot and autonomous vehicle navigation problem. The controls have been designed to deal with the uncertainty in the environment. The main contributions of this thesis in the navigation field of the mobile robot are presented below:

A novel path planning has been presented using grasshopper algorithm in unknown and dynamic environments. For this purpose, a sensor-based online technique was applied to obtain a non-collision path. The main idea for predicting obstacles is to use a range sensor with four different radii circles. These circles are classified into four groups that estimate the complex obstacle direction. The robot saves the points of intersection between the locations of the obstacle and these circles and decides which of these areas have these points of intersection. For two sampling intervals, the intersection points do not change, then the obstacle is static; otherwise, it will be dynamic. Also, by measuring the number of these intersection points, the direction of the dynamic obstacle is determined. Then, using the grasshopper algorithm, the robot tries to avoid obstacles and find the optimal path. The location, shape, and velocity of obstacles are not available to the robot. Many scenarios are used for evaluating our methodology suggested. The results of the simulation have shown which the suggested controller successfully guides the robot towards the target, effectively avoids collision and finds the shortest and optimal path in minimum time. Besides, the proposed controller is compared with PSO, GA, D*, Neuro-Fuzzy, and RRT*. The comparison results indicate the prominent features of the proposed approach. In addition, the proposed approach considers time, unexpected obstacles and velocity vector of the obstacles making it a good candidate for real-time applications. Moreover, a new hybrid controller based on grasshopper algorithm optimization and the genetic algorithm (GOA-GA) has been presented for the path planning of the mobile robot. To evaluate and analysis, the proposed controller has simulated on MATLAB software.

A path planning based on the model predictive controller for autonomous vehicles in the dynamic environments has been presented. Three different potential field mechanisms have been implemented for the road center of the lane and objects to avoid collision and ensure vehicle safety. This dissertation utilizes a vehicle model using a bicycle model and tire constraints that remain valid at tire force limits. The problem is a nonlinear optimal control problem by its nature. The MPC model was chosen to use a quadratic objective function to formulate the problem of path planning. Approximating the problem to a convex quadratic problem reduces the computational burden significantly. This function makes it easier to use the system's vehicle dynamics and constraints within the MPC structure that has been used to determine maneuvers for lane-keeping and lane-changing. The computational time and performance of the quadratic problem have been evaluated by simulations. The obtained result indicates that the performance of the quadratic formulation is better nonlinear. Several simulations are carried out to evaluate different scenarios. The performance of the motion planning based on the MPC controller is simulated in the MATLAB/Simulink and CarSim. The results show that the suggested path planning method is efficient in generating safe and comfortable pathways for autonomous vehicles. Since the dynamics of the vehicle is used as the predicated model, the planned path is an optimal path based on the dynamics of the vehicle.

6.2. Future Work

In this section, to improve accuracy and efficiency of the suggested path planning of the mobile robot and autonomous vehicle, several suggestions for future work have proposed.

- To improve the efficiency of the proposed controller with GOA for the mobile robot, it can be hybridized with another algorithm. Also, it can be tested in different unknown dynamic environments.
- The proposed controller and its hybridization can be utilized for navigation of multiple mobile robots on the unknown environment.
- To improve performance motion planning based on MPC, comforting passenger and selecting lane can also be considered in the objective function of MPC so that these rules can optimize the trajectories of the vehicle.
- To increase the accuracy of motion planning based on MPC, controlling tire slips can also be included in the objective function. To control tire slips, the wheel

dynamics in the vehicle model of MPC are considered. The obtained trajectory plans based on a more accurate vehicle model.

- The two above mentioned can also be tested for the rainy road.

REFERENCE

[1] H. Choset and J. Burdick, Sensor-based exploration: The hierarchical generalized Voronoi graph. The International Journal of Robotics Research, 19 (**2000**) 96.

[2] P. Hart, N. Nilsson, and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics, 4 (**1968**) 100.

[3] T. L. P´erez and M. A. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, Communications of the ACM, 22 (**1979**) 560.

[4] M. D. Berg, M. V. Kreveld, M. M. Overmars and O. C. Schwarzkopf, Computational geometry, Springer Berlin Heidelberg,1 (**2000**) 17.

[5] M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, SIAM Journal on Computing, 15 (**1986**) 193.

[6] Overmars, M.H. and Welzl, E., New methods for constructing visibility graphs, In Proceedings of the fourth annual symposium on Computational geometry, 6 Jan, Urbana-Champaign, United States, 164-171, **1988**.

[7] S. K. Debnath, R Omar, and N. B. A. Latip, Comparison of Different Configuration Space Representations for Path Planning Under Combinatorial Method. Indonesian Journal of Electrical Engineering and Computer Science, 1 (**2019**) 401.

[8] JC. Latombe, Motion planning: A journey of robots, molecules, digital actors, and other artifacts. The International Journal of Robotics Research, 11(**1999**)1119.

[9] Ma, Y., Zheng, G. and Perruquetti, W., Cooperative path planning for mobile robots based on visibility graph. In Proceedings of the 32nd Chinese Control Conference Jul 26, China, 4915–4920, **2013**.

[10] Canny, J. and Reif, J. New lower bound techniques for robot motion planning problems, Proceedings of the 28th Annual Symposium on Foundations of Computer Science, 12-14 October LOS ANGELES, CA, USA, 49–60, **1987**.

[11] S. Garrido, L. Moreno, D. Blanco, and P. Jurewicz, Path planning for mobile robot navigation using Voronoi diagram and fast marching. International Journal of Robotics and Automation, 2 (**2011**) 42.

[12] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavriki and S. Thrun, Principles of Robot Motion, MIT Press, Cambridge, MA, **2005**.

[13] S. Garrido, L. Moreno, D. Blanco and P. Jurewicz, Path planning for mobile robot navigation using Voronoi diagram and fast, marching, International Journal of Robotics and Automation, 2 (**2011**) 42.

[14] D. H. Yang and S. K. Hong, A roadmap construction algorithm for mobile robot path planning using skeleton maps, Advanced Robotics, 21(**2007**) 51.

[15] R. Wein, J. P. Van Den Berg and D. Halperin, The visibility-Voronoi complex and its application, Computational Geometry, 36 (**2007**) 66.

[16] L. E. Kavraki, P. Svestka, J. C. Latombe and M. H. Overmars, Probabilistic roadmaps for path planning in high dimensional configuration spaces, IEEE Transaction on Robotics and Automation, 12 (**1996**) 566.

[17] G. Sanchez and J. Latombe, A Single-query Bidirectional Probabilistic Roadmap Planner with Lazy Collision Checking, Robotics research. Springer, 6 (**2003**) 403.

[18] Keerthi, S.S., Ong, C.J., Huang, E. and Gilbert, E.G., Equidistance diagram-a new roadmap method for path planning, IEEE International Conference on Robotics and Automation, 10 May, USA, **1999,** p 682.

[19] Chen, P., Xiaoqing, L., Jiyang, D. and Linfei, Y., Research of path planning method based on the improved Voronoi diagram, In IEEE 25th Chinese Control and Decision Conference, 25-27 May, China, **2013**, p 2940.

[20] Gómez, E.J., Santa, F.M. and Sarmiento, F.H.M., A comparative study of geometric path planning methods for a mobile robot: Potential field and Voronoi diagrams. In IEEE II international congress of engineering mechatronics and automation, 23-25 October, Colombia, **2013**, p 1.

[21] Gonzalez, R., Mahulea, C. and Kloetzer, M., A matlab-based interactive simulator for mobile robotics. In IEEE International Conference on Automation Science and Engineering, 24-28 August, Sweden, **2015**, p 310.

[22] A. Abbadi and R. Matousek. Path Planning Implementation Using MATLAB, in Technical Computing Bratislava, **2014**, p 1.

[23] S. M. Lavalle, Planning algorithms, Cambridge University Press, New York, **2006**.

[24] W. Regli, Robot Lab: Robot path planning, Lecture notes of department of computer science, Drexel University, **2007**.

[25] T. J. Schwartz and M. Sharir, On the “Piano Movers” Problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers, Communications on Pure and Applied Mathematics, 36 (**1983**) 345.
[26] D. Zhu and J. C. Latombe, “New heuristic algorithms for efficient hierarchical path planning,” IEEE Transaction on Robotics and Automation, 7 (**1991**) 9.
[27] H. Samet, An overview of quadtree octrees and related hierarchical data structures. In Theoretical Foundations of Computer Graphics and CAD, (**1988**) 51.
[28] H. Noborio, T. Naniwa, and S. Arimoto, A quadtree-based path-planning algorithm for a mobile robot, Journal of Robotic Systems, 7 (**1990**) 555.
[29] Kim, D.H., Hoang, G., Bae, M.J., Kim, J.W., Yoon, S.M., Yeo, T.K., Sup, H. and Kim, S.B., Path tracking control coverage of a mining robot based on exhaustive path planning with exact cell decomposition, In 14th International Conference on Control, Automation and Systems, 22-25 October, South Korea, **2014**, p 730.
[30] Kloetzer, M., Mahulea, C. and Gonzalez, R., Optimizing cell decomposition path planning for mobile robots using different metrics, In19th International Conference on System Theory, Control and Computing, 14-16 October, Romania, **2015**, p 565.
[31] Scheurer, C. and Zimmermann, U.E., Path planning method for palletizing tasks using workspace cell decomposition, In IEEE International Conference on Robotics and Automation, 9-13 May, China, **2011**, p 1.
[32] O. Khatib, Real time Obstacle Avoidance for manipulators and Mobile Robots, Autonomous Robot Vehicles, (**1986**) 396.
[33] Garibotto, G. and Masciangelo, S., Path planning using the potential field approach for navigation, Fifth International Conference on Advanced Robotics, 19-22 June, Pisa, Italy, 1991, p 1679.
[34] J. O. Kim and P. K. Khosla, Real-time obstacle avoidance using harmonic potential functions, IEEE Transactions on Robotics and Automation, 8 (**1992**) 338.
[35] J. Borenstein and Y. Koren, Real-time obstacle avoidance for fast mobile robots, IEEE Transactions on Systems, Man and Cybernetics, 19 (**1989**) 1179.

[36] S. S. Ge and Y. J. Cui, Dynamic motion planning for mobile robots using potential field method, Autonomous Robots, 13 (**2002**) 207.

[37] K. P. Valavanis, T. Hebert, R. Kolluru and N. Tsourveloudis, Mobile robot navigation in 2D dynamic environments using an electrostatic potential field, IEEE Transactions on Systems Man and Cybernetics, Part A: Systems and Humans, 30 (**2000**) 187.

[38] L. Huang, Velocity planning for a mobile robot to track a moving target-apotential field approach, Robotics and Autonomous Systems, 57 (**2009**) 55.

[39] Shi, P. and Zhao, Y., An efficient path planning algorithm for mobile robot using improved potential field, IEEE International Conference on Robotics and Biomimetics, 19-23 December, *Guilin, China*, **2009**, p 1704.

[40] Sfeir, J., Saad, M. and Saliah-Hassane, H., An improved artificial potential field approaches to real-time mobile robot path planning in an unknown environment, IEEE International Symposium on Robotic and Sensors Environments, 17-18 September, Montreal, Canada, **2011**, p 208.

[41] Biswas, K. and Kar, I., On reduction of oscillations in target tracking by artificial potential field method, 9th IEEE International Conference on Industrial and Information Systems, 15-17 December, Gwalior, India, **2014**, p 1.

[42] Hui, N. B., Coordinated motion planning of multiple mobile robots using potential field method, IEEE International Conference on Industrial Electronics, Control and Robotics, 27-29 December, India, **2010**, p 6.

[43] L. C. Pimenta, A. R. Fonseca, G. A. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas and M. F. Campos, Robot navigation based on electrostatic field computation, IEEE Transactions on Magnetics, 42 (**2006**) 1459.

[44] S. K. Pradhan, D. R. Parhi, A. K. Panda and R. K. Behera, Potential field method to navigate several mobile robots, Applied Intelligence, 25 (**2006**) 321.

[45] Y. B. Chen, G. C. Luo, Y. S. Mei, J. Q. Yu, X. L. Su. UAV path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science. 47 (**2016**) 1407.

[46] R Raja, A Dutta, K. S. Venkatesh, New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover, Robotics and Autonomous Systems, 72 (**2015**) 295.

[47] D. R. Parhi and A. K. Jha, Review and analysis of different methodologies used in mobile robot navigation, IJAAIES, 4 (**2012**) 1.

[48] H. J. Bremermann, The evolution of intelligence. The Nervous system as a model of its environment, Technical Report no.1, University of Washington, Department of Mathematics; **1958**.

[49] J. H. Holland, Adaptation in natural and artificial systems, 18 (**1976**) 529.

[50] Shibata, T. and Fukuda, T., Robot motion planning by genetic algorithm with fuzzy critic, Proceedings of 8th IEEE International Symposium on Intelligent Control, 25-27 August, Chicago, USA, **1993**, p 565.

[51] M. T. Shing and G. B. Parkar, Genetic algorithm for the development of real-time multi-heuristic search strategies," In ICGA, (**1993**) 565.

[52] J. Xia, Z. Michalewicz, L. Zhang and K. Trojanowski, Adaptive evolutionary planner/navigator for mobile robot, IEEE Transaction on Evolutionary Computation, 1 (**1997**) 18.

[53] X. Kang, Y. Yue, D. Li and C. Maple, Genetic algorithm-based solution to dead problems in robot navigation, International Journal of Computer application in technology, 41 (**2011**) 177.

[54] Shi, P. and Cui, Y., Dynamic path planning for mobile robot based on genetic algorithm in unknown environment, In 2010 Chinese control and decision conference, 26-28 May, Xuzhou, China, **2010**, p 4325.

[55] D. K. Pratihar, K. Deb and A. Ghosh, Fuzzy-Genetic algorithm and time optimal obstacle free path generation for mobile robots, Engineering Optimization, 32(**1999**)117.

[56] N. B. Hui and D. K. Pratihar, A comparative study on some navigation schemes of a real robot tackling moving obstacles, Robotics and Computer-Integrated Manufacturing, 25 (**2009**) 810.

[57] X. Wang, Y. Shi, D. Ding and X. Gu, Double global optimum genetic algorithm-particle swarm optimization-based welding robot path planning, Engineering Optimization, 48 (**2016**) 299.

[58] R. Kala, Coordination in navigation of multiple mobile robots, Cybernetics and systems, 45 (**2014**) 1.

[59] F. Liu, S. Liang and X. Xian, Optimal robot path planning for multiple goals visiting based on Tailored genetic algorithm, International Journal of Computational Intelligence Systems, 7 (**2014**) 1109.

[60] T. Kuyucu, I. Tanev and K. Shumohara, Incremental evolution of fast moving and simulated snake-like robot with multi-objective GP and strongly-typed crossover, Memetic Computing, 4 (**2012**) 183.

[61] Yang, S.X., Hu, Y. and Meng, M.Q.H., A knowledge-based GA path planning of multiple mobile robots in dynamic environments, in 2006 IEEE Conference on Robotics, Automation and Mechatronics, 1-3 June, Bangkok, Thailand, **2006**, p 1.

[62] Q. U. Hong, K. E. Xing and T. Alexander, An improved genetic algorithm with coevolutionary strategy for global path planning of multiple mobile robots, Neurocomputing, 120 (**2013**) 509.

[63] G. Carlos, S. Antonia and X. H. Vazquez, Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction, An application to fish aggregating devices, Computers and Operations Research, 56 (**2015**) 22.

[64] A. H. Karami and M. Hasanzadeh, An adaptive genetic algorithm for robot motion planning in 2D complex environment, Computers and electrical engineering, 43 (**2015**) 317.

[65] L. A. Zadeh, Fuzzy sets, Information and control, 8 (**1965**) 338.

[66] D. Herrero-perez, H. Martinez-Barbara, K. LeBlanc and A. Saffiotti, Fuzzy uncertainty modeling for grid-based localization of mobile robots, International Journal of Approximate Reasoning, 51(**2010**) 912-932.

[67] F. Abdessemed, K. Benmahammed and E. Monacelli, A Fuzzy-based reactive controller for a nonholonomic mobile robot, Robotics and Autonomous Systems, 47, (**2004**) 31.

[68] P. G. Zavlangas and S. G. Tzafestas, Motion control for mobile robot obstacle avoidance and navigation: a fuzzy logic-based approach, Systems Analysis Modeling Simulation, 43(**2003**) 1625.

[69] G. Castellano, G. Attolico and A. Distante, Automatic generation of fuzzy rules for reactive robot controllers, Robotics and Autonomous Systems, 22 (**1997**) 133.

[70] A. Homaifar and E. McCormick, Simultaneous design of membership function and rule sets for fuzzy controllers using genetic algorithm, IEEE Transaction on Fuzzy systems, 3(**1995**) 129.

[71] H. R. Berenji and P. Khedkar, Learning and tuning fuzzy logic controllers through reinforcement, IEEE transaction on neural network, 3 (**1992**) 724.

[72] R. Abiyev, D. Ibrahim and B. Erin, Navigation of mobile robot in presence of obstacles, Advances in Engineering Software, 41(**2010**) 1179.

[73] S. S. Ge, X. Lai and A. Mamun, Sensor-based path planning for nonholonomic mobile robots subject to Dynamic environment, Robotics and Autonomous System, 55 (**2007**) 513.

[74] O. Motlagh, T. S. Hong and N. Ismail, Development of new minimum avoidance systems for a behavior-based mobile robot, Fuzzy Sets and Systems, 160 (**2009**) 1929.

[75] K. Park and N. Zhang, Dual fuzzy logic controllers for autonomous robot navigation. International Journal of Systems, Control and Communication, 3(**2011**) 322.

[76] R. Huq, G. Mann and R. Gosine, Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation, Applied soft computing, 8(**2008**) 422.

[77] G. P. Moustris and S. G. Tzafestas, Switching fuzzy tracking control for mobile robots under curvature constraints, Control Engineering Practice, 19 (**2011**) 45.

[78] J. Gasos and A. Rosetti, Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments, Fuzzy sets and systems, 107(**1999**) 1.

[79] P. Parasuraman, Sensor fusion for mobile robot navigation: Fuzzy associative memory, Procedia Engineering, 41(**2012**) 251.

[80] C. Flanagan, D. Toal, and M. Leyden, Subsumption and fuzzy logic, experiments in behavior-based control of mobile robot, International Journal of smart engineering system design, 5(**2003**) 161.

[81] M. Khatib and J. Saade, An efficient data-driven fuzzy approach to the motion planning problem of a mobile robot, Fuzzy sets and systems, 134 (**2003**) 65.

[82] H. Lee, J. Jung, K. Choi, P. Jiyoung and H. Myung, Fuzzy-logic-assisted interacting multiple model (FLAIMM) for mobile robot localization, Robotics and Autonomous Systems, 60 (**2012**) 1592.

[83] M. Hoy, A. S. Matveev and A. V. Savkin, Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments, Robotics and Autonomous Systems, 60 (**2012**) 1253.

[84] T. K. Kang, H. Zhang, G. T. Park and D. W. Kim, Ego-motion compensated object recognition using type-2 fuzzy set for a moving robot, Neurocomputing, 120 (**2013**) 130.

[85] K. Al-Mutib, E. Mattar and M. Alsulaiman, Implementation of fuzzy decision based mobile robot navigation using sereo vision, Procedia Computer Sciences, 62 (**2015**) 143.

[86] D. N. M. Abadi and M. H. Khooban, Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots, Journal of King Saud–Engineering Sciences, 27 (**2015**) 92.

[87] O. Castillo, H. Neyoy, J. Soria and F. Valdez, A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot, Applied soft computing, 28 (**2015**) 150.

[88] S. F. Toloue, M. R. Akbarzadeh, A. Akbarzadeh and M. Jalaeian-F, Position tracking of a 3-PSP parallel robot using dynamic growing interval type 2-fuzzy neural control, Applied Soft Computing, 37 (**2015**) 1.

[89] R. Al-Jarrah, A. Shahzad and H. Roth, Path planning and motion coordination for multi-robots' systems using probabilistic neuro fuzzy, IFAC-papers on line, 48 (**2015**) 46.

[90] F. Yili, L. Han, Z. Jiang and S. Wang, Double layers fuzzy logic based mobile robot path planning in unknown environment, Intelligence Automation and Soft Computing, 15(**2009**) 275.

[91] W.S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5 (**1943**) 115.

[92] D. Janglova, Neural networks in mobile robot motion, International Journal of Advanced Robotic Systems, 1(**2004**) 15.

[93] B. Siemiatkowska and A. Dubrawski, Cellular neural networks for navigation of a mobile robot, Rough Sets and Current Trends in Computing, 1424 (**1999**) 147.

[94] X. Wang, Z. G. Hou, A. Zou, M. Tan and C. Long, A behavior controller based on spiking neural networks for mobile robots, Neurocomputing, 71(**2008**) 655.

[95] J. Qiao, R. Fan, H. Han and X. Ruan, Q-learning based on dynamical structures neural network for robot navigation in unknown environment, Advances in neural network, 5553(**2009**) 188.

[96] Q. L. Li, Y. Song and Z. G. Hou, Neural network based Fast SLAM for automobile robots in unknown environments, Neurocomputing, 165 (**2015**) 99.

[97] Y. K. Na and S. Y. Oh, Hybrid control for autonomous mobile robot navigation using neural network-based behavior modules and environment classification, Autonomous robots, 15 (**2003**) 193.

[98] J. K. Pothal and D. R. Parhi, Navigation of multiple robots in a highly clutter terrains using adaptive neuro-fuzzy inference system, Robotics and Automation, 72 (**2015**) 48.

[99] A. Abu Baker, A novel mobile robot navigation system using neuro-fuzzy rule-based optimization technique, Research Journal of Applied Sciences, Engineering and Technology, 4 (**2012**) 2577.

[100] P. K. Pal and A. Kar, Sonar-based mobile robot navigation through supervised learning on a neural net, Autonomous Robots, 3 (**1996**) 355.

[101] A. Medina-Santiago, J. L. Campus-Anzueto, J. A. Vazquez-Feijoo, H. R. Hernandez-de-Leon and R. Mota-Grajales, Neural control systems in obstacle avoidance in mobile robots using untrasonic sensors, Journal of Applied Research and Technology, 2 (**2014**) 104.

[102] G. Capi, S. Kaneko and B. Hua, Neural network-based guide robot navigation: An evolutionary approach, Procedia Computer Science, 76 (**2015**) 74.

[103] U. A. Syed, F. Kunwar and M. Iqbal, Guided autowave pulse coupled neural network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots, Robotics and Autonomous Systems, 62 (**2014**) 474.

[104] Z. Hendzel, Neural network reactive navigation and control of wheeled mobile robot, Neural Networks and Soft Computing, 19 (**2003**) 686.

[105] B. Markoski, S. Vukosavavijev, D. Kukolj and S. Pletl, Mobile robot control using self-learning neural network, Intelligent systems and informatics (**2009**) 45.

[106] Y. Quinonez, M. Ramirez, C. Lizarraga, I. Tostado and J. Bekios, Autonomous robot navigation based on pattern recognition techniques and artificial neural networks, Advanced Intelligent Computing, 6838 (**2015**) 210.

[107] Eberhart, R. and Kennedy, J., A new optimizer using particle swarm theory, Proceedings of the sixth international symposium on micro machine and human science, 4-6 October, Nagoya, Japan, **1995**, p 39.

[108] X. Tang, L. Li and B. Jiang, Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter, The Journal of China Universities of Posts and Telecommunications, 21(**2014**)78.

[109] V. H. Xuan, H. Cheolkeun and L. Jewon, Novel hybrid optimization algorithm using PSO and MADS for the trajectory estimation of a four-track wheel skid-steered mobile robot, Advanced Robotics, 27 (**2013**) 1421.

[110] A. Atyabi, S. Phon-Amnuaisuk and C. K. Ho, Applying area extension PSO in robotic swarm, Journal of Intelligent Robot System, vol. 58 (**2010**) 253.

[111] Q. Tang and P. Eberhard, Cooperative motion of swarm mobile robots based on particle swarm optimization and multibody system dynamics, Mechanics based Design of Structures and Machines, 39 (**2011**) 179.

[112] M. S. Couceiro, R. P. Rocha and F. Nuno, A PSO multi-robot exploration approach over unreliable MANETs, Advanced Robotics, 27 (**2013**)1221.

[113] Y. Chen, J. Cheng, C. Lin, X. Wu, Y. Ou and Y. Xu, Classification based learning by particle swarm optimization for wall-following robot navigation, Neurocomputing, 113 (**2013**) 27.

[114] P. K. Das, H. S. Behera and B. K. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, Swarm and Evolutionary Computation, 28 (**2016**) 14.

[115] He Bo, L. Ying, S. Zhang, X. Feng, T. Yan, R. Nian and Y. Shen, Autonomous navigation based on Unscented-Fast SLAM using particle swarm optimization for autonomous underwater vehicles, Measurement, 71 (**2015**) 89.

[116] A. Khan, I. Noreen, and Z. Habib, On Complete Coverage Path Planning Algorithms for Non-Holonomic Mobile Robots: Survey and Challenges. Journal of Information Science and Engineering, 33 (**2017**)101.

[117] T. K. Lee, S. H. Baek, Y. H. Choi and S. Y. Oh, Smooth coverage path planning and control of mobile robots based on high resolution grid map representation, Robotics and Autonomous Systems, 59 (**2011**) 801.

[118] G. Pessin, D. O. Sales, M. A. Dias, R. I. Klaser, D. F. Wolf, J. Ueyama, F. S. Osoria and P. A. Vargas, Swarm intelligence and the quest to solve a garbage and recycling collection problem, Soft Computing, 17 (**2013**) 2311.

[119] Akhand, M.A.H., Akter, S., Rahman, S.S. and Rahman, M.H., Particle Swarm Optimization with Partial Search to Solve Traveling Salesman Problem, in International Conference on Computer and Communication Engineering, 3-5 July, Kuala Lumpur, Malaysia, **2012**, p 118.

[120] Rokbani, N., Abraham, A. and Alimi, A.M., Fuzzy Ant Supervised by PSO and Simplified Ant Supervised PSO Applied to TSP, In International Conference on Hybrid Intelligent Systems, 4-6 December, Gammarth, Tunisia, **2013**, p 251.

[121] D. Macro, Ant colony system: A Cooperative learning approach to the travelling salesman problem," IEEE transaction on evolutionary computation, 1 (**1997**) 53.

[122] T. Guan-Zheng, H. E. Huan and S. Aaron, Ant colony system algorithm for real time globally optimal path planning of mobile robots, Acta Automatica Sinica, 33 (**2007**) 279.

[123] Liu, S., Mao, L. and Yu, J., Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems, Proceedings of the 2006 IEEE International Conference of Mechatronics and Automation, 25-28 June, Luoyang, China, **2006**, p 1733.

[124] O. Castillo, H. Neyoy, J. Soria, P. Melin and F. Valdez, A New approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot, Applied soft computing, 28 (**2015**) 150.

[125] Purian, F.K. and Sadeghian, E., Mobile robots path planning using ant colony optimization and fuzzy logic algorithm in unknown dynamic environment, International Conference on Control, Automation, Robotics and Embedded systems, 16-18 December, Jabalpur, India, **2013**, p 1.

[126] Englot, B. and Hover, F., Multi-goal feasible path planning using ant colony optimization, IEEE International Conference on Robotics and Automation, 9-13 May, Shanghai, China, **2011**, p 2255.

[127] Iser, R. and Wahl, F.M., AntSLAM: Global map optimization using swarm intelligence, International Conference on Robotics and Automation, 3-7 May, Alaska, USA, 2010, p 265.

[128] X. Chen, Y. Kong, X. Fang, Q. Wu, A fast two–stage ACO algorithm for robotic path planning, Neural Computing and Applications, 22 (**2013**) 313.

[129] M.A.P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, Optimal path planning for autonomous mobile robot navigation using ant colony optimization and a fuzzy cost function evaluation, Adv. Soft Computing, 41 (**2007**) 790.

[130] Hoff, N.R., Sagoff, A., Wood, R.J. and Nagpal, R., Two foraging algorithms for robot swarms using only local communication, In IEEE International Conference on Robotics and Biomimetics, 14-18 December, Tianjin, China, **2010**, p 123.

[131] Zou, Y. and Luo, D., A modified ant colony algorithm used for multi-robot odor source localization, In International Conference on Intelligent Computing, Berlin Heidelberg, **2008**, p 502.

[132] X. Feng, Y. Liang, L. Jiao, Bio-inspired optimisation approach for data association in target tracking, International Journal of Wireless and Mobile Computing, 6 (**2013**) 299.

[133] M. Dorigo, L. Gambardella, Ant colonies for the traveling salesman problem, BioSystems, 43 (**1997**) 73–81.

[134] S. Xavier-De-Souza, J. A. Suykens, J. Vandewalle, and D. Bollé, Coupled simulated annealing. IEEE transactions on systems, Man, and Cybernetics, 40 (**2009**) 320.

[135] O. Ekren and B. Y. Ekren, Size optimization of a PV/wind hybrid energy conversion system with battery storage using simulated annealing. Applied energy, 87 (**2010**) 592.

[136] Y. Kuo, Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Computers & Industrial Engineering, 59 (**2010**) 157.

[137] H. Miao and Y. C. Tian, Dynamic robot path planning using an enhanced simulated annealing approach. Applied Mathematics and Computation, 222 (**2013**) 420.

[138] L. P. Behnck, D. Doering, C. E. Pereira, and A. Rettberg, A modified simulated annealing algorithm for SUAVs path planning. IFAC-PapersOnLine, 48 (**2015**) 63.

[139] Ganeshmurthy, M.S. and Suresh, G.R., Path planning algorithm for autonomous mobile robot in dynamic environment, In 3rd International Conference on Signal Processing, Communication and Networking, 26-28 March, Chennai, India, **2015**, p 1.

[140] Turker, T., Sahingoz, O.K. and Yilmaz, G., 2D path planning for UAVs in radar threatening environment using simulated annealing algorithm, In International Conference on Unmanned Aircraft Systems, 9-12 June, Denver, USA, **2015**, p 56.

[141] B. Li, L. G. Gong, and W. L. Yang, An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning. The Scientific World Journal, (**2014**) 1.

[142] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Journal of global optimization, 39 (**2007**) 459.

[143] D. O. Boyer, C. H. Martnez, N. G. Pedrajas, Crossover Operator for Evolutionary Algorithms Based on Population Features, Journal of Artificial Intelligence Research, 24 (**2005**)1.

[144] J. H. Friedman, An overview of predictive learning and function approximation, In from Statistics to Neural Networks, 136 (1994) 1.

[145] Srinivasan, D. and Seow, T.H., Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problems, InThe Congress on Evolutionary Computation, 8-12 December, Canberra, Australia, **2003**, p 2292.

[146] Mira, J. and Álvarez, J.R. (Eds.), Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach: First International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2005, Vol. 3562, Springer, Las Palmas, Canary Islands, Spain, **2005**.

[147] H. F. Wedde, M. Farooq, Y. Zhang, BeeHive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior, ant colony, optimization and swarm intelligence, In: Proceedings of the 4th International Workshop, 5–8 September, Brussels, Belgium, **2004** p 83.

[148] C. Xu, H. Duan, and F. Liu, Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning, Aerospace Science and Technology, 14 (**2010**) 535.

[149] A. S. Bhagade and P. V. Puranik, Artificial bee colony (ABC) algorithm for vehicle routing optimization problem. International Journal of Soft Computing and Engineering, 2 (**2012**) 329.

[150] N. H. Abbas and F. M. Ali, Path Planning of an Autonomous Mobile Robot using Directed Artificial Bee Colony Algorithm, International Journal of Computer Applications, 96 (**2014**).

[151] Bhattacharjee, P., Rakshit, P., Goswami, I., Konar, A. and Nagar, A.K., Multi-robot path-planning using artificial bee colony optimization algorithm, In Third World Congress on Nature and Biologically Inspired Computing, 19-21 October, Salamanca, Spain, **2011**, p 219.

[152] X. S. Yang, Nature-inspired metaheuristic algorithm, Luniver press, **2008**.

[153] Christensen, A.L., O'Grady, R. and Dorigo, M., Synchronization and fault detection in autonomous robots, IEEE/RSJ International Conference on Intelligent Robots and Systems, 22-26 September, Nice, France, **2008**, p 4139.

[154] A. Theofanis and A. Vlachos, Application of the firefly Algorithm for solving the economic emission load dispatch problem, International Journal of Combinatorics, 29 (**2011**).

[155] dos Santos Coelho, L., de Andrade Bernert, D.L. and Mariani, V.C., A chaotic firefly algorithm applied to reliability-redundancy optimization, In IEEE congress of evolutionary computation, 5-8 June, New Orleans, USA, 2011, p 517.

[156] A. H. Gandomi, X. S. Yang and A. H. Alavi, Mixed variable optimization using firefly algorithm, Computers and Structures, 89 (**2011**) 2325.

[157] Falcon, R., Li, X., Nayak, A. and Stojmenovic, I., A Harmony-seeking firefly swarm to the periodic replacement of damaged sensors by a team of mobile robots, 2012 IEEE International Conference on Communications, 10-15 June, Ottawa, Canada, **2012**, p 4914.

[158] A. Rahmani and S. A. MirHassani, A hybrid firefly-genetic algorithm for the capacitated facility location problem, Information Sciences, 283 (**2014**) 70.

[159] M. Mitic and Z. Mijkovic, Bio inspired approach to learning robot motion trajectories and visual control commands, Expert Systems with Application, 42 (**2015**) 2624.

[160] N. Babak and M. R. Meybodi, Speciation based firefly algorithm for optimization in dynamic environment, International Journal of Artificial Intelligence, 8 (**2012**) 118.

[161] M. Farahami, A. A. Abshouri, B. Nasiri and M. R. Meybodi, A Guassian firefly algorithm, International Journal of Machine Learning and Computing, 1 (**2011**) 448.

[162] S.L.Tilahun and J.M.T. Ngnotchouye, Firefly algorithm for discrete optimization problems: A survey, KSCE Journal of civil Engineering, 21 (2017) 535.

[163] A. Baykasoglu and F. B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimization problems, Applied Soft Computing, 36 (**2015**) 152.

[164] X. S. Yang, Multi-objective firefly algorithm for continuous optimization, Engineering with Computers, 29 (**2013**)175.

[165] A. Baykasoglu and B. O. Fehmi, An improved firefly algorithm for solving dynamic multidimensional knapsack problems, Expert systems with application, 41 (**2014**) 3712.

[166] M. Alweshah and S. Abdullah, Hybridizing firefly algorithm with a probabilistic neural network for solving classification problems, Applied Soft Computing, 35 (**2015**) 513.

[167] M. K. Kaushik and V. Aroro, A Hybrid data clustering using firefly algorithm based improved genetic algorithm, Procedia Computer Science, 58 (**2015**) 249.

[168] D. Zouache, F. Nouioua and A. Moussaoui, Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems, Soft Computing, 20 (**2015**) 2781.

[169] A. Baykasoglu and F. B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimisation problems, Applied soft computing, 36 (**2015**) 152.

[170] I. Fister, I. Fister Jr., X.-S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm and Evolutionary Computation, 13 (**2013**) 34.

[171] A. Hidalgo-Paniagua, A. Miguel, Vega–Rodriguez, J. Ferruz and N. Pavon, Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach, Soft Computing, 21 (**2015**) 949.

[172] Brand, M. and Yu, X.H., Autonomous robot path optimization using firefly algorithm, International Conference on Machine Learning and Cybernetics, 14-17 July, Tianjin, China, **2013**, p 14.

[173] D. Sutantyo and P. Levi, Decentralized underwater multi robot communication using bio-inspired approaches, Artificial Life Robotics, 20 (**2015**) 152.

[174] Sutantyo, D., Levi, P., Möslinger, C. and Read, M., Collective-Adaptive Leavy Flight for Underwater multi-robot exploration, International conference on mechatronics and automation, 4-7 August, Takamatsu, Japan, **2013**, p 456.

[175] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J., Kamruzzaman, and M. Murshed, Search and tracking algorithms for swarms of robots: A survey. Robotics and Autonomous Systems, 75 (**2016**) 422.

[176] G. Wang, L. Guo, H. Duan, L. Liu, H. Wang, A modified firefly algorithm for UCAV path planning International Journal of Hybrid Information Technology, 5 (**2012**) 123.

[177] I. Fister, I. Fister Jr., X.-S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm and Evolutionary Computation, 13 (**2013**) 34.

[178] Abshouri, A.A., Meybodi, M.R. and Bakhtiary, A., New firefly algorithm based on multi swarm & learning automata in dynamic environments, In 3rd IEEE International Conference on Signal Processing Systems, 27-28 August, Yantai, China, **2011**, p 989.

[179] S. Santokh, Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report, **2015**.

[180] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, A survey of motion planning and control techniques for self-driving urban vehicles, IEEE Transactions on intelligent vehicles, 1 (**2016**) 33.

[181] Q. Li, L. Chen, M. Li, S. L. Shaw, and A. Nüchter, A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios, IEEE Transactions on Vehicular Technology, 63 (**2013**) 540.

[182] V. John, K. Yoneda, Z. Liu, and S. Mita, Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching. IEEE transactions on computational imaging, 1 (**2015**) 159.

[183] B. Kim, B. Choi, S. Park, H. Kim, and E. Kim, Pedestrian/vehicle detection using a 2.5-D multi-layer laser scanner. IEEE Sensors Journal, 16 (**2015**) 400.

[184] E. Hashemi, A. Kasaiezadeh, S. Khosravani, A. Khajepour, N. Moshchuk, and S. K. Chen, Estimation of longitudinal speed robust to road conditions for ground vehicles. Vehicle System Dynamics, 54 (**2016**) 1120.

[185] D. González, J. Pérez, V. Milanés, and F. Nashashibi, A review of motion planning techniques for automated vehicles, IEEE Transactions on Intelligent Transportation Systems, 17 (**2015**) 1135.

[186] M. Ardelt, C. Coester, and N. Kaempchen, Highly automated driving on freeways in real traffic using a probabilistic framework, IEEE Transactions on Intelligent Transportation Systems, 13 (**2012**)1576.

[187] P. Falcone, H. Eric Tseng, F. Borrelli, J. Asgari, D. Hrovat, MPC-based yaw and lateral stabilisation via active front steering and braking. Vehicle System Dynamics, 46 (**2008**) 611.

[188] Erlien, S. M., Shared vehicle control using safe driving envelopes for obstacle avoidance and stability, Doctoral dissertation, Stanford University, **2015**.

[189] Hsieh, M. F., and Ozguner, U., A parking algorithm for an autonomous vehicle, In IEEE Intelligent Vehicles Symposium, 4-6 June, Eindhoven, Netherlands, **2008**, p.1155.

[190] M. Brezak and I. Petrović, Real-time approximation of clothoids with bounded error for path planning applications, IEEE Transactions on Robotics, 30 (**2013**) 507.

[191] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrila, Active pedestrian safety by automatic braking and evasive steering. IEEE Transactions on Intelligent Transportation Systems, 12 (2011) 1292.

[192] Liang, Z., Zheng, G., & Li, J., Automatic parking path optimization based on bezier curve fitting. In IEEE International Conference on Automation and Logistics, 15-17 August, Zhengzhou, China, **2012**, p. 583.

[193] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist, Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles, IEEE Transactions on Automation Science and Engineering, 7(**2009**) 167.

[194] Y. H. Kim, J. B. Park, W. S. Son, and T. S. Yoon, Modified turn algorithm for motion planning based on clothoid curve. Electronics Letters, 53 (**2017**)1574.

[195] Yang, S., Wang, Z., & Zhang, H., Kinematic model based real-time path planning method with guide line for autonomous vehicle, In IEEE 36th Chinese Control Conference, 26-28 July, Dalian, China, **2017**, p. 990.

[196] Sudhakara, P., Ganapathy, V., & Sundaran, K., Optimal trajectory planning based on bidirectional spline-RRT∗ for wheeled mobile robot, In IEEE Third

International Conference on Sensing, Signal Processing and Security, 4-5 May, Chennai, India, **2017**, p. 65.

[197] R. Kala and K. Warwick, Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization, Journal of Intelligent & Robotic Systems, 72 (**2013**) 559.

[198] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, Path planning for autonomous vehicles in unknown semi-structured environments, The International Journal of Robotics Research, 29 (**2010**) 485.

[199] M. Likhachev and D. Ferguson, Planning long dynamically feasible maneuvers for autonomous vehicles. The International Journal of Robotics Research, 28 (**2009**) 933.

[200] Gu, T., Snider, J., Dolan, J. M., & Lee, J. W., Focused trajectory planning for autonomous on-road driving, In 2013 IEEE Intelligent Vehicles Symposium, 23-26 June, Gold Coast, Australia, **2013**, p. 547.

[201] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, Real-time motion planning with applications to autonomous urban driving, IEEE Transactions on Control Systems Technology, 17 (**2009**) 1105.

[202] Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., & Teller, S., Anytime motion planning using the RRT, In 2011 IEEE International Conference on Robotics and Automation, 9-13 May, Shanghai, China, **2011**, p. 1478.

[203] J. Wang, J. Wu, and Y. Li, The driving safety field based on driver–vehicle–road interactions, IEEE Transactions on Intelligent Transportation Systems, 16 (**2015**) 2203.

[204] Wolf, M. T., & Burdick, J. W., Artificial potential functions for highway driving with collision avoidance, In 2008 IEEE International Conference on Robotics and Automation, 19-23 May, Pasadena, USA, **2008**, p. 3731.

[205] Gao, Y., Lin, T., Borrelli, F., Tseng, E., & Hrovat, D., Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads, In ASME 2010 dynamic systems and control conference, American Society of Mechanical Engineers, **2010**, p. 265.

[206] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, Receding horizon maneuver generation for automated highway driving, Control Engineering Practice, 41(**2015**) 124.

[207] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, Automated driving: The role of forecasts and uncertainty—A control perspective, European Journal of Control, 24 (**2015**)14.

[208] K. Zhang, J. Sprinkle, and R. G. Sanfelice, Computationally aware control of autonomous vehicles: a hybrid model predictive control approach. Autonomous Robots, 39 (**2015**) 503.

[209] Frasch, J. V., Gray, A., Zanon, M., Ferreau, H. J., Sager, S., Borrelli, F., & Diehl, M., An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles, In 2013 European Control Conference, 17-19 July, Zurich, Switzerland, **2013**, p. 4136.

[210] Y. Yoon, J. Shin, H. J. Kim, Y. Park, and S. Sastry, Model-predictive active steering and obstacle avoidance for autonomous ground vehicles, Control Engineering Practice, 17 (**2009**) 741.

[211] A. Liniger, A. Domahidi, and M. Morari, Optimization-based autonomous racing of 1: 43 scale RC cars, Optimal Control Applications and Methods, 36 (**2015**) 628.

[212] Zhang, K., Sprinkle, J., & Sanfelice, R. G., A hybrid model predictive controller for path planning and path following, In Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems,14-16 April, Seattle, Washington, **2015**, p. 139-148.

[213] Shen, C., Shi, Y., & Buckham, B., Model predictive control for an AUV with dynamic path planning, In 2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 28-30 July, Hangzhou, China, **2015**, p. 475.

[214] Liu, C., Lee, S., Varnhagen, S., & Tseng, H. E., Path planning for autonomous vehicles using model predictive control, In 2017 IEEE Intelligent Vehicles Symposium (IV), 11-14 June, Los Angeles, USA, **2017**, p. 174-179.

[215] Kong, J., Pfeiffer, M., Schildbach, G., & Borrelli, F., Kinematic and dynamic vehicle models for autonomous driving control design, In 2015 IEEE Intelligent Vehicles Symposium (IV), 28 June-1 July, Seoul, South Korea, **2015**, p. 1094.

[216] Yi, B., Gottschling, S., Ferdinand, J., Simm, N., Bonarens, F., & Stiller, C., Real time integrated vehicle dynamics control and trajectory planning with MPC

for critical maneuvers, In 2016 IEEE Intelligent Vehicles Symposium (IV), 19-22 June, Gothenburg, Sweden, **2016**, p. 584.

[217] Turri, V., Carvalho, A., Tseng, H. E., Johansson, K. H., & Borrelli, F., Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads, In 16th international IEEE conference on intelligent transportation systems, 6-9 October, The Hague, Netherlands, **2013**, p. 378.

[218] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles, Vehicle System Dynamics, 52 (**2014**) 802.

[219] S. M. Erlien, S. Fujita, and J. C. Gerdes, Shared steering control using safe envelopes for obstacle avoidance and vehicle stability, IEEE Transactions on Intelligent Transportation Systems, 17(**2015**) 441.

[220] Funke, J., Brown, M., Erlien, S. M., & Gerdes, J. C., Prioritizing collision avoidance and vehicle stabilization for autonomous vehicles, In 2015 IEEE Intelligent Vehicles Symposium (IV), 28 June-1 July**,** Seoul, South Korea**, 2015**, p. 1134.

[221] Gray, A., Gao, Y., Hedrick, J. K., & Borrelli, F., Robust predictive control for semi-autonomous vehicles with an uncertain driver model, In 2013 IEEE Intelligent Vehicles Symposium (IV), 23-26 June, Gold Coast, Australia, **2013**, p. 208.

[222] M. A. Abbas, R. Milman, and J. M. Eklund, Obstacle avoidance in real time with Nonlinear Model Predictive Control of autonomous vehicles, Canadian journal of electrical and computer engineering, 40 (**2017**)12.

[223] L. Cherroun and M. Boumehraz, Fuzzy behavior-based navigation approach for mobile robot in unknown environment, Journal of Electrical Engineering, 13 (**2013**) 1.

[224] B. K. Patle, A. Pandey, A. Jagadeesh, and D. R. Parhi, Path planning in uncertain environment by using firefly algorithm. Defence technology, 14 (**2018**) 691.

[225] W. Shi, K. Wang, S. K. Yang, A fuzzy-neural network approach to multi-sensor integration for obstacle avoidance of a mobile robot, Intelligent Automation & Soft Computing,15(**2009**) 289.

[226] H. Mo, Q. Tang and L. Meng, Behavior-Based Fuzzy Control for Mobile Robot Navigation, Mathematical Problems in Engineering, (**2013**).

[227] J. M. Park, D. W. Kim, Y. S. Yoon, H. J. Kim, and K. S. Yi, Obstacle avoidance of autonomous vehicles based on model predictive control, Proceedings of the Institution of Mechanical Engineers, Journal of Automobile Engineering, 223 (**2009**) 1499.

[228] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles, Vehicle System Dynamics, 54 (**2016**)1629.

[229] Gotte, C., Keller, M., Hass, C., Glander, K. H., Seewald, A., & Bertram, T., A model predictive combined planning and control approach for guidance of automated vehicles, In 2015 IEEE International Conference on Vehicular Electronics and Safety, 5-7 November, Yokohama, Japan, **2015**, p. 69.

[230] Liao, Y. W., & Hedrick, J. K., A discrete-time integral sliding model predictive control for obstacle avoidance of ground vehicles, In ASME 2015 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 28–30 October, Columbus, USA, **2015**.

[231] Qian, X., De La Fortelle, A., & Moutarde, F., A hierarchical model predictive control framework for on-road formation control of autonomous vehicles, In 2016 IEEE Intelligent Vehicles Symposium (IV), 19-22 June, Gothenburg, Sweden, **2016**, p. 376.

[232] Jalalmaab, M., Fidan, B., Jeon, S., & Falcone, P., Model predictive path planning with time-varying safety constraints for highway autonomous driving, In 2015 International Conference on Advanced Robotics, 27-31 July, Istanbul, Turkey, **2015**, p. 213.

[233] Carvalho, A., Gao, Y., Lefevre, S., & Borrelli, F., Stochastic predictive control of autonomous vehicles in uncertain environments, In 12th International Symposium on Advanced Vehicle Control, 22-26 September, Tokyo, Japan, **2014**, p. 712.

[234] S. Saremi, S. Mirjalili, and A. Lewis, Grasshopper optimisation algorithm: theory and application, Advances in Engineering Software, 105 (**2017**) 30.

[235] F. Kamil, T. S. Hong, W. Khaksar, M. Y. Moghrabiah, N. Zulkifli, and S. A. Ahmad, New robot navigation algorithm for arbitrary unknown dynamic environments based on future prediction and priority behavior, Expert Systems with Applications, 86 (**2017**) 274.

[236] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, Safe driving envelopes for path tracking in autonomous vehicles, Control Engineering Practice, 61(**2017**) 307.

[237] Franklin, G. F., Powell, J. D., & Workman, M. L., Digital control of dynamic systems, Vol. 3, Menlo Park, CA: Addison-wesley, **1998**.

[238] Silva-Ortigoza, R., Márquez-Sánchez, C., Carrizosa-Corral, F., Hernández-Guzmán, V.M., García-Sánchez, J.R. and Bennett, K., Obstacle avoidance task for a wheeled mobile robot–a matlab-simulink-based didactic application, In MATLAB Applications for the Practical Engineer, IntechOpen, **2014**.

[239] M. Tanaka, K. Nakata, Positive definite matrix approximation with condition number constraint, Optimization Letters, 8 (**2014**) 939.

[240] P. T. Boggs and J. W. Tolle, Sequential quadratic programming, Acta numerica, 4 (**1995**) 1.