Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Kerbala/College of Engineering

Electrical and Electronics Department

Optimization of Path Tracking of Self-Acting Mobile Robotic System

A Thesis

Submitted to the Department of Electrical and Electronics Engineering, University of Kerbala in Partial Fulfillment of the Requirements for the Degree of Master of Science (MSc) in Electrical Engineering

Written By:

Ghaidaa Hadi Salih

B.Sc. in Electrical and Electronics Engineering /University of Kerbala (2014)

Supervised By:

Assist. Prof. Dr. Ahmed Abdulhadi Ahmed

Assist. Prof. Dr. Haider Galil Kamil

202٢A.D **١٤٤3A.H**

Supervisor Certification

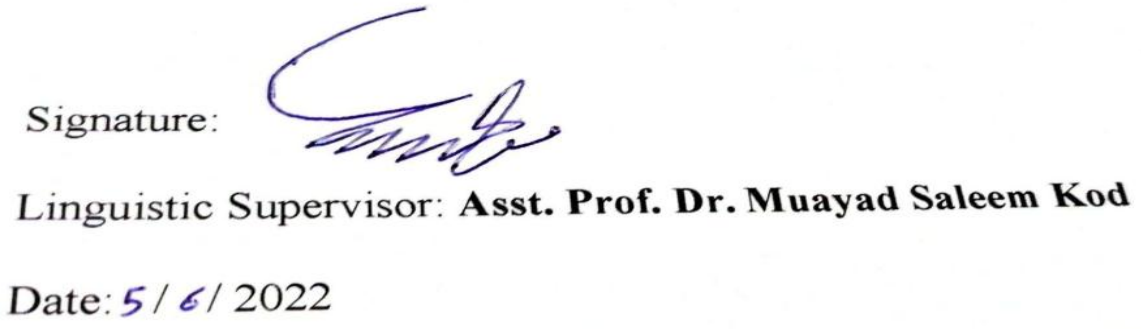
We certify that the thesis entitled **"Optimization of Path Tracking of Self-Acting Mobile Robotic System"** which is being submitted by **Ghaidaa Hadi Salih,** has been carried out completely under our supervision at College of Engineering, University of Kerbala, in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering.

Signature:
Asst. Prof. Dr. **Ahmed Abdulhadi Ahmed**
Date: / / 2022

Signature:
Asst. Prof. Dr. **Haider Galil Kamil**
Date: / / 2022

Linguistic Certification

I hereby certify that this thesis entitled "**Optimization of Path Tracking of Self-Acting Mobile Robotic System**" has been proofread and entitled under my linguistic supervision and was amended to meet proper English language.



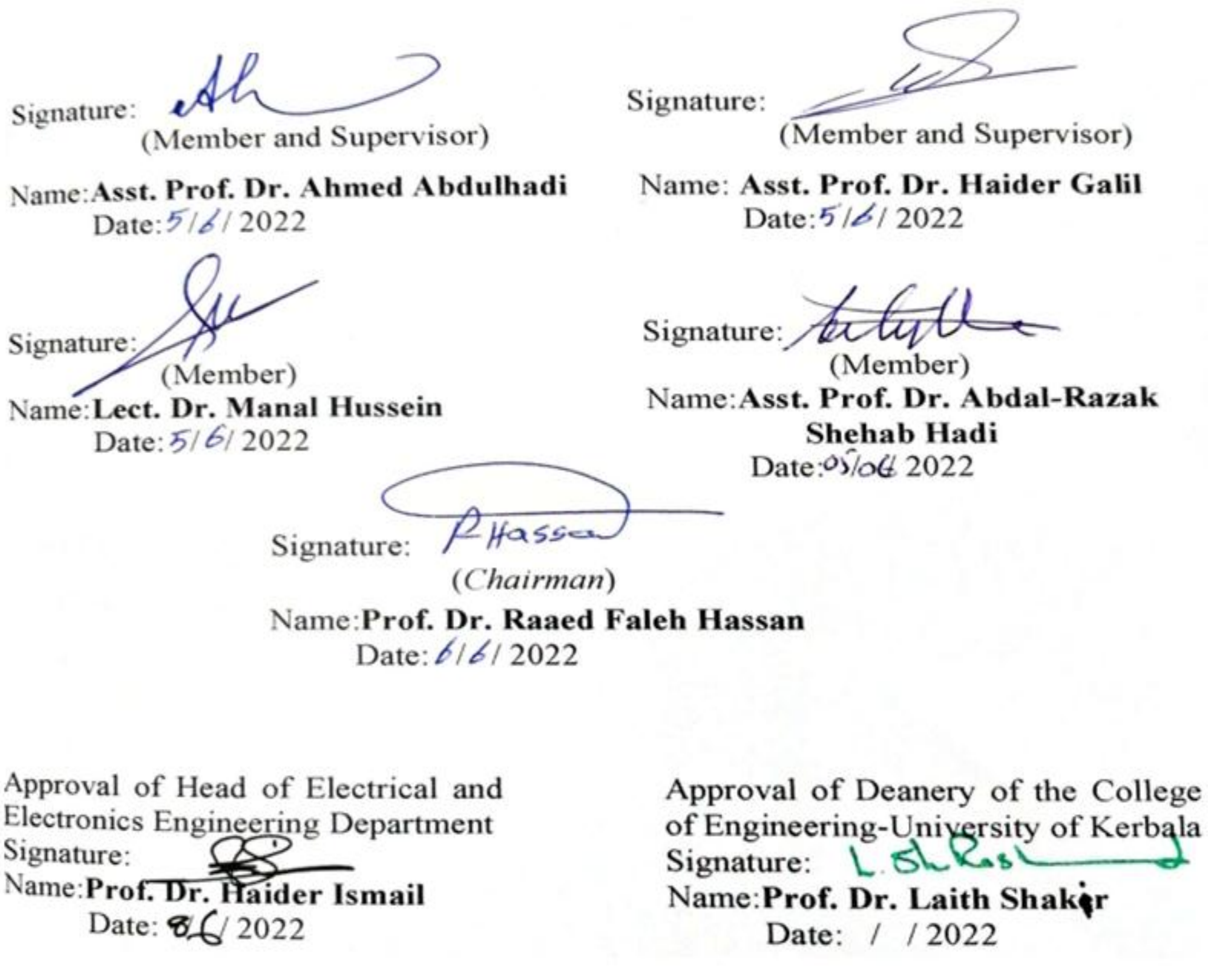
Signature:

Linguistic Supervisor: **Asst. Prof. Dr. Muayad Saleem Kod**

Date: 5 / 6 / 2022

Examining Committee Certificate

We certify that we have read the thesis entitled “**Optimization of Path Tracking of Self-Acting Mobile Robotic System**” and as an examining committee, we examined the student “**Ghaidaa Hadi Salih,**” in its content and in what is connected with it, and that in our opinion it is adequate as a thesis for the degree of Master of Science in Electrical Engineering.



Signature:
(Member and Supervisor)
Name: **Asst. Prof. Dr. Ahmed Abdulhadi**
Date: 5/6/ 2022

Signature:
(Member and Supervisor)
Name: **Asst. Prof. Dr. Haider Galil**
Date: 5/6/ 2022

Signature:
(Member)
Name: **Lect. Dr. Manal Hussein**
Date: 5/6/ 2022

Signature:
(Member)
Name: **Asst. Prof. Dr. Abdal-Razak Shehab Hadi**
Date: 05/06/ 2022

Signature: R Hassan
(*Chairman*)
Name: **Prof. Dr. Raaed Faleh Hassan**
Date: 6/6/ 2022

Approval of Head of Electrical and Electronics Engineering Department
Signature:
Name: **Prof. Dr. Haider Ismail**
Date: 8 6/ 2022

Approval of Deanery of the College of Engineering-University of Kerbala
Signature:
Name: **Prof. Dr. Laith Shakir**
Date: / / 2022

بسم الله الرحمن الرحيم

وَقُلْ رَبِّي زِدْنِي

عِلْــــــــماً

صَدَق اللّه العلي العظيم

سورة طه الاية ١١٤

Dedication

To my Homeland

And

To my Family

For their Love and Support and Encourage

Acknowledgement

Praise be to Allah, Lord of the Worlds, for endowing me with health, patience, and knowledge to complete this work.

I would like to take this opportunity to express my sincere thanks and gratitude to my supervisors, **Dr. Ahmed Abdulhadi Ahmed** and **Dr. Haider Galil Kamil** for their advice, support, suggestions, and encouragement throughout the work.

I wish to thank my family, particularly my parents, for generous and continuous support and encouragement during the study and research periods.

My thanks also go to my friends and colleagues who have given me their support and help.

Ghaidaa Hadi Salih
202٢

Abstract

An autonomous mobile robot system such as an autonomous vehicle requires high criteria of accuracy to track the desired speed profiles and predefined trajectory. Some challenges in the development of self-acting systems, like building a mathematical model, designing longitudinal and lateral control systems, and the optimization technique are still significant topics.

This thesis is concerned with constructing the (vehicle and bicycle) models which govern the motion of the autonomous vehicle in terms of the longitudinal dynamic vehicle model and both longitudinal, lateral motions for the kinematic bicycle model. To set the motion of the driverless vehicle, various control strategies are proposed.

The Proportional Integral Derivative (PID) controller was suggested to manipulate throttle/brake actuators of the longitudinal dynamic vehicle model to track various reference speeds. Besides, the proportional (P) controller was proposed for longitudinal motion while four controllers for lateral motion were proposed to steer the kinematic bicycle model at various speeds correctly which are: Path Tracking Modified Proportional Integral Derivative (PTMPID), PID, Stanley, and Modified Stanley (MS) controllers. The comparison between lateral controllers was accomplished to know which one is the best.

The parameters of the controllers are optimized with two new optimization methods: Hybrid Salp Swarm and Butterfly Optimization Algorithms (HSSABOA1 and HSSABOA2). However, to investigate the performance competence of the suggested algorithms, they have been compared with the basic algorithms: Butterfly Optimization Algorithm (BOA) and Salp Swarm Algorithm (SSA) in addition to commonly used algorithms; Genetic Algorithm (GA), Ant

Colony Optimization (ACO) algorithm, and Particle Swarm Optimization (PSO) algorithm. In these optimization techniques, the Integral Absolute Error (IAE) and Root Mean Square Error (RMSE) are used as objective functions to minimize tracing errors of the speed and steering, respectively.

The simulation results found that the PID controller based on HSSABOA1 for the dynamic vehicle model has the best solution to reduce the speed error with an improvement percentage of (8.6088%), (5.0226%), (2.5074%), (0.0306%), and (0.2295%) than PID based on GA, ACO, PSO, BOA, and SSA, in the order. Moreover, it also has traced various speed profiles successfully.

In contrast, the PTMPID based on HSSABOA1 has better performance; in the minimization of the lateral error beside a percentage enhancement by (10.523%), (19.456%), (83.276%), (18.263%), and (94.005%), than PTMPID-SSA, PTMPID-BOA, Stanley-HSSABOA2, modified Stanley-HSSABOA2, and PID-HSSABOA1, respectively. In addition, it also managed to track the road maneuver with various longitudinal speeds without large vacillation.

List of Contents

List of Abbreviations

Abbreviation	Meaning
ACO	Ant Colony Optimization
ADCs	Autonomous driving cars
AUV	Autonomous Underwater Vehicle
BOA	Butterfly Optimization Algorithm
COVID-19	Corona Virus Disease 2019
DARPA	Defense Advanced Research Projects Agency
DOF	Degree of Freedom
ELROB	European Land Robot Trial
GA	Genetic Algorithm
Gbest	Global best
GCDC	Grand Cooperative Driving Challenge
GNSS	Global Navigation Satellite System
GSA	Gravitational Search Algorithm
GWO	Gray Wolf Optimization
HSSABOA	Hybrid Salp Swarm Algorithm and Butterfly Optimization
IAE	Integral Absolute Error
IMUs	Inertial Measurement Units
ISE	Integral Square Error
ITAE	Integral Time Absolute Error
Lb	Lower bounds
MPC	Model Predictive Control
MS	Modified Stanley
MSE	Mean Square Error
OF	Objective Function
P	Proportional
Pbest	Particle best
PI	Proportional Integral
PID	Proportional Integral Derivative
POP	Proximally Optimal Predictive
PSO	Particle Swarm Optimization
PTMPID	Path Tracking Modified Proportional Integral Derivative
RMSE	Root Mean Square Error
RPM	Revolutions Per Minute

SAE	Society of Automotive Engineers
SLAM	Simultaneous Localization and Mapping
SSA	Salp Swarm Algorithm
SSE	Steady State Error
ub	Upper bounds
V-REP	Virtual Robot Testation Platform

List of Symbols

Symbol	Description
a	Acceleration of the vehicle (rad/s^2)
a_0, a_1, a_2	Constant coefficients of the maximum torque
A	Cross-sectional area (m^2)
B	Heuristic exponent
β	Vehicle side slip angle (rad)
c	The sensory modality
c_1	The main coefficient gives a balance between the exploration and exploitation
c_2, c_3	Random numbers between 0 and 1
C_a	Drag coefficient
C_r	Rolling resistance coefficient
C_s	Longitudinal stiffness of the tire
d$_{ij}$	A distance from nodes i to the joined node j
Dim	Length of each agent
D	Air density (kg/m^3)
e	Error between current and target measurements (rad/s)
f	The perceived magnitude for the fragrance
F_{aero}	Aerodynamic force (N)
F_j	The food source position in the j^{th} dimension
F_x	Longitudinal force (N)
F_{xf}	Front tire force (N)
F_{xr}	Rear tire force (N)
g	Acceleration of gravity (m/s^2)
g_i	Best particle between members of the group
GR	Gear ratio
I	The stimulus intensity
I_f	Distance from the bicycle center to the center of the front wheel (m)
I_r	Distance from the bicycle center to the center of the rear wheel (m)
J_e	Inertia moment of an engine (kg/m^2)
k_d	Derivative gain of the PID controller
k_i	Integral gain of the PID controller

k_p	Proportional gain of the PID controller
$k_1, k_2\ , k_3\ , k_4$	Modified Stanley controller’s gain
K	Stanley controller’s gain
lb_j	lower bound of the j^{th} dimension
L_k	The length of the round generated by ant k
m	Mass of the vehicle (kg)
N	Maximum iteration
$N(S^P)$	The set of nodes
OS	Overshoot
P	Switch probability
p_i	Best position of the particles
Q	Maximum number of iterations
r_{eff}	The radius of the wheel (m)
r_1, r_2	Random values between the range 0 and 1
R	A random number between the interval 0 and 1
R_x	Total rolling resistance force (N)
R_{xf}	Front rolling resistance force (N)
R_{xr}	Rear rolling resistance force (N)
S	Longitudinal slip
S^P	The partial solution
t	Iterations number
T	Maximum number of the time steps
T_e	Engine torque (N-m)
T_{load}	Friction torque (N-m)
T_{emax}	Maximum of the engine torque (N-m)
T_{emin}	Minimum of the engine torque (N-m)
t_r	Rise time (s)
t_s	Settling time (s)
u	Control input
ub_j	Upper bound of the j^{th} dimension
v	Vehicle speed (m/s)
v_i	Speed of the particle
w_e	Engine angular speed (rad/s)

w_w	Wheel angular speed (rad/s)
x_c	Vehicle coordinate in the global frame (m)
x_i	Position of the particle
X_j^1	The leader position in the j^{th} dimension
X_j^i	Position of the follower salp in the j^{th} dimension
X^*	The best solution found
y_c	Vehicle coordinate in the global frame (m)
z_1	Cognitive learning factor
z_2	social learning factor
$\square_{ij}$	The heuristic information
τ	The power exponent attached to modality
θ	Vehicle yaw angle (rad)
θ_e	Heading error of the vehicle (rad)
θ_p	Orientation of the reference path (rad)
α	Pheromone exponent
δ	Steering angle (rad)
φ	Steering rate (rad/s)
ϕ	Drag angle between the ground and vehicle (rad)

List of Tables

List of Figures

Chapter One: Introduction

Chapter One
Introduction

1.1 Introduction

1.1.1 Background

The term “Autonomous mobile robot,” means a system that can navigate autonomously by itself from one place to other to achieve a set of actions or perform a task without assistance from external human operators [1-3]. Autonomous mobile robots can accomplish a range of locomotion techniques such as flying, swimming, crawling, walking, or rolling. Therefore, it is used in many fields such as factories, hospitals, and transportation [1]. This thesis focuses on one of the important applications of wheeled mobile robots which is autonomous driving cars (ADCs).

Autonomous cars (also called self-driving cars or driverless vehicles) are a complex system that introduces many services to society like reducing road accidents, increasing safety, and providing convenient transportation for passengers [4-6]. ADCs can reduce the influence of driver errors as the reason for car collisions. In addition, it can provide contrivances special mobility to persons who are incapable to drive as cause visual disability or physical [6].

It is also contributed to limit the spread of infectious diseases such as Corona Virus Disease 2019 (COVID-19) by delivering food and medicine to the infected people and transporting the non-critical cases of patients infected with that disease to keep the health of the drivers [7].

1.1.2 Levels of Autonomous Vehicles

According to the Society of Automotive Engineers (SAE) J3016 standard, autonomous vehicles have been classified into five levels of autonomous as follows [6][8]:

Level 0 (no automation): At that level, the driving is entirely done by the driver [6][8].

Level 1 (driver assistance): It includes one automated function such as adaptive cruise control [6][8][9].

Level 2 (partial automation): It contains several automated functions such as longitudinal or lateral control in addition to contingency braking [6][8].

Level 3 (conditional automation): The vehicle can perform many automated functions under definite conditions, but the driver is necessary to perform control if the vehicle leaves the automated functions [6][8][9].

Level 4 (high automation): The vehicle accomplishes all driving functions under specific situations. As well, the driver is not necessary with that level [6][8][9].

Level 5 (full automation): At that level, the vehicle performs wholly driving functions in all conditions without driver [6][8][9].

1.1.3 Architecture of an Autonomous Vehicle System

The architecture of an ADC system is divided into two groups: Hardware architecture and software architecture as shown in Figure (1.1) [10].

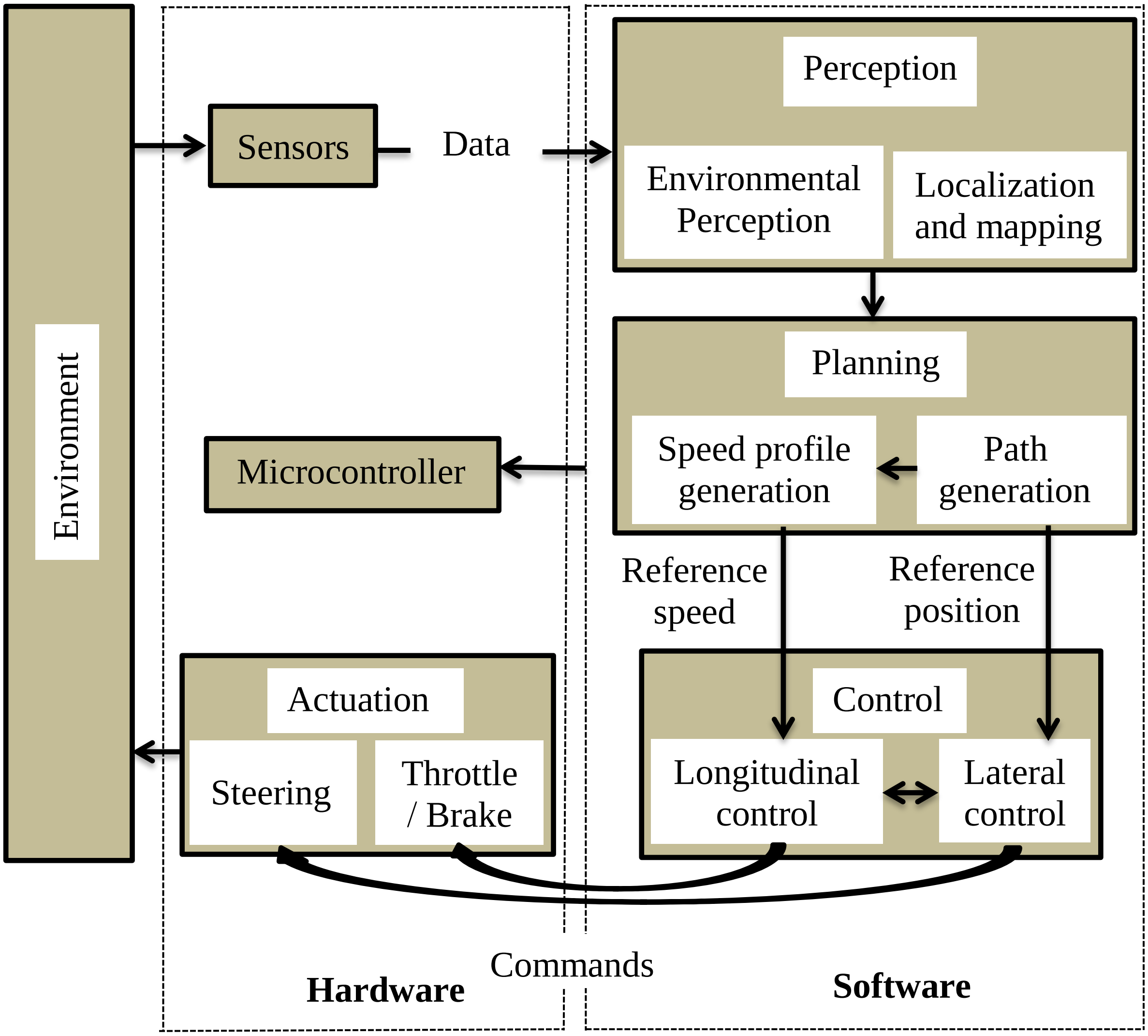


Figure (1.1): Architecture of an ADC.

The hardware architecture can be categorized into three parts: The sensor devices, microcontroller, and actuation systems. On the other hand, the software architecture can be classified into three parts: Perception, planning, and control. Python software is utilized for microcontroller programming [10][11].

Perception: The perception part receives the information from the sensor devices; including the environmental perception, localization, and mapping [10][12][13].

The environmental perception collects information about the autonomous vehicle's environment such as objects detection [10][12]. On the other hand, localization allows an ADC to locate itself inside a map of its surrounding environment. This requires stating its orientation inside the map. Localization depends mainly on sensor devices such as Inertial Measurement Units (IMUs), Global Navigation Satellite System (GNSS), odometry, cameras, or lidar [13].

The mapping part of ADC is used to build a correct picture from the environment around it. This possibly depends on a global or local map. This map should be extremely accurate to enable an ADC to run safely [13].

Planning: The planning functions are described by three layers: route planning, behavioral planning, and motion planning. At route planning, the ADC accomplishes the calculation for determining the best route to navigate from the current position to the target location based on the map information [13]. After finding a route plan, the ADC can navigate the specified route according to road rules and driving conventions. This action is accomplished chiefly by the behavior layer [6]. Motion planning is employed to generate the desired trajectory and speed that the control system must follow [6].

Vehicle Control: The control system of an ADC is categorized into longitudinal and lateral control. Where the longitudinal control is used to manipulate the (throttle/brake) actuators. Besides, lateral control is employed to set the steering angle of the vehicle. The main function of the control system is to execute the decision provided by the planning part by sending commands to the actuators to drive the vehicle correctly in the environment [6][13].

This thesis focused on studying the control part of ADCs for tracking road’s speed and trajectory accurately in the environment. The accuracy of the control system is required to attain the purpose of safely and efficiently driving. That accuracy is achieved by finding the best gains of the controllers. The adjusting methods based on mathematical and trial and error are not guaranteed to reach best gains, therefore, the optimization techniques are employed to obtain the best gains.

1.1.4 Optimization Techniques

The optimization method has been employed in many various domains, like engineering applications, data mining, networks, energy, medicine, and economics [14].

It is principally used to discover several best values to generate a solution that can solve the problem optimally [14].

The main objective of employing the optimization algorithms is to find the best problem solutions by optimizing (maximizing or minimizing) its Objective Function (OF) or fitness function. Optimization algorithms can be categorized into four main groups: continuous or discrete, constrained or unconstrained, single or multi-objective function, and dynamic or static [14].

The general classes of these techniques are displayed in Figure (1.2) [14]. Heuristic algorithms require special knowledge on the problem to be optimized. In contrast, the application of meta-heuristic does not require special knowledge related to the optimization problem, but it is used to state the idea of the general issue solving model [15].

The meta-heuristic techniques can be classified into the following categories [14]:

1. Techniques based on swarm search,
2. Techniques based on evolutionary search,
3. Techniques based on local search.

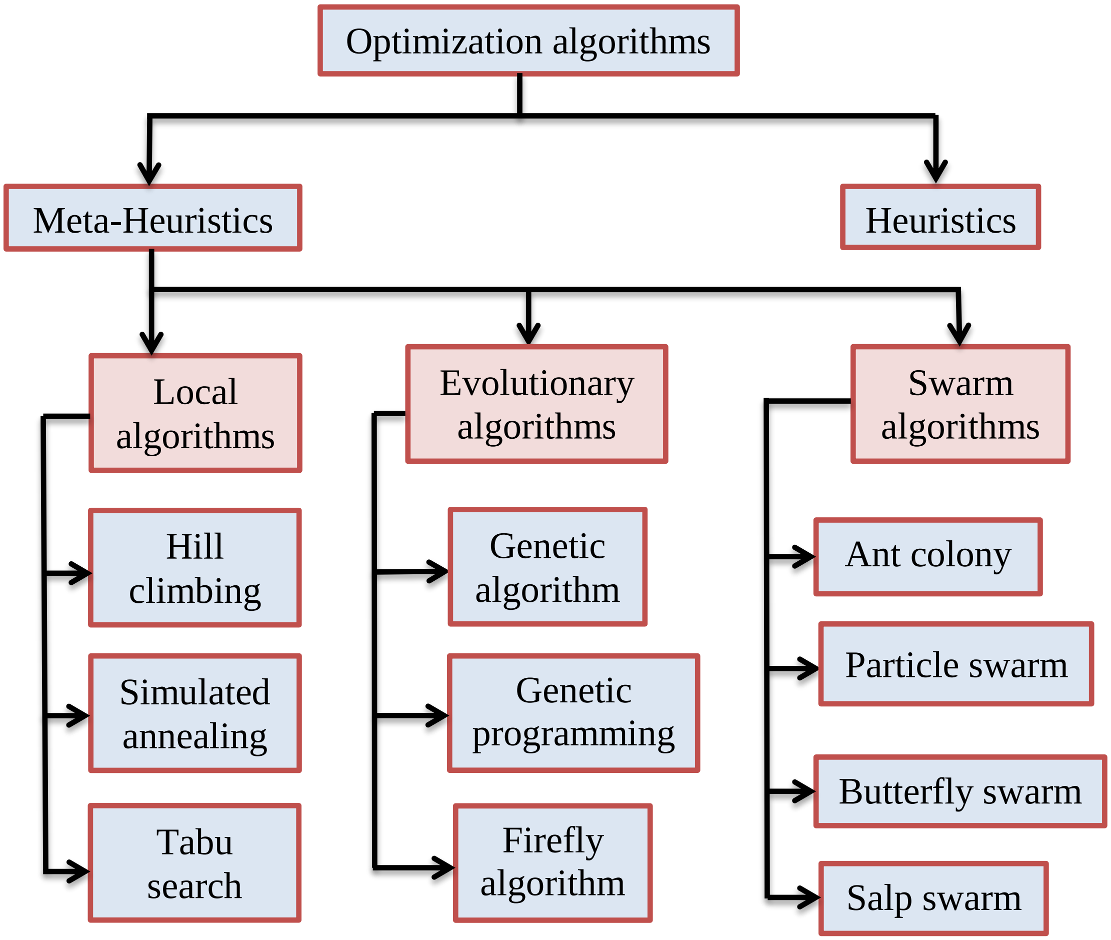


Figure (1.2): General categories of optimization algorithms.

The first class; is the swarm algorithms that are based on a population, where, at each repetitive, the present solutions are usually produced based on information found by previous generations. Several techniques in this group are included, the famous examples: PSO, ACO, BOA, and SSA [14].

The second class; is the evolutionary algorithms based on population strategy, which are a collection of randomly generated solutions. These solutions are mixed iteratively, until the best solution is reached, for getting on better and new solutions in means of its objective function. Examples involve genetic algorithms, firefly algorithms, and genetic programming [14].

The third class; is the local algorithms which works by one potential solution, that will be iteratively developed (increase its objective function) until stopping criteria is reached. Examples include tabu search, hill climbing, and simulated annealing [14].

1.2 Problem Statement

The problem of longitudinal and lateral control for an autonomous car is concentrated on adjusting the actuators (Throttle, brake, and steering angle) for tracking the desired speed profiles and path defined by the motion planning. In many cases, the researchers encountered a large challenge to find the best values of the controllers' parameters. The classical adjusting methods based on mathematical functions or trial and error have been found inappropriate in many cases. To avoid this issue, optimization methods have been suggested lately for optimizing controllers' gains. On the other hand, many researchers have faced a problem in steering the ADCs on the desired road at various speeds without large oscillation.

1.3 Research Aim and Objectives

The goals of this thesis are organized as follows:

1. Modeling the longitudinal dynamic vehicle and the kinematic bicycle to describe the longitudinal and lateral motion of an ADC, respectively.

2. Designing a combined PID controller with two powerful suggested optimization methods (HSSABOA1 and HSSABOA2) to the longitudinal dynamic vehicle model in order to trace different reference speeds efficiently.
3. The comparison between PID controller based on GA, ACO, PSO, BOA, and SSA with PID controller based on HSSABOA1 or HSSABOA2 to examine the effectiveness of the proposed optimization algorithms to obtain best gains for the PID controller.
4. Designing the proportional and PTMPID controllers based on the HSSABOA1 or HSSABOA2 to the kinematic bicycle model for tracing road maneuvers at various speeds without large oscillation. Where the P-HSSABOA is used for controlling the throttle/brake of the ADC, while the PTMPID-HSSABOA is employed to set the ADC steering angle for reference path tracking correctly in the environment. In addition, exploring the performance effectiveness of the proposed lateral controller (PTMPID based on HSSABOA1/HSSABOA2) by comparing it with three other lateral controllers which are the PID-HSSABOA, Stanley-HSSABOA, and MS-HSSABOA.

1.4 Contributions

This thesis has the following contributions:

1. Finding two forms of new and powerful optimization algorithms for the first time called the hybrid salp swarm algorithm with assistant of the butterfly optimization algorithm (HSSABOA1 and HSSABOA2) for optimizing the parameters of the longitudinal and lateral controllers.

2. Developing a new efficient controller called PTMPID for the first time based on HSSABOA1/HSSABOA2 to control the steering angle of the kinematic bicycle model.
3. Design a controller for the kinematic bicycle model able to track the trajectory with speed equivalent or less than 20 m/s correctly without the needing to design an adaptive or predictive controller to solve that matter.

1.5 Thesis Structure

The rest of this thesis is regulated as follows:

Chapter Two: This chapter involves preview to the related works of this study.

Chapter Three: This chapter presents the modeling, control, and optimization algorithms.

Chapter Four: This chapter presents the results and discussions of the applied methods in chapter three.

Chapter Five: This chapter introduces the main conclusions, and recommendations for future works that are related to this thesis.

Chapter Two: Review of the Related Literature

Chapter Two
Review of the Related Literature

2.1 Introduction

Autonomous vehicles have been developed and studied by many researchers, study centers, and vehicle companies since the middle 1980s. Since then until today, the projects about autonomous vehicles are increasing day by day [4].

In section 2.2 the related works about the control strategies of driverless vehicles are discussed.

2.2 Related Works

In the previous studies, several kinds of algorithms were used for the trajectory tracking of ADCs. These studies included geometric methods which have used geometric properties for the car, kinematic/dynamic methods which have used a kinematic/dynamic model of the vehicle, classical methods such as PID, intelligent methods such as neural network and fuzzy controllers, sliding mode controllers and adaptive controller methods.

The geometric methods are flexible and simple to implement. Therefore, many researchers have proposed these methods for tracking the desired trajectories. The two commonly used geometric methods are pure pursuit and Stanley controllers that gave good results for tracking defined trajectory. In addition, they are better at disturbance elimination in general. But they are somewhat problematic at high speeds. To overtake this problem, the gain scaling or adaptive look-forward methods are proposed [16][17].

Dynamic, kinematic, nonlinear, and linear models are used in several works of literature. They are comparatively complicated compared to geometrical controllers.

To perform the longitudinal control for an ADC, the researchers were also found various algorithms for solving this matter including adaptive control, PID control, fuzzy control, sliding mode control, and Model Predictive Control (MPC). One of the commonly employed controllers applied in longitudinal control of the dynamic, kinematic models for the vehicle is the PID controller because of its simplicity and inexpensive computation.

The gains of the longitudinal and lateral controllers were improved using various types of optimizations techniques; the widespread methods are GA, PSO, ACO, SSA, and BOA. Some of the previous works relative to the longitudinal and lateral control of ADCs are arranged as the following:

1. **Ping et al., in 2012 [18];** have proposed a strategy for the control with a vehicle model containing 9 Degree of Freedom (9DOF). Where the heading angle error was reduced by an adaptive fuzzy logic controller and the crosstrack error is recompensed with a Proportional Integral (PI) controller. Simulation results showed that the suggested automatic directing control can minimize lateral trajectory error and enhancing lateral trajectory maneuver largely.

2. **Kong et al., in 2015 [19];** have published a study about kinematic and dynamic models for an autonomous vehicle. Through which, they applied the MPC for the kinematic and dynamic bicycle models. So, they found the kinematic model has better prediction errors when discretized at 200 ms

than to 100 ms. In addition, they proved the success of the proposed methodology at several speeds in windy ways.

3. **Dominguez et al., in 2016 [20];** have published a study that compares three lateral controllers (Pure pursuit, Stanley, and sliding mode controllers). Also, they have discussed the fourth one called the kinematic controller-based lateral control. Thus, they have proved that the new method is the best one among other controls. On the other hand, they have applied all controllers on a real car as well its represented version on a simulator. They have pre-recorded the paths in a real-world environment using many Simultaneous Localization and Mapping (SLAM) maps systems. Also, they showed that Stanley method has little stability in high speed, low smoothness and it has good accuracy at low speed than the pursuit controller.

4. **Amer et al., in 2018 [21];** have designed the Stanley controller for an autonomous vehicle to track the reference path. They modified the basic Stanley controller and applied it to a non-linear, 7DOF armored vehicle model. The parameters of the controller are optimized using the PSO algorithm to select the best values. The performance of the controller was examined by comparing it with the original Stanley controller to guide the vehicle along several paths. It was found that the optimized controller is succeeded to improve the overall lateral error throughout the tests with 24% to 96% reduction in lateral error.

5. **Farag, in 2018 [22];** has designed the PID controller for ADCs maneuver tracking correctly. Three different methods are employed to tune the controller's parameters. One of them is trial and error based technique that

is proposed in this research for this specific application. Extensive studies in complex paths with many sharp turns have been performed to evaluate the suggested controller at various speeds. The simulation results showed that the proposed technique overtook the other ones.

6. **Herlambang et al., in 2019 [23];** suggested the PID controller to control the Autonomous Underwater Vehicle (AUV). The gains of the PID controller are optimized by two meta-heuristic methods: the ACO and PSO. IAE is used as the fitness function by these optimization methods to find the best gains. Based on the results of the simulation, the responses found by the meta-heuristic methods have small values of the overshoot, settling time, and rise time.

7. **El Hajjami et al., in 2019 [24];** proposed the PID controller combined with the BOA for controlling the lateral dynamics of an ADC. The simulation results showed that the suggested controller gave good results in path tracking compared to the PID based on PSO and GA.

8. **Ma'ani et al., in 2020 [25];** employed the PID-controller for manipulating the throttle/brake of the KIA Soul autonomous vehicle to follow the desired speed profiles. They discussed two methods of tuning parameters of the longitudinal control which are Cohen Coon method and the genetic algorithm. The work outcomes proved that PID based on GA provides dynamic optimizing of PID gains and follows the desired speed.

9. **AbdElmoniem et al., in 2020 [26];** proposed a new approach based on a discrete prediction model called the predictive Stanley controller. They verified their proposed approach performance through some tests on the Virtual Robot Testation Platform (V-REP) simulator with various kinds of

maneuvers and a wide range of speeds. In addition, predictive Stanley controller was compared to the basic Stanley controller. The results of the suggested controller proved the benefit and the performance of the method in terms of reducing the lateral error and ensuring yaw stability by an average of 53% and 22%, respectively.

10. **Ma'ani et al., in 2020 [27];** have published a study about the optimization of PID parameters for controlling an ADC using a pollination algorithm based on a data-driven approach. They designed a PID controller for controlling the throttle of an autonomous vehicle. Thus, they concluded the proposed design can track the speed profiles acceptably in an anon-straight path.

11. **Chen et al., in 2020 [28];** discussed a coupled longitudinal and lateral control for an ADC to trace the reference speed and path. Where the PID controller is employed for the longitudinal control and the MPC is employed for the lateral control. The results of that work demonstrated that the suggested approach achieved good tracing for both defined path and speed.

12. **Gutiérrez et al., in 2020 [29];** have published a study about design a control system for an ADC for tracking the waypoints that output from the planning layer. They have proposed a controller that includes a cubic spline interpolator, for finding a smooth path and reduces the number of waypoints. Also, they added two improvements to the control system; the first one sets the linear speed according to the curvature of the trajectory. The second one corrects the instability due to delays in the measurement of car localization and the actuation systems by adding a delay compensator.

13. **Samak et al., in 2021 [5];** have published a study about designing several controllers for an autonomous vehicle for tracking the defined path. They proposed Proximally Optimal Predictive (POP), PID, Stanley, and pure pursuit controllers for tracking the reference path. Moreover, they applied for their work by CARLA simulator and Python language. The simulation outcomes proved that the POP controller is the best for tracking for reference waypoints of the path among other controllers.

The comparison of this work with some literature is listed in Table 2.1.

Table (2.1): The comparison of this work with some literature.

Reference	Longitudinal Control	Lateral Control	Tuning method	Accuracy	
				Longitudinal	Lateral
[21]	Nan	Modified Stanley	PSO	Nan	93
[24]	Nan	PID	BOA	Nan	99.3521
[25]	PID	Nan	GA	98.09	Nan
Proposed work	PID	PTMPID	HSSABOA1	98.26	99.9628

In the next chapter, the methodology of the proposed work will be presented.

Chapter Three: Methodology

Chapter Three
Methodology

3.1 Introduction

This chapter introduces the mathematical models for the vehicle motion (longitudinal dynamic vehicle model and kinematic model). The longitudinal dynamic vehicle is divided into vehicle coordinate system, longitudinal vehicle model, powertrain model, and longitudinal tire model. On the other hand, the kinematic model includes the basics of kinematic and coordinates in addition to the kinematic bicycle model. This chapter also includes the control strategies of the longitudinal and lateral motion for the vehicle. Where the PID controller is employed for adjusting the brake/throttle position. Besides, the Stanley, MS, PID, PTMPID are employed to set the steering angle. In addition, this chapter explains some optimization methods: GA, ACO, PSO, SSA, BOA, and HSSABOA in two forms (HSSABOA1 and HSSABOA2). As well to state objective functions that are used in these optimization techniques for adjusting controllers' gains.

3.2 Mathematical Modeling of Autonomous Vehicles

This thesis focuses on the analysis of the vehicle having four wheels. The front wheels of this car have the same steering angle. But the two rear wheels have zero steering angles. The mathematical models for representing an ADC are classified in this study according to the detailed diagram in Figure (3.1):

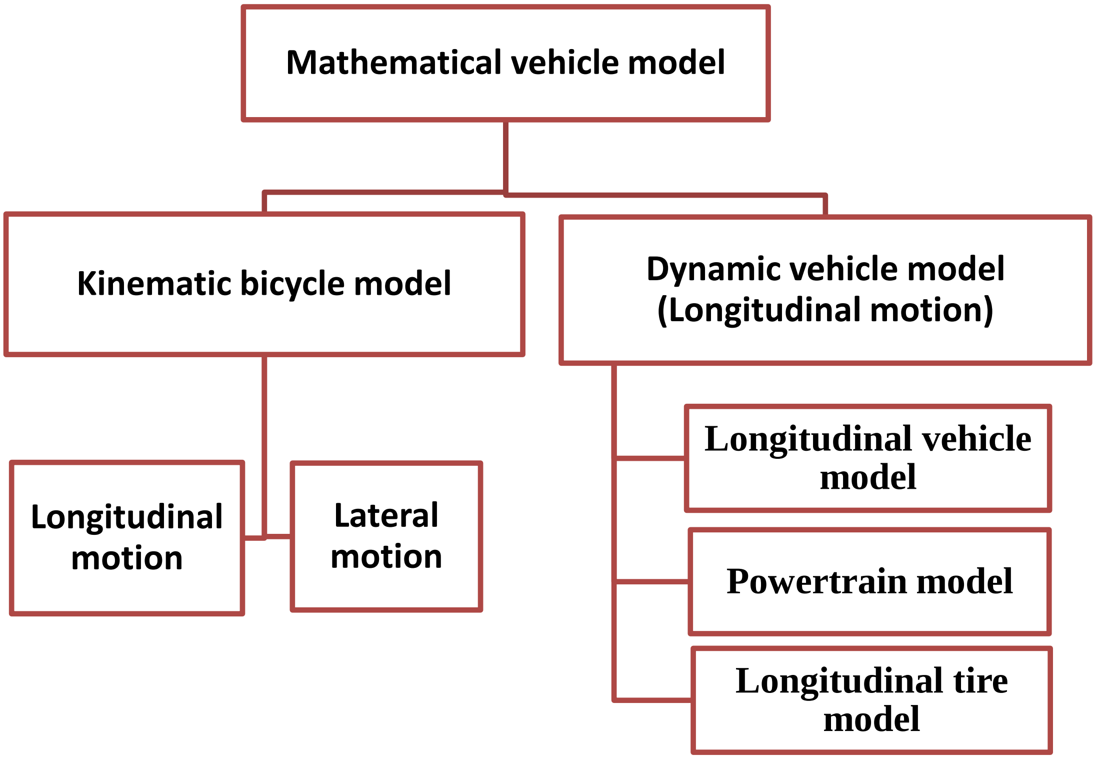

Mathematical vehicle model
Kinematic bicycle model
Dynamic vehicle model (Longitudinal motion)
Longitudinal motion
Lateral motion
Longitudinal vehicle model
Powertrain model
Longitudinal tire model

Figure (3.1): Mathematical models categories of an ADC.

3.2.1 Dynamic Vehicle Model (Longitudinal Motion)

The dynamic model defines motion by considering all of forces and moments that are acting on a vehicle, and it has higher computational complexity than the kinematic model. It is employed when the speed of the vehicle increases and the no-slip hypothesis between the ground and tire becomes unacceptable. At this point, the use of the kinematic model will be inaccurate [6][30][31]

3.2.1.1 Vehicle Coordinate System

The coordinate system of the vehicle motion is selected according to the right-hand rules as offered in Figure (3.2). By SAE convention the coordinates are [42]:

x - Longitudinal: Vehicle forward longitudinal motion,

y - Lateral: Vehicle forward lateral motion,

z - Vertical: Going down relative to the vehicle,

p - Roll: Vehicle rotation around the x-axis,

q - Pitch: Vehicle rotation around the y-axis,

r - Yaw: Vehicle rotation around the z-axis.

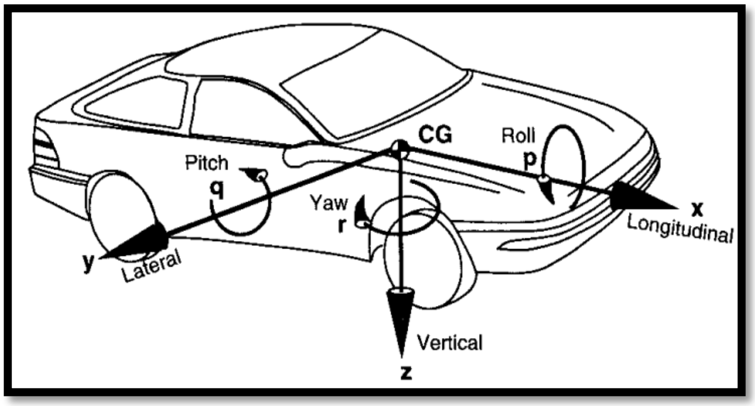


Figure (3.2): SAE Coordinate system [32].

3.2.1.2 Longitudinal Vehicle Model

The longitudinal vehicle motion diagram of an autonomous vehicle is related to x-direction as displayed in Figure (3.3) [32]:

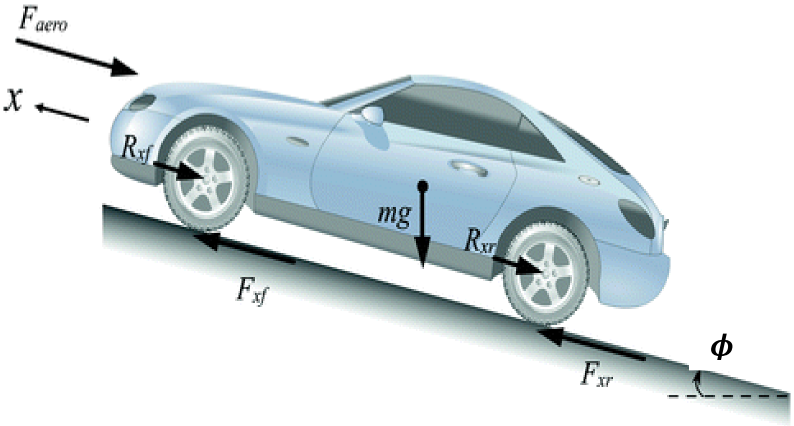


Figure (3.3): Longitudinal vehicle dynamics [33].

According to Newton's second law the Equation (1) (Eq) gives [34]:

$$m\,\ddot{x} = -F_{xf} - F_{xr} + F_{aero} + R_{xf} + R_{xr} + m * g * \sin\phi \qquad (1)$$

Where,

m: Mass of the vehicle (g),

$\ddot{x}$: Acceleration of the vehicle (m/s^2),

F_{xf}, F_{xr}: Front and rear tire forces (N),

F_{aero}: Drag force (N),

R_{xf}, R_{xr}: Front and rear rolling resistances (N),

$m * g * sin\,\phi$: X-component of the gravitational force, where g is the acceleration of gravity (m/s^2), and ϕ is the incline angle (rad),

The simplification of Equation (1) is as follows:

The front and rear tire forces are represented by longitudinal force F_x and the front and rear rolling resistances are denoted by R_x. Therefore, Equation (1) becomes:

$$m\,\ddot{x} = -F_x + F_{aero} + R_x + m * g * sin\,\phi \qquad (2)$$

The aerodynamic force F_{aero} is given by Equation (3):

$$F_{aero} = 0.5 * C_a * D * A * (v)^2 \qquad (3)$$

Where,

C_a: Drag coefficient,

D: Air density (g/cm^3),

A: Maximum vehicle cross-sectional area (m^2),

v: Vehicle speed (m/s).

The total rolling resistances R$_x$ is given by the Equation (4) [34]:

$$R_x = m * g * C_r * v \qquad (4)$$

Where, C_r is the rolling resistance coefficient. For more simplification of the Equation (2), aerodynamic force $F_{aero,}$ total rolling resistances R$_x$ and the x-component of the gravitational force $m * g * sin\,\phi$ are represented by the friction forces F_{fr} as in Equation (5) [34]:

$$a = \ddot{x} = (-F_x + F_{fr})/m \qquad (5)$$

3.2.1.3 Powertrain Model

Powertrain describes the transmission of the torque and speed from the engine to the vehicle's wheels as displayed in Figure (3.4). The torque produced by the engine is transmitted to the transmission (gearbox) across the torque converter [35]. The automatic transmission has gearsets that match engine speed to desired road speed. The power passes from the transmission to the differential by the drive shaft. The differential turns power flow 90 degrees and allows one wheel to rotate faster from the other on curves. Finally, the axle shaft transmits power from the differential to the rear wheels [32].

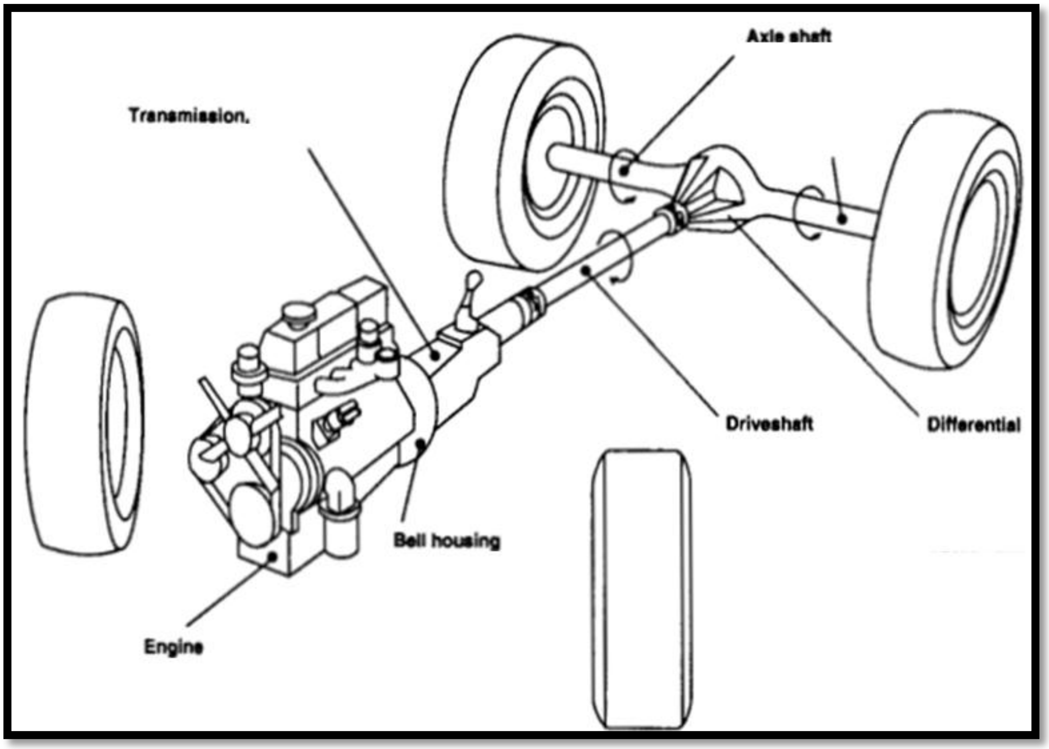


Figure (3.4): Vehicle powertrain model [32].

The Equations (6 to 8) of powertrain model can be calculated as follows [34][36]:

$$\dot{w}_e * J_e = T_e - T_{load} \quad (6)$$

$$T_{load} = GR * w_e * F_{fr} \quad (7)$$

$$v = r_{eff} * w_w \quad (8)$$

Where,

J_e : Inertia moment of engine (kg.m^2),

$\dot{w}_e$: Change in engine angular speed (rad/s^2),

T_e: Engine torque (N.m),

T_{load}: Friction torque (N.m),

GR : Gear ratio,

w_e : Engine angular speed (rad/s),

w_w:Wheel angular speed (rad/s),

r_{eff} : The radius of the wheel (m).

Figure (3.5) refers to the engine map, which represents the engine torque relative to engine rotation speed in the unit of rpm (Revolutions Per Minute) and the throttle opening percentage [34][36].

The engine rpm and engine torque are related in the form of a polynomial equation such as in Equation (9), where a_0, a_1, and a_2 are constant values [34].

$$T_{emax} = (a_0 + a_1 * \text{engine rotation speed} + a_2 * \text{engine rotation speed}^2) \quad (9)$$

The engine torque is a linear function of throttle opening, therefore the engine torque is written in Equation (10) as [36]:

$$T_e = \text{Throttle angle} * T_{emax} + (1\text{-Throttle angle}) * T_{emin} \quad (10)$$

Where T_{emin} and T_{emax} denote the minimum and the maximum of the engine torque T_e, respectively.

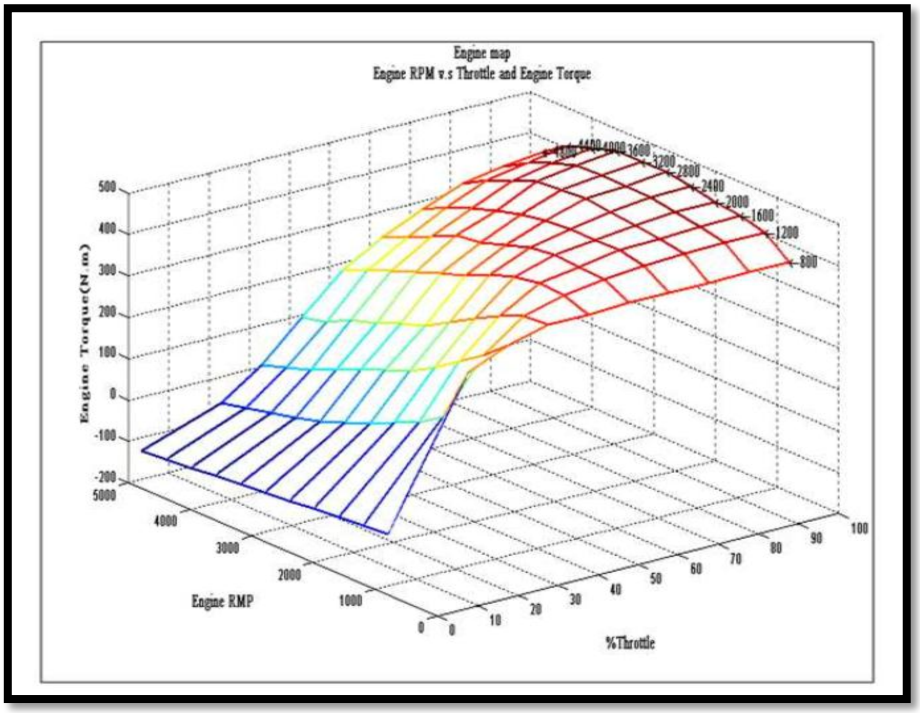


Figure (3.5): Engine map (Engine RPM vs. Engine torque and % Throttle) [35].

3.2.1.4 Longitudinal Tire Model

The main source to calculate the road forces acting on the vehicle is the tire model [35]. One of the famous models that are used extensively in vehicle dynamic modeling is the linear model as illustrated in Figure (3.6). Where it offers a combination of accurate force prediction and convenient computation. According to that model, the longitudinal force is described by Equation (11) [37]:

$$F_x = C_s * S \tag{11}$$

Where,

C_s: Longitudinal stiffness of the tire (N),

S: Longitudinal slip (percent).

The slip ratio is dissimilar in case of the vehicle is accelerating or braking, as given in Equation (12) [35]:

$$S = \begin{cases} \dfrac{r_{eff} * w_w - v}{r_{eff} * w_w} & \text{During acceleration} \\ \\ \dfrac{r_{eff} * w_w - v}{v} & \text{During braking} \end{cases} \qquad (12)$$

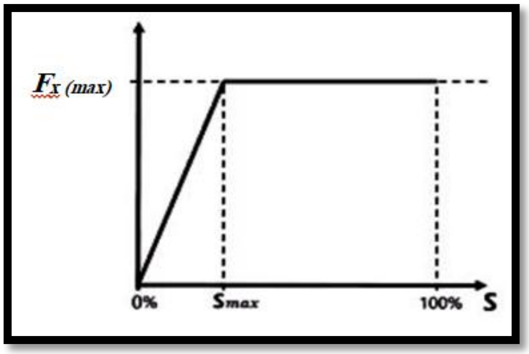


Figure (3.6): Linear tire model.

The block diagram of the longitudinal dynamic vehicle model is presented in Figure (3.7).

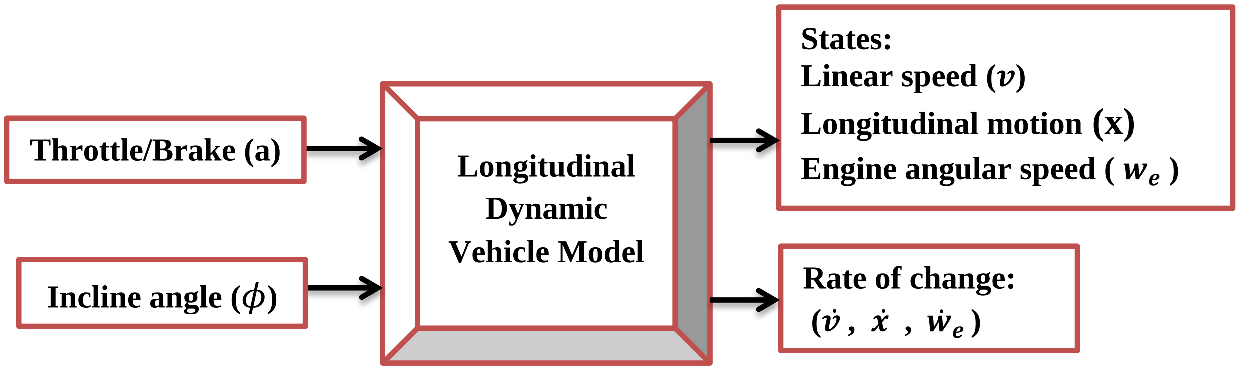


Figure (3.7): Block diagram of longitudinal dynamic vehicle model.

Where, $\dot{v}$, ẋ are the change in the longitudinal speed and longitudinal motion relative to the time, in the order. The parameters of the longitudinal dynamic vehicle employed for the simulation are specified in Table (A.1) of the Appendix.

3.2.2 Kinematic Model

The kinematic model describes the motion of a vehicle, neglecting the forces and moments acting on the vehicle. It is used for motion planning and control of maneuvers at low speeds [30].

3.2.2.1 Basics of Kinematic and Coordinates

The position of each robot in the actual world represents by its position (x, y, and z) and orientation (Pitch, Yaw, and Roll) along the three main axes of a cartesian coordinate system. The direction of these coordinate axes and rotation around them are selected according to the right-hand rules as shown in Figure (3.8) [38]. The kinematic model requires the concepts of coordinate frame transformations. The common coordinate frames of the robot are the global or inertial coordinate frame and the body or local frame. The body or local frame is related to the robot body itself. To move the system, the transformation variables from one coordinate frame to the other are required [39].

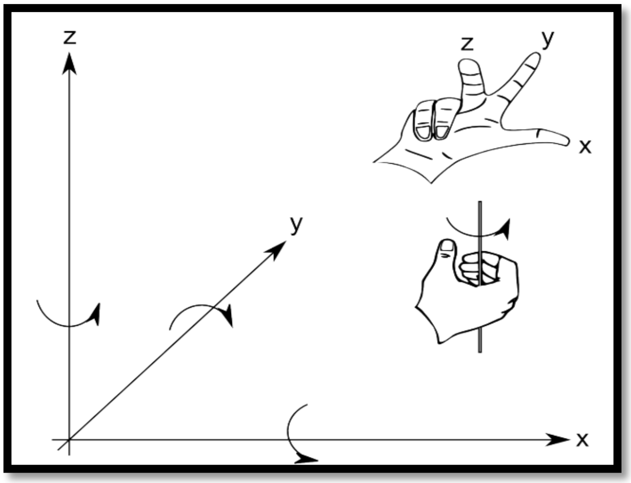


Figure (3.8): The direction of coordinate axes to the robot system [38].

3.2.2.2 Kinematic Bicycle Model

The kinematic bicycle model as illustrated in Figure (3.9) is commonly employed to represent the driverless car motion at low speed. According to a transformation from the local coordinate frame to the global reference frame, the Equations (13 to 18) of the kinematic bicycle model are represented as follows [30][40]:

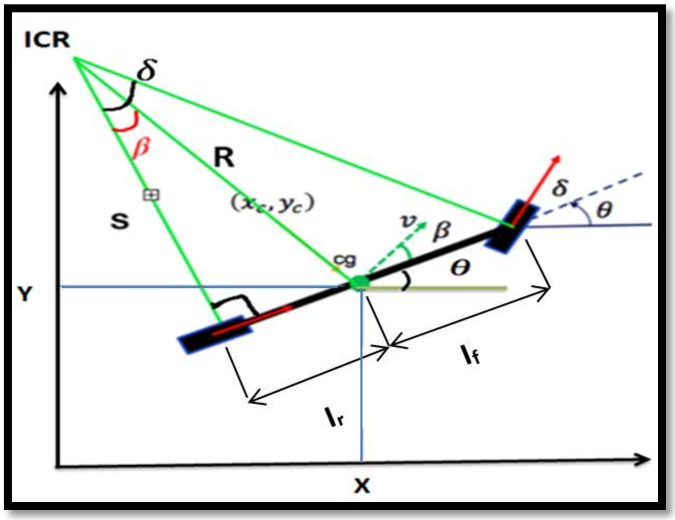


Figure (3.9): Kinematic bicycle model.

$$\dot{x}_c = v\cos(\theta + \beta) \tag{13}$$

$$\dot{y}_c = v\, sin\ (\theta + \beta) \tag{14}$$

$$\dot{\theta} = \frac{v\, cos\, \beta}{L}\ tan\, \delta \tag{15}$$

$$\dot{v}\ = a \tag{16}$$

$$\beta = tan^{-1}\ (\ \frac{I_r}{L}\ \ tan\, \delta) \tag{17}$$

$$\dot{\delta}\ = \varphi \tag{18}$$

Where,

x_c, y_c: Vehicle coordinates in the global frame (m),

Θ: Vehicle yaw angle (rad),

v: The magnitude of the speed vector from the bicycle center (rad/s),

β : Vehicle side slip angle (rad),

δ: Steering angle (rad),

a: Vehicle acceleration (rad/s),

I_f: Distance from the bicycle center to the center of the front wheel (m),

I_r : Distance from the bicycle center to the center of the rear wheel (m),

L: Wheelbase (m) and it is equal to the collection of I_f and I_r ,

φ: Steering rate (rad/s).

The state space model of the kinematic bicycle model is represented in Equation (19)

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\Theta} \\ \dot{\delta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \cos(\Theta+\beta) \\ \sin(\Theta+\beta) \\ cos\ (\beta/L)\ tan\,\delta \\ 0 \\ 1 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \varphi \qquad (19)$$

The parameters of the kinematic bicycle model employed for the simulation are expressed in Table (A.2) of the Appendix:

The block chart of the kinematic bicycle model is offered in Figure (3.10).

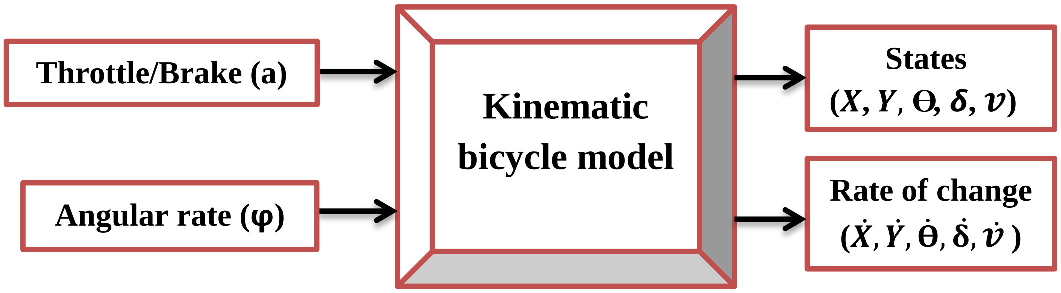

Throttle/Brake (a)
Angular rate (φ)
Kinematic bicycle model
States
$(X, Y, \Theta, \delta, v)$
Rate of change
$(\dot{X}, \dot{Y}, \dot{\Theta}, \dot{\delta}, \dot{v})$

Figure (3.10): Block chart of the kinematic bicycle model.

Where, the $(\dot{X}, \dot{Y},$ ⍰, $\dot{\delta}, \dot{v})$ are the change of these states $(X, Y,$ ⍰, $\delta, v)$ with relative to the time, respectively.

3.3 Control Strategies

The control system is an important part of an ADC to execute the reference path and desired speed generated by the motion planning processes [10]. The control of autonomous vehicles can be separated into two parts: Longitudinal and lateral control as given in Figure (3.11).

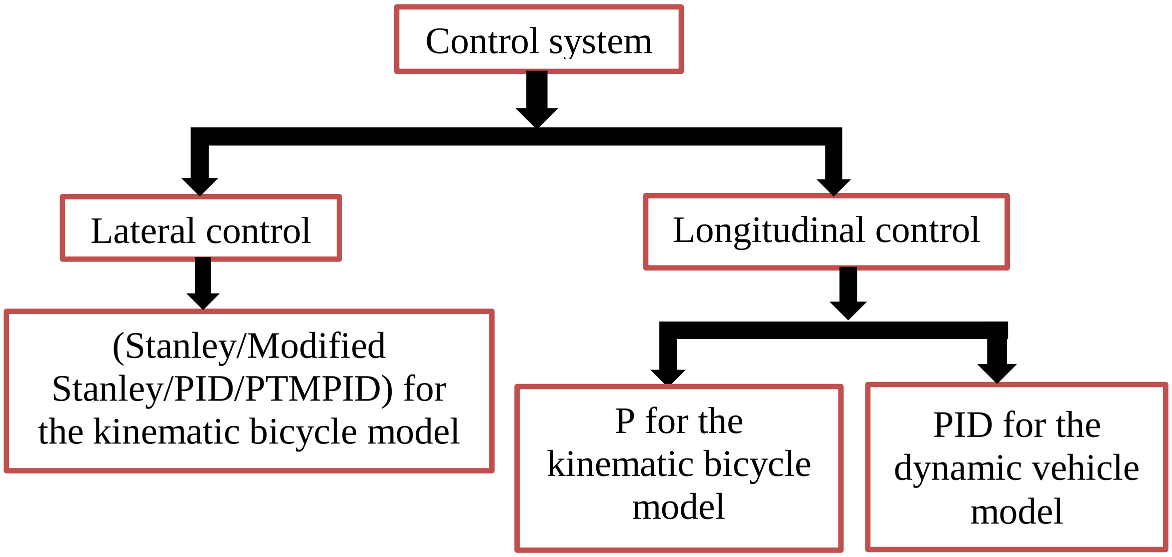

Control system
Lateral control
Longitudinal control
(Stanley/Modified Stanley/PID/PTMPID) for the kinematic bicycle model
P for the kinematic bicycle model
PID for the dynamic vehicle model

Figure (3.11): Control system diagram.

3.3.1 Longitudinal Control

The common controller that is employed in many applications is the feedback controller. It can reduce the bad effects of parameters changes, errors, and measurement disturbances between the desired and observed measurements [10].

One of the common kinds of feedback controllers that employed for the longitudinal control of an ADC is the PID controller. The PID controller contains three items; proportional, integral, and derivative. The speed profile of the longitudinal autonomous vehicle control involves throttle and brake values. For the driverless car to follow the speed profiles, a longitudinal controller is used to minimize the error between current and target values of the throttle and brake. The PID controller can be represented as in Equation (20) [35][36]:

$$u(t) = k_p * e(t) + k_d * \frac{de(t)}{dt} + k_i * \int e(t)\, dt \tag{20}$$

Where,

$\mathrm{k_p}$: The proportional gain of the PID controller,

$\mathrm{k_i}$ and $\mathrm{k_d}$: Integral and derivative gains of the PID controller, respectively,

e(t): The error between current and target measurements.

The control input is employed for regulatory the throttle and brake positions and in that test, the throttle values are limited between (-1 and 1), as the block diagram of the PID controller is presented in Figure (3.12).

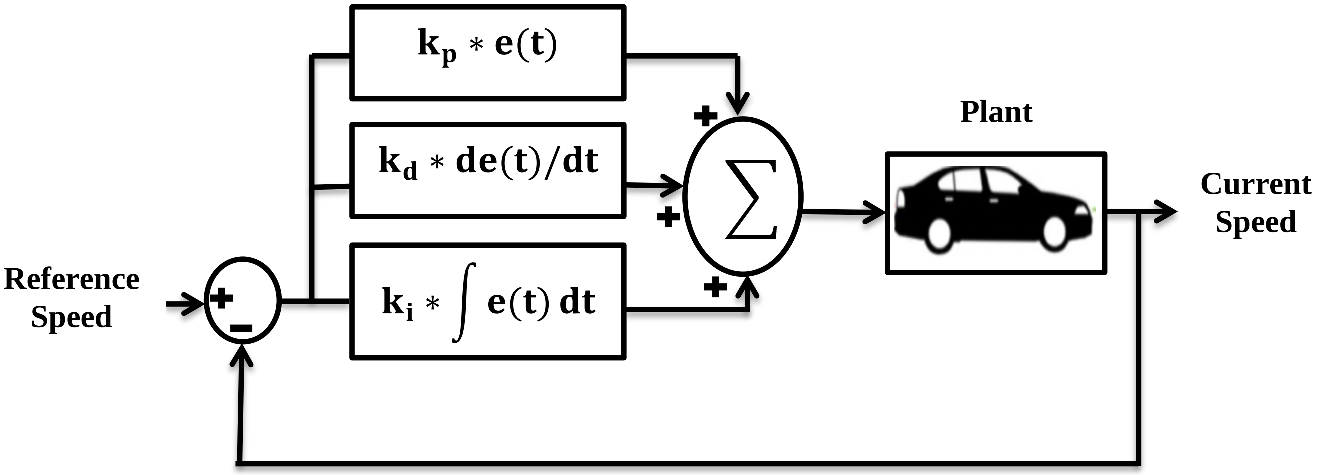


Figure (3.12): Block diagram of the PID controller.

The proportional component can correct the proportional response by multiplying the error by proportional gain $\mathrm{k_p}$, and it produces an output value that is proportional to the current error value. It reduces the steady-state error and increases the response speed. When the P gain is high enough, the system becomes unstable. In contrast, if the proportional gain is too low, the response to system disturbances may be very small [41][42].

The advantage of using the integral component is to minimize the steady-state error by multiplying the accumulated error by the integral gain k_i. On the other hand, the basic function of the derivative term is to put-out the overshoot by multiplying the slope of the error by differential gain k_d. Moreover, the differential component reduces the speed of the system, and it makes small changes on the steady-state error [41][42].

The effects of the $\mathrm{k_p}$, $\mathrm{k_d}$, and $\mathrm{k_i}$ on the system response have explained as in the Table (3.1) [42]:

Table (3.1): Effects of PID controller parameters.

Controller parameter	Closed loop system response			
	Rise time (t_r)	Overshoot (OS)	Settling time (t_s)	Steady state error (SSE)
k_p	Diminution	Increasing	Slight changing	Decreasing
k_i	Diminution	Increasing	Increasing	Remove
k_d	Slight changing	Diminution	Diminution	Slight changing

3.3.2 Lateral Control

This section discusses two kinds of controllers: geometric and feedback controllers.

3.3.2.1 Geometric controller

To set a steering angle for the kinematic bicycle model; geometric path tracking algorithms are used to form simple relations. These algorithms use lookahead distance to measure the error in front of the vehicle and their complexity range from simple circular arc calculations to much more complicated geometric methods. There are two common geometric controller algorithms are pure pursuit and Stanley controllers [10]. This thesis focuses on the Stanley controller where two types of Stanley controller are stated in this section as follow:

A) Stanley Controller

One of the common geometric control methods used to steer a vehicle is a Stanley controller, as presented in Figure (3.13). This controller computes the crosstrack error e_{cte} from the front wheel center to the reference path at point P.

Moreover, this method calculates the heading error between the orientation of the vehicle and the reference yaw angle θ_p as in the Equation (21) [10]:

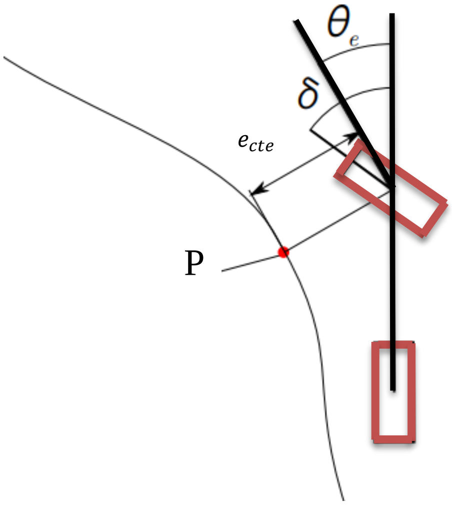


Figure (3.13): Steering geometry of Stanley method.

$$\theta_e = \theta - \theta_p \qquad (21)$$

The steering angle of the Stanley controller is described by the Equation (22) [10]:

$$\delta = \theta_e + tan^{-1}(\frac{K * e_{cte}}{v}) ; \qquad \delta \in [-\delta_{max}, \delta_{max}] \qquad (22)$$

Where, K is the gain of the Stanley controller.

B) Modified Stanley Controller (MS)

The modified Stanley controller is proposed by Amer, et al.[21]. In this controller, the steering angle of the Stanley controller is modified by adding one item and three gains as, specified by Equation (23):

$$\delta = k_1\theta_e + k_2 tan^{-1}(\frac{k_3 * e_{cte}}{1+v}) + k_4(\dot{\theta} - \dot{\theta}_{ref}); \quad \delta \in [-\delta_{max}, \delta_{max}] \quad (23)$$

Where $k_1, k_2,\ k_3$, and k_4 are the controller's gains, $\dot{\theta}$ and $\dot{\theta}_{ref}$ are the vehicle's yaw rate and path's yaw rate, respectively.

3.3.2.2 Feedback Controller

In this section, two kinds of controllers are identified:

A) PID controller

It's a kind of feedback controllers that can be used in lateral control. The crosstrack error is used as an input to the PID controller as in Equation (24) [22]:

$$\delta = k_p * e_{cte} + k_d * \frac{de_{cte}}{dt} + k_i * \int e_{cte}\, dt; \qquad \delta \in [-\delta_{max}\,, \delta_{max}] \qquad (24)$$

B) Path Tracking Based on Modified PID controller (PTMPID)

To make the PID controller efficient to trace the trajectories at different speeds without large oscillation, the PID controller has been modified in this work. So that the vehicle speed is embedded with the PID controller formula, therefore, the steering angle is altered as in Equation (25):

$$\delta = (k_p * e_{cte} + k_d * \frac{de_{cte}}{dt} + k_i * \int e_{cte}\, dt)/(1 + v); \quad \delta \in [-\delta_{max}\,, \delta_{max}] \ (25)$$

3.4 Optimizing Techniques

The algorithms that have been employed in this work to tune the gains of the longitudinal and lateral controllers are presented in the following subsections:

3.4.1 Genetic Algorithm (GA)

GA is one of the evolutionary algorithms that depend on genetics, it was proposed by Holland in 1975 [43]. In GA the individuals (chromosomes) are evolved by objective or fitness function value. Then according to the objective function, the selection is performed on the chromosomes for reproduction. On the selected chromosomes the crossover and mutation are implemented to create children or offspring which forms the population of the following generation. This process is looped until reaches the maximum iteration [44][45]. The main problem of the GA is that the best solution is very hard to obtain because it easily falls into premature convergence [46]. The parameters of the GA algorithm are stated in Table (3.2).

Table (3.2): The parameters of GA algorithm.

Description	Value
Crossover type	Multipoint crossover
Crossover rate	1
Mutation rate	0.01
Selection strategy	Roulette wheel selection

3.4.2 Ant Colony Optimization (ACO)

ACO algorithm is introduced by Marco Dorigo and colleagues in the early 1990s for solving optimization problems. This algorithm is inspired by the behavior of ants for collecting food. It is a probabilistic algorithm used for finding the shortest way in graphs. When ants search for food, they take several paths and put down pheromone to the surface. So the shortest path has the greatest density of pheromones. In this algorithm three different main stages are followed:

initialization, constructing ant solution, and updating pheromone [47-49]. The mathematic expression of the pheromone τ_{ij} is updated by the Equations (26) and (27):

$$\tau_{ij} = (1-\rho) * \tau_{ij} + \sum_{k=1}^{n} \Delta\tau_{ij}^{k} \qquad (26)$$

$$\Delta\tau_{ii}^{k} = \begin{cases} \frac{Q}{L_k}\ if\ ant\ k\ employed\ edge\ (i,j) in\ its\ tour, \\ 0\ \ otherwise \end{cases} \qquad (27)$$

Where:

i, j: The edge connection nodes,

ρ: The evaporation rate for pheromone quantity,

$\Delta\tau_{ij}^{k}$: The pheromone quantity,

n: The number of ants,

Q: Constant,

L_k: The length of the round generated by ant k,

The probability ρ to move from node i to node j is calculated by the Equations (28) and (29):

$$\rho_{ii}^{k} = \begin{cases} \square_{ij} * \frac{\tau_{ij}^{\alpha} * \square_{ij}^{B}}{\sum_{c_{ij} \in N(S^P)} \tau_{ij}^{\alpha} * \square_{ij}^{B}}\ \ if\ c_{ij} \in N(S^P), \\ 0\ \ otherwise \end{cases} \qquad (28)$$

$$\square_{ij} = \frac{1}{\mathrm{d}_{ij}} \qquad (29)$$

S^P: The partial solution,

$N(S^P)$: The set of nodes,

$\square_{ij}$: The heuristic information,

α and B: The pheromone and the heuristic exponent in the order,

d_{ij}: A distance from nodes i to the joined node j.

The disadvantage of the ACO algorithm is be able to fall easily into the trap of local optimum [46]. The parameters of ACO algorithm are specified in Table (3.3).

Table (3.3): The parameters of ACO algorithm.

Description	Value
Pheromone exponent (α)	0.6
Heuristic exponent (B)	0.2
Pheromone evaporation factor	0.7
Number of nodes	10000

3.4.3 Particle Swarm Optimization (PSO)

PSO technique is the optimization method which has been proposed in 1995 by Kennedy and Eberhart. It is described according to the social behavior of fishes or birds. In this technique, initially, a random group of particles is created. Each particle expressed a solution and the individuals in each group of particles are assigned a speed and position. The particles are used to generate many solutions for the fitness function at every iteration. For each generation, the objective function is evaluated to find the particle best (pbest) and the global best (gbest). The pbest provides the best solution of each particle. On the other hand, the gbest represents the best solution of all generations. The pbest and gbest are updated at each iteration and they are used to calculate the new speed and position according to Equations (30 to 32) [45][50][51][52]:

$$v_i^{t+1} = \omega\, v_i^t + z_1 r_1 (p_i^t - x_i^t) + z_2 r_2 (g_i^t - x_i^t) \tag{30}$$

$$\omega = \omega_{max} - t * \frac{(\omega_{max} - \omega_{min})}{N} \tag{31}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \qquad i = 1,2,\dots,n \tag{32}$$

Here,

t: Iterations number,

n: Number of the particles,

z_1 and z_2: Cognitive and social learning factors, respectively,

ω: Inertia weight,

r_1 and r_2: Random values inside the range (0,1),

p_i: The best position of the particles,

g_i: The best particle between members of the group.

N: Maximum iteration,

ω_{min} and ω_{max}: The minimum and maximum values of inertia weight.

These steps are repeated until the maximum iteration is reached. The major drawback of the PSO algorithm is the likelihood of falling all particles at a local minimum in the search space. Thus, they cannot find the exit way from the trap with their own [45, 51]. The limitations of the PSO algorithm are given in Table (3.4).

Table (3.4): The limitations of PSO algorithm.

Description	Value
The minimum and maximum of inertia weight	0.2 and 0.9
Cognitive learning factor	2
Social learning factor	2
Maximum speed	6

3.4.4 Salp Swarm Algorithm (SSA)

The SSA is a new method of the meta-heuristic algorithms which was suggested by Mirjalili et al. in 2017 [53]. The idea inspiration of SSA is back to behavior the swarm of salps in seas [54][55]. The swarm of salps is composed of the leader and the followers. The leader searches for food while the followers track the leader sequentially [56][57].

The SSA has unique features that are not found by the other methods of optimization, such as Gray Wolf Optimization (GWO), Gravitational Search Algorithm (GSA), and PSO technique. It is flexible, easy to understand, simple construction, and it avoids the problem of being stuck in local optimum [14][54].

The population of salps can be defined by an (M × N) dimensional matrix and the mathematical expressions of the salp swarm algorithm are represented by the Equations (33 to 35) [53][54]:

$$X_j^1 = F_j + c_1 * ((ub_j - lb_j) * c_2 + lb_j) \qquad c_3 \geq 0.5 \qquad (33)$$

$$X_j^1 = F_j - c_1 * ((ub_j - lb_j) * c_2 + lb_j) \qquad c_3 < 0.5 \qquad (34)$$

$$c_1 = 2\exp(-4l/N)^2 \qquad (35)$$

Here,

X_j^1: The position leader in the j^{th} dimension,

F_j : The position's food source in the j^{th}dimension,

ub_j and lb_j: Upper and lower bound of the j^{th} dimension, respectively,

c_2 and c_3: The random numbers between 0 to 1 and it indicates the step size to select the next position in the j^{th} dimension should be towards positive or negative infinity,

c_1: The main coefficient gives a balance between the exploration and exploitation,

l: Current iteration.

The position of the followers should be updated according to Equation (36) [53]:

$$X_j^i = \frac{1}{2}\left(X_j^i + X_j^{i-1}\right) \qquad i \geq 2 \qquad (36)$$

This technique is not requiring any limitations like other methods.

3.4.5 Butterfly Optimization Algorithm (BOA)

The BOA is a recent meta-heuristic technique which was invented by Singh and Arora in 2018 [58]. It is inspired by butterflies' natural behavior in the search for food. This technique is dependent on two matters, which are the fragrance function formulation and the difference in fragrance intensity. The formulation of the fragrance function can be represented according to Equation (37) [58][59]:

$$f = cI^{\tau} \qquad (37)$$

Where:

f: The perceived magnitude for the fragrance,

c: The sensory modality,

I: The stimulus intensity,

τ: The power exponent attached to modality,

The sensory modality c can be updated as in Equation (38) [60]:

$$c_{t+1} = c_t + \frac{0.025}{c_t * N} \qquad (38)$$

When the butterfly senses the scent emanating from another butterfly, it will move towards it, and it is identified as the global search. In contrast, when the butterfly is unable to sense the scent emanating from any other butterfly, it will transfer randomly and this is called the local search. The global and local searches are formulated according to Equations (39) and (40), respectively [59]:

$$X_i^{t+1} = X_i^t + (r^2 * X^* - X_i^t) * f_i \tag{39}$$

$$X_i^{t+1} = X_i^t + \left(r^2 * X_j^t - X_k^t\right) * f_i \tag{40}$$

Where:

X^*: The best solution found,

r: A random number between the interval 0 and 1,

The mode to change between global and local search is named a switch probability and indicated the symbol P. The bounds of the BOA that were used in that work are stated in Table (3.5).

Table (3.5): The parameters of BOA algorithm.

Description	Value
Switch probability (P)	0.2
Power exponent (τ)	0.1
Sensory modality (c)	0.02

3.4.6 Hybrid Salp Swarm Algorithm and Butterfly Optimization Algorithm (HSSABOA)

Hybridization between two optimization algorithms provides a better solution to several problems of optimization and it leads to faster convergence. In this thesis, SSA is merged with the BOA in two forms to improve the performance of SSA. At each iteration, the comparison between fitness values of both methods is evaluated and the best solution was considered. Then half number of agents of the method that gave the best solution from the other way are copied to the half number of agents of another method before its updating. Two forms of that hybridization have found in this work as follows:

3.4.6.1 HSSABOA1

The basic idea in that form of hybridization is that, in each generation, if the fitness value of the BOA is better than SSA, half the number of agents of the BOA is copied to the half number of agents of SSA before its updating. In that form, we avoid the replacing of the agent leader as it is updated through the best solutions.

3.4.6.2 HSSABOA2

That form of hybridization includes reversing the suggested idea in HSSABOA1. Where, in case been the fitness value of the SSA better than BOA, half the number of agents of the SSA is copied to a half number of agents of BOA before it updated.

HSSABOA1/HSSABOA2 algorithms are used the same parameters stated in Table (3.5).

The common parameters for all algorithms are defined in Table (3.6). As well, the bounds of the upper, lower, and the length of each agent for the optimization techniques of the longitudinal and lateral controllers are signified in Tables (3.7) and (3.8), respectively.

Table (3.6): The common parameters in all algorithms.

Description	Value
Number of agents (n)	20
Maximum iteration (N)	300

Table (3.7): The parameters of longitudinal control.

Controller type	Lower bounds (lb)	Upper bounds (ub)	Length of each agent (dim)
PID	0.000001, 0.000001, 0.000001	10, 10, 10	3
P	0.1	30	1

Table (3.8): The parameters of lateral control.

Controller type	Lower bounds (lb)	Upper bounds (ub)	Length of each agent (dim)
Stanley	0.1	20	1
PID	0.001, 0.001, 0.001	20, 20, 20	3
PTMPID	0.001, 0.001, 0.001	20, 20, 20	3
Modified Stanley	0.0000001, 0.0000001, 0.0000001, 0.0000001	10, 10, 10, 10	4

3.4.7 Objective Function (OF)

The solution to most of the optimization problems is dependent on the representation of the OF.

The most widely used error integrating fitness functions in literature are the IAE, Integral Square Error (ISE), Integral Time Absolute Error (ITAE), Integral Time Square Error (ITSE) or Mean Square Error (MSE), and RMSE. These objective functions can be expressed as in Equations (41 to 45) [21][22][61][62]:

$$IAE = 1/T \sum |e(t)| \tag{41}$$

$$ISE = 1/T \sum e^2(t) \tag{42}$$

$$ITAE = 1/T \sum |e(t)| \tag{43}$$

$$MSE = 1/T \sum e^2(t) \tag{44}$$

$$RMSE = \sqrt{1/T \sum (e_{cte}(t))^2} \tag{45}$$

Where T is the maximum number of the time steps.

The certain objective functions in the proposed algorithms are the IAE and RMSE. Where the IAE is employed for minimizing the speed error; while the RMSE is used to reduce the lateral error.

The IAE is used to minimize the speed error in this study instead of ITAE, ISE, and MSE as a reason, it has less overshoot and minimum value of the Steady State Error (SSE) as clarified in Tables (A.3), (A.4) and Figure (A.1) of the Appendix. In addition, the RMSE is also a good choice for employing for performance evaluation of path tracking.

The flowcharts of BOA and SSA for optimization controllers' gains are offered in Figures (3.14) and (3.15), respectively. On the other hand, the hybridization approach of SSA and BOA is expressed by the two flowcharts as presented in Figures (3.16) and (3.17).

The block diagrams of algorithms for optimizing the controller's parameters of the (longitudinal dynamic vehicle and kinematic bicycle) models are also illustrated in Figures (3.18) and (3.19), respectively.

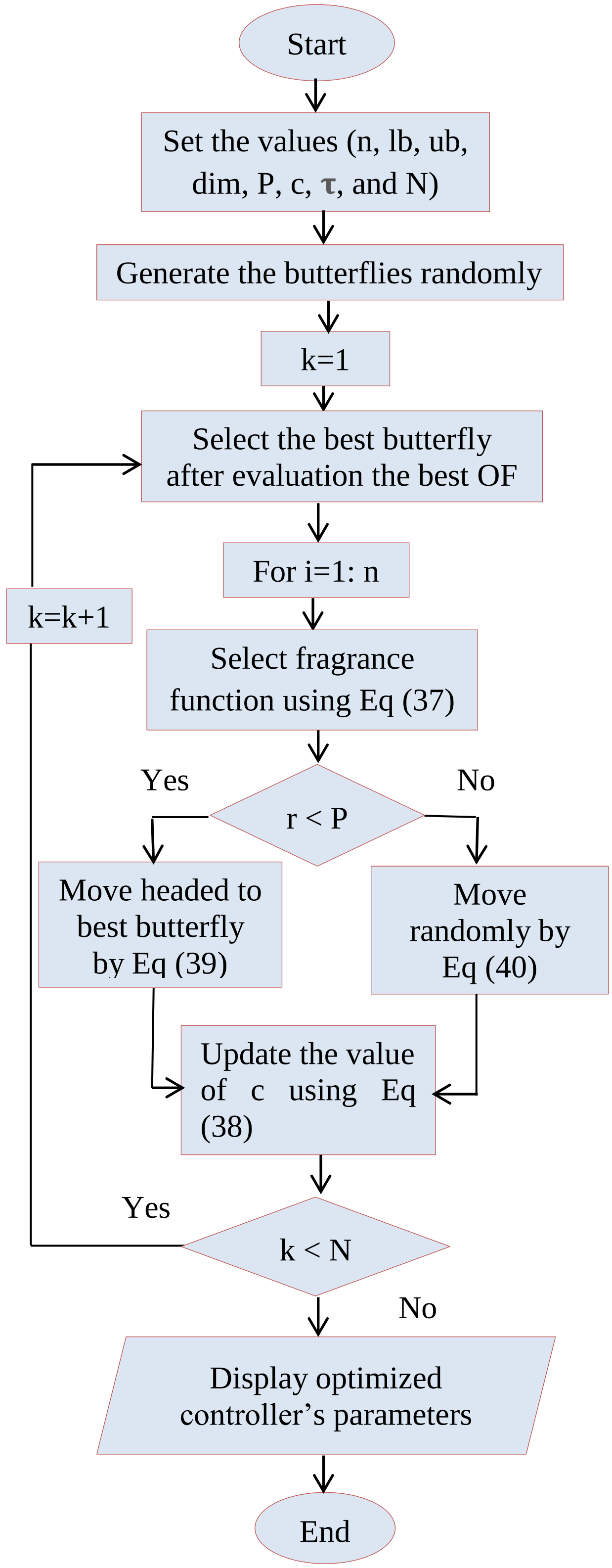

Start
Set the values (n, lb, ub, dim, P, c, τ, and N)
Generate the butterflies randomly
k=1
Select the best butterfly after evaluation the best OF
For i=1: n
k=k+1
Select fragrance function using Eq (37)
Yes
No
r < P
Move headed to best butterfly by Eq (39)
Move randomly by Eq (40)
Update the value of c using Eq (38)
Yes
k < N
No
Display optimized controller's parameters
End

Figure (3.14): Controller based on BOA flowchart.

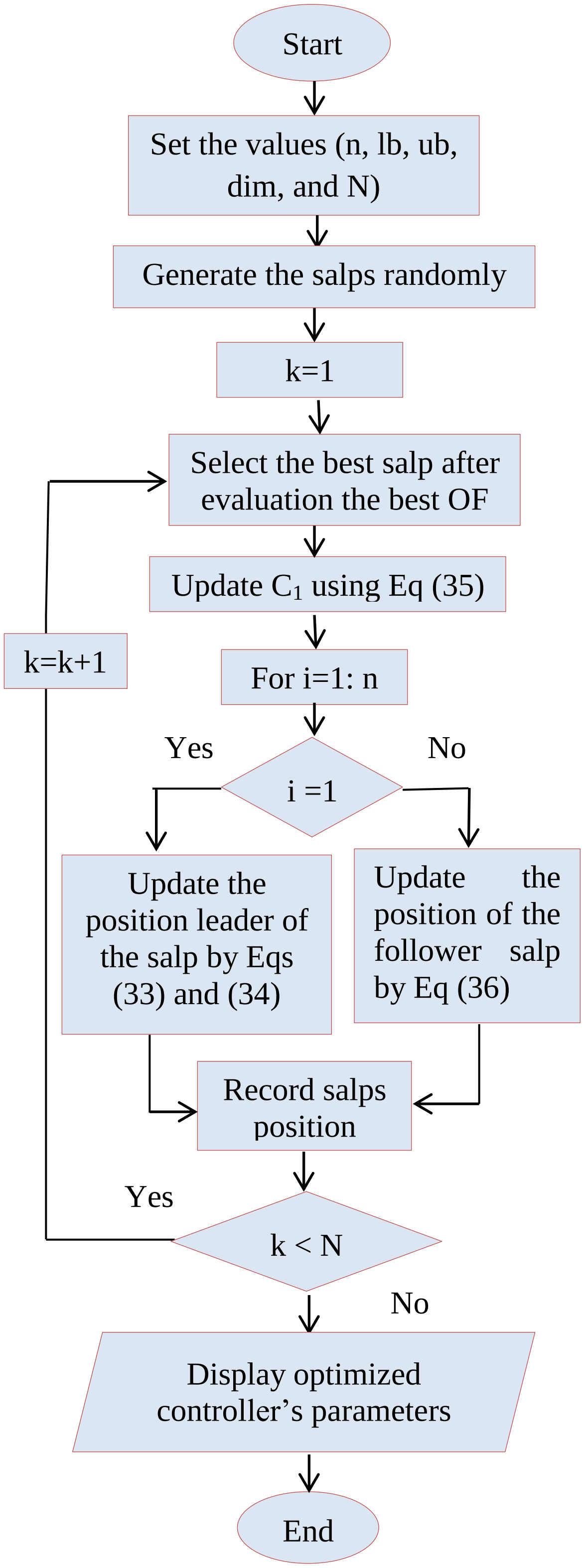

Start
Set the values (n, lb, ub, dim, and N)
Generate the salps randomly
k=1
Select the best salp after evaluation the best OF
Update C1 using Eq (35)
k=k+1
For i=1: n
Yes
No
i =1
Update the position leader of the salp by Eqs (33) and (34)
Update the position of the follower salp by Eq (36)
Record salps position
Yes
k < N
No
Display optimized controller's parameters
End

Figure (3.15): Controller based on SSA flowchart.

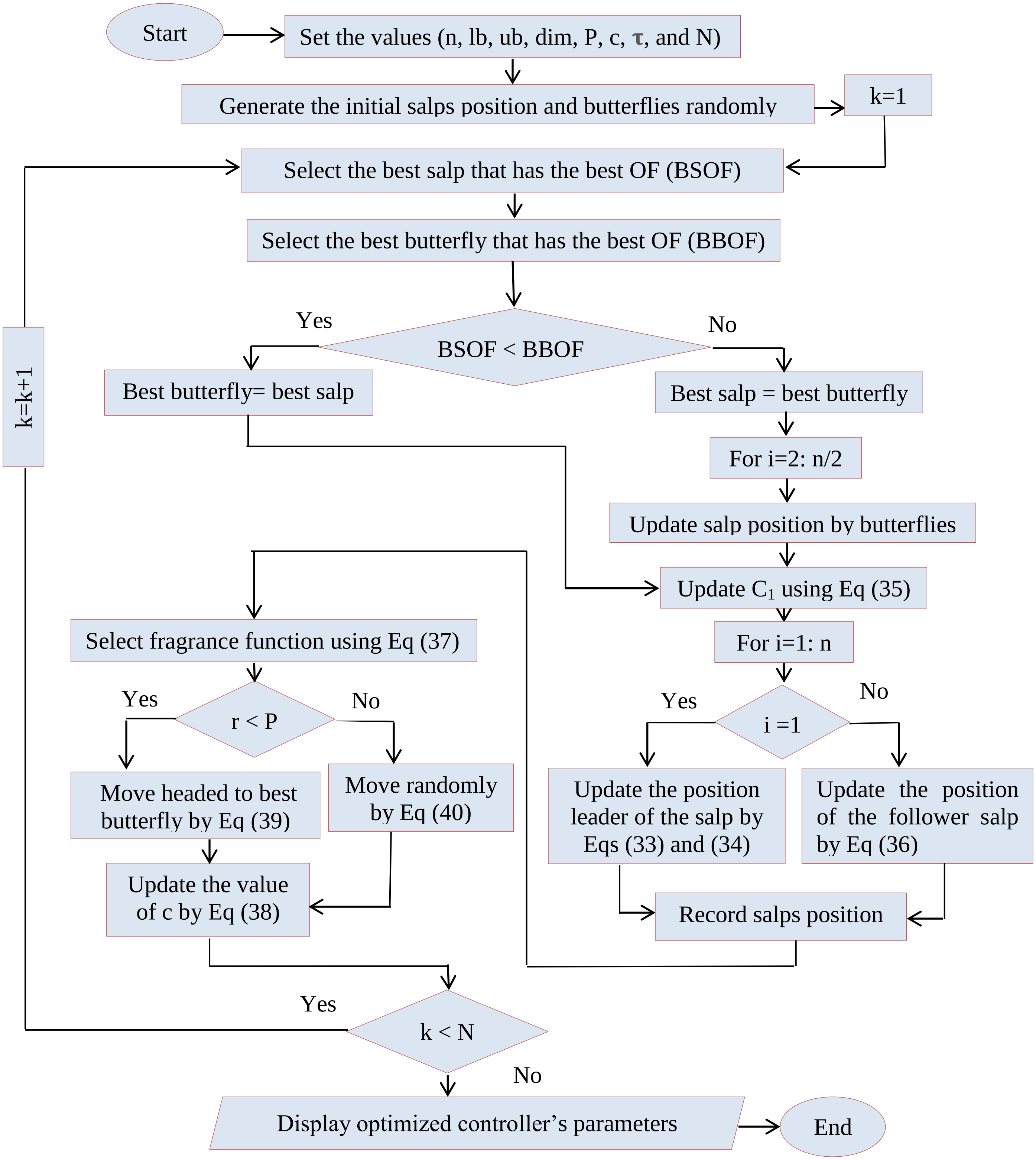

Start
Set the values (n, lb, ub, dim, P, c, τ, and N)
Generate the initial salps position and butterflies randomly
k=1
Select the best salp that has the best OF (BSOF)
Select the best butterfly that has the best OF (BBOF)
Yes
BSOF < BBOF
No
Best butterfly= best salp
Best salp = best butterfly
For i=2: n/2
Update salp position by butterflies
Update C1 using Eq (35)
For i=1: n
Yes
i =1
No
Update the position leader of the salp by Eqs (33) and (34)
Update the position of the follower salp by Eq (36)
Record salps position
Select fragrance function using Eq (37)
Yes
r < P
No
Move headed to best butterfly by Eq (39)
Move randomly by Eq (40)
Update the value of c by Eq (38)
k=k+1
Yes
k < N
No
Display optimized controller's parameters
End

Figure (3.16): Controller based on HSSABOA1 flowchart.

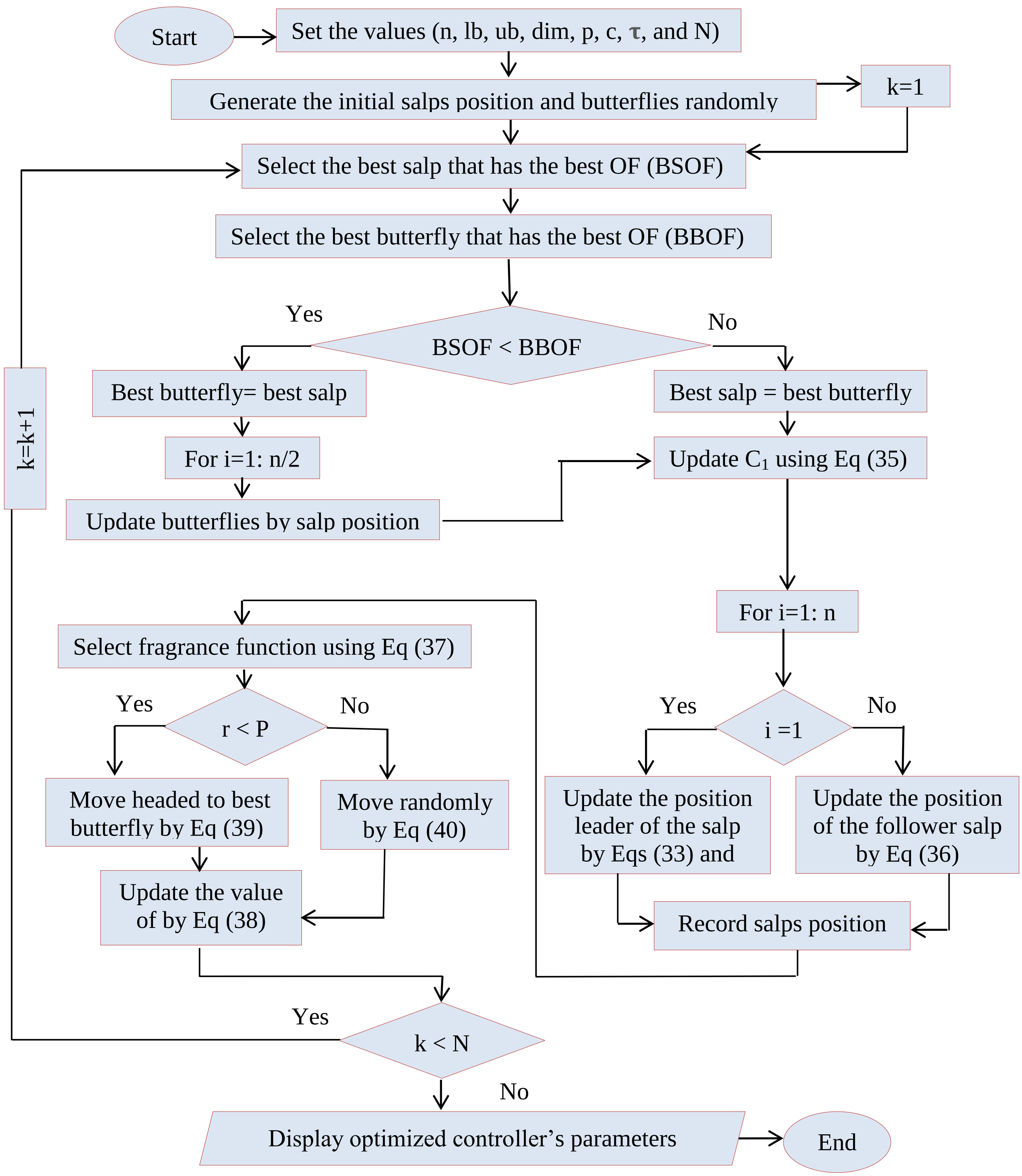

Start
Set the values (n, lb, ub, dim, p, c, τ, and N)
Generate the initial salps position and butterflies randomly
k=1
Select the best salp that has the best OF (BSOF)
Select the best butterfly that has the best OF (BBOF)
Yes
BSOF < BBOF
No
Best butterfly= best salp
Best salp = best butterfly
For i=1: n/2
Update C1 using Eq (35)
Update butterflies by salp position
k=k+1
For i=1: n
Select fragrance function using Eq (37)
Yes
r < P
No
Yes
i =1
No
Move headed to best butterfly by Eq (39)
Move randomly by Eq (40)
Update the position leader of the salp by Eqs (33) and
Update the position of the follower salp by Eq (36)
Update the value of by Eq (38)
Record salps position
Yes
k < N
No
Display optimized controller's parameters
End

Figure (3.17): Controller based on HSSABOA2 flowchart.

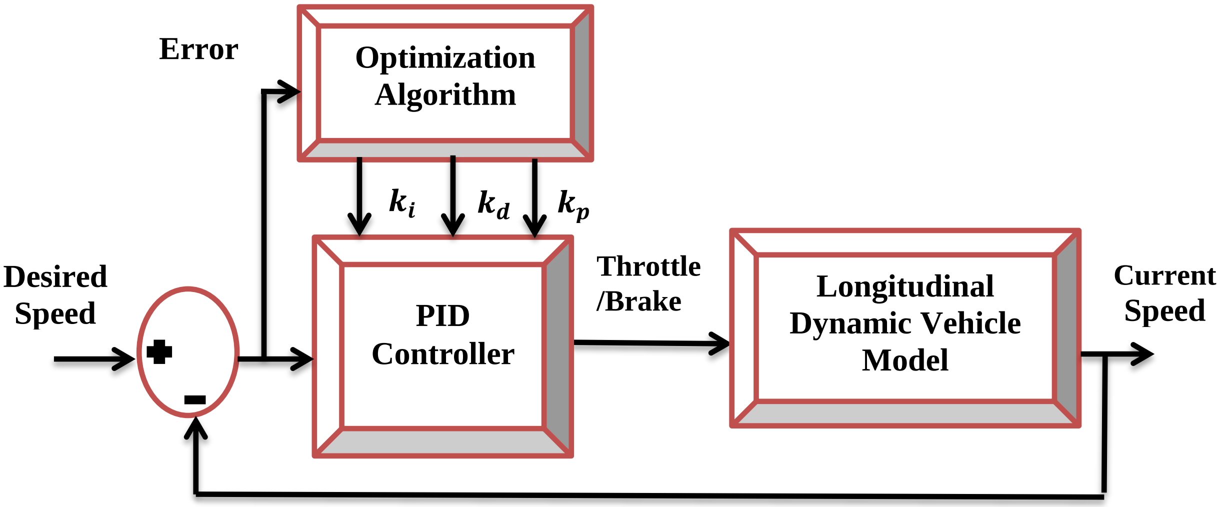


Figure (3.18): Block diagram for optimizing PID parameters.

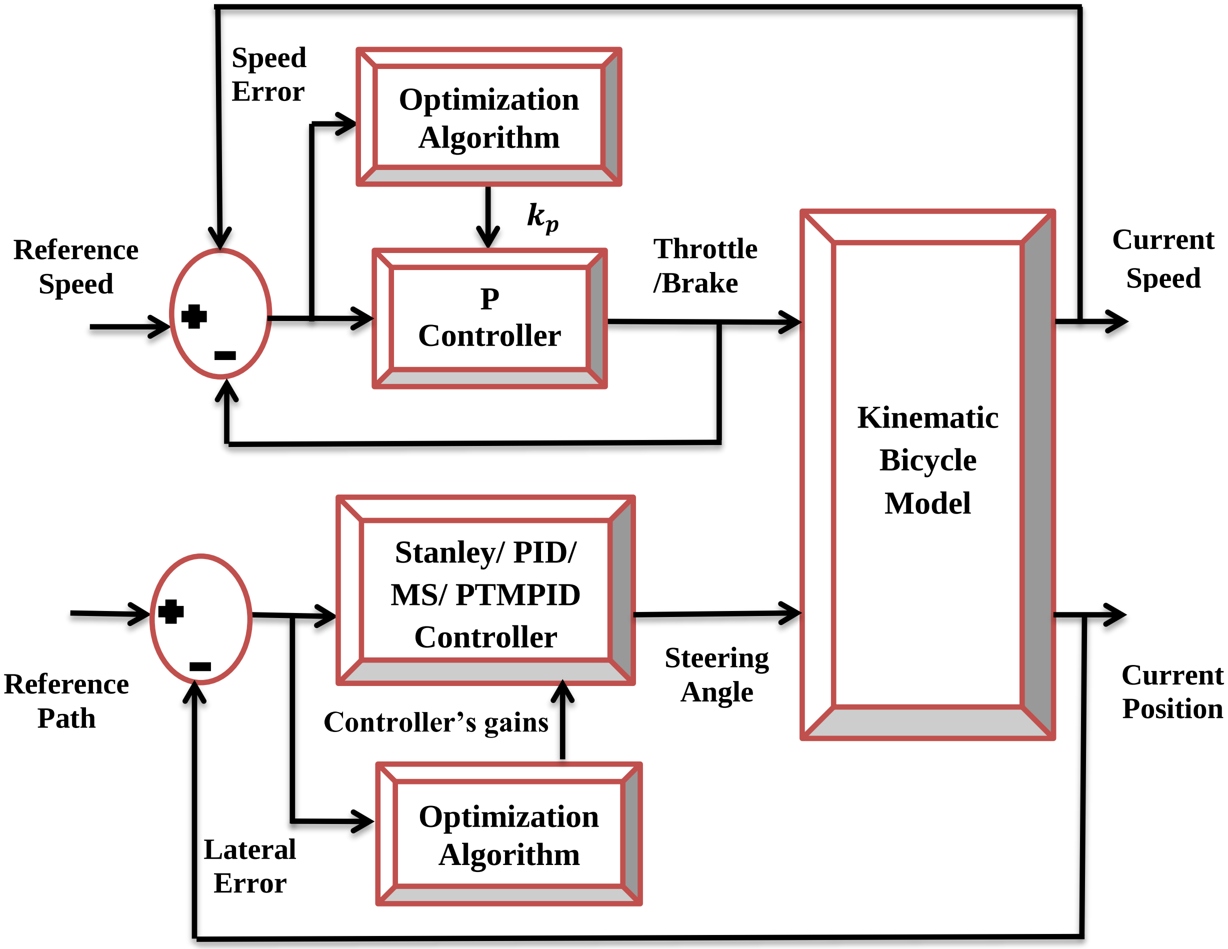


Figure (3.19): Bicycle model diagram to optimize its controller's gains.

In the following chapter, the results and discussions of the tests related to the modeling, control strategies and optimization techniques that have been introduced in this chapter will be presented.

Chapter Four: Results and Discussion

Chapter Four
Results and Discussion

4.1 Introduction

In this chapter, the outcomes of tests that were performed are presented and discussed to evaluate the performance effectiveness of the controllers and optimization algorithms mentioned in chapter three. These simulation methodologies are executed in a computer with the following specifications: (Processor AMD PRO A10, RAM 8 GB, Storage 500 GB, Freq. 1.8 GHz), Operating System (Windows 10, 64 bit), and Programming Language (Python 3.6).

4.2 Longitudinal Dynamic Vehicle Model

4.2.1 Speed Generation by Longitudinal Dynamic Vehicle Model

Initially, the parameters effect of the longitudinal dynamic vehicle model, on the speed generation have been tested several times before performing the techniques of control and optimization.

In the first test, constant values of throttle position (0.2, 0.5, 0.7, and 0.9), have been taken one by one individually. The simulation results are shown in Figure (4.1).

In the second test, various values for throttle are set at the same time. Then, the simulation output of that test is illustrated in Figure (4.2).

The results in Figure (4.1) indicate that when the throttle is set to a constant value then, the speed approaches a certain constant value according to the throttle value. In another word, there is a direct relationship between the speed and throttle values. But, the outcome in Figure (4.2) explains that, when the throttle input values changed, the speed curve has changed according to these values.

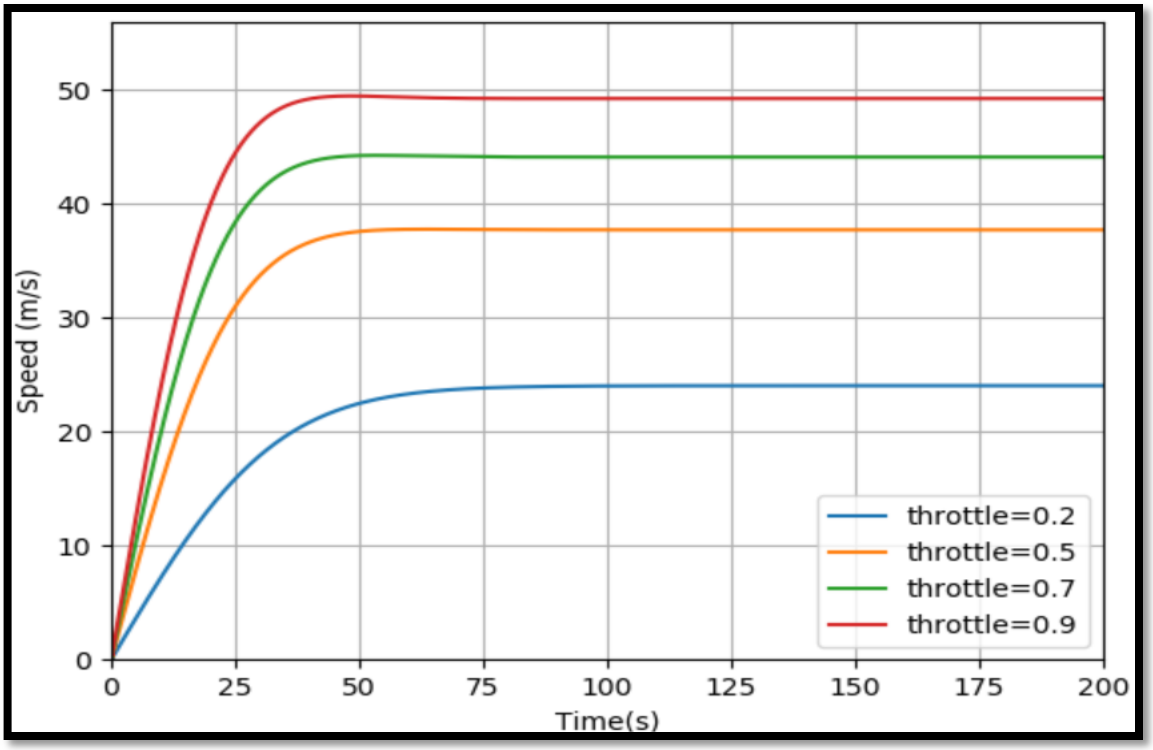


Figure (4.1): Speed vs time with constant values of throttle.

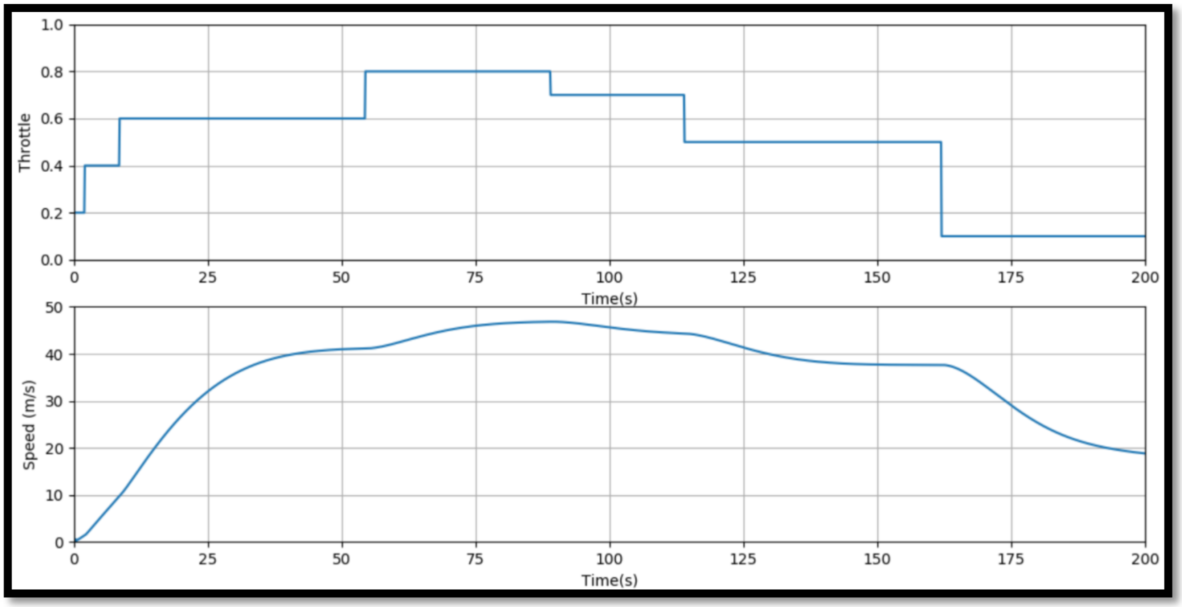


Figure (4.2): Speed vs time with variable values of throttle.

4.2.2 Longitudinal Control of Dynamic Vehicle Model

The tests in this section are divided into three tests: In the first test, the longitudinal control system of the dynamic vehicle was provided with a constant reference speed of 30 m/s. Then, the gains of the PID controller were optimized using: GA, ACO, PSO, BOA, SSA, HSSABOA1, and HSSABOA2 algorithms to solve the issue of desired speed tracing. The convergence curve of the objective functions for these algorithms is presented graphically in Figure (4.3). In that Figure, the HSSABOA1, and HSSABOA2 algorithms were converged faster than other optimization algorithms, which indicates the efficiency of the proposed algorithms.

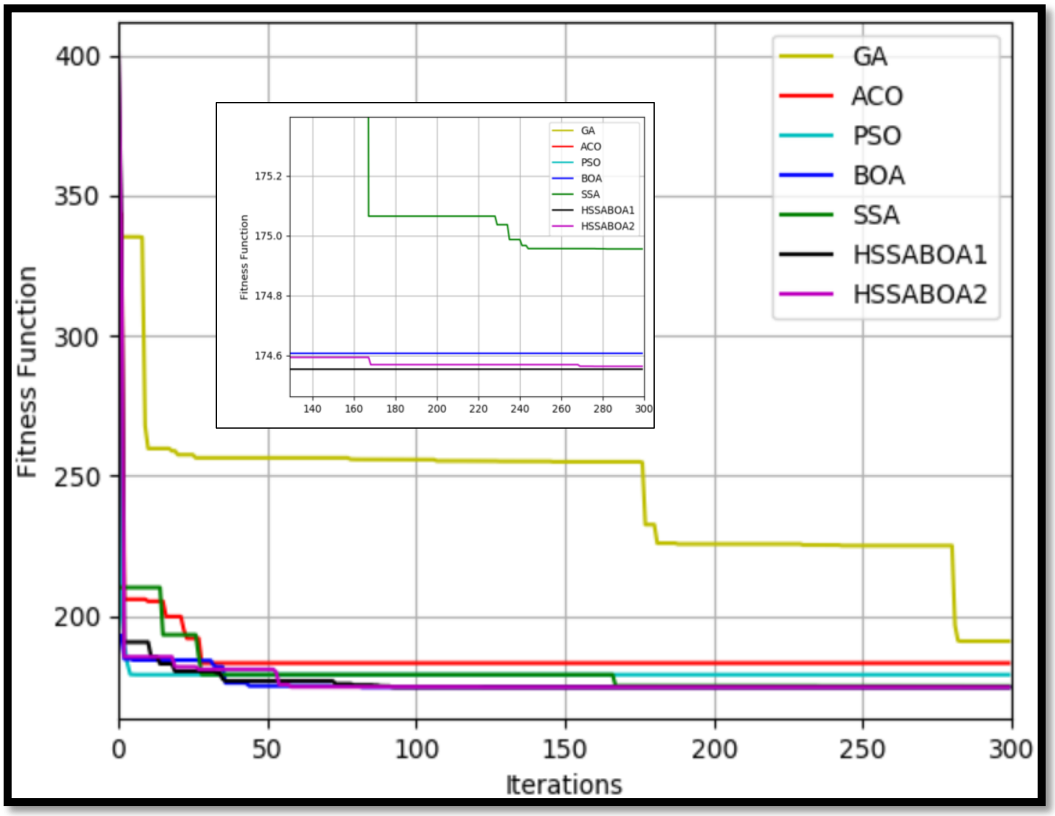


Figure (4.3): Convergence curve for PID based on optimization algorithms.

The results of Table (4.1) contain the best controllers' parameters and minimum fitness function values provided by the optimization algorithms. According to the outcomes of that table, the PID-HSSABOA1, and PID-HSSABOA2 included the best fitness values compared to the PID based on other algorithms.

Table (4.1): Best values of PID gains at the best IAE value.

Type of Optimized PID Controller	k_p	k_i	k_d	IAE (m/s)
PID-GA	7.37609	0.00826	9.65858	190.99628
PID-ACO	4.97145	0.00009	6.23189	183.78443
PID-PSO	5.59039	0.000001	1	179.04310
PID-BOA	1.73070	0.00179	3.26505	174.60714
PID-SSA	5.60605	0.00181	9.90603	174.95530
PID-HSSABOA1	3.69088	0.00179	6.99794	174.55376
PID-HSSABOA2	2.24379	0.00179	4.21653	174.56332

On the other hand, Figure (4.4) proved that the PID-BOA, PID-HSSABOA1, and PID-HSSABOA2 have provided a faster and smoother response than the PID-GA, PID-ACO, PID-PSO, and PID-SSA. Table (4.2) includes the specifications of the optimized PID response represented in standards of steady-state error, and overshoot (OS). The results in the table indicated that PID-HSSABOA1 has less SSE compared to other controllers; further to maintain a smaller value of the overshoot.

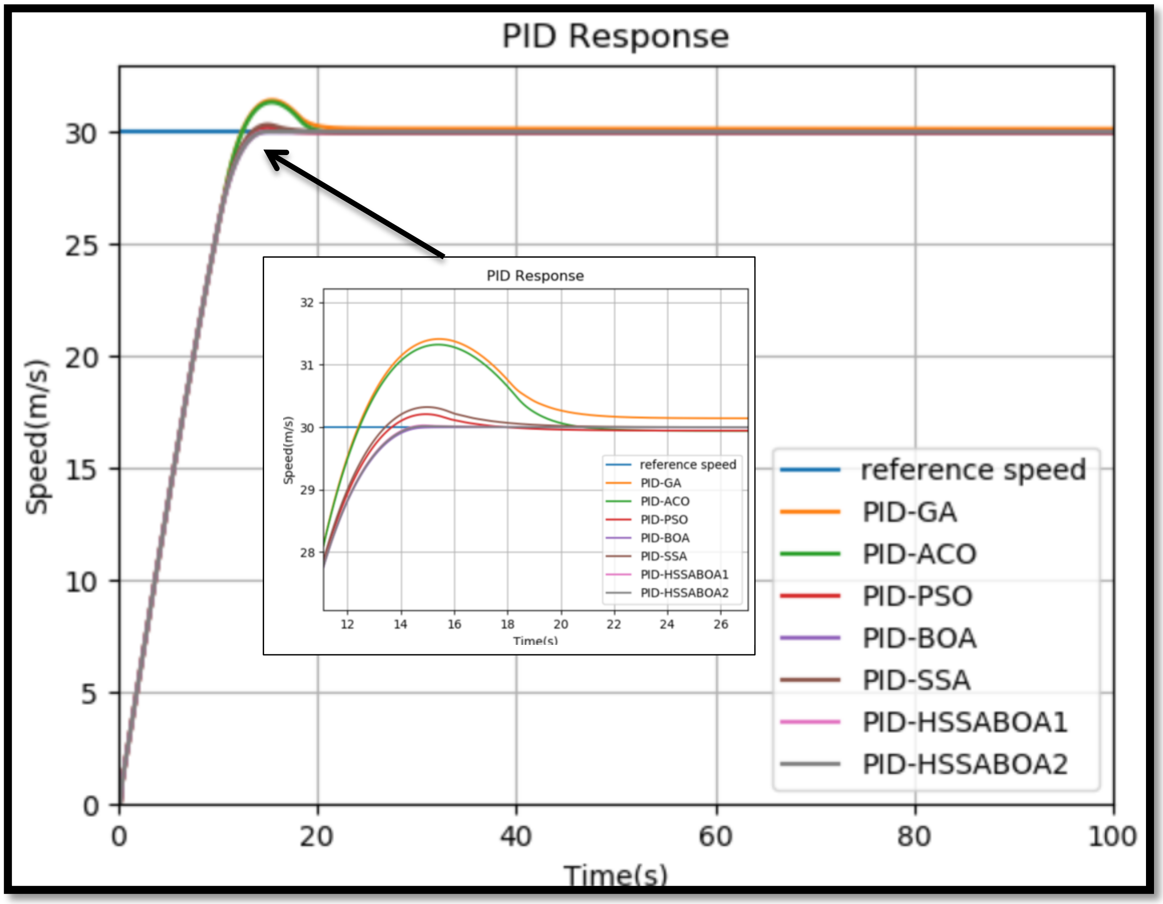


Figure (4.4): The response of the PID based on various optimization methods.

Table (4.2): The specifications of the optimized PID response.

Type of Optimized PID Controller	OS (%)	SSE (m/s)
PID-GA	4.43316	0.13123
PID-ACO	4.58772	0.05946
PID-PSO	0.87617	0.05564
PID-BOA	0.0	0.00005
PID-SSA	1.07573	0.0000002
PID-HSSABOA1	0.10108	1*10^-12
PID-HSSABOA2	0.05819	0.0000002

In the second test, the longitudinal control system of the dynamic vehicle was tested with a constant reference speed of 40 m/s and 50 m/s. The response of the PID-HSSABOA1 was depicted in Figures (4.5) and (4.6), respectively. From the simulation results, the SSE and the OS were extracted into Table (4.3).

Figures (4.5) and (4.6) demonstrate that the PID-HSSABOA1 response was following the constant reference speed correctly. This result indicates the efficiency of the optimized gains.

As noticed in Table (4.3), the PID-HSSABOA1 response has improved values of the SSE and overshoot.

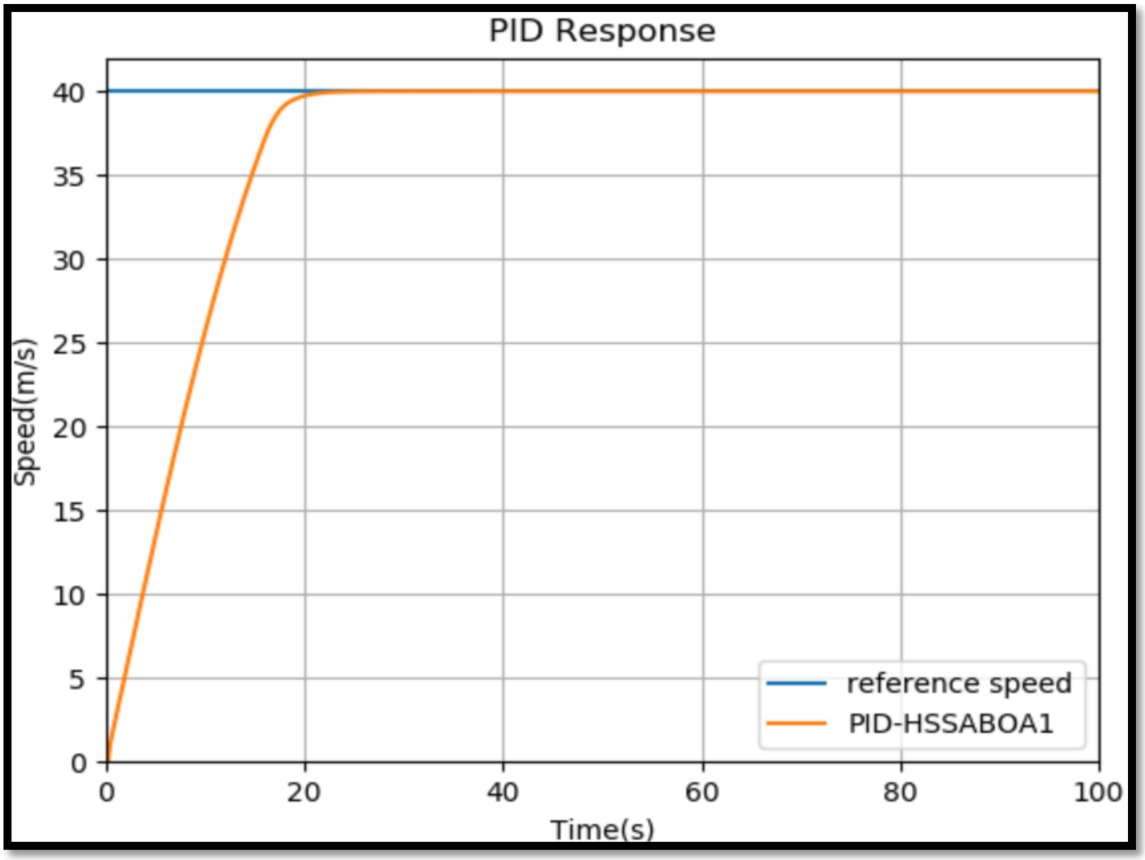


Figure (4.5): The response of the PID-HSSABOA1 at 40 m/s.

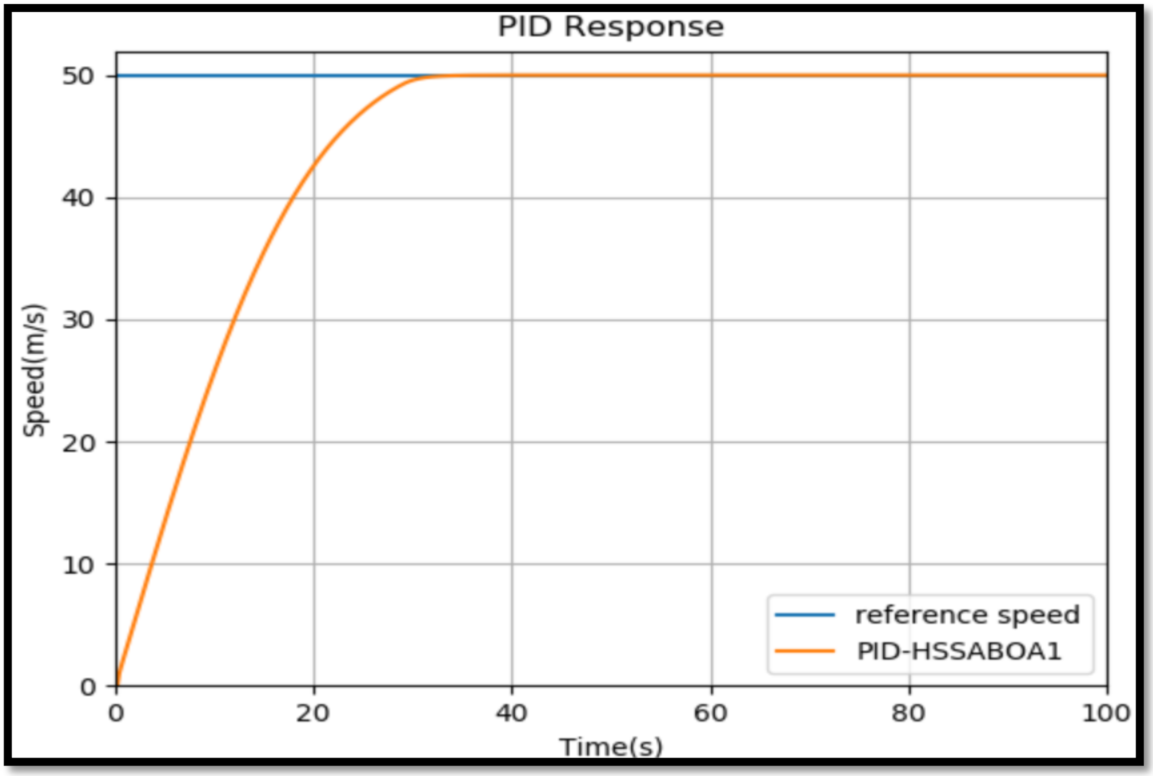


Figure (4.6): The response of the PID-HSSABOA1 at 50 m/s.

Table (4.3): The specifications of the PID-HSSABOA1 response at various speeds.

Speed	(%) OS	SSE (m/s)
40 m/s	0	0.003062484818102007
50 m/s	0	0.014044063259547102

4.3 Kinematic Bicycle Model

4.3.1 Path Generation by Kinematic Bicycle Model

Before testing the control and optimization on the kinematic bicycle model, several tests were tested initially to discover the effects of the parameters of this model on the trajectories generation.

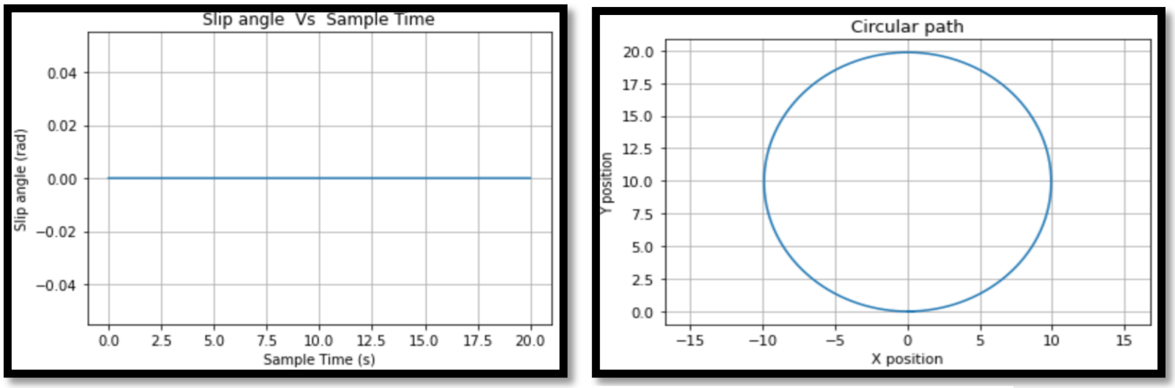
The first test is generating a circular path of radius 10 m in 20 seconds by the kinematic bicycle model. Where the test reflects the effect of slip angle on that path. The steering angle has been directly set to 0.1988 rad and the linear speed has been computed from the relationship as in Equations (41) and (42):

$$v = 2*\boldsymbol{\pi}*\text{radius / total sample time}, \tag{41}$$

$$v = 2*\boldsymbol{\pi}*10/20 = \boldsymbol{\pi}\ \text{rad/s}. \tag{42}$$

This test involves two cases and the results are shown in Figures (4.7) and (4.8), respectively.

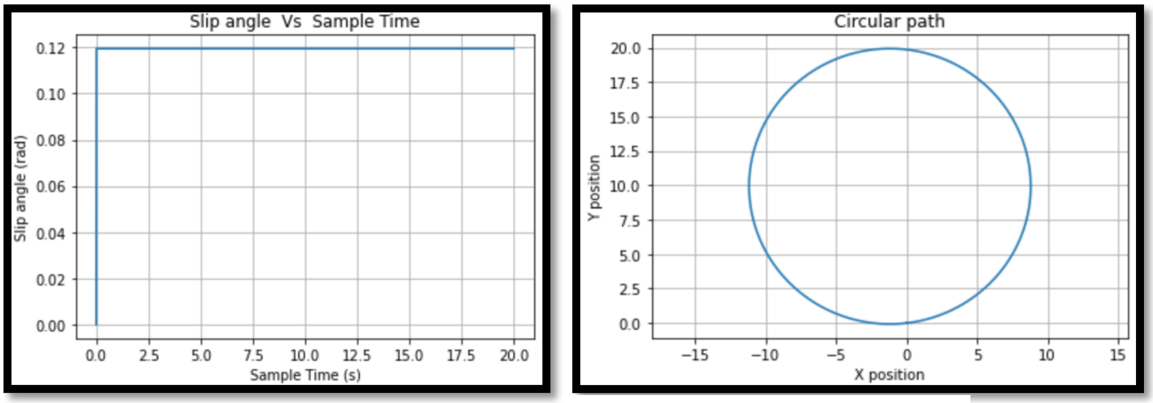
Case 1 (β = 0)



(a) **(b)**

Figure (4.7): Negligibly slip angle: **(a)** Slip angle, **(b)** Circular path.

Case 2 (β ≠ 0)



(a) **(b)**

Figure (4.8): Effect slip angle: **(a)** Slip angle, **(b)** Circular path**.**

As observed in Figures (4.7) and (4.8) when a vehicle moves with the steering is fixed at a particular angle then it will generate a circular arc. In addition by setting β to zero, the circle was centered at (0, 10). But, when the β doesn't set zero, the path was slightly shifted. This clarifies that the sideslip angle makes slight changes in the vehicle direction through the maneuver.

The result in Figure (4.7, a) shows that the slip angle has been set relative to the steering angle value. At the starting point, the steering angle was set to zero, as a result, the slip angle has automatically set to zero because it depends on the value of the steering angle. But, after one sample time, the steering angle has changed to a particular angle during the simulation; therefore, the slip angle changed according to this value.

The second test is generating a square path. The kinematic bicycle model has been provided with a linear speed of 4 m/s. Also, the angular rate has been adjusted at corners only but, in other cases; it has been set to zero. The model completed the path in a sample time of 60 seconds; the results of this test are shown in Figures (4.9) and (4.10).

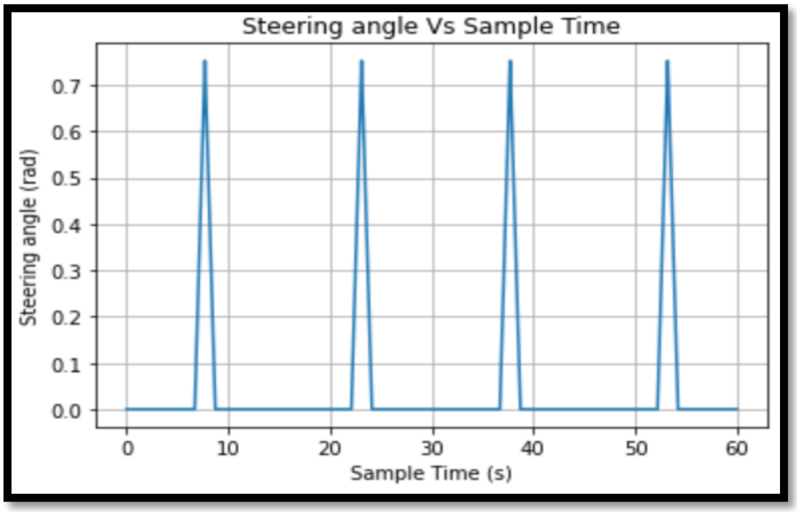


Figure (4.9): Steering angle.

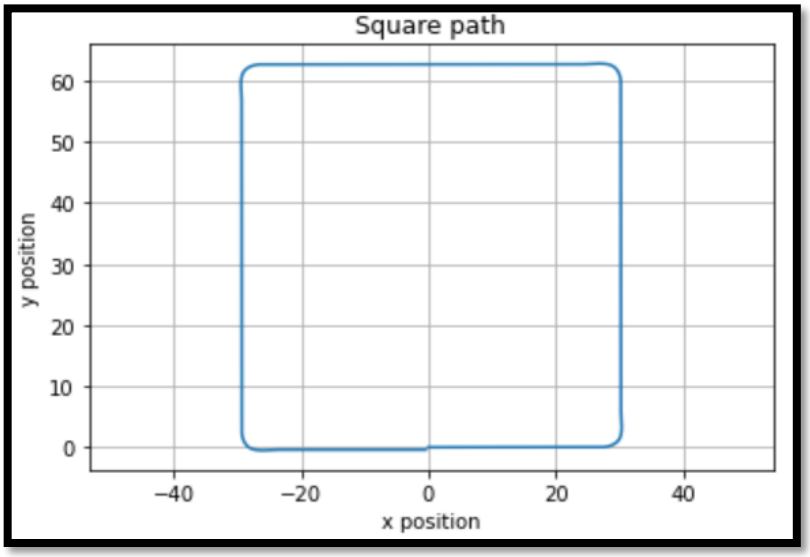
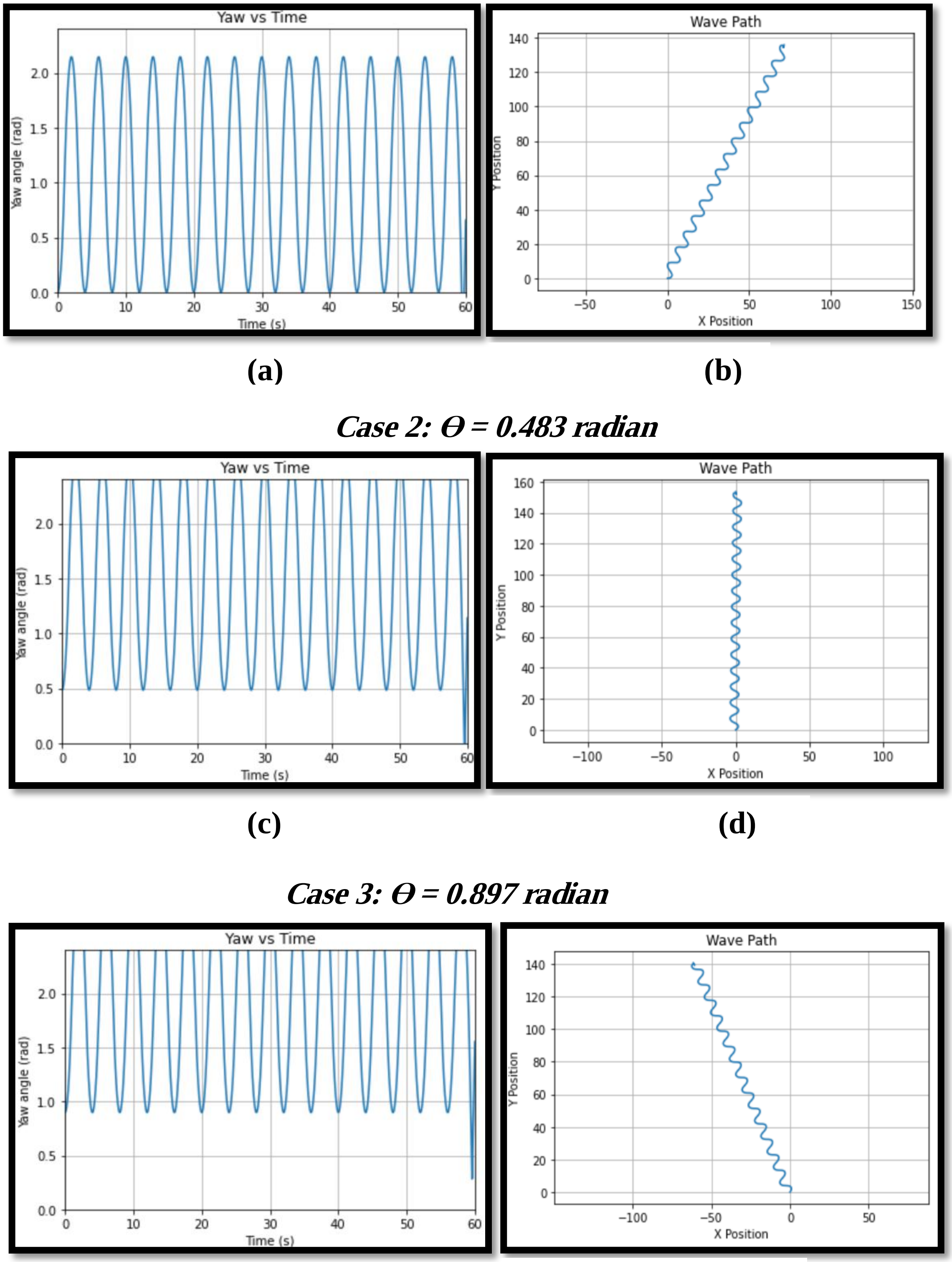


Figure (4.10): Square path.

The outcomes in Figure (4.9) show the steering angle vs time, where the angular rate and the initial value of the steering angle when set to zero, the steering angle also becomes zero, because it is dependent on the angular rate. In another word, when the angular rate is set to a positive or negative practical value, the steering angle is changed according to that value.

The third test is to generate the wave path. This test shows the effects of the initial value of yaw angle on the generated trajectory. At the starting, the kinematic bicycle model has been provided with a linear speed of 4 m/s. The angular rate has been adjusted with an alternating square wave input with amplitude one. After running the simulation, the model had completed the path in a sample time of 60 s. In this test, three cases of the initial value of the yaw angle have been tried, and the test results are presented in Figure (4.11).



Case 1: Ө = 0 radian

(a) **(b)**

Case 2: Ө = 0.483 radian

(c) **(d)**

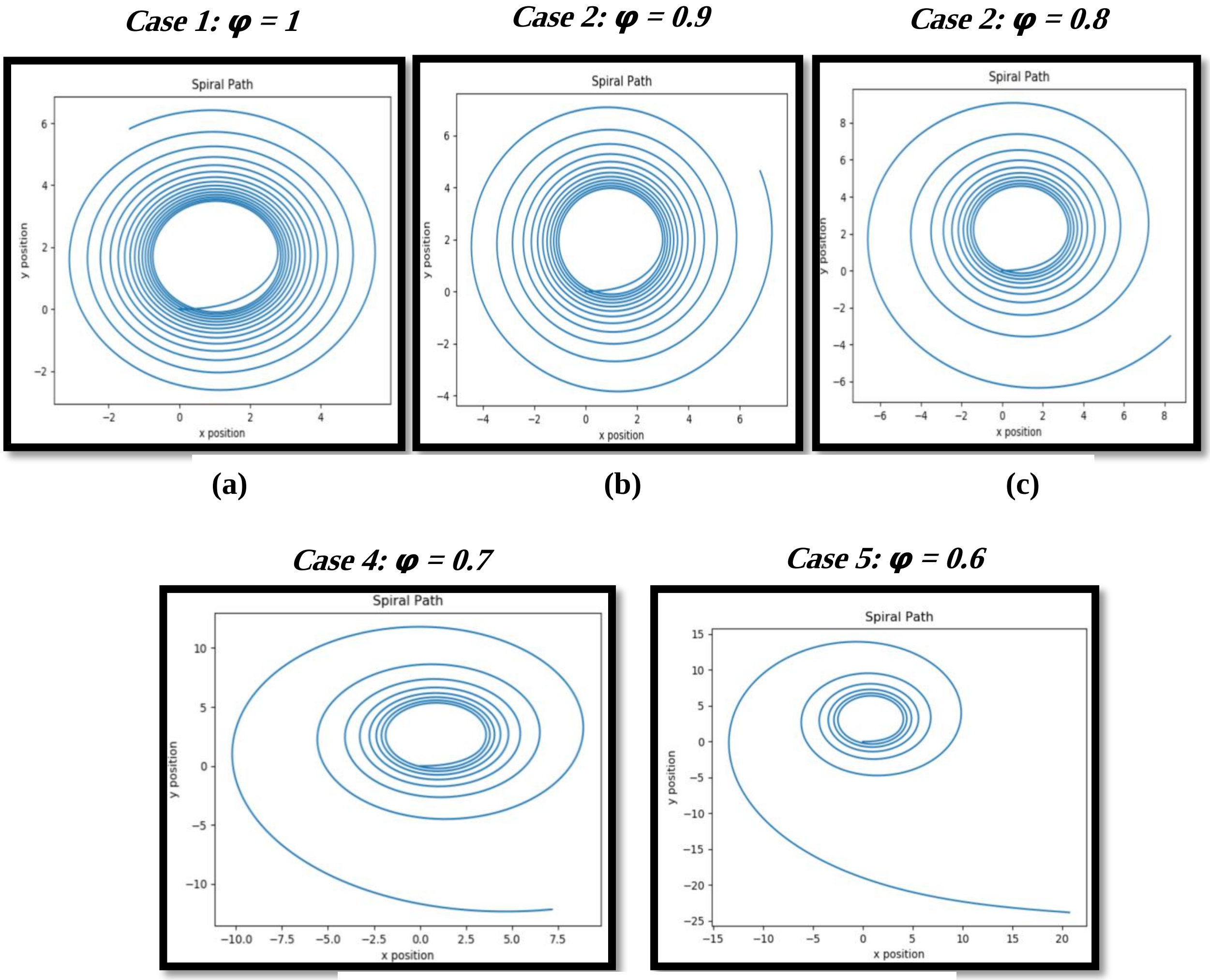
Case 3: Ө = 0.897 radian

(e) **(f)**

Figure (4.11): Effect initial value of yaw angle: **(a, c, e)** yaw angle, **(b, d, f)** the wave path.

The results of Figure (4.10) show that when yaw angle equals zero as in case one, the trajectory for the bicycle model is guided in the original direction. But in cases two and three, the outcomes have showed that the initial value of the yaw angle causes shifting in the guiding direction of the path.

The fourth test is to create a spiral path. In this test, the bicycle model has been set to a linear speed of 4 m/s. Then, the angular rate has been set, from a high positive to a small negative rate and the model has completed the path with a sample time of 60 s. Five cases of positive φ were taken, but the negative rate for each case has been set to 0.01. The test results are depicted in Figure (4.12).


Case 1: φ = 1

Case 2: φ = 0.9

Case 2: φ = 0.8

Case 4: φ = 0.7

Case 5: φ = 0.6

(a) **(b)** **(c)**

(d) **(e)**

Figure (4.12): (a), (b), (c), (d), and **(e)** Spiral path.

Figure (4.12) shows that the bicycle model has followed a spiral trajectory with decreasing radius of the curvature. It started from the zero position and continues to follow a spiral trajectory until the sample time has finished. On another side, the steering angle values have increased gradually according to angular rate values. This proves that when the angular rate is set to a positive large value, the path includes many spiral trajectories compared to the small positive value.

4.3.2 Longitudinal Control of Kinematic Bicycle Model

At starting, the bicycle model was provided by the linear input speed of 20 m/s. Then, the BOA, SSA, HSSABOA1, and HSSABOA2 algorithms were used to improve the gain of the proportional controller.

Table (4.4) involves the best controller's parameters and minimum fitness function values provided by the optimization algorithms. According to the results of that table, P-BOA, P-SSA, P-HSSABOA1, and P-HSSABOA2 included the same fitness function value due to the linearity of the speed equation of the bicycle model.

Table (4.4): Best values of P controller gain at the best IAE value

Optimization Method	k_p	IAE (m/s)
BOA	21.95726	201.00000000000028
SSA	16.43888	201.00000000000028
HSSABOA1	10	201.00000000000028
HSSABOA2	10	201.00000000000028

One of the gains related to HSSABOA in Table (4.4) is applied for the P controller to manipulate the throttle/brake of the kinematic bicycle model. The, response of the P controller based on HSSABOA1/HSSABOA2 is displayed in Figure (4.13). The simulation outcome in Figure (4.13) illustrated that the response of the optimized controller by the proposed algorithm has achieved good tracing for the desired speed.

Figure (4.14) includes the optimized P controller response to follow target speed profiles of 15 m/s, 10 m/s, and 5 m/s. The outcome in Figure (4.14) clarifies that the improved longitudinal controller could match the required speed correctly, which refers to robust the optimized controller gain.

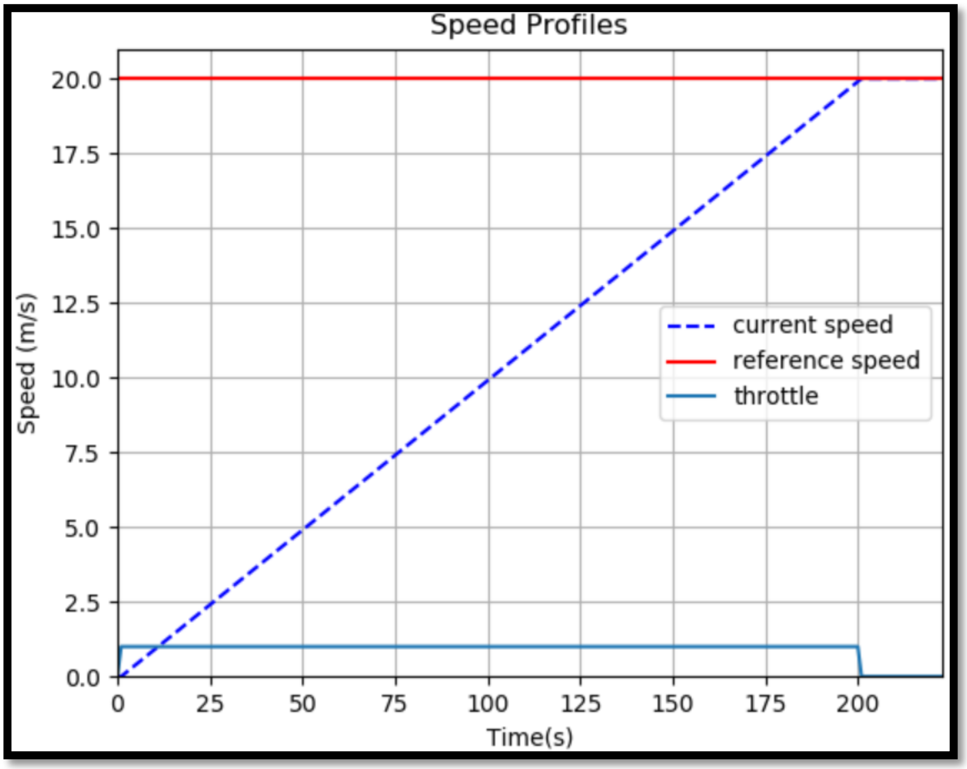
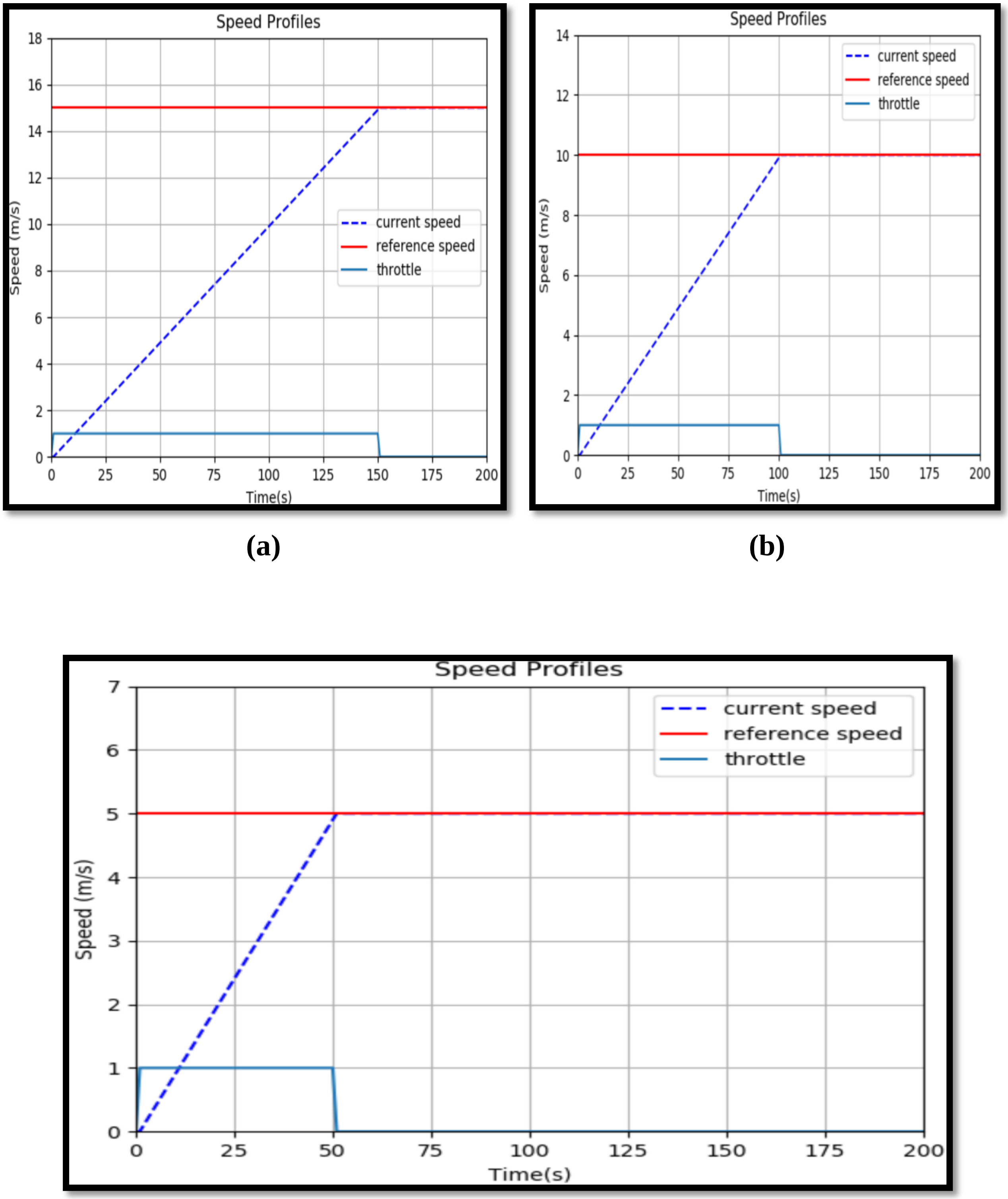


Figure (4.13): Speed curve of the P controller based on HSSABOA1 at 20 m/s.



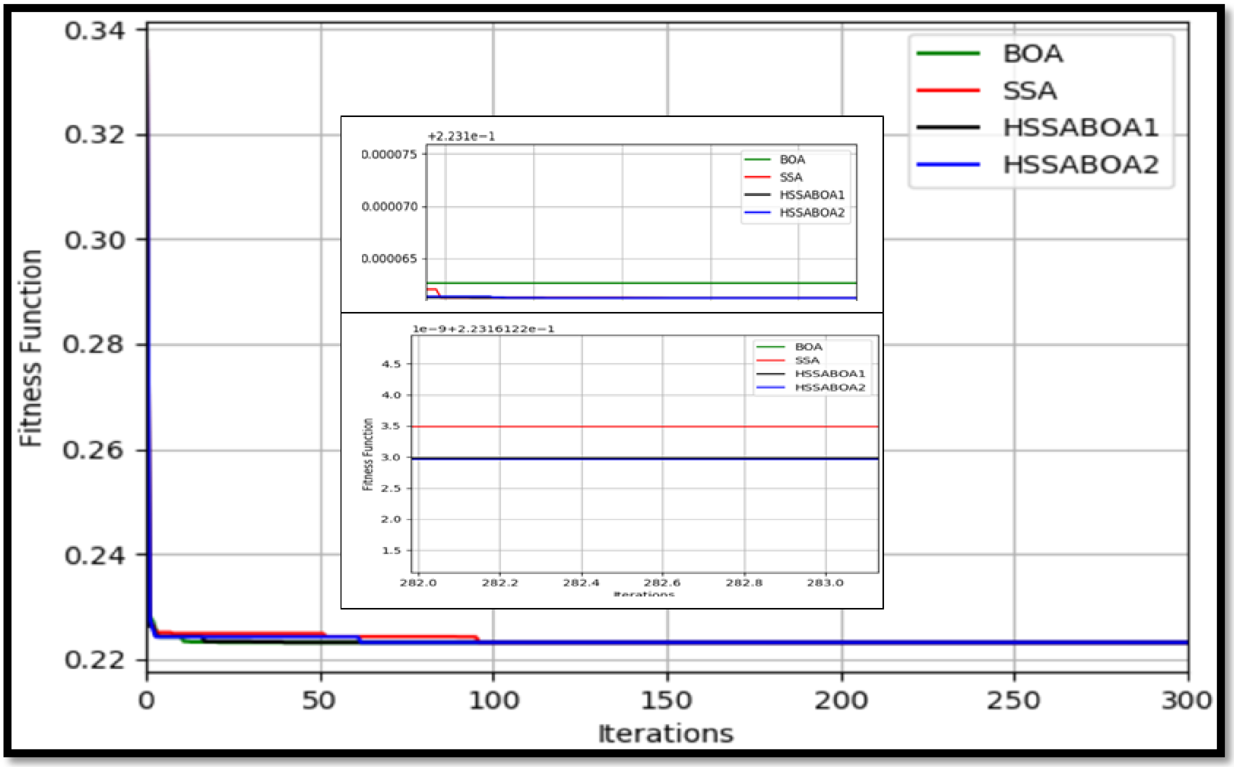
(a) **(b)**

(c)

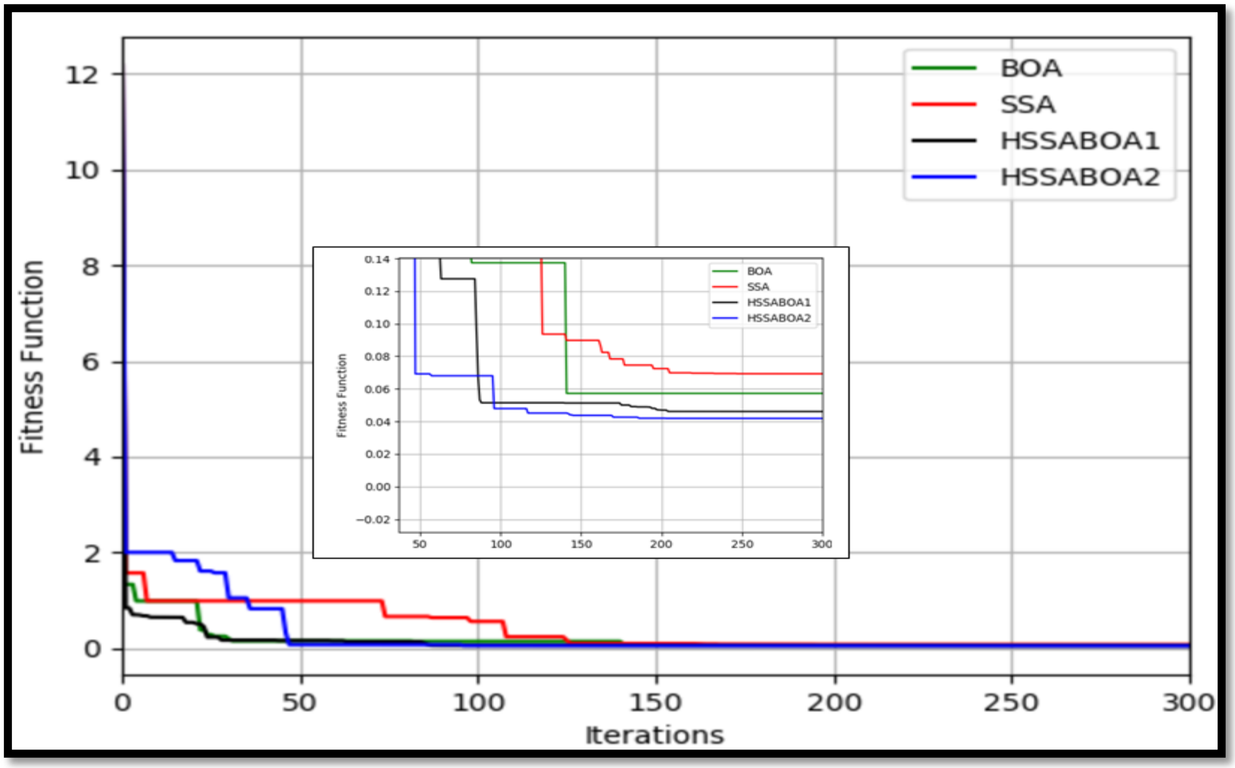
Figure (4.14): Speed curve of the P controller based on HSSABOA1 at: **(a)** 15 m/s, **(b)** 10 m/s, and **(c)** 5 m/s

4.3.3 Lateral Control of Kinematic Bicycle Model

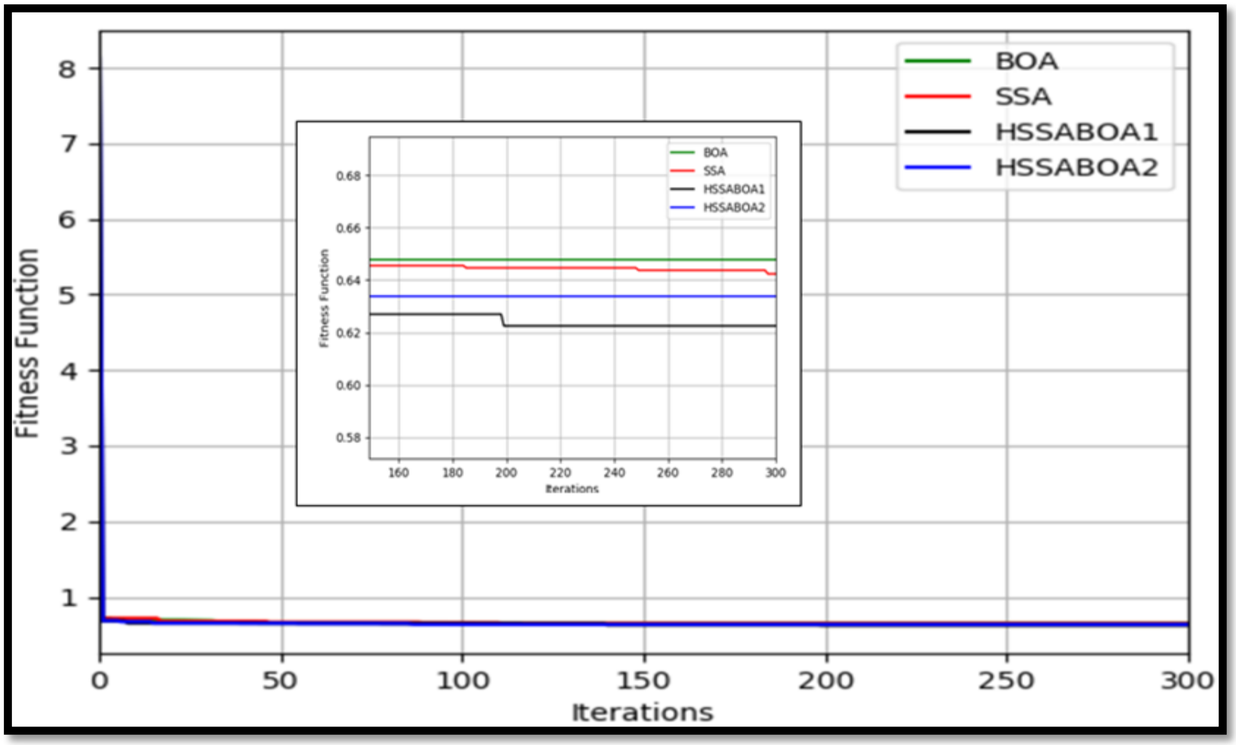
After finishing the longitudinal control optimization of the kinematic bicycle model; the lateral control was adjusted by the BOA, SSA, and the proposed algorithms. So, in this section, four tests were performed, in the first test, the Stanley, MS, PID, and PTMPID controllers are tuned by the BOA, SSA, HSSABOA1, and HSSABOA2 algorithms. The comparisons between these controllers are performed by representing the convergence curve as shown in Figure (4.15).



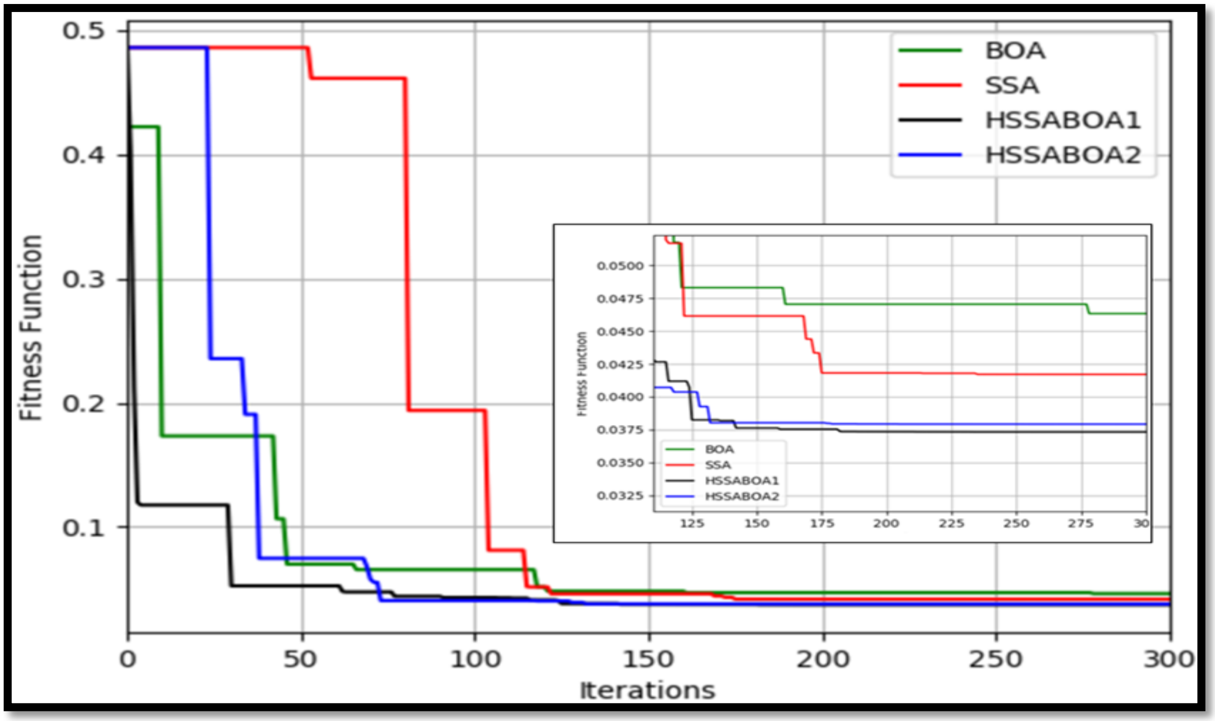
(a)


BOA
SSA
HSSABOA1
HSSABOA2
Fitness Function
12
10
8
6
4
2
0
0
50
100
150
200
250
300
Iterations
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
50
100
150
200
250
300

(b)


BOA
SSA
HSSABOA1
HSSABOA2
Fitness Function
8
7
6
5
4
3
2
1
0
50
100
150
200
250
300
Iterations
0.68
0.66
0.64
0.62
0.60
0.58
160
180
200
220
240
260
280
300

(c)



(d)

Figure (4.15): Convergence curve of the: **(a)** Stanley, **(b)** MS, **(c)** PID, and **(d)** PTMPID controllers based on optimization algorithms.

Figure (4.15) indicated that the controllers based on the proposed algorithms are converged faster compared to the original algorithms. The results of the improved gains that it obtained at the lowest value of the fitness function are written in Tables (4.5 to 4.8).

Table (4.5): Best gains of Stanley controller.

Optimization method	K	RMSE (m)
BOA	15.49408234	0.22316268389806113
SSA	15.49310536	0.22316122296204102
HSSABOA1	15.49310538	0.2231612229837286
HSSABOA2	15.49310534	0.2231612229245861

Table (4.6): Best gains of MS controller.

Optimization Method	k_1	k_2	k_3	k_4	RMSE (m)
BOA	1*10^-7	3.86419	2.84269	0.12377	0.05725
SSA	0.03207	4.43253	1.329	0.03313	0.06929
HSSABOA1	0.10878	6.58594	2.36279	0.15505	0.04935
HSSABOA2	0.07345	3.43565	6.79116	0.23099	0.04566

Table (4.7): Best gains of PID controller.

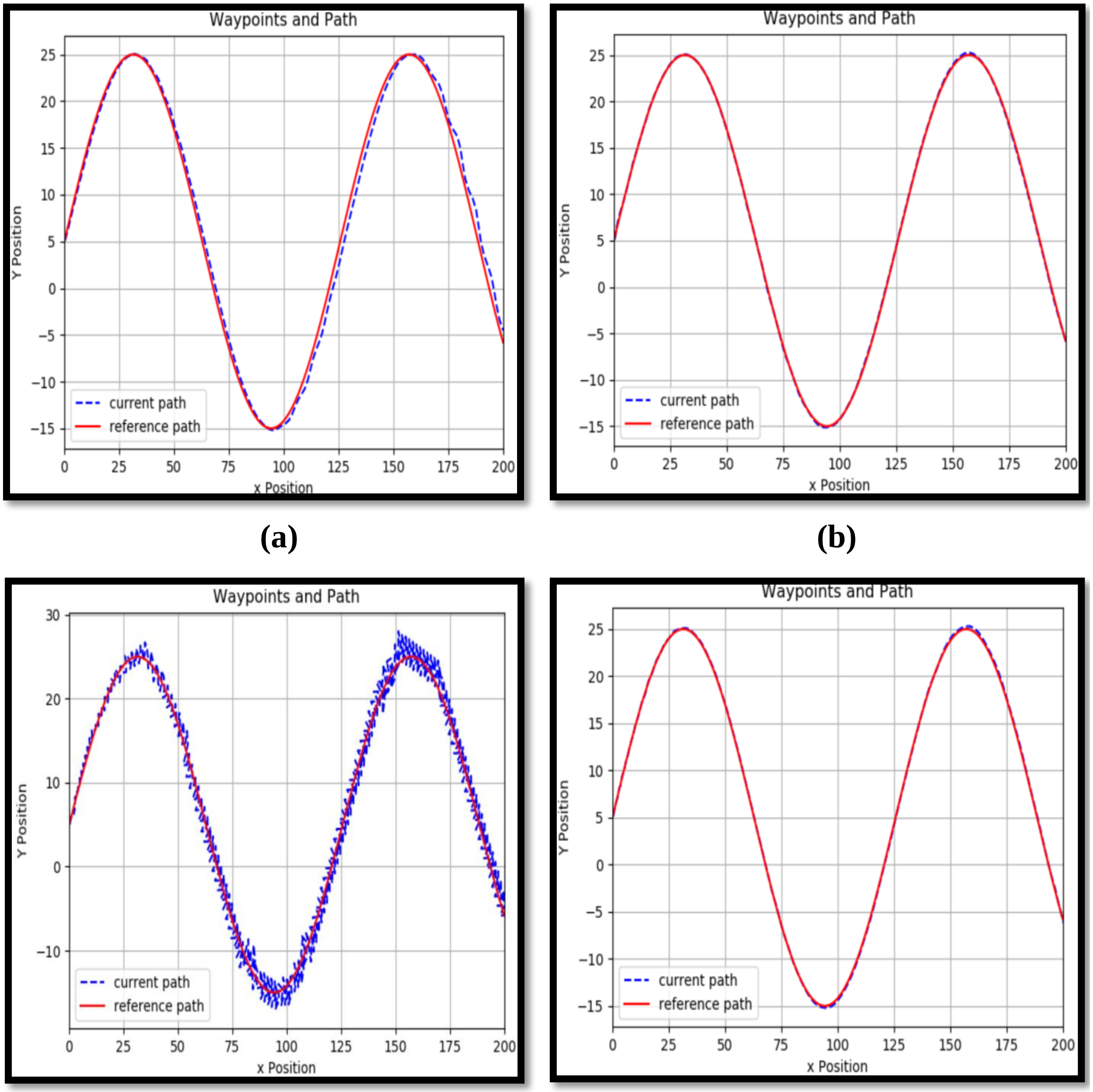
Optimization Method	k_p	k_i	k_d	RMSE (m)
BOA	9.19209	1.34555	1.33993	0.64791
SSA	13.38832	2.28402	10.72765	0.64239
HSSABOA1	11.52305	0.06096	2.06341	0.62256
HSSABOA2	12.24485	7.88071	2.63422	0.63389

Table (4.8): Best gains of PTMPID controller.

Optimization Method	k_p	k_i	k_d	RMSE (m)
BOA	3.420327	9.40376	0.64390	0.046336
SSA	3.653217	12.97717	0.66948	0.04171
HSSABOA1	4.63956	7.51872	0.70629	0.037321
HSSABOA2	4.57061	6.92445	0.51407	0.037925

Tables (4.5 to 4.8) have demonstrated that the PTMPID based on HSSABOA1/ HSSABOA2 offered the best value of the objective function among other controllers.

In the second test, the best improved parameters of each controller which are stated in Tables (4.5 to 4.8) have been applied to the lateral controller to set the steering angle of the kinematic bicycle model. The responses of the four controls are illustrated in Figure (4.16).



(a) (b)

(c) (d)

Figure (4.16): The response of trajectory tracking by: **(a)** Stanely-HSSABOA2, **(b)** MS-HSSABOA2, **(c)** PID-HSSABOA1, and **(d)** PTMPID-HSSABOA1.

As observed in Figure (4.16) that the MS-HSSABOA2 and PTMPID-HSSABOA1 have tracked the desired trajectory with more accuracy than that with the Stanely-HSSABOA2, and PID-HSSABOA1.

Moreover, the result of Figure (4.16) demonstrated that the response of the PID-HSSABOA1 to follow the target trajectory has more oscillation than optimized other controllers. In contrast, the PID-HSSABOA1 has achieved good performance to track the desired path at a very low speed as shown in Figure (4.17).

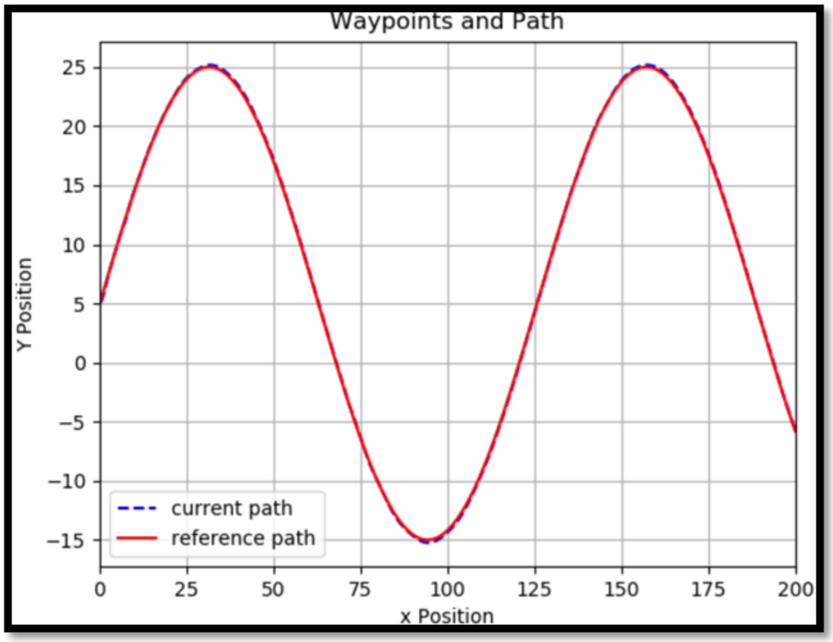
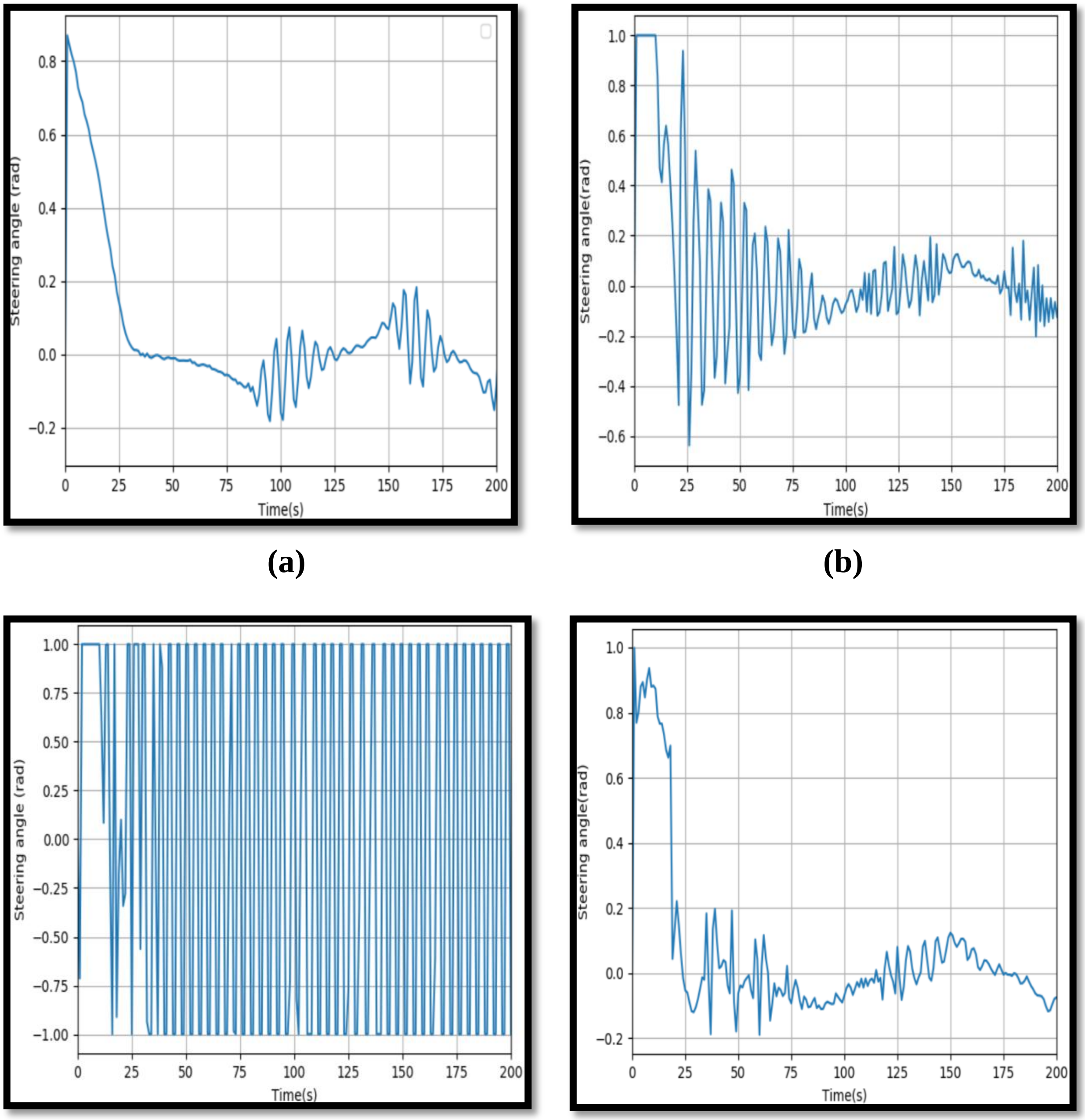


Figure (4.17): The response of PID-HSSABOA1 at 5 m/s.

The reason for that is the formula of the PID controller is not related to the vehicle's speed. The better solution to that matter is the optimized PTMPID, where it managed to track the desired path correctly at different speeds without large oscillation.

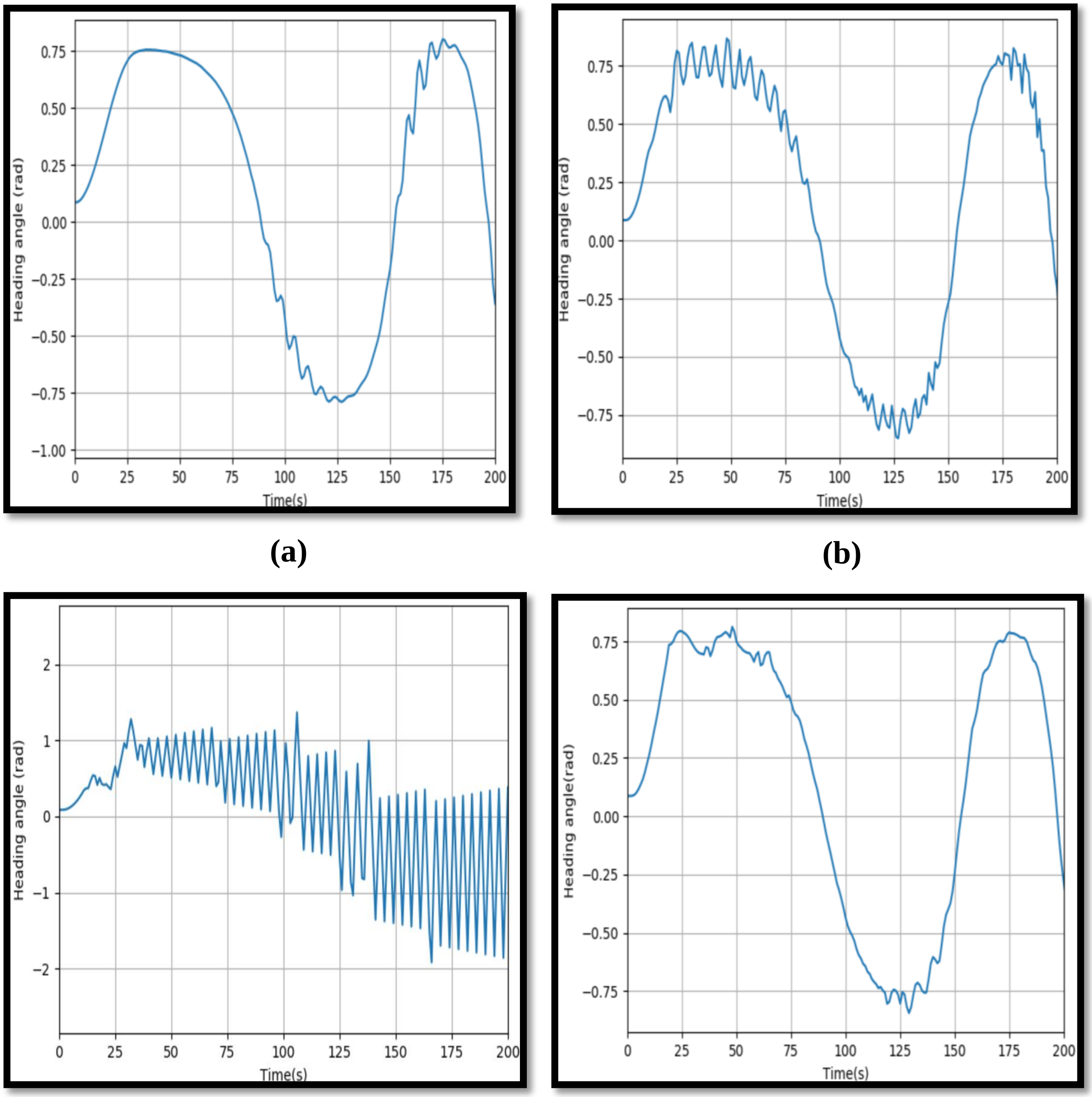
Figure (4.18) shows the steering angle of path tracking controllers to estimate the stability of each controller. In contrast, Figure (4.19) clarified the yaw or heading angle of path tracking controllers to evaluate the stability of each controller.



(a) (b)

(c) (d)

Figure (4.18): The response of steering angle by: **(a)** Stanley-HSSABOA2, **(b)** MS-HSSABOA2, **(c)** PID-HSSABOA1, and **(d)** PTMPID-HSSABOA1.

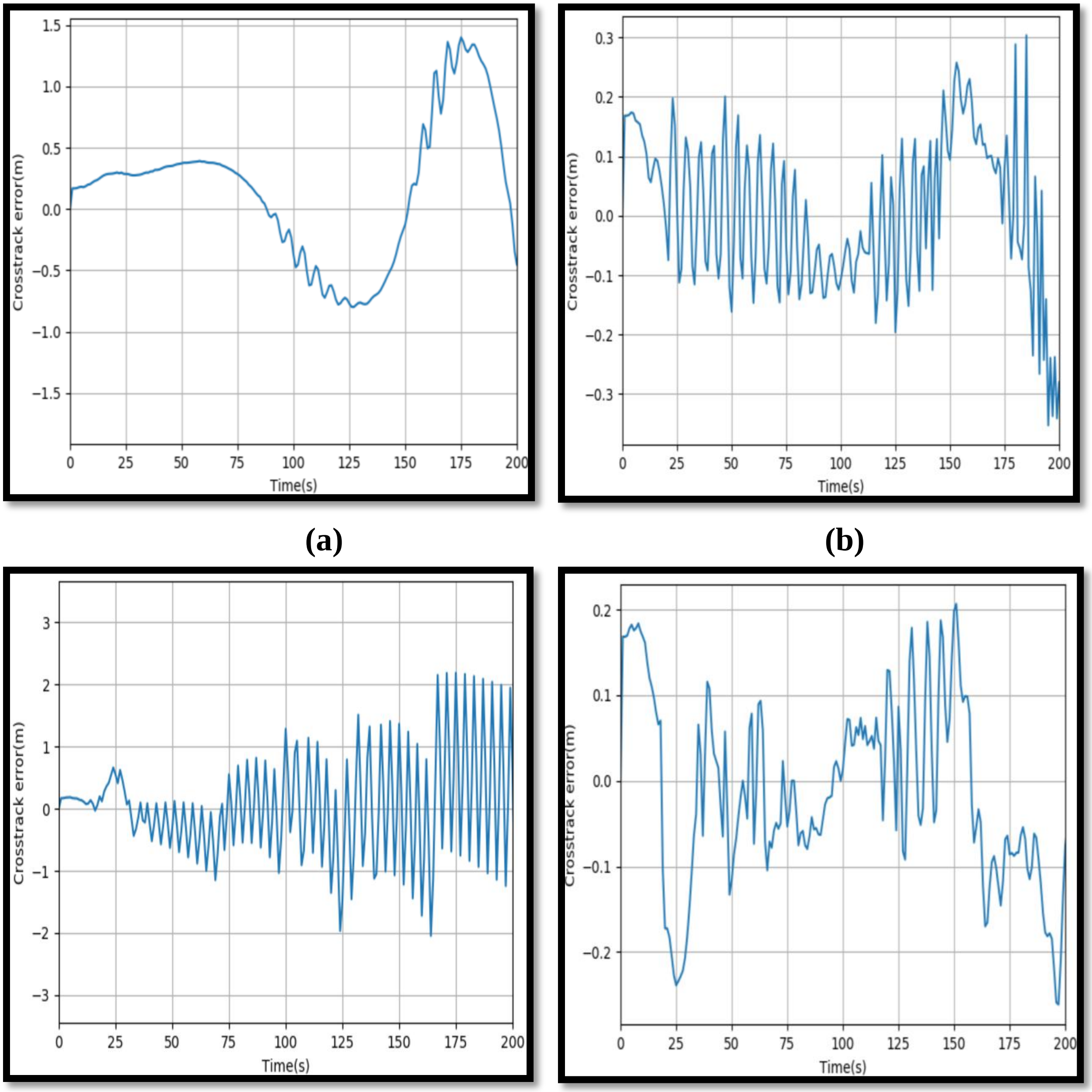


(a) **(b)**

(c) **(d)**

Figure (4.19): The response of heading angle by: **(a)** Stanley-HSSABOA2, **(b)** MS-HSSABOA2, **(c)** PID-HSSABOA1, and **(d)** PTMPID-HSSABOA1.

Figure (4.20) shows the crosstrack error of path tracking controllers. Where the result showed that PTMPID-HSSABOA1 offers a smaller crosstrack error than other controllers.

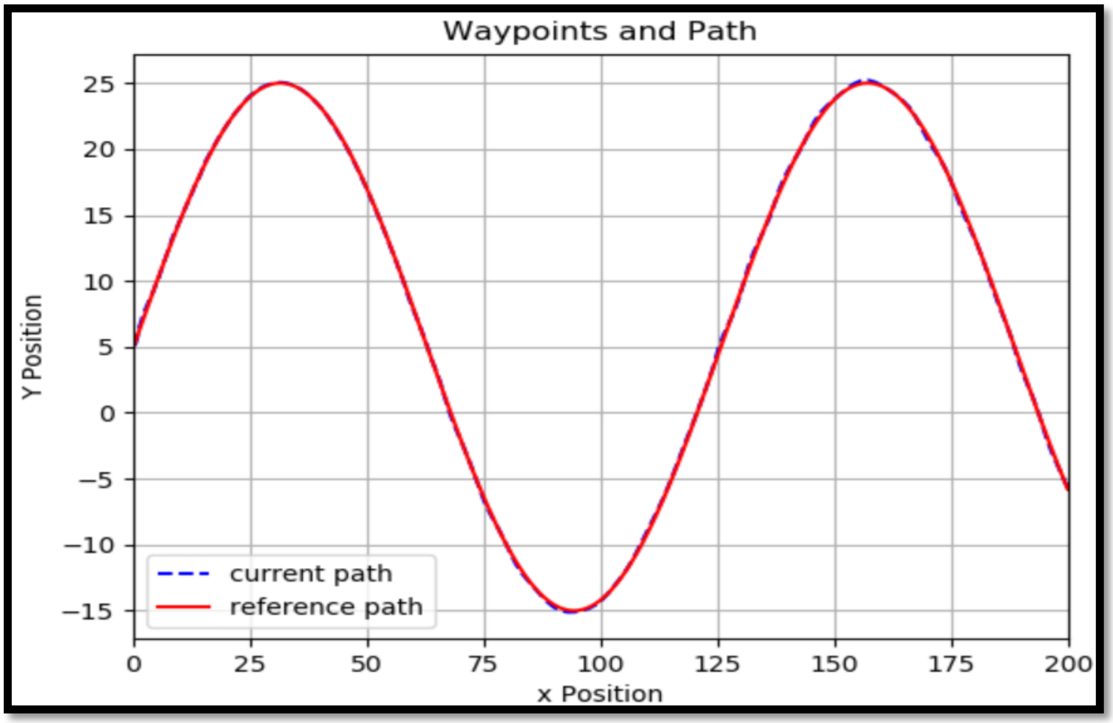


(a) **(b)**

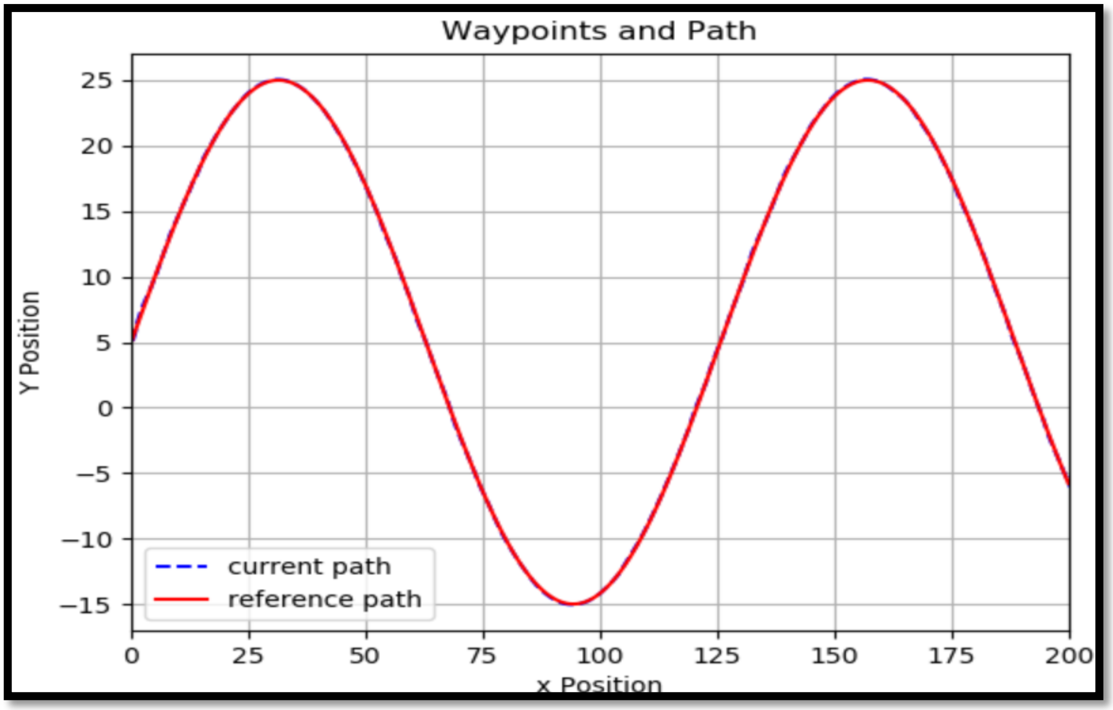
(c) **(d)**

Figure (4.20): The response of crosstrack error by: **(a)** Stanley-HSSABOA2, **(b)** MS-HSSABOA2, **(c)** PID-HSSABOA1, and **(d)** PTMPID-HSSABOA1.

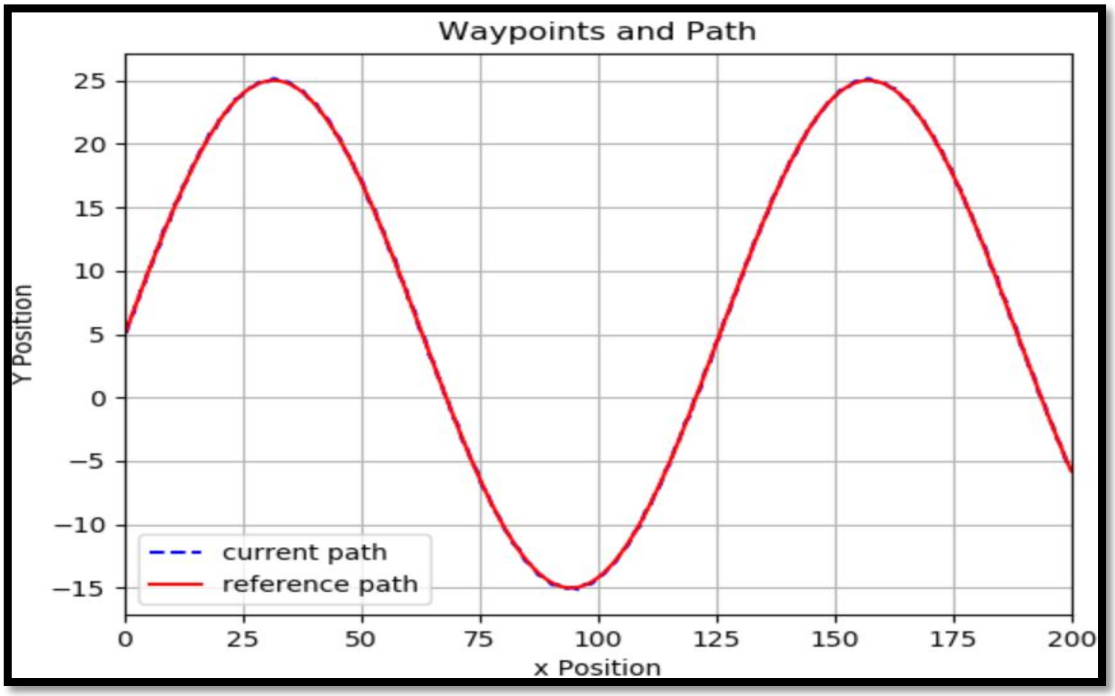
In the third test, PTMPID-HSSABOA1 was examined with speeds of 15 m/s, 10 m/s, and 5 m/s. The outcomes of that test are presented in Figure (4.21).


Waypoints and Path
Y Position
x Position
25
20
15
10
5
0
−5
−10
−15
0
25
50
75
100
125
150
175
200
current path
reference path

(a)


Waypoints and Path
Y Position
x Position
25
20
15
10
5
0
−5
−10
−15
0
25
50
75
100
125
150
175
200
current path
reference path

(b)



(c)

Figure (4.21): PTMPID-HSSABOA1 response at: **(a)** 15 m/s, **(b)** 10 m/s, **(c)** 5 m/s.

Figure (4.21) showed that the ADC can follow the trajectory at different speeds very well without any large oscillation.

In the fourth test, the PTMPID was tuned by HSSABOA1 with a longitudinal reference speed of 10 m/s, the obtained response of this test is shown in Figure (4.22). After optimizing PTMPID gains, the desired speed was adjusted to 20 m/s and the output response is depicted in Figure (4.23). The simulation result of Figure (4.22) clarified that the response of the PTMPID-HSSABOA1 was good but it has started to oscillate when set the reference speed to 20 m/s as presented in Figure (4.23). This led to conclude important point that the lateral control can trace the predefined path at various desired speeds without large oscillation only if it is tuned at the maximum target speed.

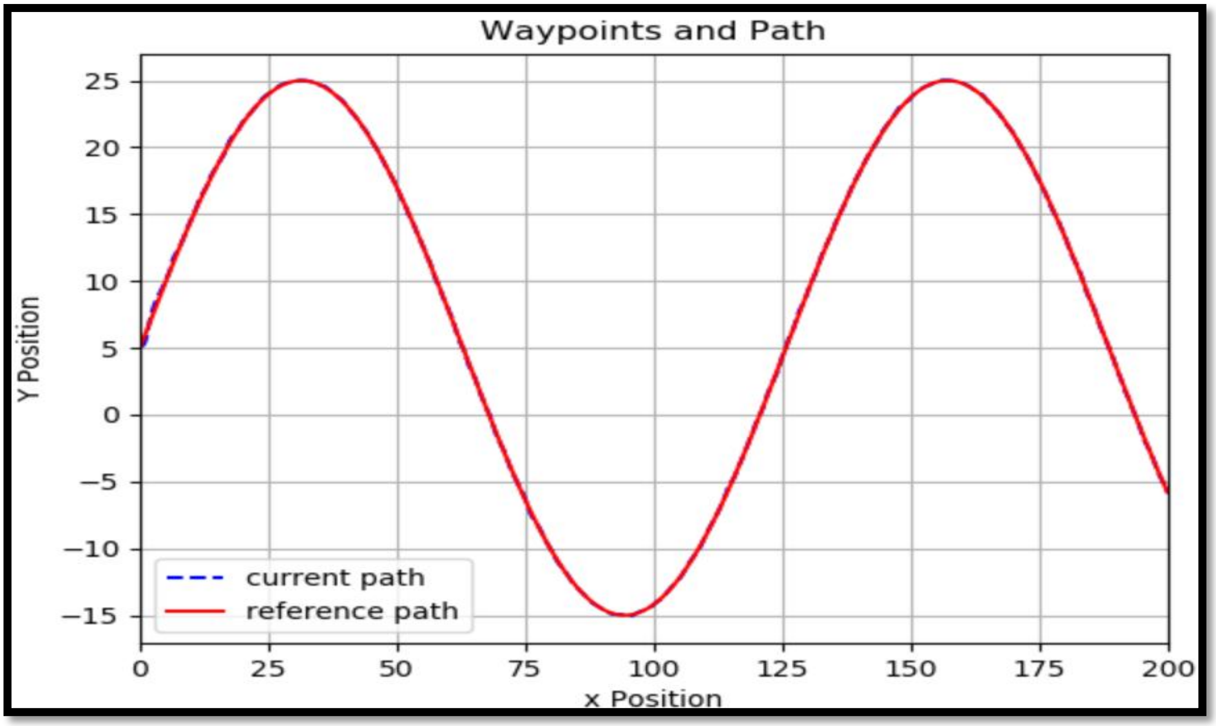


Figure (4.22): The response of PTMPID-HSSABOA1 at 10 m/s.

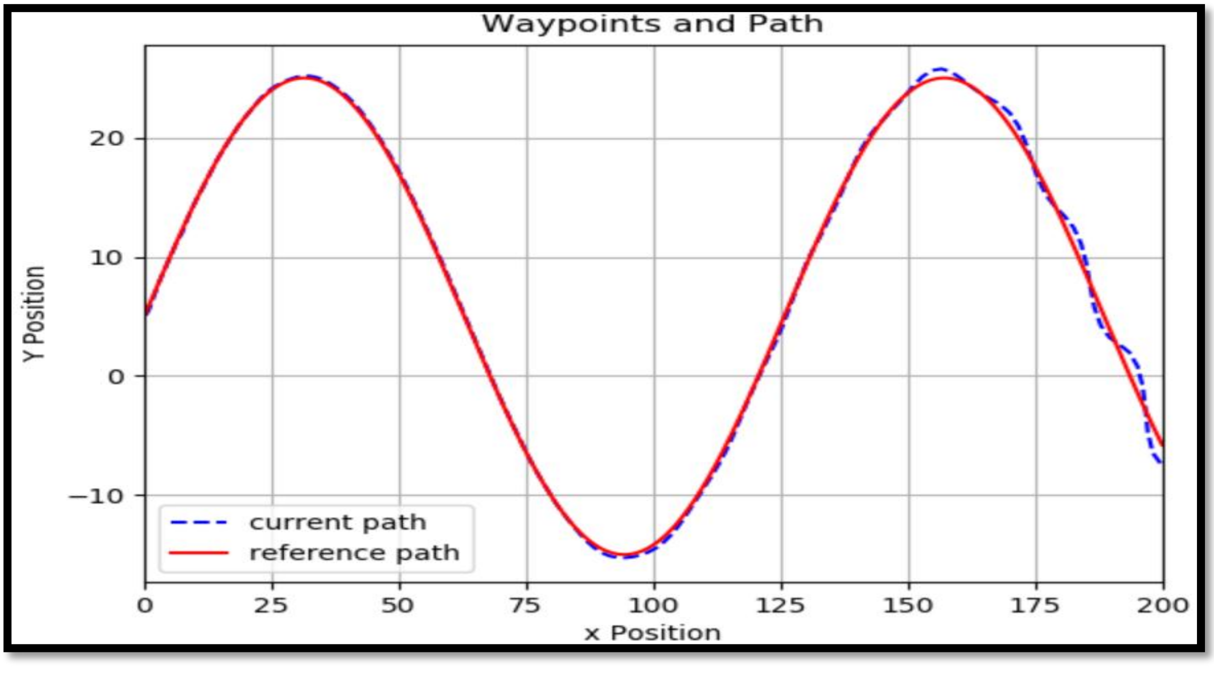


Figure (4.23): The response of PTMPID-HSSABOA1 at 20 m/s.

In the next chapter, the main conclusions and recommendations for future work will be presented.

Chapter Five: Conclusions and Recommendations

Chapter Five
Conclusions and Recommendations

5.1 Introduction

In this chapter, the conclusions of this work will be introduced. In addition to provide some suggestions and recommendations for future work.

5.2 Conclusions

There are set of important conclusions that have been extracted from the results of performing tests are scheduled below:

1. When the throttle and brake input values change, the speed curve has changed according to those values.

2. The sideslip of the kinematic bicycle model causes a slight shift in the vehicle direction through the maneuver.

3. The optimization techniques (SSA and BOA) have a better performance than GA, ACO algorithm, and PSO algorithm. On the other hand, HSSABOA1 and HSSABOA2 have faster convergence and are more guaranteed to get the best controller's gains compared to the original techniques (SSA and BOA).

4. The PTMPID based on HSSABOA1 and HSSABOA2 has a better performance in reducing lateral control error for the kinematic bicycle model than Stanley-HSSABOA, MS-HSSABOA, and PID-HSSABOA.

5. When the lateral control is optimized at the maximum road speed, the output response can trace the desired path at various road speeds correctly without large oscillation.

5.3 Recommendations for Future Work

In this section, some of suggestions to develop the current work in future are listed as follows:

1. Adding a feedforward controller to the PID controller or designing an adaptive PID controller for solving some of the problems such as delay response to errors and the response to the disturbances.
2. Embedded PTMPID design with the (HSSABOA1 and HSSABOA2) optimization algorithms for the dynamic bicycle model to examine maneuvers tracking at high speeds.
3. To verify the effectiveness of proposed work, simulation experiments will be conducted for longitudinal and lateral controls on the prototype real vehicle in order to examine path tracking at various speeds.

References

1. M. Shneier and R. Bostelman, Literature Review of Mobile Robots for Manufacturing. Gaithersburg, MD: U.S. Department of Commerce, National Institute of Standards and Technology, 2015.
2. M. B. Alatise and G. P. Hancke, “A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods,” IEEE Access, vol. 8, pp. 39830–39846, 2020, doi: 10.1109/ACCESS.2020.2975643.
3. F. Rubio, F. Valero, and C. Llopis-albert, “A review of mobile robots : Concepts , methods , theoretical framework , and applications,” Int. J. Adv. Robot. Syst., vol. 16, no. 2, pp. 1–22, 2019, doi: 10.1177/1729881419839596.
4. C. Badue et al., “Self-Driving Cars: A Survey,” Expert Syst. Appl., p. 113816, 2021.
5. C. V. Samak, T. V. Samak, and S. Kandhasamy, “Proximally Optimal Predictive Control Algorithm for Path Tracking of Self-Driving Cars,” in Advances in Robotics-5th International Conference of The Robotics Society, 2021, pp. 1–5, [Online]. Available: http://arxiv.org/abs/2103.13240.
6. 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 Trans. Intell. Veh., vol. 1, no. 1, pp. 33–55, 2016, doi: 10.1109/TIV.2016.2578706.
7. M. Khalid et al., “Autonomous Transportation in Emergency Healthcare Services: Framework, Challenges, and Future Work,” IEEE Internet Things Mag., vol. 4, no. 1, pp. 28–33, 2021, doi: 10.1109/iotm.0011.2000076.
8. T. Litman, “Autonomous Vehicle Implementation Predictions: Implications for Transport Planning,” pp. 1–48, 2020, [Online]. Available: https://www.vtpi.org/avip.pdf.
9. J. Hecht, “Lidar for Self-Dr riving Cars,” Opt. Photonics News, vol. 19, no. 1, pp. 26–33, 2018.
10. S. D. Pendleton et al., “Perception, planning, control, and coordination for autonomous vehicles,” Machines, vol. 5, no. 1, p. 6, 2017, doi: 10.3390/machines5010006.
11. J. Betz et al., “A software architecture for an autonomous racecar,” in IEEE 89th Vehicular Technology Conference (VTC2019-Spring), 2019, pp. 1–6, doi: 10.1109/VTCSpring.2019.8746367.
12. S. Liu, J. Tang, Z. Zhang, and J. Gaudiot, “CAAD : Computer Architecture for Autonomous Driving,” Computer, vol. 50, no. 8, pp. 18–25, 2017.
13. H. Sjafrie, Introduction to Self-Driving Vehicle Technology, 1st Edition. Chapman and Hall/CRC, 2019.
14. L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, “Salp swarm algorithm : a comprehensive survey,” Neural Comput. Appl., vol. 32, no. 15, pp. 11195–11215, 2020, doi: 10.1007/s00521-019-04629-4.
15. M. Gavrilas, “Heuristic and Metaheuristic Optimization Techniques with Application to Power Systems,” Tech. Univ. Iasi, D. Mangeron Blvd., Iasi, Rom., pp. 95–103, 2010.
16. "Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm." In 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), pp. 1443-1447. IEEE, 2014.
17. J. M. Snider, “Automatic Steering Methods for Autonomous Automobile Path Tracking,” Robot. Institute, Pittsburgh, PA, Tech. Rep. C., 2009, [Online]. Available: http://www.ri.cmu.edu/pub_files/2009/2/Automatic_Steering_Methods_for_Autonomo

us_Automobile_Path_Tracking.pdf.

18. E. P. Ping and S. K. Swee, “Simulation and experiment of automatic steering control for lane keeping manoeuvre,” in I2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012), 2012, vol. 1, pp. 105–110, doi: 10.1109/ICIAS.2012.6306169.
19. J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design,” in 2015 IEEE intelligent vehicles symposium (IV), 2015, pp. 1094–1099, doi: 10.1109/IVS.2015.7225830.
20. S. Dominguez, A. Ali, G. Garcia, and P. Martinet, “Comparison of lateral controllers for autonomous vehicle : Experimental results,” in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 1418–1423, doi: 10.1109/ITSC.2016.7795743.
21. N. H. Amer, H. Zamzuri, K. Hudha, V. R. Aparow, Z. A. Kadir, and A. F. Z. Abidin, “Path tracking controller of an autonomous armoured vehicle using modified Stanley controller optimized with particle swarm optimization,” J. Brazilian Soc. Mech. Sci. Eng., vol. 40, no. 2, pp. 1–17, 2018, doi: 10.1007/s40430-017-0945-z.
22. W. Farag, “Track Maneuvering using PID Control for Self-driving Cars,” Recent Adv. Electr. Electron. Eng. (Formerly Recent Patents Electr. Electron. Eng., vol. 13, no. 1, pp. 91–100, 2020, doi: 10.2174/2352096512666190118161122.
23. T. Herlambang, D. Rahmalia, and T. Yulianto, “Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) for optimizing PID parameters on Autonomous Underwater Vehicle (AUV) control system,” in Journal of Physics: Conference Series, 2019, vol. 1211, no. 1, p. 012039, doi: 10.1088/1742-6596/1211/1/012039.
24. L. El Hajjami, E. M. Mellouli, and M. Berrada, “Optimal PID control of an autonomous vehicle using butterfly optimization algorithm BOA,” in Proceedings of the 4th international conference on big data and internet of things, 2019, pp. 1–5, doi: 10.1145/3372938.3372980.
25. S. Azam, F. Munir, and M. Jeon, “Dynamic Control System Design for Autonomous Car,” in VEHITS, 2020, pp. 456–463, doi: 10.5220/0009392904560463.
26. A. AbdElmoniem, A. Osama, M. Abdelaziz, and S. A. Maged, “A path-tracking algorithm using predictive Stanley lateral controller,” Int. J. Adv. Robot. Syst., vol. 17, no. 6, p. 1729881420974852, 2020, doi: 10.1177/1729881420974852.
27. F. A. Ma’ani and Y. Y. Nazaruddin, “Optimization of Longitudinal Control of an Autonomous Vehicle using Flower Pollination Algorithm based on Data-driven Approach,” IInternational J. Sustain. Transp. Technol., vol. 3, no. 2, pp. 58–65, 2020, doi: 10.31427/ijstt.2020.3.2.4.
28. S. Chen and H. Chen, “MPC-based path tracking with PID speed control for autonomous vehicles,” in IOP Conference Series: Materials Science and Engineering, 2020, vol. 892, no. 1, p. 012034, doi: 10.1088/1757-899X/892/1/012034.
29. R. Gutiérrez et al., “A waypoint tracking controller for autonomous road vehicles using ROS framework,” Sensors, vol. 20, no. 14, p. 4062, 2020, doi: 10.3390/s20144062.
30. J. Filip, “Trajectory tracking for autonomous vehicles,” MSc thesis,Czech Technical University,Electrical Engineering,Control Engineering, 2018.
31. W. Jiahao, “Autonomous driving car model and its control,” MSc thesis,Automotive Engineering.
32. T. D. Gillespie, Fundamentals of vehicle dynamics, 1st Edition. Society of Automotive

Engineers, 1992.

33. R. Rajamani, “Longitudinal Vehicle Dynamics,” in Vehicle Dynamics and Control, 2nd Editio., Mechanical Engineering Series,Springer, Boston, MA, 2012, pp. 87–111.

34. F. Ahmad, S. A. Mazlan, H. Zamzuri, H. Jamaluddin, K. Hudha, and M. Short, “Modeling and Validation of the Vehicle Longitudinal Model,” Int. J. Automot. Mech. Eng., vol. 10, pp. 2042–2056, 2014, doi: http://dx.doi.org/10.15282/ijame.10.2014.21.0172.

35. P. Shakour, A. Ordys, and D. S. Laila, “Longitudinal vehicle dynamics using Simulink / Matlab,” in Control 2010, UKACC International Conference, Designing of the adaptive cruise control system-switching controller, 2010, pp. 955–960, doi: 10.1049/ic.2010.0410.

36. R. Di Martino, “Modelling and simulation of the dynamic behaviour of the automobile,” Université de Haute Alsace-Mulhouse, 2005.

37. M. Kissai, B. Monsuez, A. Tapus, and D. Martinez, “A new linear tire model with varying parameters,” in 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE), 2017, pp. 108–115.

38. R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to autonomous mobile robots, 2nd Edition. MIT press, 2011.

39. N. Correll, "Introduction to Autonomous Robots.", 1st Edition., vol. 1. Magellan Scientific, 2016.

40. A. Patnaik et al., “Design and Implementation of Path Trackers for Ackermann Drive based Vehicles,” 2020.

41. D.C. Herráez, “Self-Driving Car Autonomous System Overview,” MSc thesis,de Navarra University,Industrial Electronics Engineering, 2020.

42. S.Y. Syed Hussien, H. I. Jaafar, R. Ghazali, and N. R. Abdul Razif, “The effects of auto-tuned method in PID and PD control scheme for gantry crane system,” Int. J. Soft Comput. Eng., vol. 4, no. 6, pp. 121–125, 2015.

43. M. G. H. Omran, “Particle Swarm Optimization Methods for Pattern Recognition and Image Processing,” PhD thesis, University of Pretoria,Engineering,Built Environment and Information Technology, 2006.

44. M. A. Panduro, C. A. Brizuela, L. I. Balderas, and D. A. Acosta, “A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays,” Prog. Electromagn. Res. B, vol. 13, pp. 171–186, 2009.

45. K. Chandrasekar and N. V. Ramana, “Performance comparison of GA, DE, PSO and SA approaches in enhancement of total transfer capability using FACTS devices,” J. Electr. Eng. Technol., vol. 7, no. 4, pp. 493–500, 2012, doi: 10.5370/JEET.2012.7.4.493.

46. S. binti Samsuddin, M. S. Othman, and L. M. Yusuf, “A review of single and population-based metaheuristic algorithms solving multi depot vehicle routing problem,” Int. J. Softw. Eng. Comput., vol. 4, no. 2, pp. 80–93, 2018, doi: 10.15282/ijsecs.4.2.2018.6.0050.

47. C. Blum, “Ant colony optimization : Introduction and recent trends,” Phys. Life Rev., vol. 2, no. 4, pp. 353–373, 2005, doi: 10.1016/j.plrev.2005.10.001.

48. M. Dorigo, M. Birattari, and T. St, “Ant Colony Optimization,” IEEE computational intelligence magazine, vol. 1, no. 4, pp. 28–39, 2006.

49. M. Dorigo and G. Di Caro, “Ant Colony Optimization : A New Meta-Heuristic,” Proc. 1999 Congr. Evol. Comput. (Cat. No. 99TH8406), vol. 2, pp. 1470–1477, 1999.

50. M. A. Sen and M. Kalyoncu, “Grey Wolf Optimizer Based Tuning of a Hybrid LQR-PID Controller for Foot Trajectory Control of a Quadruped Robot,” Gazi Univ. J. Sci., vol. 32, no. 2, pp. 674–684, 2019.

51. K. D. Abeyrathna and C. Jeenanunta, “Escape Local Minima with Improved Particle Swarm Optimization Algorithm,” Nor. IKT-konferanse Forsk. ogutdanning, 2020, doi: 10.13140/RG.2.2.17153.28009.

52. H. Yu, Y. Gao, and J. Wang, “A Multiobjective Particle Swarm Optimization Algorithm Based on Competition Mechanism and Gaussian Variation,” Complexity, p. 23, 2020, [Online]. Available: https://doi.org/10.1155/2020/5980504.

53. N. Patnana, S. Pattnaik, and V. P. Singh, “Salp Swarm Optimization Based PID Controller Tuning for Doha Reverse Osmosis Desalination Plant,” Int. J. Pure Appl. Math., vol. 119, no. 12, pp. 12707–12720, 2018, Accessed: May 30, 2021. [Online]. Available: http://www.ijpam.eu.

54. H. Faris, S. Mirjalili, I. Aljarah, M. Mafarja, and A. A. Heidari, “Salp Swarm Algorithm: Theory, Literature Review, and Application in Extreme Learning Machines,” Nature-inspired Optim., pp. 185–199, 2020, doi: 10.1007/978-3-030-12127-3.

55. S. Kassaymeh et al., “Salp Swarm Optimizer for Modeling the Software Fault Prediction Problem,” J. King Saud Univ. Inf. Sci., 2021, doi: 10.1016/j.jksuci.2021.01.015.

56. R. Abbassi, A. Abbassi, A. A. Heidari, and S. Mirjalili, “An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models,” Energy Convers. Manag., vol. 179, pp. 362–372, 2019, doi: 10.1016/j.enconman.2018.10.069.

57. F. Tian, H. Wei, X. Li, M. Lv, and P. Wang, “An improved salp optimization algorithm inspired by quantum computing,” in Journal of Physics: Conference Series, 2020, vol. 1570, no. 1, p. 012016, doi: 10.1088/1742-6596/1570/1/012016.

58. A. S. Assiri, “On the performance improvement of butterfly optimization approaches for global optimization and Feature Selection,” PLoS One, vol. 16, no. 1, p. e0242612, 2021, doi: 10.1371/journal.pone.0242612.

59. K. G. Abdulhussein, N. M. Yasin, and I. J. Hasan, “Comparison between butterfly optimization algorithm and particle swarm optimization for tuning cascade PID control system of PMDC motor,” Int. J. Power Electron. Drive Syst., vol. 2088, p. 8694, 2021, doi: 10.11591/ijpeds.v12.i2.pp736-744.

60. PYogananda, L. R. Aravindbabu, and A. A. Giri, “Oppositional Butterfly Optimization Algorithm with Multilayer Perceptron for Medical Data Classification,” Turkish J. Comput. Math. Educ., vol. 12, no. 10, pp. 2721–2731, 2021.

61. I. A. Khan, A. S. Alghamdi, T. A. Jumani, A. Alamgir, A. B. Awan, and A. Khidrani, “Salp Swarm Optimization Algorithm-Based Fractional Order PID Controller for Dynamic Response and Stability Enhancement of an Automatic Voltage Regulator System,” Electronics, vol. 8, no. 12, p. 1472, 2019, doi: 10.3390/electronics8121472.

62. A. Mirzal, S. Yoshii, and M. Furukawa, “PID Parameters Optimization by Using Genetic Algorithm,” arXiv Prepr. arXiv1204.0885, 2012.

Appendix

Table (A.1): The parameters of longitudinal dynamic vehicle model.

Symbol	Description	Value
$a_0, a_1,$ and a_2	Constants used in vehicle engine second order polynomial equation	400, 0.1, and - 0.0002
GR	Gear ratio	0.35
r_{eff}	Effective radius	0.3 m
J_e	Inertia of engine	10 kg.m^2
m	Mass	2000 g
g	Gravitational acceleration	9.81 m/s^2
C_a and C_r	Drag and rolling coefficients	1.36, and 0.01
C_s	Longitudinal stiffness of the tire	10000
F_{max}	Maximum tire force	10000 N
T_s	Sample time	0.1 s
ϕ	Incline angle	0 rad
Limitation of the throttle	-1 to 1	Percentage

Table (A.2): The parameters of the kinematic bicycle model.

Description	Value	Unit
Wheelbase	2	m
Distance from the bicycle center to the center of the rear wheel	1.2	m
Limitation of the steering angle	-1 to 1	rad
Limitation of the steering rate	-1.22 to 1.22	rad/s
Limitation of the throttle	-1 to 1	percentage

الخلاصة

يتطلب الروبوت المتنقل المستقل مثل المركبة المستقلة معاييرعالية الدقة لتتبع ملفات تعريف السرعة المطلوبة والمسار المحدد مسبقًا. بعض التحديات في تطوير أنظمة التمثيل الذاتي، مثل بناء نموذج رياضي، تصميم أنظمة التحكم الطولية والجانبية، وتقنية التحسين لا تزال مواضيع مهمة.

تهتم هذه الأطروحة ببناء نموذج السيارة والدراجة الذي يحكم حركة السيارة المستقلة من حيث نموذج السيارة الديناميكي الطولي والحركة الطولية والجانبية لنموذج الدراجة الحركية. لضبط حركة السيارة ذاتية القيادة ، يتم اقتراح استراتيجيات تحكم مختلفة.

تم اقتراح جهاز التحكم النسبي المتكامل المشتق (PID) للتعامل مع مشغلات الخانق / الفرامل لنموذج السيارة الديناميكي الطولي لتتبع السرعات المرجعية المختلفة. علاوة على ذلك ، تم اقتراح وحدة التحكم التناسبية (P) للحركة الطولية بينما اقترح أربع وحدات تحكم للحركة الجانبية لتوجيه نموذج الدراجات الحركية بسرعات مختلفة بشكل صحيح هي: المشتق المتكامل النسبي المعدل لتعقب المسار (PTMPID) ، و PID ، و Stanley ، ووحدات تحكم Stanley المعدلة (MS). تم إجراء المقارنة بين أجهزة التحكم الجانبية لمعرفة الأفضل.

تم تحسين معلمات وحدات التحكم بطريقتين جديدتين للتحسين هما خوارزميات تحسين Hybrid Salp Swarm وال Butterfly (HSSABOA1 و HSSABOA2). ومع ذلك ، للتحقق من كفاءة أداء الخوارزميات المقترحة ، تمت مقارنتها بالخوارزميات الأساسية: خوارزمية تحسين الفراشة (BOA) و خوارزمية تحسين Salp (SSA) بالإضافة إلى الخوارزميات الشائعة الاستخدام ، وهي الخوارزمية الجينية (GA) ، وخوارزميات تحسين مستعمرة النمل (ACO) ، و تحسين سرب الجسيمات (PSO). في تقنيات التحسين هذه ، يتم استخدام (IAE) و (RMSE) كدوال هدف لتقليل أخطاء تتبع السرعة والتوجيه على التوالي.

وجدت نتائج المحاكاة أن المتحكم PID القائم على HSSABOA1 لنموذج السيارة الديناميكي لديه الحل الأفضل لتقليل خطأ السرعة بنسبة تحسن (٨.٦٠٨٨٪) , (٥.٠٢٢٦٪) , (٢.٥٠٧٤٪) , (٠.٠٣٠٦٪) , و (٠.٢٢٩٥٪) من PID القائم على (GA و ACO و PSO و BOA و SSA) بالترتيب. علاوة على ذلك ،

فقد تتبع أيضًا ملفات تعريف السرعة المختلفة بنجاح. في المقابل ، فإن PTMPID القائم على HSSABOA1 لديه أداء أفضل ، في تقليل الخطأ الجانبي إلى جانب زيادة النسبة المئوية بنسبة (١٠.٥٢٣٪) , (١٩.٤٥٦٪) , (٨٣.٢٧٦٪) , (١٨.٢٦٣٪) , و (٩٤.٠٠٥٪) من PTMPID-SSA PTMPID-BOA, Stanley-HSSABOA٢ و Stanley-HSSABOA٢ المعدل و PID-HSSABOA1 على التوالي. بالإضافة إلى ذلك ، تمكنت من تتبع مناورة الطريق بسرعات طولية مختلفة دون تذبذب كبير.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة كربلاء / كلية الهندسة

قسم الهندسة الكهربائية والالكترونية

تحسين تعقب المسار لنظام روبوت متنقل ذاتي الحركة

رسالة
مقدمة الى قسم الهندسة الكهربائية والالكترونية , جامعة كربلاء
وهي جزء من متطلبات نيل درجة الماجستير(MSC)
في علوم الهندسة الكهربائية

من قبل

غيداء هادي صالح

بكالوريوس هندسة الكهربائية والالكترونية / جامعة كربلاء (٢٠١٤)

بإشراف

أ.م.د. أحمد عبد الهادي أحمد

أ.م.د. حيدر جليل كامل

١٤٤٣هـ **٢٠٢٢ م**