Design and Development of Advanced Control Techniques for an Unmanned Ground Vehicle

by

Amr Mohamed

A thesis presented in partial fulfillment

Of the requirements for the degree of

Doctor of Philosophy

In

Electrical Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

November 2018

THESIS EXAMINATION INFORMATION

Submitted by: **Amr Mohamed**

Degree Name in Program Name

Thesis title
Design and Development of Advanced Control Techniques for an Unmanned Ground Vehicle

An oral defense of this thesis took place on November 6, 2018 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Vijay Sood
Research Supervisor	Dr. Jing Ren
Research Co-supervisor	Dr. Moustafa El-Gindy
Examining Committee Member	Dr. Yuping He
Examining Committee Member	Dr. Haoxiang Lang
Examining Committee Member	Dr. Mohamed Youssef
University Examiner	Dr. Ying Wang
External Examiner	Dr. Mehrdad R. Kermani, University of Western Ontario

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Recent years have seen considerable progress towards the goal of autonomous and unmanned ground vehicles which became essential for conducting military operations. These autonomous vehicles have the capability to operate and react to their environments without external control. Autonomous multi-wheeled combat vehicles are crucial for military applications which offer numerous leverages on modern battlefields. Applying autonomy features to such vehicles significantly increases its combat capabilities and expands its applications to work-day and night for risky missions compared with traditional manned ground vehicles. However, it is associated with some challenges because of their large dimension, heavy weight, and complex geometry. Therefore, the development of autonomous combat vehicles has become a cutting-edge research topic in robotics and automotive engineering.

This thesis focuses on the control issues related to applying autonomous features for the multi-wheeled combat vehicles due to their significant influence especially when navigating in the presence of obstacles. The primary concern of path planning is to compute collision-free paths. Another equally important issue is to compute a realizable path and, if possible, achieving an optimal path bringing the vehicle to the final position. For these purposes, the developed methodology considers the combination between the optimal control theory using Pontryagin's Minimum Principle (PMP) and Artificial Potential Filed (APF). In addition, a four-axle bicycle model of the actual multi-wheeled combat vehicle considering the vehicle body lateral and yaw dynamics is developed.

To generate the vehicle optimal path in real time, an Artificial Neural Network (ANN) model is proposed. The introduced ANN model allows the vehicle to carry out an autonomous navigation in real time with maintaining the path optimality by considering the vehicle parameters in terms of yaw rate, lateral velocity, heading angle and steering angle. Subsequently, a comparative study and performance analysis of the developed optimal path algorithm using PMP with Dynamic Programming (DP) method was carried out in order to guarantee the global optimum solution.

Determining the accurate vehicle position offers sufficient capabilities which increase the autonomy and safety features, especially in case of off-road locomotion. In this regard, a hybrid framework for positioning technique based on the integration of GPS/INS for combat vehicles is developed. The developed algorithm is able to provide an accurate and reliable vehicle positioning information, even if the number of visible satellites is less than four, due to the harsh vehicle operation environments.

In this work, a scaled multi-wheeled combat vehicle model was developed using system identification methodology. Different system identification methods are considered and applied to solve and identify this problem. An advanced control system in terms of fuzzy logic, robust, and PID control systems are designed. In addition, the Processor-In-the-Loop co-simulation (PIL) is considered, which permits and achieves a more realistic situation where the developed control algorithms running on a dedicated processor. The performance and effectiveness of the developed controllers are evaluated for vehicle heading angle tracking using different predefined heading angles. Furthermore, a comparative evaluation to assess the feasibility of the developed control algorithms is discussed. Finally, it should be stated that this work offers the first attempt in the open literature to control the scaled multi-wheeled combat vehicle using different advanced control techniques such as, fuzzy logic, H_∞.

Acknowledgements

I would like to express my special thanks and gratitude to my supervisors, Dr. Jing Ren, and Dr. Moustafa El-Gindy, for their great support and encouragement to conduct this research. Their extensive knowledge and significant experience have not only helped me throughout my research but has also allowed me to overcome any obstacles throughout this process. I would like appreciatively to acknowledge the Egyptian Ministry of Defense for the financial support extended to this research project. I would also like to thank Dr. Alhossein Mostafa Sharaf, the chair of department of automotive Engineering, military technical college for his valuable technical advice. I would like to express my genuine appreciation to my parents Mr. Sayed Selim and Mrs. Mona Salem for their love, guidance, and inspiration throughout this journey. I would like to express my deepest gratitude to my wife Mona, who allowed me to focus on my research through her love and looked after our kids Abdelrahman and Kenda.

Contents

Nomenclature

a	The distance from the vehicle CG to the front axle
a_x, a_y	The centre position of the attractor
b	The distance from the vehicle CG to the rear axle
$C_{\alpha f}$	The front tyre cornering stiffness
$C_{\alpha r}$	The rear tyre cornering stiffness
d_{xo} , d_{yo}	The initial vehicle positions in longitudinal and lateral directions
d_{xf} , d_{yf}	The final vehicle positions in longitudinal and lateral directions
D^{-1}	The transformation matrix from rectangular coordinates into curvilinear coordinates in the Earth-Centered Earth-Fixed Frame
F_{yf}	The front tyre force
F_{yr}	The rear tyre force
$\boldsymbol{F}^l$	The skew-symmetric matrix corresponding to the specific force
$F(t)$	The dynamic matrix
f^b	The specific force measured by the accelerometers in the body-frame
G_{k-1}	The noise distribution matrix
$G(t)$	The noise distribution matrix
g	The normal component of gravity
H_k	The measurement design matrix associated with the sensor
I	The identity matrix
I_{zz}	The moment of inertia
k	The measurement epoch
K_k	The Kalman gain
m	The vehicle mass
r	The yaw rate
r_x, r_y	The centre position of the repulsor
R_k	The process noise covariance
R	The mean radius of the Earth
R_N	The normal radius of curvature for the east-west direction

R_M	The meridian radius of curvature for the north-south direction
R_b^l	The transformation matrix from the body frame and the LLF
R_b^l	The transformation matrix from the body-frame to the inertial-frame
Δt	The sampling interval
$\hat{V}^l$	The velocity vector in the local level frame
U	The forward velocity
V	The lateral velocity
x	The longitudinal coordinates of the vehicle body motion
$\boldsymbol{X}(t)$	The state vector
$\hat{X}_k$	The estimated states from the epoch $k-1$ to k
$W(t)$	The random forcing functions
W_{k-1}	The process noise vector
$\mathrm{w(t)}$	The white Gaussian noise
y	The lateral coordinates of the vehicle body motion
Z_k	The measurement vector of the sensor output

Greek Symbols

δ	The steering angle
α_f	The front slip angle
α_r	The rear slip angle
θ	The angle between vehicle x-axis and the direction of travel
Ψ	The heading angle
Ψ_o	The initial heading angle
Ψ_f	The final heading angle
$\varphi_{k,k-1}$	The state transition matrix
$\delta_N \delta_E \delta_U$	The north, east and up positions respectively
$\delta_{VN}\ \delta_{VE}\ \delta_{VU}$	The north, east and up velocities respectively
$\delta_{Offset}\ \ \delta_{drift}$	The receiver clock offset and drift
ξ_k	The white gaussian observation noise for the sensor
$\delta_\varphi \delta_\lambda \delta_h$	The three position states
$\delta_p \delta_r \delta_A$	The three attitude states in east, north and up directions
$\delta_{f_x} \delta_{f_y} \delta_{f_z}$	The three accelerometer biases
$\delta_{\omega_x} \delta_{\omega_y} \delta_{\omega_z}$	The three gyro drifts
δ_r^l	The position error
ω_{el}^l	The effect of the change of the orientation of the LLF w.r.t the earth
Ω_{ie}^l	The skew-symmetric matrices corresponding to the rotation of the Earth about its spin axis
Ω_{el}^l	The change of orientation of the local-level frame with respect to the Earth
ω_{ie}^l	The angular velocity vector in the LLF
f_e, f_n, f_u	The body acceleration in east, north and up directions
δg^l	The error in normal gravity due primarily to the error in the altitude
ε^l	The attitude errors $[\delta p, \delta r, \delta A]^T$
ω_{il}^l	The angular velocity of the LLF w.r.t the inertial frame

ω_{ib}^{b}	The angular velocity of the body frame to the LLF
Ω_{il}^{l}	The skew-symmetric matrix corresponding to the vector ω_{il}^{l}
β	The reciprocal of correlation time
ω	The zero mean Gaussian noise with unit variance
σ^2	The variance of white noise
$\beta_{fx}, \beta_{fy}, \beta_{fz}$	The reciprocals of the correlation times associated with the autocorrelation sequence of $\delta_{fx}, \delta_{fy}, \delta_{fz}$
$\sigma_{fx}^2, \sigma_{fy}^2, \sigma_{fz}^2$	The variances associated with the accelerometer errors
$\beta_{\omega x}, \beta_{\omega y}, \beta_{\omega z}$	The reciprocals of the correlation times associated with the autocorrelation sequence of $\delta_{\omega x}, \delta_{\omega y}, \delta_{\omega z}$
$\sigma_{\omega x}^2, \sigma_{\omega y}^2, \sigma_{\omega z}^2$	The variances associated with the gyroscope errors

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Research Overview

Autonomous multi-wheeled combat vehicles have many potential applications, especially in the military area, due to their capability to operate, react, and navigate without human intervention. The knowledge of autonomous vehicle behaviors under control commands plays an important role in improving their autonomy. The vehicles offer advantages over more traditional battle tanks and four-wheeled vehicles. They are able to maneuver at relatively high speeds in road and off-road applications. Unlike traditional four-wheeled armoured vehicles, additional wheels allow for a uniform distribution of vehicle weight across the entire wheelbase offering mobility advantages in softer soils.

Nowadays, many researchers are interested in applying advanced control strategies for control and improve the performance of the autonomous vehicle. On the other hand, a multi-wheeled combat vehicle is considered as one of the most difficult types of vehicles in applying autonomy due to their large dimensions, heavy weight, complex geometry and dynamics. Combat vehicles are mainly essential for military applications. Therefore, building autonomous multi-wheeled combat vehicles has drawn dramatic attention as the autonomous vehicles are one of the boosting research topics in both robotics and automotive engineering. The expected outcomes of the autonomous multi-wheeled combat vehicles are set to increase the combat capability and protect soldiers in a full spectrum of battlefield scenarios.

Path planning is one of the challenges in building an autonomous multi-wheeled combat vehicle. It is often necessary to deal with unknown workspaces such as road conditions, uncertainty in the vehicle states, and limited knowledge of the surrounding environment. However, the main challenge is to provide accurate and reliable vehicle positioning information due to the harsh vehicle operation environments. Consequently, the integration

of the navigation sensors, such as Global Positioning System (GPS) and Inertial Navigation System (INS), attempts to solve this problem and increases the positioning accuracy.

1.2 Scope and Objectives

1.2.1 Scope

This thesis is focused on studying the benefits of advanced control systems for developing and enhancing both the multi-wheeled combat vehicle and the scaled vehicle, which is electrically driven with eight independently controlled wheel drive motors. The effects of the imposed obstacles on the vehicle path planning and its maneuverability between two points are to be investigated. The mathematical model of the scaled vehicle was identified using system identification techniques, where various control algorithms are applied to control the vehicle heading angle. In addition, developing an accurate positioning technique is based on fusing the data of the GPS and INS using the Kalman filter (KF).

1.2.2 Objectives

The aim of the current research is to provide insight into an autonomous multi-wheeled combat vehicle path planning and heading angle control. Subsequently, the benefits of applying these control systems to an electrically driven 8 x 8 scaled combat vehicle have been investigated. In addition, a sensor fusion technique is applied to improve the navigation solution of the vehicle. Various simulations are conducted with a validated multi-wheeled combat vehicle TruckSim model and MATLAB/Simulink to check the effectiveness and the feasibility of the developed algorithms. A set of well-defined objectives has been accomplished and is outlined as follows:

- Developing a multi-wheeled combat vehicle dynamic model based on real combat vehicle characteristics in terms of dimensions, and weights for introducing an optimal path planning algorithm to generate an optimal path for the vehicle to navigate between two points while avoiding the imposed obstacles. The algorithm is based on the combination of optimal control theory and artificial potential field.

- Proposing a real time path planning algorithm for the autonomous multi-wheeled combat vehicles navigation using Artificial Neural Network model (ANN). The ANN is trained offline using a back-propagation supervised learning algorithm. The training data are multiple optimal paths collected using the path planning algorithm in MATLAB using optimal control theory.

- Conducting a comparative study and performance analysis of the generated optimal paths using Dynamic Programming (DP) and Pontryagin's Minimum Principle (PMP). PMP and DP are two major branches of the optimal control theory. Up to now, there is a research gap in comparing the performance effectiveness of the mentioned techniques in the applications of navigating autonomous vehicles.

- Developing a hybrid positioning technique based on the integration of GPS/INS for autonomous navigation. This framework combined both loosely and tightly coupled Kalman filter algorithms in order to provide an accurate and reliable vehicle positioning information, even if the number of visible satellites is less than four, due to the harsh vehicle operation environments.

- Development and modeling of the remotely operated scaled multi-wheeled combat vehicle using system identification methodology for vehicle heading angle tracking. For this purpose, an experimental test will be carried out to record and analyze the vehicle input/output signals during a maneuver.

- Developing different control systems based on fuzzy logic, classical, and robust controllers in MATLAB/Simulink to enhance the scaled vehicle heading angle tracking performance. Afterwards, introducing a processor-in-the-loop co-simulation to permit and achieve a more realistic situation, where the control algorithms running on a dedicated processor.

- Investigating and Evaluating the heading angle tracking controllers of the scaled multi-wheeled combat vehicle in the presence of noise and disturbance, as well as for tracking different predefined heading angles.

CHAPTER 2

Literature Survey

2.1 Introduction

In recent years, Unmanned Ground Vehicles (UGVs) have drawn dramatic attention. The applications of UGVs range from using UGVs in collecting soil samples on Mars to being deployed in military missions. They are primarily either teleoperated by an operator or autonomous. In the case of teleoperated vehicles, an operator and high-bandwidth remote video are required to control and ensure that the vehicle is performing its task fittingly [1]. On the other hand, autonomous vehicle is not limited by these requirements. It is equipped with various sensors, hardware, and software systems to perform missions and navigate without human intervention. Consequently, reliability and robustness are two major demands for such vehicle in particular for the field of combat operations. Accordingly, the challenge is to develop an advanced control system to handle and control the vehicle subsystems at different conditions. In addition, motion planning needs to improve so that the vehicle can navigate autonomously by generating its own path and achieve the desired maneuver. Generally, UGV consist of the following components:

Sensors: UGVs are fitted with sensors in order to observe their surrounding environment. Consequently, they permit controlled movements especially in highly unpredictable environments, such as the battlefield.

Control: The robustness of autonomy and intelligence of the UGV mainly depends on the developed control systems algorithms. It has the range from classical control to more advanced and intelligent control methods such as robust control, adaptive control, and neural networks.

Communication: Communication is an essential part of Remote Operated Vehicle (ROV) used in military operations. The communications occur between humans and such vehicles and involve a human in the decision-making cycle during the vehicle operation.

System integration: The choice of system level formation, configuration, sensors and other components provides momentous interaction within a robotic system. Well-designed UGV systems are self-reliant and adaptable, which increasing the autonomy level.

In this chapter, the following sections critically analyze the most appropriate reported work as follows. Section 2.2 provides a survey of the recent work on advanced control techniques applied to UGVs, electric vehicles, and mobile robots. Section 2.3 introduces a survey of the recent work on path planning using different techniques both in static and in a dynamic environment. Section 2.4 summarizes state of the art of sensor fusion methods and present the most relevant studies Section 2.5 provides a survey of system identification methods and applications.

2.2 Advanced Control Techniques for UGV

This section discusses the related research on developing and improving the advanced control techniques for UGVs. There are a significant number of variations in the control technologies to approximate vehicle behaviors in response to control actions in relevant operating conditions. Some of these are listed below:

PEEIE et al. [1] enhanced the stability of Electric Vehicle (EV) with two in-wheel motors during turning and braking using direct yaw moment control (DYC). For this purpose, a linearized model has been used as a reference for yaw angular velocity and side slip angle. The system input is the steering angle, while the outputs are the yaw angular velocity and side slip angle. There are more conditions required in order to improve the vehicle stability. First, the vehicle velocity must be constant. Second, the rate of change of the lateral velocity must be small. Consequently, the stability of the EV is increased by satisfying these constraints.

Gasbaoui & Nasri [2] presented an intelligent adaptive fuzzy PI controller for improving the stability of a 4WD EV under different road condition. The vehicle's wheels are individually controlled by a direct torque control. The torque is not fixed due to the nonlinearity of the vehicle. For this purpose, a fuzzy logic controller is combined with the

PI controller in order to adjust their gains in response to the nonlinearity as shown in Figure 2-1. Consequently, the vehicle stability during different road conditions can be improved.

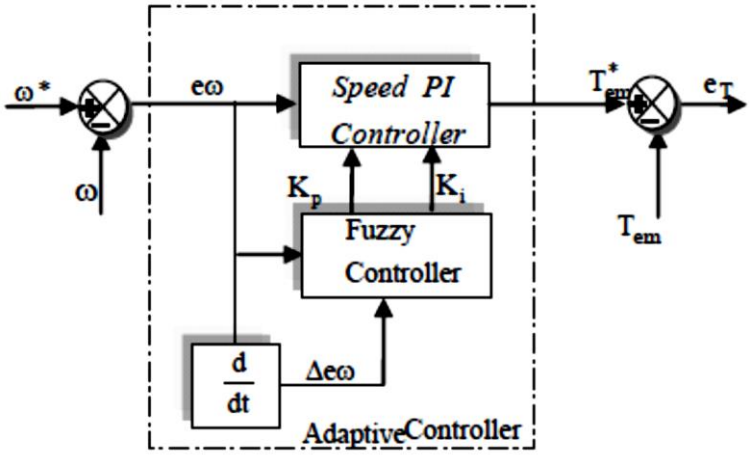


Figure 2-1 PI gains online tuning by fuzzy logic controller [2]

Z. Sun et al. [3] introduced a good path tracking control algorithm for UGV in the presence of uncertain dynamics. This algorithm using an adaptive control method based on a virtual current vector algorithm. The proposed controller has a simple structure and does not require high computational time. Furthermore, it can directly control the left and right driving motors. The computer simulation results showed that the proposed algorithm is sensitive and can maintain good path tracking.

Lei Yuan et al. [4] designed a slip controller for 5 Degree of Freedom (DOF) EV using a nonlinear model predictive controller (NMPC). The developed control algorithm has the capability to prevent the wheels from lock up during braking and spin-out or when accelerating on a low-friction coefficient road. It has been noticed that, the controller parameters are easy to be tuned based on the situation of braking or spinning, which has the capability to deal with nonlinearities. Figure 2-2 shows NMPC slip control block diagram. The controller was evaluated using an AMESim model connected to Simulink, which is tested under acceleration and braking maneuvers. The obtained results show that the algorithm can efficiently prevent the wheel from spinning during accelerating or locking.

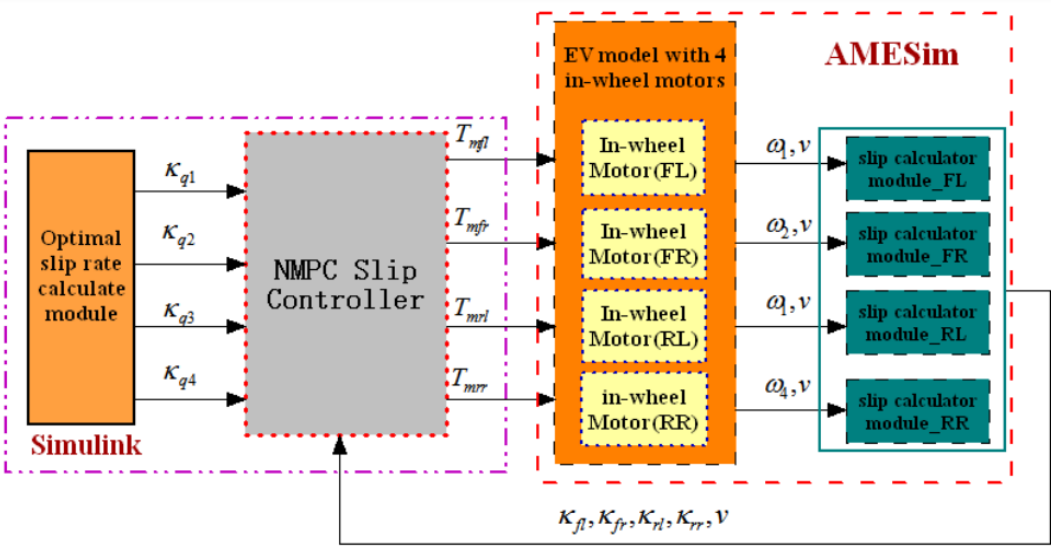


Figure 2-2 Block diagram of the NMPC slip control system [4]

Zhang et al. [5] developed a trajectory tracking controller for a 6x6 multi-wheel unmanned ground vehicle with 12-DOF that developed by Kanayama [6]. The vehicle dynamic constraints are considered in this work in order to ensure the smooth motion of the vehicle. The proposed controller comprised of two PD controllers in order to improve the velocity and yaw rate of the vehicle. Additionally, the tire model in the driving system is designed by using a Pacejka tire model (Magic formula). The proposed tracking algorithm is shown in Figure 2-3. The simulation results showed that the vehicle is able to track the desired path.

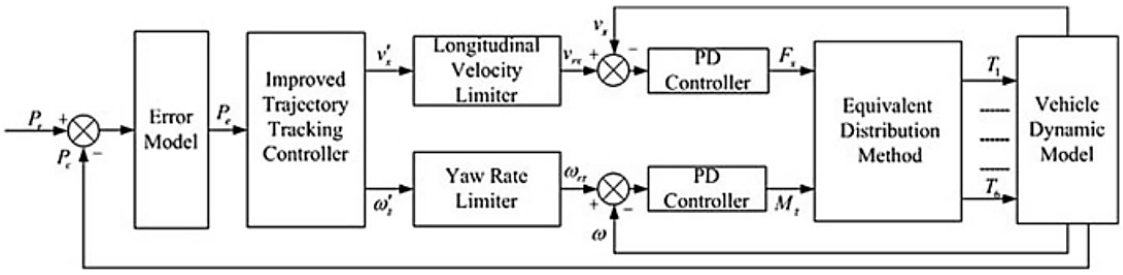


Figure 2-3 Trajectory tracking control algorithm [5]

Noor et al. [7] developed a teleoperated omnidirectional small UGV. The vehicle development is concerns in the movement of multidirectional UGV by using Mecanum

wheels and brushless DC electric motors. The block diagram of the proposed UGV is shown in Figure 2-4.

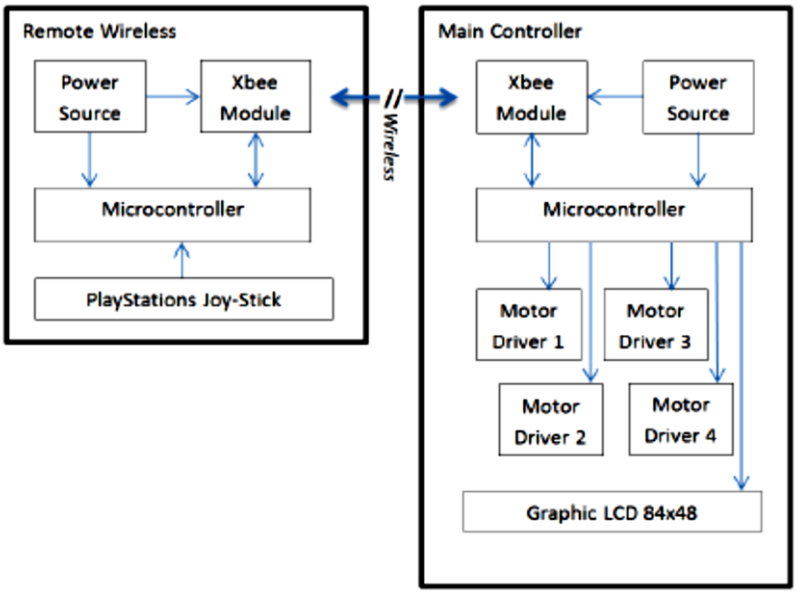


Figure 2-4 Block Diagram of UGV's Electronic System [7]

In addition, the movements of UGV are configured as shown in Figure 2-5, according to the following table:

Table 2-1 Movements configuration of UGV

Motion Direction	**Action**
Forward	All four wheels move forward at the same speed
Backward	All four wheels move backward at the same speed
Left slide	Front-right wheel and back-left wheel are moving backward
Backward	Front-left wheel and back-right wheel are moving forward
Right slide	Front-right wheel and back-left wheel are moving forward, front-left wheel and wheel back-right are move backward
Clockwise	All right wheels are moving forward, all left wheels are moving backward.
Counter-Clockwise	All right wheels are moving backward, all left wheels are moving forward

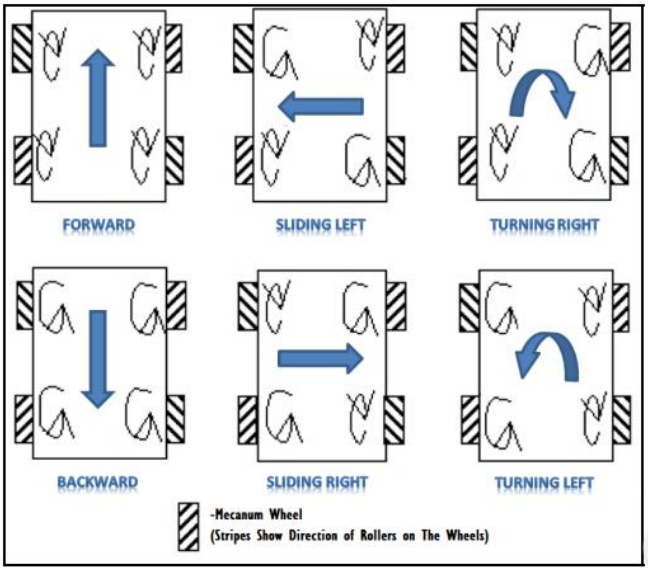


Figure 2-5 Movements of UGV [7]

J. Kim, et al. [8] designed a yaw rate controller for an independent in-wheel motor vehicle. The yaw rate control algorithm generates the desired direct yaw moment by adjusting the traction torque at each wheel, which is obtained from the sliding mode control. A CarSim model was employed for simulating the vehicle. Furthermore, a PID controller was developed in order to control the lateral acceleration error to not exceed the desired value. The yaw rate controller block diagram is shown in Figure 2-6. The controller performance was evaluated by the slalom test, which generates the desired direct yaw moment by adjusting the traction torque of each wheel.

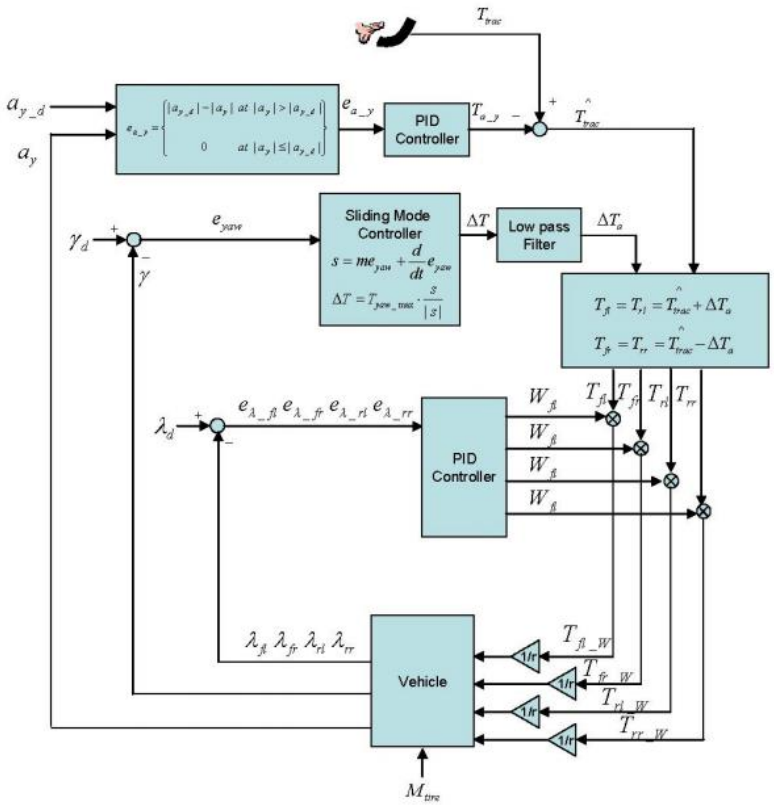


Figure 2-6 Yaw rate controller [8]

Matsumura S et al. [9] developed a driving system for EVs, where a test system using motors was considered in order to simulate the driving performance of the EV. In addition, a PID controller was developed to control the rotating speed of the motors. The parameters of the PID controller was adjusted and tuned using a neural network model. In this way, the performance of the speed controller was improved. The experimental results show that the neural network was able to reduce the error effectively, while the PI controller parameters are being tuned online.

Yin G et al. [10] developed an autonomous driving system for an EV of 7-DOF. The introduced system consists of two components: a simplified motion planning algorithm and a model predictive-control system. The motion planning algorithm computes the path according to the target position within a specified time. On the other hand, the model predictive controller was developed to control the front steering wheel individually in order to follow the desired path and satisfy the actuator physical constraints. Figure 2-7 shows

the control scheme of the autonomous vehicle. Simulation results using CarSim show that the proposed controllers achieved the requirements for the autonomous vehicle.

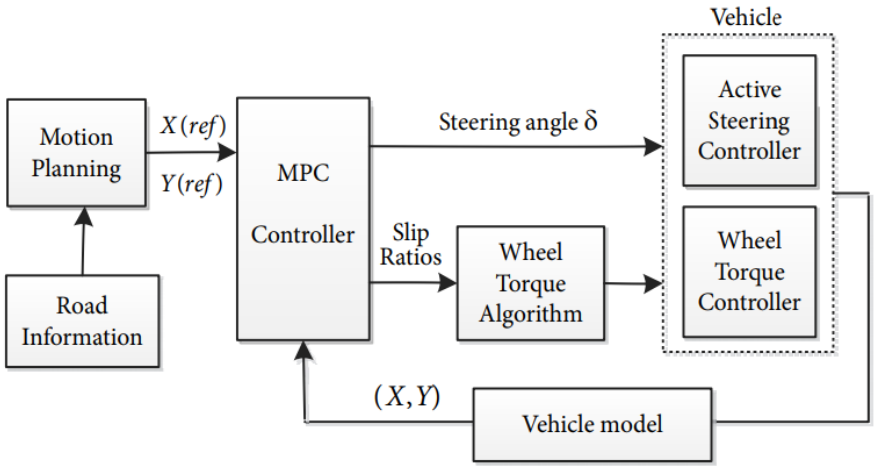


Figure 2-7 The control scheme for autonomous vehicle [10]

Sakai et al. [11] developed a motion controller for an EV with four independent in-wheel motors using DYC. The vehicle's lateral motion was controlled using a robust dynamic yaw-moment controller. This controller generates the yaw motion of torque differences between the right and left wheels. On the other hand, the problem of instability due to the slippery and low friction coefficient road still exists. For this purpose, a skid detection method was developed to increase the vehicle stability in different conditions. The proposed algorithm is shown in Figure 2-8. The experimental tests showed that the new skid detection system has the capability to detect the skidding wheel without any information on the chassis velocity.

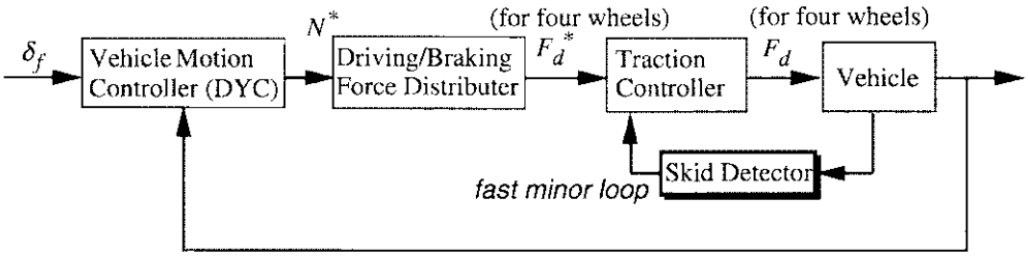


Figure 2-8 Schematic diagram of control system [11]

Yin G et al. [12] developed a μ-synthesis robust controller for a four-wheel independently actuated EV using a full-vehicle model in CarSim. This controller aims to maintain the vehicle stability and increase vehicle handling performance. The proposed controller has the capability to deal with unmodeled dynamics and uncertainties. The vehicle yaw and lateral motions were generated by using torque differences between the left and right wheels. The proposed controller was validated using CarSim, which has verified that it can achieve the required performance under various driving scenarios.

Elshazly et al. [13] introduced an LQR controller augmented with feedforward part in order to control a skid steering mobile robot. The main objective of this work was to design one controller for augmented dynamic drive model in a reduced order form, then enable the direct control of the motors. The proposed LQR controller has the advantage of simplicity in design and easy implementation compared with nonlinear controllers. Consequently, it can improve the mobile robot movement and increase the trajectory tracking accuracy using skid steering.

Lucet E et al. [14] developed a robust sliding mode controller for a four-wheel skid steering ground robot based on the dynamic model. This controller increases the robustness of the robot maneuver when moving over slippery road by controlling the mobile robot yaw angle and longitudinal velocity. It has the capability to handle the nonlinearity variable of the system while tracking the predefined path at a comparatively high speed. The evaluation of the designed controller is accomplished with real-life conditions using a four-wheel skid-steering robot, which proves to be effective on the slippery road.

A skid steer based autonomous driving controller for UGV was carried out by Kang, et al. [15]. The proposed algorithm consists of four parts as follows: the first is a speed controller, which maintains the desired driving speed. The second controller is devoted to maintaining the vehicle direction and follow the desired generated tracking path. The third is a longitudinal tire force distribution algorithm, and the last control algorithm is a wheel torque controller, which maintains the slip ratio under the limit by obtaining the torque command at each wheel. The control algorithm has shown good path tracking results.

A robust fuzzy logic controller for a four-wheel skid steer vehicle was introduced by Aslam et al. [16]. It has been noticed that the controller takes its action under high-speed cornering motion. In addition, it has the capability to balance the consequence of forces generated between the wheel and soil interaction. In addition, it eliminates the chattering phenomena (the phenomenon of oscillations with finite frequency and amplitude) encountered in the practical applications of sliding mode control.

An adaptive tracking algorithm for EV was developed by Kececi et al. [17]. This controller was developed to work when the road condition information is unavailable. Two adaptive control algorithms were developed to improve the stability of the vehicle, which controls the vehicle wheels' slippage without braking. In addition, it controls the vehicle when the type of road surface is unavailable. They have the capability to maintain the desired speed of the vehicle by applying more power to the drive wheel. Furthermore, they control the vehicle direction by changing the steering angle of the vehicle`s front wheels.

Al-Mayyahi et al. [18] introduced an Adaptive Neuro-Fuzzy Inference System (ANFIS) control algorithm for autonomous ground vehicles. This controller improved the autonomous navigation of the vehicle using four ANFIS algorithms. Two algorithms are presented in the paper to control the left and right angular velocities of the autonomous vehicle in order to maintain the planning path to the target. The data was collected from three distance sensors in terms of front, right, and left distance sensors. The other two algorithms are used to keep the vehicle on the optimal heading angle in order to avoid obstacles. They are taking the difference between the vehicle heading angle and the angle to the target point. The introduced controllers are evaluated using MATLAB, which shows that they are able to improve the vehicle`s navigation.

A robust adaptive torque control algorithm for skid steer wheel mobile robot was carried out by Mohammad pour et al. [19]. This algorithm improves the robot stability on the slippery road and in the situation of turning at high speed. Moreover, a fuzzy logic control algorithm was developed to maintain the stability of the internal dynamics of the system. The performance evaluation of the proposed control algorithm was carried out by modeling the vehicle on ADAMS, which shows the effectiveness of the algorithm.

A four-wheel drive control algorithm based on inverse longitudinal dynamics in straight line motion for UGV are introduced by Salama et al. [20]. It is indirectly provided the required torque for each wheel to overcome wheel load torque produced from the stochastic terrain. The proposed algorithm has the capability to provides a wheel with both the specified/required angular velocity and rolling radius. The developed algorithm is evaluated with the disturbance and the changes in stochastic terrain properties. The results show the robustness of the proposed algorithm.

A motion and stability control algorithm for a four-wheel drive skid steer robot was developed by Tu et al. [21]. For this purpose, a ratio control algorithm was introduced in order to keep the ratio of the vehicle lateral and angular velocities constant. Additionally, a high order deferential feedback controller conjoining with the ratio controller in order to enhance the vehicle motion. The evaluation has been performed under different scenarios using different levels of road friction, where the results showed that the system was robust.

An obstacle avoidance system using a laser scanner for perception was introduced by Jiménez et al. [22]. The proposed algorithm consists of two systems; first, a sensor system which used in order to discover the surrounding environment. Second, a control action system which used in order to control the steering and the speed of the vehicle based on the detected information. This system was developed using a fuzzy logic controller. Based on the obtained information from the sensor, there are two control actions, which are moving or braking the vehicle. In order to decide the safer action for the vehicle, it should take into account the required time for collision and avoidance by comparing them.

A robust motion control of a four-wheel drive skid steer vehicles was proposed by Arslan et al. [23]. The proposed methodology consists of two controllers; first, kinematic path-tracking controller, which is based on the vector field orientation technique. Second, a dynamic velocity control algorithm, this is based on sliding mode approach. In addition, a PID controller was developed in order to overcome the performance limitation. The evaluation has been done using Simulink/ADAMS, which showed the robustness and the stability of the developed motion control algorithm.

Sahoo et al. [24] designed and implemented a heading angle tracking algorithm for a UGV. The proposed tracking algorithm has been implemented using a two-degree-of freedom vehicle model. The closed-loop control for the steering motors and the heading angle are combined in order to achieve more tracking accuracy. Consequently, the inner loop is a proportional controller to minimize the error signal of current rotation of the steering motor according to the steering angle input. The outer loop is a PD controller, which works mainly to reduce the error of the vehicle heading angle. The main contribution in this work is to incorporate the dynamics of the actuator as the response time in order to steer the front wheel in the same order as that of the heading angle dynamics of the vehicle.

Yasuno et al. [25] designed an autonomous navigation system for UGV (RV-SCOT2) in order to control its motion and avoid the obstacles in an outdoor environment. The proposed algorithm consists of three control modules. The first module is a self-tuning fuzzy path tracking control, which estimates a steering angle for reference path tracking. The second one is an obstacle avoidance controller, which estimates the steering angle in order to obtain the effective avoidance action. The last one is a two-degree-of-freedom speed control module. Consequently, the autonomous navigation performance can be obtained by solving the trade-off relationship between each module using the collision danger-degree.

Ullah et al. [26] introduced an automatic distance control algorithm to control and keep a constant distance between the wheeled mobile robot and the objects in order to avoid collision with them. The developed control algorithm was implemented in a wheeled robot to track a moving object, while maintaining the distance constant. The perception of the surrounding environment is obtained using range sensors. These sensors are fitted to the front side of the robot to provide the controller with the required data in order to take the right decision on moving forward or backward. Figure 2-9 shows the system block diagram, which is divided into three parts; perception sensors, controllers, and actuators (motors).

The proposed algorithm was verified by carrying out indoor/outdoor experiments using different objects. The experimental result shows that the robot was able to track and maintain a constant distance from the object at the same time.

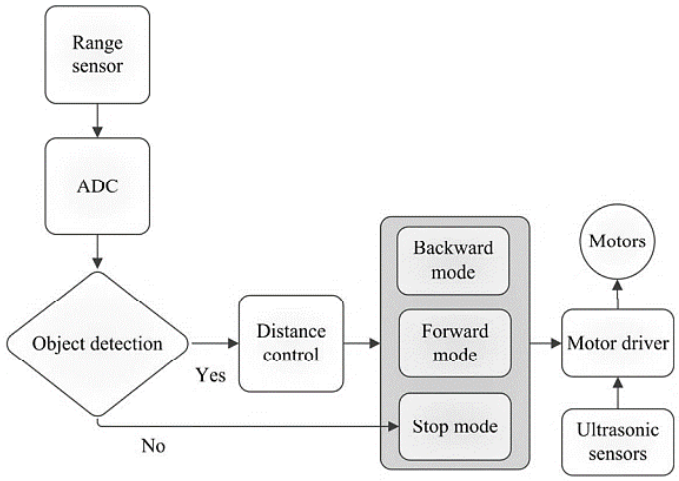


Figure 2-9 Tracking object block diagram [26]

Prayudhi et al. [27] designed a wall following control algorithm based on a differential wheel drive robot car using the Lyapunov method. The proposed controller enables the robot to maintain its orientation the same as the wall during navigation with a predefined distance from the wall. For this purpose, encoder sensors are fitted to the robot in order to control the speed of each wheel, which reflects the steering angle. In addition, a laser range finder was considered in order to detect the obstacles and their distances from the robot. The system states are the distance error of the robot position to the preferred distance from the wall and the orientation error between the robot and wall.

2.3 Motion Planning

This section discusses the work related to path generation and obstacle avoidance of the UGV. Path planning for such vehicles is considered an enormous challenge which aims to find the shortest or the optimal path for a given start and goal points. The problem becomes more complicated if it is required to generate the path across several obstacles, which is requiring to be combined with an obstacle avoidance controller in order to avoid the

obstacles. Generally, motion planning algorithms should satisfy the obstacle avoidance and other navigation requirements, such as smooth motion and shorter traveling time.

Autonomous vehicle motion planning can be divided into two main types, local and global path planning. In the case of global path planning, all required data about the environment is known in advance including locations of the obstacles. Therefore, the path planning algorithm is able to compute and generate the vehicle path from its current location to the target location before the vehicle starts navigation Thomaz et al. [28]. In the case of local path planning, the environment is completely unknown to the vehicle as investigated in A. Chakravarthy and D. Ghose [29]. In this case, it is necessary for the vehicle to collect the required data about the environment in real time in order to reach the goal point safely.

Mashadi B et al. [30] developed an optimal path planning algorithm for the autonomous vehicle using the linear bicycle model. The proposed algorithm generates the optimal path for a double lane change maneuver using the optimal control theory. It is able to generate the appropriate control law for steering an autonomous vehicle between two points. Figure 2-10 shows the generated optimal path after detecting the obstacle and then makes the lane change manoeuver. The proposed algorithm was validated in CarSim and MATLAB/ Simulink.

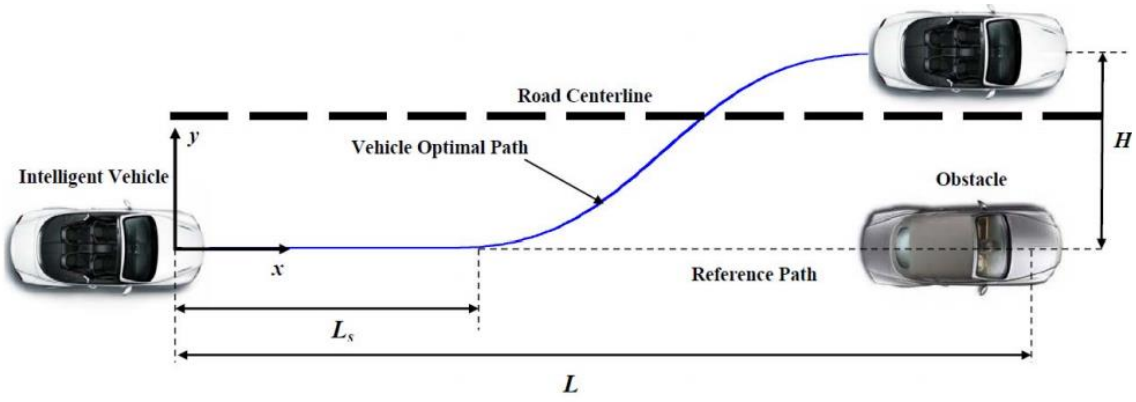


Figure 2-10 Schematic diagram of the lane-change maneuver [30]

Benamati et al. [31] solved the path planning problem for UGV in the static world using the flat potential field technique. This technique combines the efficiency of the potential

field approach with the simplicity of exact cell decomposition. The hierarchy of flat potential field algorithm is shown in Figure 2-11. The developed methodology has the capability to pass different situations while generating the optimal path. In addition, the potential field method does not introduce any local minima problems so that it is suitable for real time applications.

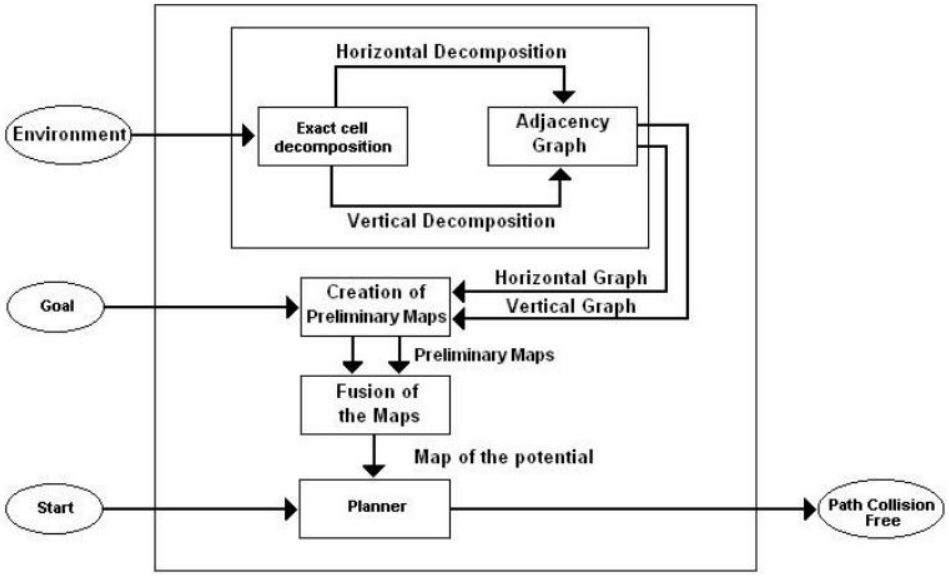


Figure 2-11 Hierarchy of Flat Potential Field algorithm [31]

Luh et al. [32] proposed dynamic robot motion planning using Potential Field Immune Network (PFIN). The proposed algorithm has the capability to deal with moving obstacles in addition to the fixed or moving target. The velocity obstacle method was utilized in order to determine the imminent collision obstacle. Subsequently, the PFIN was implemented to guide robots to avoid collision objects at any instant.

A new motion planning algorithm for a 6x6 wheel rover of 10-DOF over rough terrain was proposed by Conkur et al. [33]. A gradient function was derived to include both the tangential and gradient forces in order to ensure the safety of the generated path. Furthermore, the cost function was introduced and optimized using genetic algorithms to find the optimal path of the rover. The objective function was evaluated by considering the energy consumed, wheel slip, and traction, as well as the length of the path.

A motion planning algorithm using the potential field method for a mobile robot moving in a dynamic environment was introduced by S. GE et al. [34]. The proposed algorithm has the capability to deal with both moving obstacles and targets, where the problem of local minima was considered. The evaluation of the designed algorithm was performed using extensive computer simulations, which have validated the effectiveness of the algorithm.

An obstacle avoidance and navigation motion planning methodology for a mobile robot using fuzzy logic approach was carried out by Panagiotis et al. [35]. The authors combined fuzzy-rules of the proposed controller with the repulsive and attractive influences of the obstacle, as well as the goal using potential field method. Two fuzzy logic controllers were designed for velocity and steering control respectively. The performance of the proposed controllers was evaluated using MATLAB environment.

Jaradat et al. [36] proposed a fuzzy potential field path planning technique for an autonomous vehicle that dealing with the dynamic environment. Two fuzzy logic controllers with the Mamdani and Sugeno models were developed in order to deal with the attractive and repulsive forces of the target and obstacle respectively. In addition, an adaptive neuro fuzzy inference system was considered in order to implement the Sugeno model and improve functionality via the knowledge learning. The system was verified using different scenarios, which showed its effectiveness.

Tarokh [37] introduced a motion planning hybrid intelligent algorithm for an articulated wheeled robot moving in rough terrain. This algorithm combines both the global and the local path planning algorithms to generate the path between the rough terrains and fetch the data from the sensors fitted to the robot. The proposed methodology consists of a fuzzy logic-based algorithm and a two-stage genetic algorithm. The algorithm has the capability to produce a new path in response to environmental changes. The main contribution of this work is the fuzzy logic adaptation algorithm, which enables the adjustment of the probabilities of the genetic operators based on the terrain changes.

Sezer et al. [38] introduced a new velocity planning method used for autonomous vehicle navigation and obstacle avoidance as shown in Figure 2-12. A fuzzy logic algorithm was proposed for online speed planning which consists of two cascades connected Mamdani

approaches. In addition, an artificial potential field approach was introduced for obstacle avoidance. The developed system has the feature of work with any obstacle avoidance system.

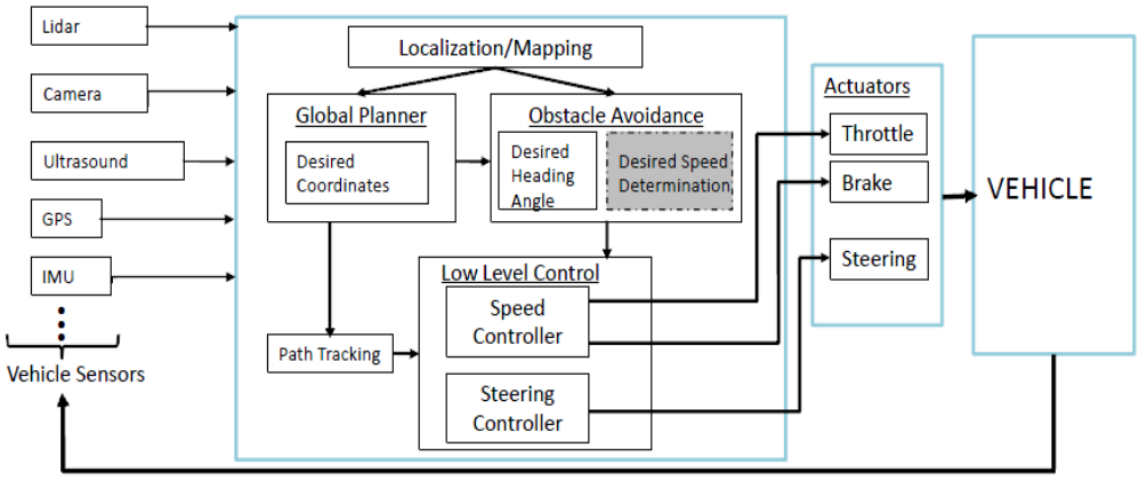


Figure 2-12 Desired Speed Determination in a UGV [38]

Lu Yin and Yixin Yin [39] developed a new artificial potential field methodology for autonomous mobile robot motion planning in dynamic environments. The main idea of this algorithm is to define the attractive potential field function according to the position, velocity, and acceleration of the mobile robot and the goal point. A similar function is defined for a repulsive potential field based on the distance between the robot and the obstacles. This algorithm can improve the robot capability to move on the right path, avoid obstacles, and maintain a suitable velocity according to the generated forces.

2.4 Sensor Fusion

Data fusion is the process of integrating multiple data from various sensors and related information in order to enhance the accuracy of the obtained data that cannot be achieved using a single sensor. Data fusion first appeared in the 1960s, as mathematical models for data manipulation, then it was implemented in the 1970s in the fields of robotics and defense. This section will summarize the state of art of sensor fusion methods and present the most relevant studies. Most of the state estimation methods are based on control theory and employ the laws of probability to compute a vector state from a vector measurement

or a stream of vector measurements. The estimation methods considered in this literature are as follows 1) maximum likelihood, 2) Kalman filter methods, and 3) particle filter method.

2.4.1 Maximum Likelihood method

The Maximum Likelihood (ML) technique is an estimation method based on the probabilistic theory. ML is applicable when the state variable follows unknown probability distribution Brown et al. [40]. The main disadvantage of this method in practice is that it requires an analytical or empirical sensor model to provide the prior distribution and compute the likelihood function. In addition, it can systematically underestimate the variance of the distribution, which leads to bias problem. However, the bias of the ML solution becomes less significant as the number of data points N increases and is equal to the true variance of the distribution that generated the data at the limit $N \rightarrow \infty$.

Okello et al. [41] developed a maximum likelihood registration algorithm for spatial alignment of multiple dissimilar sensors in order to improve the fusion performance. Based on the collected data by distributed sensors, the algorithm is able to estimate and display the states of the objects. The developed algorithm has the capability to overcome the following two limitations;first, estimates the sensor biases only for a pair of sensors, second, if the sensors have to be commensurate or alike. This is in addition to the capability to handle any number of dissimilar sensors.

Chen et al. [42] introduced the maximum likelihood approach for joint image registration and fusion, where the expectation maximization algorithm is employed to solve this joint optimization problem. experiments tests were carried out for performance evaluation. Several types of sensory images such as visual images, IR thermal images, and hyperspectral images are considered. The algorithm has the capability to automatically tune the registration parameters. Consequently, the optimal fusion performance can be achieved. On the other hand, Kok [43] developed the same approach for magnetometers and inertial sensors to estimate 3D orientation and obtain the calibration parameters.

2.4.2 The Kalman filter

Kalman filters (KF) are widely used in integrated navigation systems, which combines the collected data from the navigation sensors such as GPS and INS for accurate and useful information, which is better than using individual sensor. KF algorithm is considered a powerful mathematical tool for analyzing and solving localization estimation problems. It was originally proposed by Kalman [44] and has been widely studied and applied since then. Kalman filter estimates the state X of a discrete time process governed by the following space-time model Equation 2-1.

$$X_k = \varphi_{k,k-1}\ X_{k-1} + G_{k-1}W_{k-1} \qquad 2\text{-}1$$

where X_k is the state vector, $\varphi_{k,k-1}$ is the state transition matrix, G_{k-1} is the noise distribution matrix, W_{k-1} is the process noise vector, and k is the measurement epoch.

KF is mainly employed to fuse low-level data, which is used to measure the statistical properties of the model in order to determine the optimal fusion and data estimation. If the system can be described as a linear model and the error could be modeled as the Gaussian noise, then the recursive KF obtains optimal statistical estimations [45]. However, other methods are required to address nonlinear dynamic models and nonlinear measurements. The modified KF known as the Extended Kalman Filter (EKF) is considered an optimal approach for implementing nonlinear recursive filters [46]. The EKF is one of the most often employed methods for fusing data in robotic applications. However, it has some disadvantages due to the computations of the Jacobians which are extremely extensive. Some attempts have been made to reduce the computational cost, such as linearization, but these attempts introduce errors in the filter and make it unstable.

Caron et al. [47] developed a GPS/IMU sensor fusion algorithm using the Kalman filter estimation methods based on fuzzy subsets for UGV. The contextual variables are introduced to define fuzzy validity domains of each sensor. The fused data comes from GPS and IMU fitted to a real vehicle test. It has been noticed that due to the lack of credibility of GPS signal in some cases and the drift of the INS, KF is directly fed with the acceleration provided by the IMU. The author claims that the multi-sensor filter has the

capability to integrate a high number of sensors without changing their structure and the algorithm.

Li et al. [48] introduced a hybrid intelligent multi-sensor positioning methodology for reliable vehicle navigation based on KF. The proposed hybrid positioning technique is fusing the data from low-cost sensors such as GPS, MEMS-based strapdown inertial navigation system (SINS), and electronic compass, which enhances the performance of the integration scheme of these sensors. The improved KF with sequential measurement-update processing was developed to realize the filtering fusion.

Ryu et al. [49] proposed extended and unscented Kalman filters in order to integrate the GPS with INS for autonomous navigation purposes. The developed algorithm aims to reduce the error of the low-cost sensor and estimate an accurate heading and position data for vehicle navigation. For the validation purpose, a GPS, electronic compass, and inertial measurement unit were considered. The obtained result shows that the extended KF able to provide accurate data compared with unscented KF.

Xian et al. [50] integrated the stereo camera and GPS in order to provide the exact motion estimation for an autonomous vehicles navigation. For this purpose, an Iterative Extended Kalman Filter (IEKF) was developed. Figure 2-13 shows the relationship between the world (W), stereo camera (CL) (CR) left and right respectively, and IMU reference frames, where the stereo camera and the MIMU are rigidly attached. In addition, Figure 2-14 shows the integration flowchart between the camera and IMU.

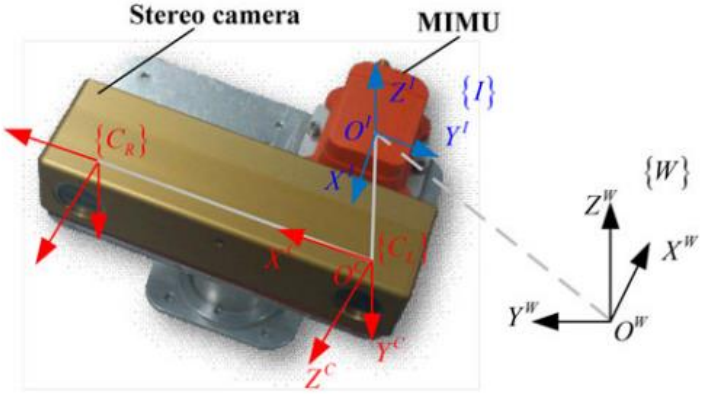


Figure 2-13 Fusing stereo camera and IMU for mobile vehicle navigation [50]

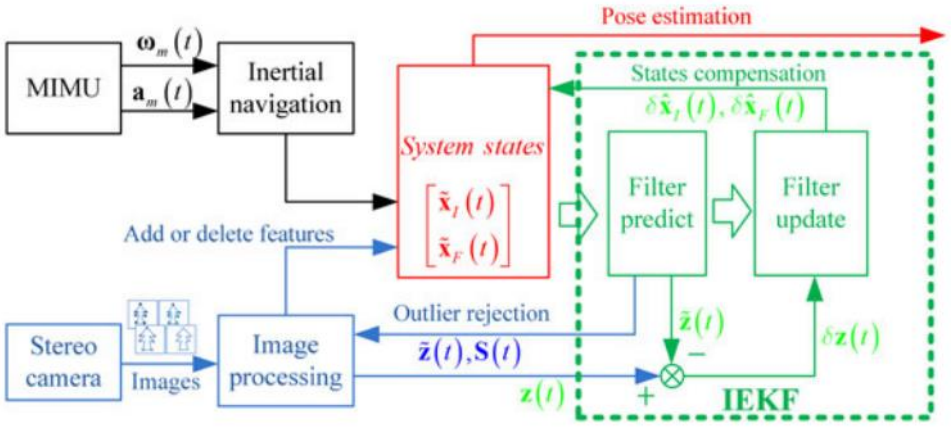


Figure 2-14 The flowchart of the tight integration system of the stereo camera and the MIMU [50]

Wang et al. [51] introduced a navigation system which integrating GPS-INS-Vision for Unmanned Aerial Vehicle (UAV). Subsequently, a CCD camera and laser range finder based on vision system are combined with inertial sensors. The system has the capabilities to provide accurate information on the vertical and horizontal movements of the UAV relative to the ground. In addition, two KF are developed to operate separately and provide a reliable check on the navigation solutions, where the GPS are used to update the KF error states. The integrated GPS-INS-vision navigation system flow chart is shown in Figure 2-15 with two KFs.

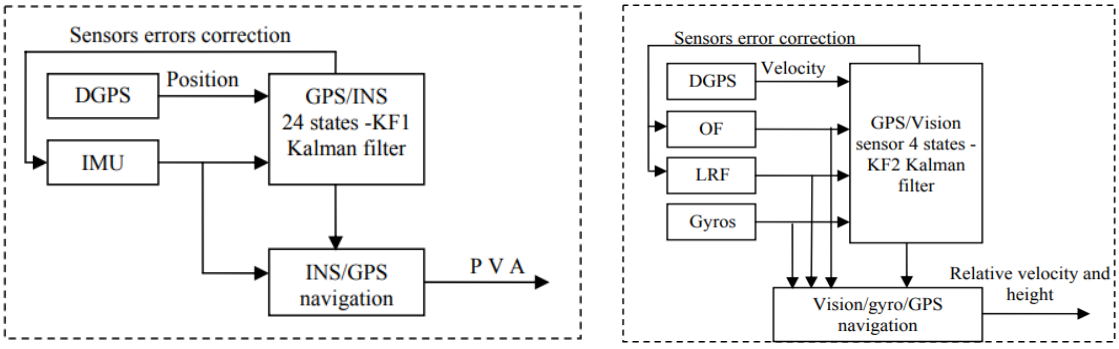


Figure 2-15 Integrated GPS/INS/vision system flowchart [51]

2.4.3 Particle filter.

Particle filters (PF) are iterative implementations of the sequential Monte Carlo methods [52] to solve the filtering problem. This method builds the density function using several

random samples called particles. These particles are increased over the time by the integration of sampling and resampling steps. The sampling step is employed to discard some particles and increasing the relevance of regions with a higher posterior probability. On the other hand, for the filtering process, the particles that have the same state variable are employed and each particle has an associated weight that indicates the quality of the particle. Therefore, the estimation is the result of a weighted sum of all the particles.

The standard particle filter algorithm has two phases: (1) the predicting phase, (2) the updating phase. In the predicting phase, each particle is modified according to the existing model and accounts for the sum of the random noise to simulate the noise effect. Subsequently, in the updating phase, the weight of each particle is re-evaluated using the last available sensor observation, and particles with lower weights are removed.

Kurashiki [53] developed a self-localization algorithm including a two-dimensional laser range finder based on the particle filter for mobile robot navigation. A robust controller and path generation algorithm were developed, then integrated with the localization method in order to improve the vehicle stability under disturbance on rough terrain. The author claims that the particle filter was applied instead of the EKF in order to handle the sensing uncertainties due to unexpected objects.

Rigatos et al. [54] discussed the problem of dynamic ship positioning by developing a sensor fusion algorithm based on Kalman and Particle filters. The combination of these two filters is used to estimate the ship state vector, which fuses the ship position and heading measurements data. These data were received from onboard sensors together with distance measurements data coming from sensors located at the coast. The estimated state vector is used to regulate the horizontal position and the heading direction of the ship. Figure 2-16 show the integrated navigation and dynamic positioning system.

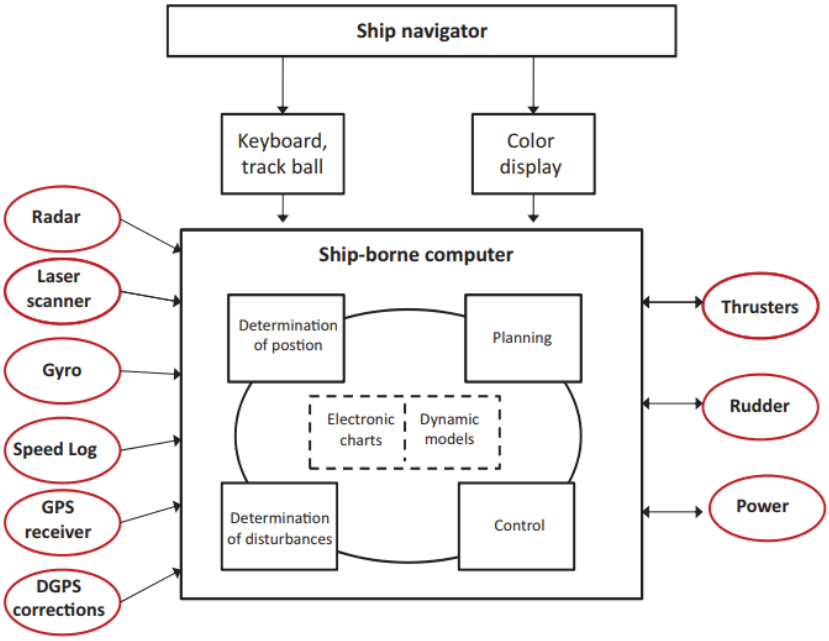


Figure 2-16 Integrated navigation and dynamic positioning system with EKF and Particle Filtering [54]

Hiremath et al. [55] developed an autonomous navigation system using a particle filter based on navigation algorithm for a mobile robot equipped with a LIDAR moving in a maze filed. The developed PF algorithm has the capability to estimate the robot environment state of the system such as robot heading and lateral deviation. These estimated values are used to steer the robot in the right direction. The obtained results showed that the Root Mean Squared Error of the robot heading and lateral deviation were equal to 2.4 degrees and 0.04 m, respectively.

2.5 System Identification Control

This section provides a survey of system identification techniques and its applications. System identification is the methodology of determining a mathematical model of a dynamic system by analyzing the measured input and output signals of the underlying system. The process of system identification uses much of the same theory as in optimal estimation [56] and control [57]. However, instead of estimating the states of a system or observing the states to drive a controller, system identification uses the inputs and output

to develop the model. This model describes the relationship between the input signals to the system and the system output or response.

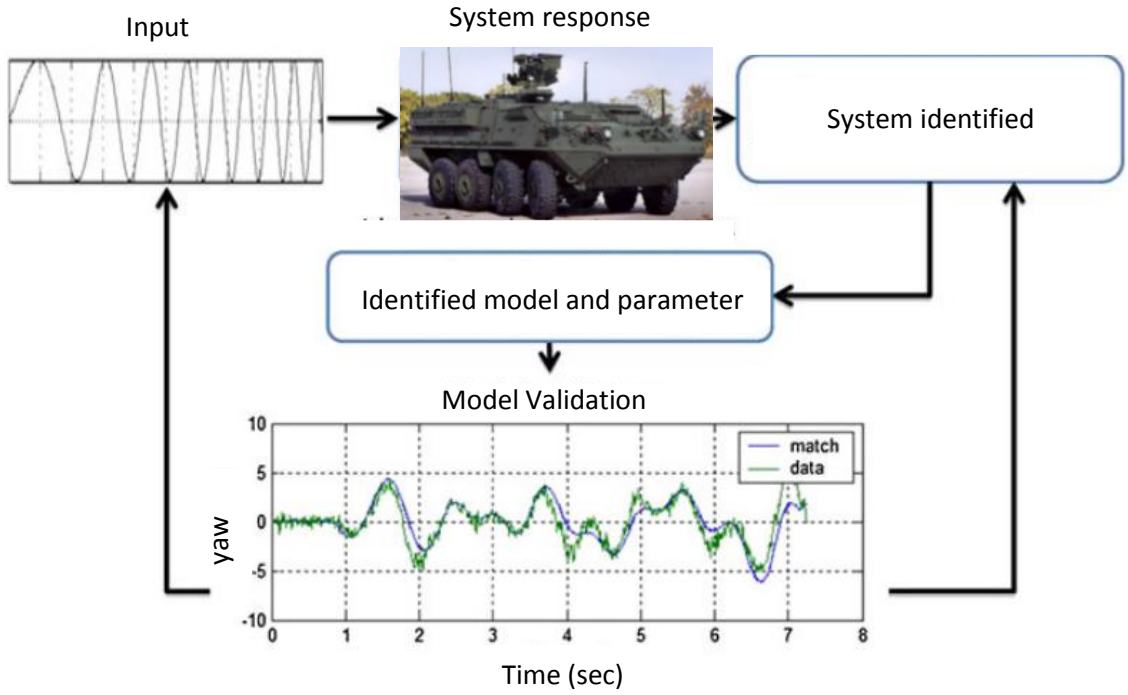


Figure 2-17 Overview of the system identification process

The process of system identification is summarized in Figure 2-17, which clarify how the identified model and parameters are estimated. System identification has five main elements as follows: (1) experiment design, (2) data collection, (3) parameter estimation algorithm and system identification model selection, (4) model validation (5) model implementation. System Identification flowchart is shown in Figure 2-18 , which starts by selecting a model structure followed by the computation of an appropriate model in the structure. The selected model will be evaluated afterward.

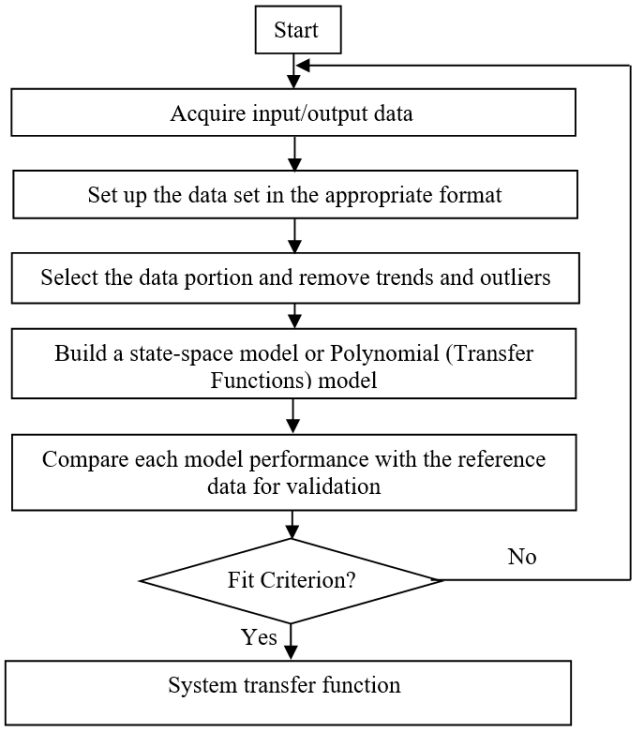


Figure 2-18 System identification procedure flowchart

Nourizadeh et al. [58] developed a discrete time model using system identification techniques in order to describe the motion of a wheeled mobile robot. Two system identification models were considered in this study: 1) Auto Regression Moving Average Exogenous Input (ARMAX), 2) Nonlinear ARMAX (NARMAX) models. The authors claim that the proposed controller based on the ARMAX model is applicable to nonlinear dynamics of the wheeled mobile robot in a wide range of variations. Additionally, there are more advantages of using this method such as that the model does not depend on the platform of the wheeled mobile robot and the model parameters can be estimated using recursive algorithms. The obtained results show that the ARMAX model has the capability to achieve the same performance as the NARMAX model. However, the ARMAX is simpler and less computational time than the NARMAX model, in addition to the capability to adapt itself according to any changes that may arise to the mobile robot.

Aras et al. [59] developed and modelled a low cost underwater Remotely Operated Vehicle (ROV) for depth control using system identification technique. The ROV was developed by Underwater Technology Research Group (UTeRG). First, the system input/output

signals were collected and recorded, then the system identification toolbox in MATLAB was applied to generate the model of the ROV. Subsequently, the identified ROV model was used to design the PID controller for depth control. Consequently, the ROV was able to remain stationary at the desired depth using the pressure sensor data as feedback.

A system identification based on the prediction error method for a large-scale unmanned helicopter was carried out by Hashimoto et al. [60]. The helicopter was compensated by the attitude control that permitted the experiments during the flight. Based on the measured input/output data, the system identification method was applied to the helicopter as a single-input-single-output (SISO) system. Afterwards, it is applied as a multi-input-multi-output (MIMO) system to derive the mathematical model of the helicopter. The block diagram of the helicopter for system identification is shown in Figure 2-19.

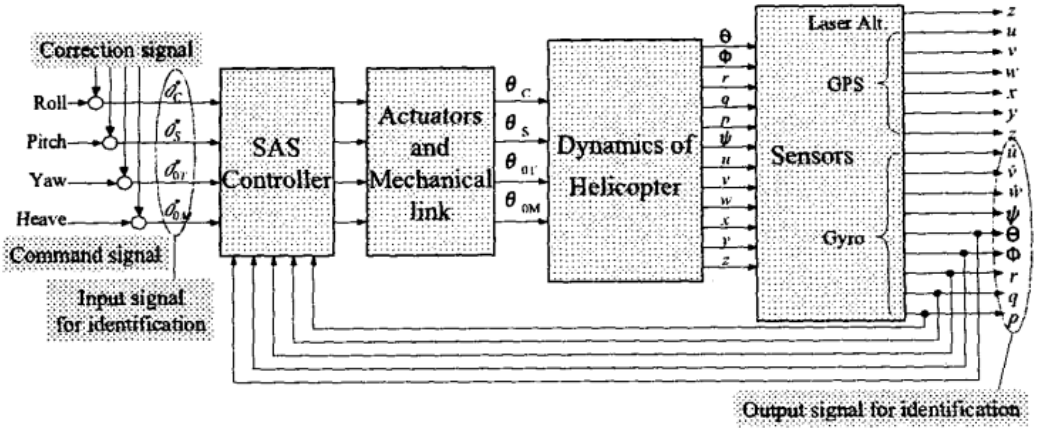


Figure 2-19 Construction for system identification experiments on the helicopter [60]

Karras et al. [61] developed an online identification system for autonomous underwater vehicles, which estimates their dynamic parameters based on a global derivative-free optimization algorithm. The introduced identification method consists of three modules as follows.

1- System excitation module: this module sends excitation inputs to the autonomous vehicle.

2- Optimization algorithm module: this module calculates a candidate parameter vector, which feds to the third module (metric calculator module).

3- Metric calculator module: this module evaluates the received candidate parameter vector from the optimization algorithm module, using a metric based on the residual of the actual and the predicted commands.

The authors claim that the developed algorithm is a global, unlike Unscented Kalman Filter (UKF) and Extended Kalman Filter (EKF). It does not depend on the initialization and does not require the designer to know a good starting point. Figure 2-20 shows the On-line identification scheme.

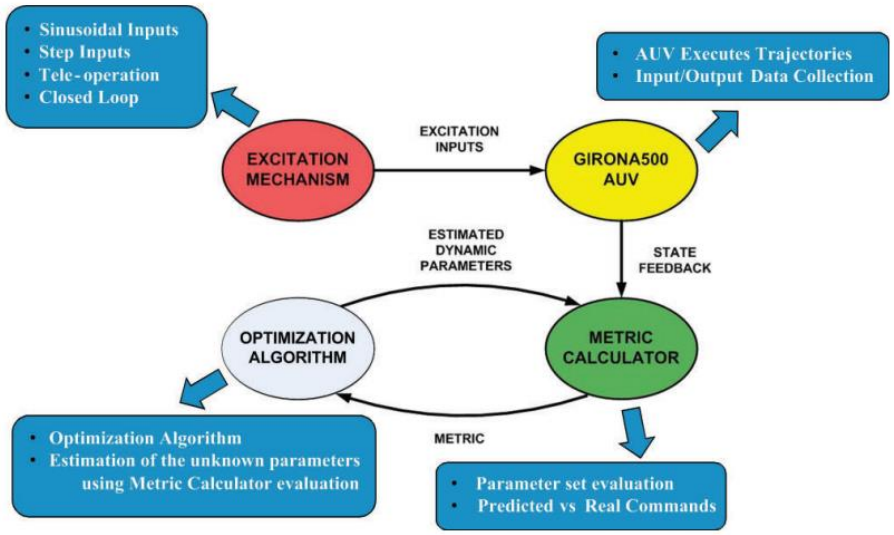


Figure 2-20 On-line identification scheme [61]

Eng [62] developed an on-line system identification method for autonomous underwater vehicle dynamics via in-field experiments. The introduced identification process has two stages: the training stage and the validation stage. The unknown parameters are estimated using a state variable filter and recursive least square (SVF-RLS) estimator in the training stage. However, in the validation stage, the prediction capability of the developed model is examined using a new data set. The validation results show that the identified model satisfies 78% to 92% of the output variation. The comparison between the SVF-RLS estimator and conventional identification method shows that the SVF-RLS estimator is better in terms of prediction accuracy, computational cost, and training time.

Samal et al. [63] introduced an off-line and on-line neural network identification algorithm for modeling the dynamics of an autonomous miniature Eagle helicopter. Both coupled and decoupled dynamics of the helicopter is identified using the flight test data. The artificial neural network based black-box method is used to model the UAV dynamics. The interaction between the inputs and outputs of the (MIMO) system is considered. Figure 2-21 shows the series-parallel model for neural network identification. The predicted responses from these neural network models and the actual responses of the Eagle helicopter were compared. The obtained results show that the off-line model performs better compared to the online model. The authors claim that the additional training time and bigger batch size available for off-line training increases the system performance.

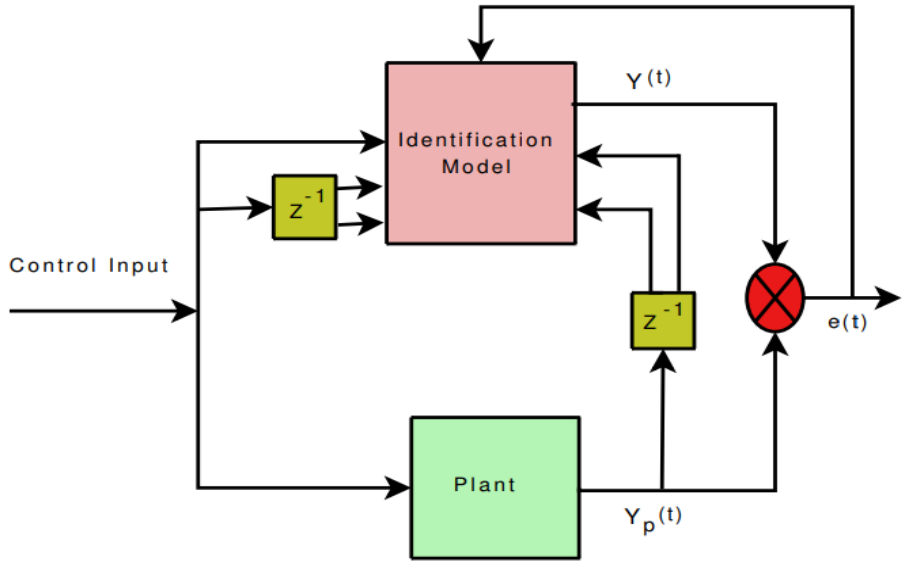


Figure 2-21 Series-Parallel Model for NN Identification [63]

XU Jian'an et al. [64] developed a kinematic model identification algorithm for an autonomous mobile robot using the dynamical recurrent neural network. Based on the structure analysis and training algorithm of the dynamical recurrent neural networks, the kinematic forward model identification of the autonomous mobile robot is realized. The experiments on the AS-R mobile robot show that the dynamical recurrent neural network has the capability to identify the robot's kinematic model accurately. The block diagram of mobile robot kinematic model identification is shown in Figure 2-22

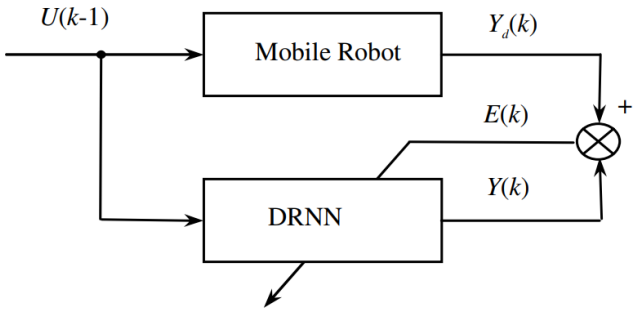


Figure 2-22 Block diagram of kinematic model identification [64]

2.6 Outline of Thesis

Chapter 3 of this thesis introduces the development of an optimal control path planning algorithm for the multi-wheeled combat vehicle. The proposed path planning algorithm is based on the combination of optimal control theory and artificial potential field function. This algorithm is developed using Pontryagin's minimum principle, where, the vehicle states are validated using the TruckSim vehicle model. Chapter 4 describes the methodology of developing an artificial neural network model for real time optimal path planning for the multi-wheeled combat vehicle. Chapter 5 illustrates the comparative study of the dynamic programming and Pontryagin's minimum principle for autonomous vehicle path planning. Chapter 6 discusses the development of a navigation system for the multi-wheeled combat vehicle using a hybrid positioning technique. Chapter 7 introduces the development and modeling of the remotely operated scaled multi-wheeled combat vehicle using system identification methodology. Chapter 8 concerns in developing an intelligent control algorithm for heading angle tracking using the identified vehicle model, then validates it using the processor in the loop co-simulation. Chapter 9 discusses the developed robust H_∞ controller for heading angle tracking. Chapter 10 offers conclusions and considers future work.

2.7 Chapter Summary

This chapter provides a critical review of a state of the art of the published work in advanced control, motion planning techniques, sensor fusion, and system identification techniques for unmanned ground vehicles (UGVs). Practical issues and technical challenges associated with the development of UGVs are also outlined.

UGVs have many potential applications and the demand for them is ever increasing, in addition, it has drawn interest from many researchers and organizations, especially military applications. In fact, UGVs have been used in some military operations such as inspection, surveillance, and rescue operations.

Based on the literature review, it is challenging to develop an intelligent controller for a UGV due to the urgent need to maintain the UGV stability performance while navigating the vehicle autonomously. Consequently, uncertain dynamics due to various disturbances coming from the external load, ground contact, and unmolded dynamics must be considered while developing the controller. The challenges for UGVs is to develop a control system that has the capability to handle all the nonlinearities caused by the vehicle subsystems and the harsh environmental conditions.

Furthermore, motion planning for UGVs has gained substantial attention since several challenges are encountered such as road conditions, and limited knowledge about the working environment. It has been noted that most of the published work focuses on the mathematical formulation and the numerical simulation without shedding the light on the practical aspects of the typical implementation in UGVs.

On the other hand, the selection of the sensor fusion estimation technique depends on the type of the problem and the established assumptions of each technique. Most of the state estimation methods that are used for sensor fusion are based on control theory and the laws of probability. Many autonomous systems use KF to integrate GPS/INS seniors as a navigation system to increase the navigation solution accuracy. Computer vision techniques have been applied to almost all environments and all kind of mobile robots for example; ground, air, and underwater robots.

On the other hand, increasing the number of the sensor will lead to some faults due to hardware or software malfunction. These faults are difficult to predict accurately in time in order to prevent them. Therefore, fault tolerance control is used to achieve a more reliable performance in such autonomous systems. Fault tolerance control aims to prevent the simple faults, that can be developed into serious failure, in this way fault tolerance increases plant reliability and reduce the risk of hazards. in addition, from the system identification

research literature, system identification is a valuable method for determining system dynamics. Safer and more reliable autonomous vehicles can be designed with a better understanding of system dynamics based on system identification methods. With safer and more reliable autonomous vehicles, scientists can have a broader range of tools for collecting scientific data.

CHAPTER 3

Optimal Collision-Free Path Planning for Autonomous Multi-Wheeled Combat Vehicle

3.1 Introduction

In recent years, many researchers investigated and developed autonomous vehicles, which have become increasingly important assets in various civilian and military operations. Multi-wheeled combat vehicles are one of the difficult types of the vehicles in applying autonomy due to their large dimensions, heavy weight, and complex geometry. However, combat vehicles are mainly essential for military applications [65,66]. The expected outcomes of the autonomous multi-wheeled combat vehicles are set to increase combat capability while protecting soldiers in a full spectrum of battlefield scenarios. Motion planning or path planning is one of the challenges in autonomous vehicles control, which is required to have the ability to deal with unknown workspaces such as road conditions, uncertainty in the vehicle states, and limited knowledge about the surrounding environments.

Motion planning and obstacle avoidance are widely used to find the collision-free path for autonomous vehicles to move amidst a set of obstacles [67,68]. There is much path planning research done using probabilistic roadmaps [69], Rapidly-exploring Random Tree RRT [70], and Artificial Potential Field (APF) [71]. The APF approach is commonly used in path planning due to its simple algorithm and mathematical description. It is also convenient for real time control [72-75]. For example, Adeli [76] solved the path planning problem of a mobile robot using a potential function considering the distances from obstacles, the goal point, and start point. Although this algorithm was able to avoid obstacles, the generated path was not optimal because it was still based on the classic potential field method. Consequently, to solve this problem and generate the vehicle optimal path, the optimal control theory is introduced in this chapter.

Using optimal control theory, the vehicle path planning can be formulated as an optimal control problem. It is used as a set of differential equations describing the paths of the

control variables that minimize the cost function to satisfy the initial/final conditions. The Pontryagin's principle [77] is introduced to solve the optimal control problem for generating the optimal path. The proposed method is able to provide the optimal solutions for the vehicle path planning problems while satisfying the initial and final constraints.

This chapter describes the methodology of generating the optimal collision-free path planning algorithm for an autonomous multi-wheeled combat vehicle. The developed algorithm is based on the combination between optimal control theory and Artificial Potential Field function (APF). The optimal path planning algorithms are developed using Pontryagin's Minimum Principle (PMP), which is one of the major branches of the optimal control theory. The cost function of the path planning is solved together with vehicle dynamic equations to satisfy the boundary conditions. For this purpose, a simplified four-axle bicycle model of the actual vehicle considering the vehicle body lateral and yaw dynamics while neglecting roll dynamics is considered. The obstacle avoidance technique is mathematically modeled based on the proposed artificial potential field method using the Gaussian function. This potential function is assigned to each obstacle as a repulsive potential field. The inclusion of these potential fields results in a new APF which controls the vehicle to reach the goal point safely. The proposed algorithm is validated using a full nonlinear vehicle model in TruckSim software, where the time history of the vehicle states in terms of lateral acceleration, yaw rate and curvature at different speeds are compared. Several simulations are carried out to check the fidelity of the proposed technique. The obtained results demonstrate that the generated path for the vehicle is able to avoid collision with the obstacles, while it satisfies the boundary conditions. In addition, it is successfully validated with TruckSim vehicle model.

3.2 Vehicle Dynamics and Modeling

A simplified two DOF vehicle is used to develop the differential equations of the multi-wheeled combat vehicle. This linearized plant is used to generate the optimal path and represent the vehicle motion during the maneuver. The actual vehicle configuration and a simulation model of the vehicle in TruckSim are shown in Figure 3-1. The vehicle is equipped with four axles, where the front two axles are steered. The vehicle model consists

of 22 degrees of freedom, namely pitch, yaw, and roll of the vehicle sprung mass, and spin and vertical motions of each of the eight wheels.

(a) (b)

Figure 3-1 (a) Actual vehicle configuration [78] and (b) the simulation model

The simplified version of the actual vehicle model describes the lateral and yaw motion of vehicle while neglecting roll dynamics. Figure 3-2 represents the free body diagram of the simplified vehicle model. The left and right sides of the vehicle have been combined into one single track. The center of gravity is represented by a golden circle and it is located between the second and third axles. The vehicle forward speed (U), the lateral speed (V), and the yaw rate (r) are shown acting on the center of gravity. The positive x-axis points to the right of the page, the positive y-axis points down the page, the positive z-axis is directed into the page and yaw moments are clockwise positive as shown in the legend at the top right of Figure 3-2. Tire slip angles and wheel steer angles are shown in blue and red respectively. Axle distances (a_1, a_2, a_3, a_4) are measured from the center of gravity to the center of each axle.

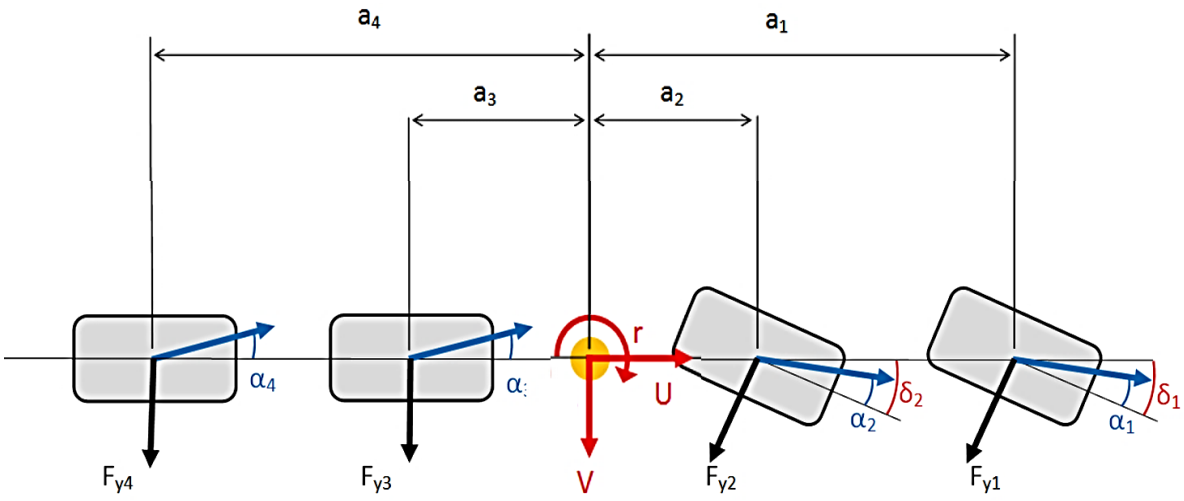


Figure 3-2 Two-degree-of-freedom Bicycle Vehicle Model

The forward velocity is assumed constant and $\delta_1$and δ_2 represents the average first and second axle's wheel steering angles respectively, which assumed small angles (Cos δ_1=1 and Cos δ_2=1). The governing equations of motion for the vehicle body lateral and yaw dynamics are given in Equations 3-1, 3-2 respectively:

$$\sum m(\dot{V} + rU) = F_{y1} + F_{y2} + F_{y3} + F_{y4} \qquad 3\text{-}1$$

$$\sum I_{zz}\dot{r} = a_1F_{y1} + a_2F_{y2} - a_3F_{y3} - a_4F_{y4} \qquad 3\text{-}2$$

where U and V stand for the longitudinal and lateral velocities of vehicle center of gravity (CG) respectively, m is the vehicle mass, I_{zz} is the yaw moment of inertia.

Tire slip angle (α) equations and cornering forces (F_y) are calculated for each axle where $[i = 1{:}\,4]$

$$\alpha_1 = \delta_1 - \left(\frac{V+a_1r}{U}\right) \;, \alpha_2 = \delta_2 - \left(\frac{V+a_2r}{U}\right) \qquad 3\text{-}3$$

$$\alpha_3 = -\left(\frac{V-a_3r}{U}\right) \quad , \; \alpha_4 = -\left(\frac{V-a_4r}{U}\right) \qquad 3\text{-}4$$

$$F_{yi} = \alpha_i C_{\alpha i} \tag{3-5}$$

where $C_{\alpha i}$ represents the axle cornering stiffness.

Lateral Motion Equation

Substituting Equations 3-3, 3-4, 3-5 into Equation 3-1 yields:

$$m(\dot{V} + rU) = C_{\alpha 1}\left[\delta_1 - \left(\frac{V + a_1 r}{U}\right)\right] + C_{\alpha 2}\left[\delta_2 - \left(\frac{V + a_2 r}{U}\right)\right]\left[-\left(\frac{V - a_3 r}{U}\right)\right] + C_{\alpha 4}\left[-\left(\frac{V - a_4 r}{U}\right)\right] \tag{3-6}$$

$$\dot{V} = -\left(\frac{(C_{\alpha 1} + C_{\alpha 2} + C_{\alpha 3} + C_{\alpha 4})}{mU}\right)V + \left(\frac{(-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4})}{mU} - U\right)r + \left(\frac{C_{\alpha 1}\delta_1 + C_{\alpha 2}\delta_2}{m}\right) \tag{3-7}$$

Yaw Motion Equation

Substituting Equations 3-3, 3-4, 3-5 into Equation 3-2 yields:

$$\dot{r} = \frac{1}{I_{zz}}\left\{a_1 C_{\alpha 1}\left[\delta_1 - \left(\frac{V + a_1 r}{U}\right)\right] + a_2 C_{\alpha 2}\left[\delta_2 - \left(\frac{V + a_2 r}{U}\right)\right] - a_3 C_{\alpha 3}\left[-\left(\frac{V - a_3 r}{U}\right)\right] - a_4 C_{\alpha 4}\left[-\left(\frac{V - a_4 r}{U}\right)\right]\right\} \tag{3-8}$$

Rearranging,

$$\dot{r} = \frac{1}{I_{zz}}\left\{a_1 C_{\alpha 1}\delta_1 + a_2 C_{\alpha 2}\delta_2 - a_1 C_{\alpha 1}\frac{V}{U} - a_2 C_{\alpha 2}\frac{V}{U} + a_3 C_{\alpha 3}\frac{V}{U} + a_4 C_{\alpha 4}\frac{V}{U} - \frac{{a_1}^2 C_{\alpha 1} r}{U} - \frac{{a_2}^2 C_{\alpha 2} r}{U} - \frac{{a_3}^2 C_{\alpha 3} r}{U} - \frac{{a_4}^2 C_{\alpha 4} r}{U}\right\} \tag{3-9}$$

$$\dot{r} = \left(\frac{(-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4})}{I_{zz} U}\right) V + \left(\frac{(-a_1^2 C_{\alpha 1} - a_2^2 C_{\alpha 2} + a_3^2 C_{\alpha 3} + a_4^2 C_{\alpha 4})}{I_{zz} U}\right) r + \left(\frac{a_1 C_{\alpha 1} \delta_1 + a_2 C_{\alpha 2} \delta_2}{I_{zz}}\right) \quad 3\text{-}10$$

Now the differential Equations 3-7, 3-10 fully describe the motion of the linear bicycle model. The final states of the vehicle model can be derived and written according to the following matrix form:

$$\begin{bmatrix} \dot{V} \\ \dot{r} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} -a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V \\ r \\ \Psi \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ 0 \end{bmatrix} \delta_1 \quad 3\text{-}11$$

where

$$a_{11} = \left(\frac{(C_{\alpha 1} + C_{\alpha 2} + C_{\alpha 3} + C_{\alpha 4})}{mU}\right)$$

$$a_{12} = \left(\frac{(-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4})}{mU} - U\right)$$

$$a_{21} = \left(\frac{(-a_1 C_{\alpha 1} - a_2 C_{\alpha 2} + a_3 C_{\alpha 3} + a_4 C_{\alpha 4})}{I_{zz} U}\right)$$

$$a_{21} = \left(\frac{(-a_1^2 C_{\alpha 1} - a_2^2 C_{\alpha 2} + a_3^2 C_{\alpha 3} + a_4^2 C_{\alpha 4})}{I_{zz} U}\right)$$

$$b_{11} = \left(\frac{C_{\alpha 1} + C_{\alpha 2} K_s}{m}\right)$$

$$b_{21} = \left(\frac{a_1 C_{\alpha 1} + a_2 C_{\alpha 2} K_s}{I_{zz}}\right)$$

where

r and Ψ are the vehicle yaw rate and yaw angle respectively, and $\delta_2 = K_s \, \delta_1$,

$C_{\alpha1}, C_{\alpha2}, C_{\alpha3}\ and\ C_{\alpha4}$ denote the cornering stiffness's of tires.

The coordinates of the vehicle body motion(x, y) in global coordinates can be derived as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix} U & -(V + a_1.r)\\ (V + a_1.r) & U\end{bmatrix}\begin{bmatrix}cos\,\Psi\\ sin\,\Psi\end{bmatrix} \qquad 3\text{-}12$$

Setting the state vector as $\boldsymbol{X} = [V\ \ r\ \ \Psi\ \ x\ \ y\ \ \delta_1\ ]^T$, then from Equations 3-11 and 3-12, the state space equations for the bicycle model can be written as follows:

$$\dot{X} = f(X) + Bu \qquad 3\text{-}13$$

where

$$f(X) = \begin{bmatrix} -a_{11}V + a_{12}V + b_{11}\delta_1\\ a_{21}V + a_{22}r + b_{21}\,\delta_1\\ r\\ Ucos\Psi\ -(V + a_1\,r)\,sin\,\Psi\\ U\,sin\,\Psi\ +(V + a_1\,r)\,cos\,\Psi\\ 0\end{bmatrix}\ ,\ B = \begin{bmatrix}0\\0\\0\\0\\0\\1\end{bmatrix}\ ,\qquad u = \dot{\delta}_1$$

3.3 Obstacle Modeling based Artificial Potential Field (APF)

In this section, the obstacles and goal locations are modeled based on the artificial potential field approach. The principle of this approach is specified in using imaginary forces acting on the vehicle direction during the maneuver. The obstacles have a repulsive field approximately them and the goal has an attractive field. The synthesis of all the fields determines the motion direction of the vehicle in the workspace [79]. The developed APF is based on two-dimensional Gaussian attractor $\left(f_A(x, y)\right)$ and repulsor $\left(f_R(x, y)\right)$ functions that can be described as follows:

$$f_A(x, y) = 1 - e^{\frac{-\left((x-a_x)^2+(y-a_y)^2\right)}{2\sigma^2}} \qquad 3\text{-}14$$

$$f_R(x, y) = e^{-0.5\left(\frac{(x-r_x)^2+(y-r_y)^2}{\sigma^2}\right)^c} \qquad 3\text{-}15$$

where

(a_x, a_y) The position center of the attractor,

(r_x, r_y) The position center of the repulsor,

(σ) Used to change the size of the obstacle,

(c) Determines the effect range of the given obstacle.

The implementation of the given potential field in the vehicle model requires the calculation of the coordinates of the vehicle body, surrounding obstacles, and target location. Figure 3-3 shows 3D Gaussian functions for the repulsor and attractor potential fields. The represented obstacles at points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ where x_i, y_i represent the coordinates of the obstacle. These obstacles are considered static with predefined coordinates.

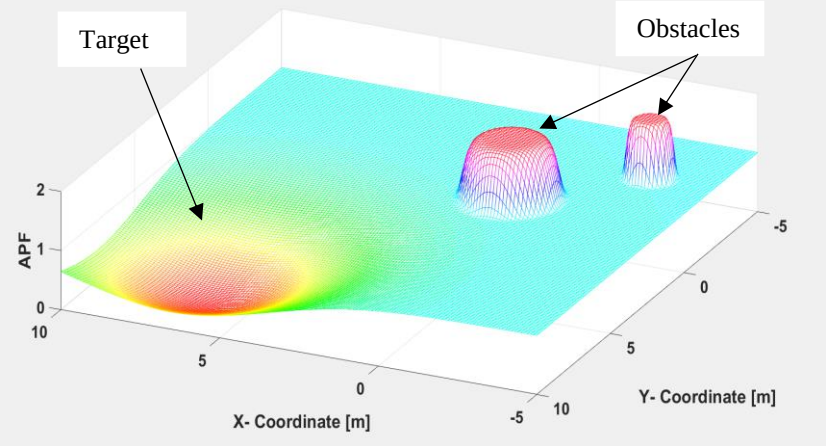


Figure 3-3 Visualization of the obstacles using APF.

3.4 Optimal Path Planning Algorithm

The proposed path planning algorithm is based on the optimal control theory to generate the vehicle optimal collision-free path. This algorithm permits the vehicle to move between two points along a predefined coordinate of the obstacles as well as initial and final locations. The control action is performed by applying the output control signal directly to the front wheel of the vehicle. For obtaining the optimal collision-free path for the vehicle, the cost function is defined in Equation 3-16. This cost function should be solved together

with vehicle dynamics Equation 3-13 to satisfy the vehicle dynamics and the initial and final conditions.

$$\min_{u(t)} J = \int_{t_o}^{t_f} L(X(t)\ , u(t)\ , t\ )\, dt \qquad 3\text{-}16$$

where $X(t)$ is the states vector $= \begin{bmatrix} V & r & \Psi & x & y & \delta_f \end{bmatrix}^T$ and $u(t)$ represent the control action for front wheel steering angle. By setting-up the Lagrangian $L(X,u)$ for the cost minimization problem considering the obstacle location will be as follows:

$$L(X,u) = L_1(X,u) + W_{po} \sum_i P_{oi}(X_c) \qquad 3\text{-}17$$

where $X_c = (x_c, y_c)$, $P_{oi}(X_c) = f_R(x_c, \mathrm{y}_c)$ is the repulsor potential field based on Gaussian function as defined in Equation 3-15, W_{po} is the weigh.

$L_1(X,u)$ is introduced to define the states vector and control signal as follows:

$$L_1(X,u) = X^T Q X + u^T R u \qquad 3\text{-}18$$

where X, X^T are the state vector and its transpose respectively, u is the control input and (Q,R) are positive weighting matrices.

From Equations 3-16, 3-17, and 3-18 the cost function that needs to be minimized in Equation 3-16 can be rewritten as follows:

$$\min_{u(t)} J = \int_{t_o}^{t_f} X^T Q X + u^T R u + W_{po} \sum_i P_{oi}(X_c) \qquad 3\text{-}19$$

In order to solve this problem, the Pontryagin's Minimum Principle (PMP) is employed as an optimal control solution. For this purpose, the Hamiltonian function is considered as shown in Equation 3-20.

$$H(X(t), u(t), \lambda(t), t) \triangleq L(X(t), u(t), t) + \lambda^T f(X(t), u(t), t) \qquad 3\text{-}20$$

where λ^T is the Lagrange multiplier vector for the vehicle dynamics constraint. The co-state equation $(\dot{P})$ can be presented in the following form:

$$\dot{P} = -\frac{\partial H}{\partial x}(X^*(t_f^*), u^*(t_f^*), \lambda(t_f^*), t_f^*) \quad 3\text{-}21$$

The control input can be obtained from the first derivative of a Hamiltonian function with respect to the control signal as follows:

$$\frac{\partial H}{\partial u}(X, \lambda, u) = 0 \;\; i.e. \;\; \frac{\partial L}{\partial u} + \sum \lambda_i \frac{\partial f_i}{\partial u} = 0 \quad 3\text{-}22$$

Both the initial and final conditions of the Hamiltonian function given in Equation 3-23 are defined as follows:

$$\left.\begin{array}{l} X(t_f) = [0 \;\; 0 \;\; \Psi_0 \;\; d_{xo} \;\; d_{yo} \;\; 0] \\ X(t_o) = [0 \;\; 0 \;\; \Psi_f \;\; d_{xf} \;\; d_{yf} \;\; 0] \end{array}\right\} \quad 3\text{-}23$$

where $(d_{xo},\ d_{yo})$, $(d_{xf},\ d_{yf})$ are the initial vehicle positions in longitudinal and lateral directions respectively. In addition, (Ψ_0, Ψ_f) are the initial and final heading angles of the vehicle. The Hamiltonian function is solved for the following states and co-states vectors which are given as follows:

$$\left.\begin{array}{l} \dot{X} = [\dot{V} \;\; \dot{r} \;\; \dot{\Psi} \;\; \dot{x} \;\; \dot{y} \;\; \dot{\delta}]^T \\ P = [\dot{P}_1 \;\; \dot{P}_2 \;\; \dot{P}_3 \;\; \dot{P}_4 \;\; \dot{P}_5 \;\; \dot{P}_6]^T \end{array}\right\} \quad 3\text{-}24$$

In order to solve the boundary value problem Equations 3-19, 3-21, 3-22, and 3-23 a MATLAB built-in boundary value problem function (bvp4c) is used to solve the state variables and solve them and generate the optimal path within the time interval from (t_o) to (t_f).

3.5 Simulation and Results

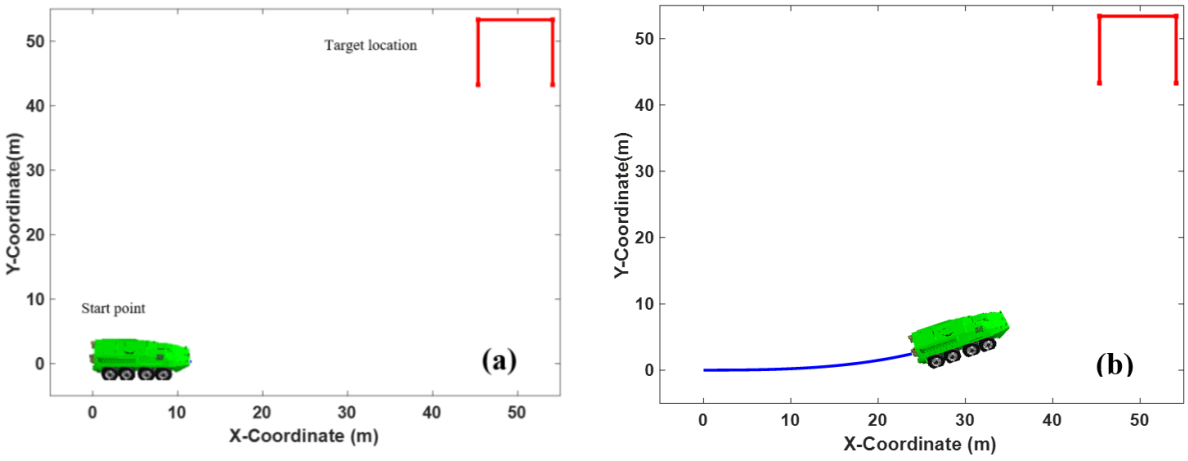
The numerical simulation is carried out in MATLAB environment in order to evaluate the robustness of the proposed collision-free path planning algorithm including different scenarios. Subsequently, the generated optimal path is applied to the TruckSim full vehicle

model to validate and compare the lateral acceleration, yaw rate, and the vehicle curvature with the linearized model at two different constant speeds of 9km/h and 28km/h. TruckSim is a mechanical simulation software used to calculate the vehicle's kinematics and dynamics, which represent the actual full vehicle model. The parameters characteristics for a typical combat vehicle is used for path planning such as actual vehicle dimension, cornering stiffness for the four axles and the yaw moment of inertia and total weight.

The proposed algorithm is experienced through the following two scenarios. The first scenario shows the vehicle optimal path between two points considering the target border as obstacles. The second scenario shows the vehicle optimal collision-free path between two points with different obstacles imposed in the path of the vehicle.

3.5.1 Optimal path between two points

The objective of this scenario is to check the capability of the proposed methodology to generate an optimal path for the vehicle from a given starting and ending points performing the maneuver safely without hitting the destination borders. For this purpose, the destination borders are simulated using repulsor potential field which is represented in red lines as shown in Figure 3-4, considering the initial and target locations at (0, 0), (50, 50) respectively. In addition, the initial and final heading angles are zero and 90 degrees respectively. Figure 3-4 (a-d) show the trace of the obtained trajectory which is close to what a real driver would do in this scenario.



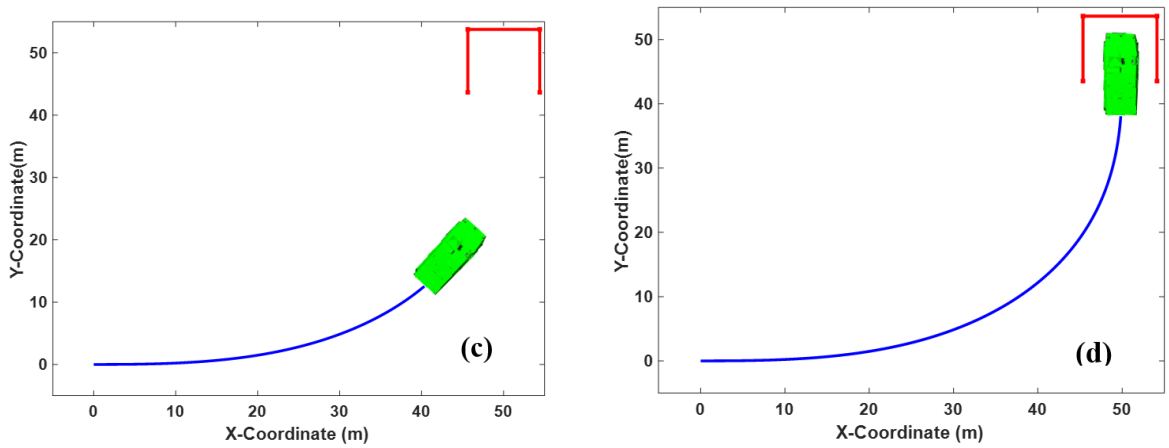
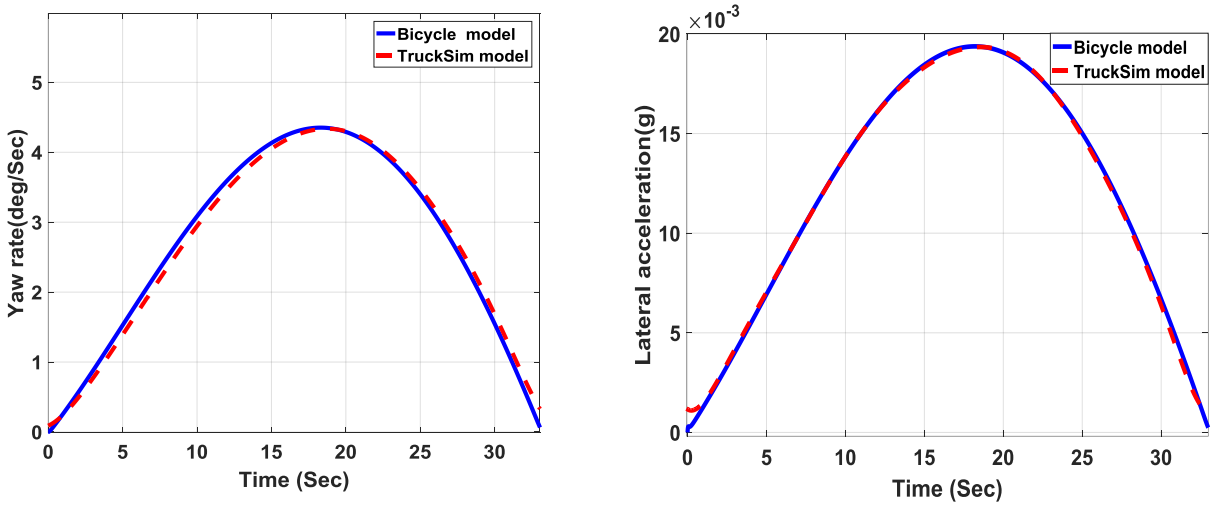
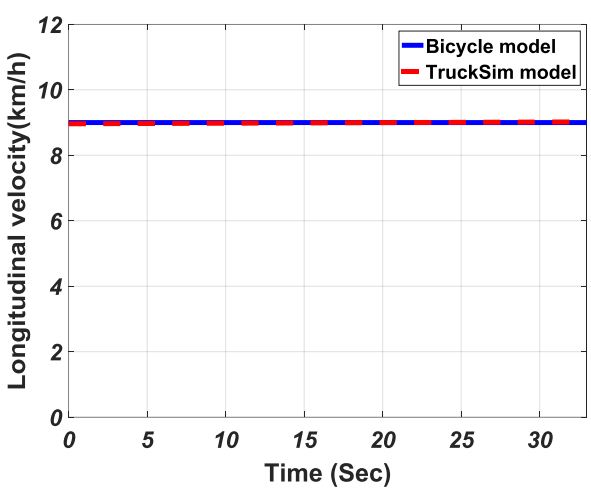


Figure 3-4 (a-d) Generated optimal Path between Two Points

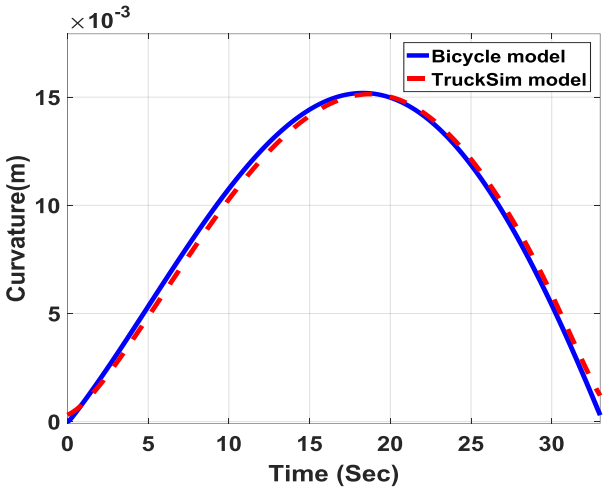
The validation of the MATLAB vehicle model with TrackSim model for performing the maneuver between two points at a constant speed of 9 km/h is shown in Figure 3-5 (a-d). These figures compare the vehicle yaw rate, lateral acceleration, curvature and longitudinal velocity of both models. The simulation result demonstrates that the vehicle parameters using both models when performing the same maneuver at a speed of 9 km/h agree with each other.



(a) (b)



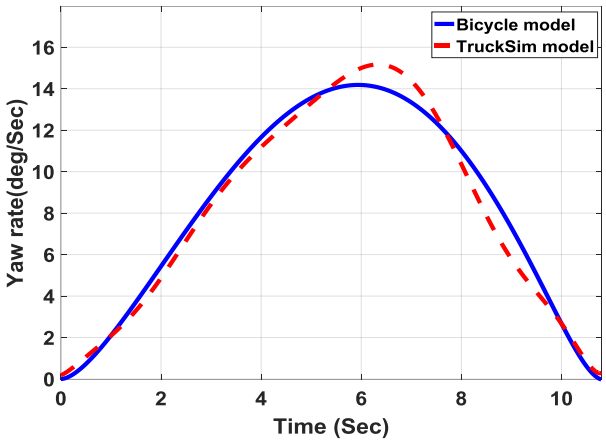
(c)



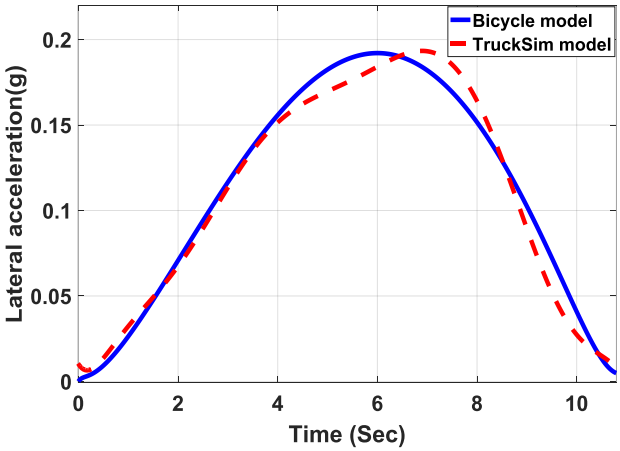
(d)

Figure 3-5 Comparison between the bicycle model and TruckSim vehicle model at speed of 9 km/h: a) Yaw Rate (deg/s), b) lateral acceleration (g), c) longitudinal velocity (km/h), d) Curvature (m).

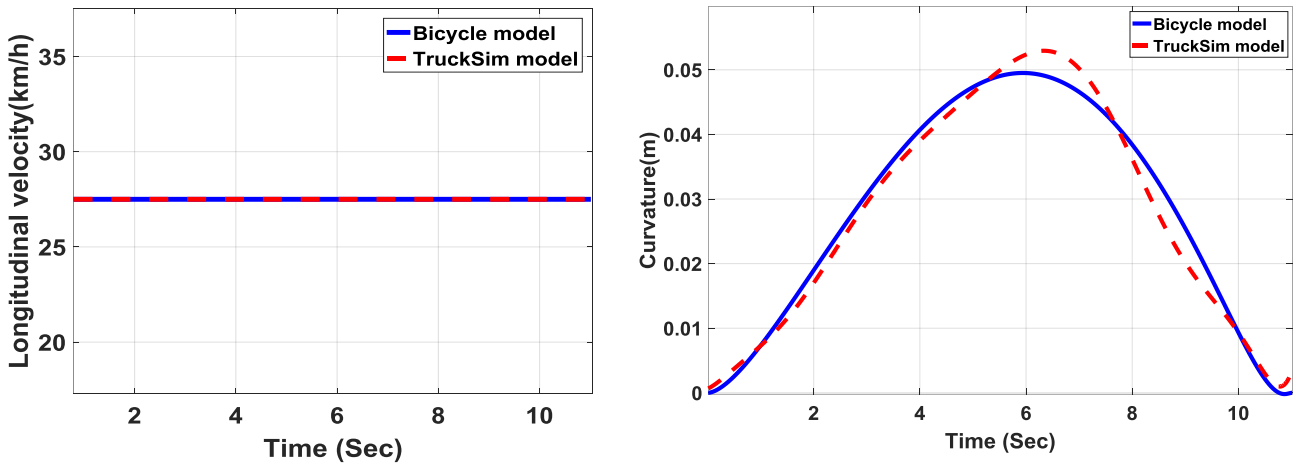
On the other hand, at speed of 28 km/h, the comparison of the vehicle states is shown in Figure 3-6 (a-d). It has been noticed that, as the longitudinal speed increased, the lateral acceleration, curvature, and yaw rate started to make a small deviation. This deviation is due to the assumption of a linear tire model characteristic in the derived model in MATLAB.



(a)



(b)



(c) (d)

Figure 3-6 Comparison between the bicycle model and TruckSim vehicle model at speed of 28 km/h longitudinal velocity: a) Yaw Rate (deg/s), b) lateral acceleration (g), c) longitudinal velocity (km/h), d) Curvature (m).

3.5.2 Optimal path between two points with obstacles

This scenario is aimed to check the generated optimal path in case that the obstacles are imposed at different locations in the way of the vehicle during the maneuver between two points. For this purpose, the simulation is carried by imposed an obstacle in the previously generated path in section 3.5.1. Furthermore, another obstacle is imposed in the new generated path.

3.5.2.1 Imposed one obstacle

In this scenario, the obstacle is imposed at (20, 1.6) on the previously generated path as mentioned above. The generated optimal collision-free path during the vehicle maneuver is shown in Figure 3-7. The simulation results show that the vehicle reaches the goal location safely while avoiding the imposed obstacle in the path and at the destination location. In addition, the generated path is similar to the human driver response in case of avoiding an obstacle.

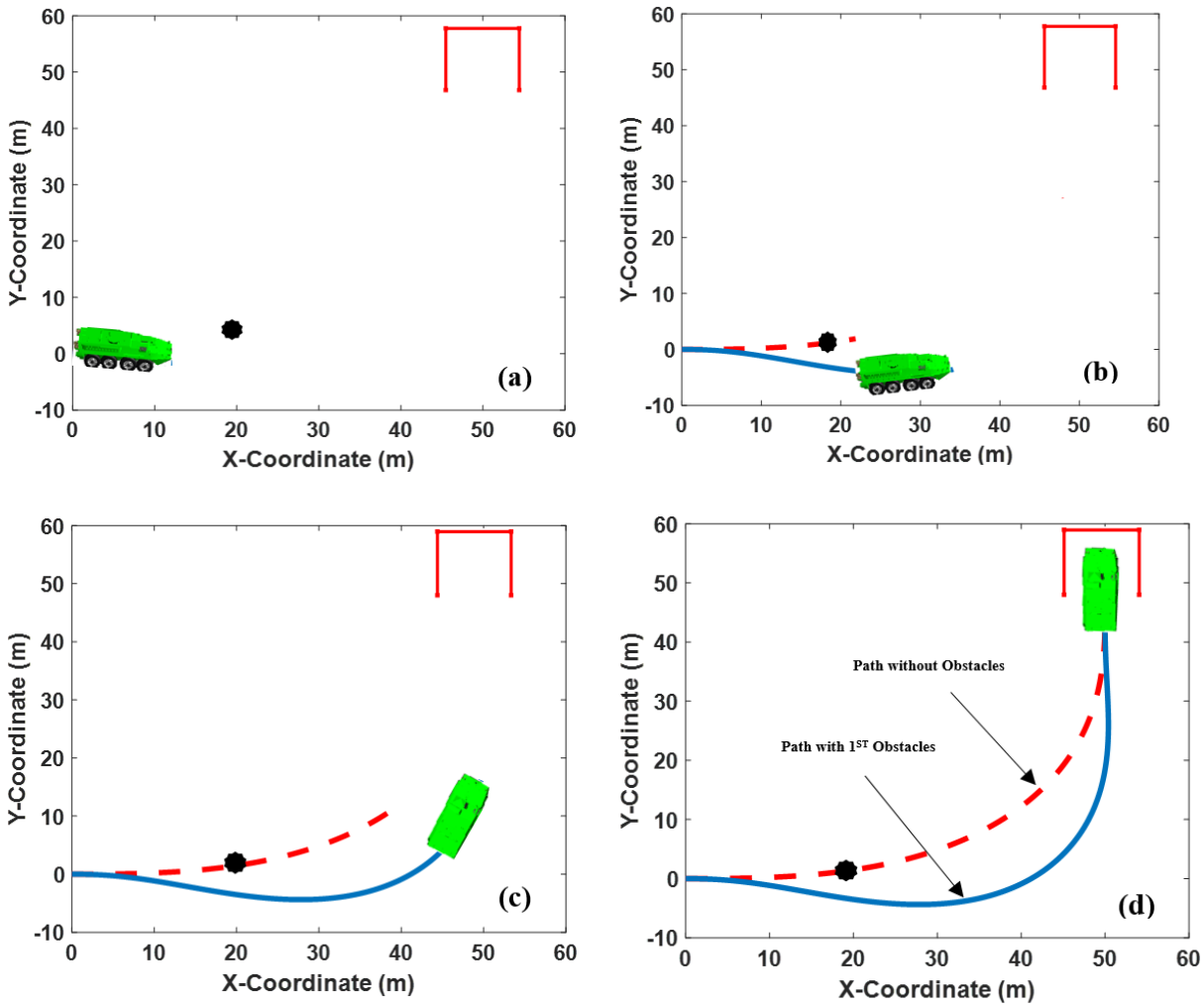


Figure 3-7 (a-d) Generated optimal path between two points and imposed 1st obstacle

3.5.2.2 Imposed two obstacles

In this scenario, two obstacles are imposed at (20, 1.6) and (50, 20) during the vehicle maneuver as shown in Figure 3-8. The simulation results show that the vehicle is able to reach the goal location safely while avoiding collision with the obstacles, generating its own optimal path.

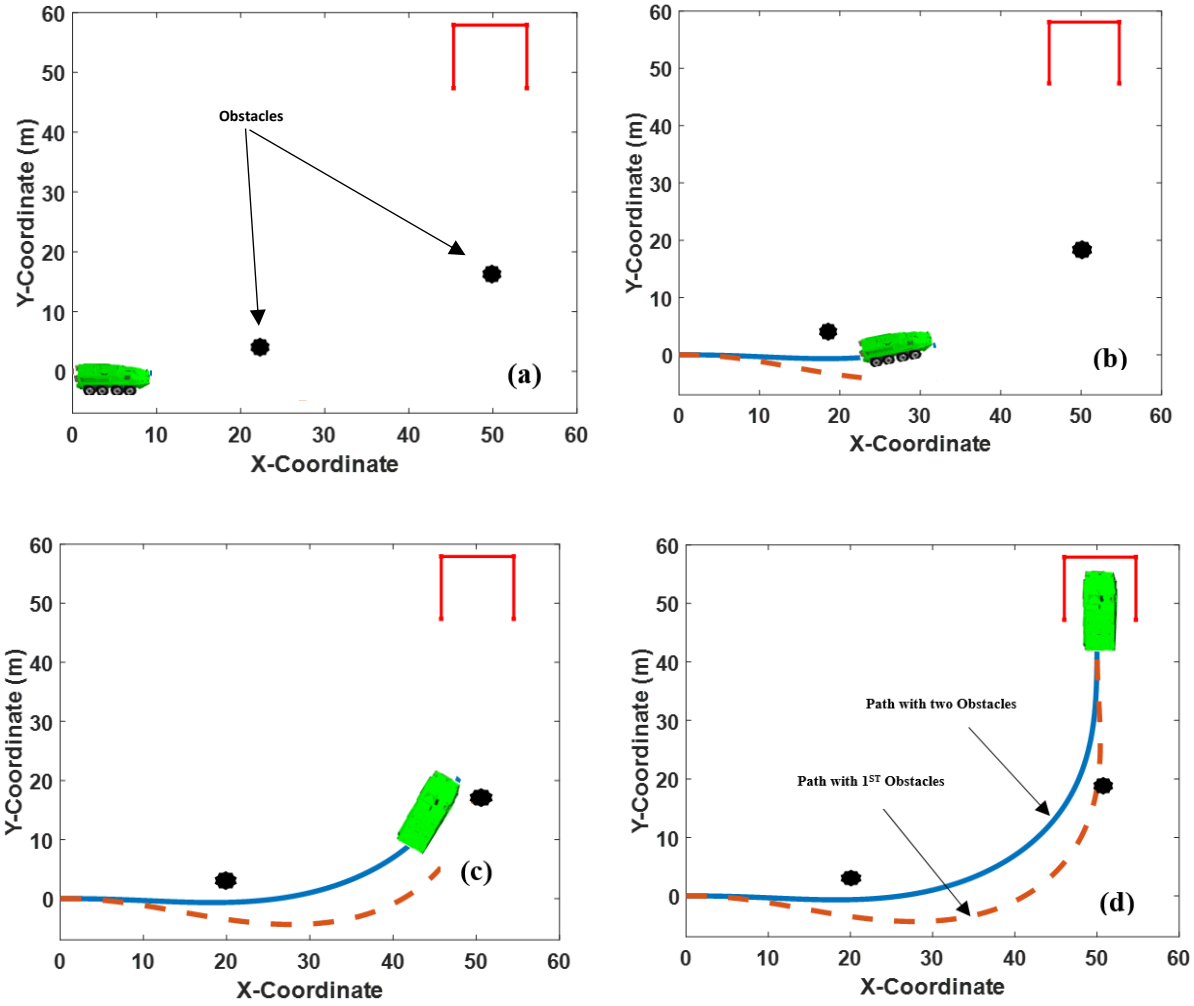


Figure 3-8 (a-d) Generated optimal path between two points and imposed 2nd obstacle

3.6 Chapter Summary

In this chapter, an optimal collision-free path planning for an autonomous multi-wheeled combat vehicle navigating through a cluttered environment has been presented. The developed algorithm is based on the integration between optimal control theory using PMP and an artificial potential field approach. The optimal control theory is successfully employed for generating the optimal path of the vehicle during the maneuver for a given starting and goal points, considering obstacles in the way. The proposed algorithm is applied to a simplified four-axle bicycle model of the actual vehicle considering the vehicle body lateral and yaw dynamics. This algorithm mainly controls the vehicle steering angle for generating the optimal path. Furthermore, an obstacle avoidance technique is

mathematically modeled based on APF for representing different obstacles using the Gaussian function which is used to simulate and represent the obstacles as repulsive potential filed.

Additionally, the obtained optimal path and the vehicle states were validated with a full nonlinear vehicle model in TruckSim software in terms of lateral acceleration, yaw rate, and curvature at two different longitudinal speed. Various numerical simulations are carried out using different locations of the obstacles. The simulation results show that the algorithm discussed in this chapter successfully generates the vehicle optimal collision-free path for the vehicle and is able to avoid the imposed obstacles at different locations.

CHAPTER 4

Real Time Path Planning and Navigation of a Multi-Wheeled Combat Vehicle Based Artificial Neural Networks

4.1 Introduction

This chapter presents a real time path planning algorithm for autonomous multi-wheeled combat vehicles based on ANN. The proposed architecture enables the vehicle to navigate autonomously from the initial location to the destination location in real time, while avoiding destination borders. The proposed ANN algorithm is based on the optimal control theory and artificial potential fields method, which are developed for path planning in Chapter 3. All possible optimal paths that cover every part of the workspace are generated using different starting points. Subsequently, these paths are used as training data for the proposed ANN considering the vehicle parameters in terms of yaw rate, lateral velocity, heading angle, and steering angle. The network is trained in offline mode using a backpropagation learning algorithm. The trained ANN model has the capability to control the movement of the combat vehicle in real time from any starting point to the desired goal location within the area of interest. The APF is introduced to prevent the vehicle from colliding with obstacles that are represented at the border of the destination. The effectiveness and efficiency of the proposed approach are demonstrated. The simulation results show that the proposed ANN model is capable of generating the optimal paths of the multi-wheeled combat vehicle.

Real time path planning for autonomous vehicle using ANN has many distinct advantages, which have been applied in autonomous driving [80-82]. For example, Ni [83] used bioinspired neural network to perform real time path planning for an autonomous underwater vehicle in a three-dimensional unknown environment. Rehder [84] developed a convolutional neural network to learn planning from imitation for path planning in complex traffic situations from both simulated and real-world data. In [85], the author introduced a neural network model via integer linear program formulation for obtaining

the optimal shortest path. [86-89] proposed the algorithm for solving shortest path problems.

4.2 Artificial Neural Networks (ANNs)

Artificial Neural Networks are very influential brain-inspired computational models, which have been employed in various areas such as computing, engineering, and many others. ANNs are composed of a certain number of simple computational elements called neurons. The neurons are organized into a structured graph topology made out of several consecutive layers and immensely interconnected through a series of links called synaptic weights. Synaptic weights are often associated with variable numerical values that can be adapted as to allow the ANN to change its behavior based on the problem being addressed [90]. Training an ANN is usually done by feeding the network's input with patterns to learn. The network then transmits the pattern through its weights and neurons until it generates a final output value.

Accordingly, the training or the learning algorithm compares the produced output value to the expected output. If the error range is high, the algorithm marginally alerts the network's weights so that if the same pattern were fed again to the network, the output error would be smaller than the previous iteration. This process will be repeated for many cycles called epochs, using a different set of input patterns until the network produces acceptable outputs for all inputs [91]. This learning progression allows the network to identify many patterns and further generalize to new and unseen patterns. Such a type of training is called supervised learning, which uses classified pattern information to train the network in offline mode.

4.3 Proposed Neural Network Model

The proposed ANN consists of three layers, an input layer, a hidden layer, and an output layer. The input layer has five input source nodes, which are physically fed by the multi-wheeled combat vehicle parameters in terms of position in X, Y, vehicle heading angle, vehicle yaw rate, and lateral velocity. The hidden layer has twenty neurons, which receive the data from the input layer. Subsequently, multiply it by the weights denoted by W_{ij} and

bias, then forward the result values to the output layer. The output layer has one neuron that directly represents the front wheel steering angle that controls the vehicle's motion. The weights W_{ij} represents the interconnection between the different neurons of the network.

The proposed neural network shown in

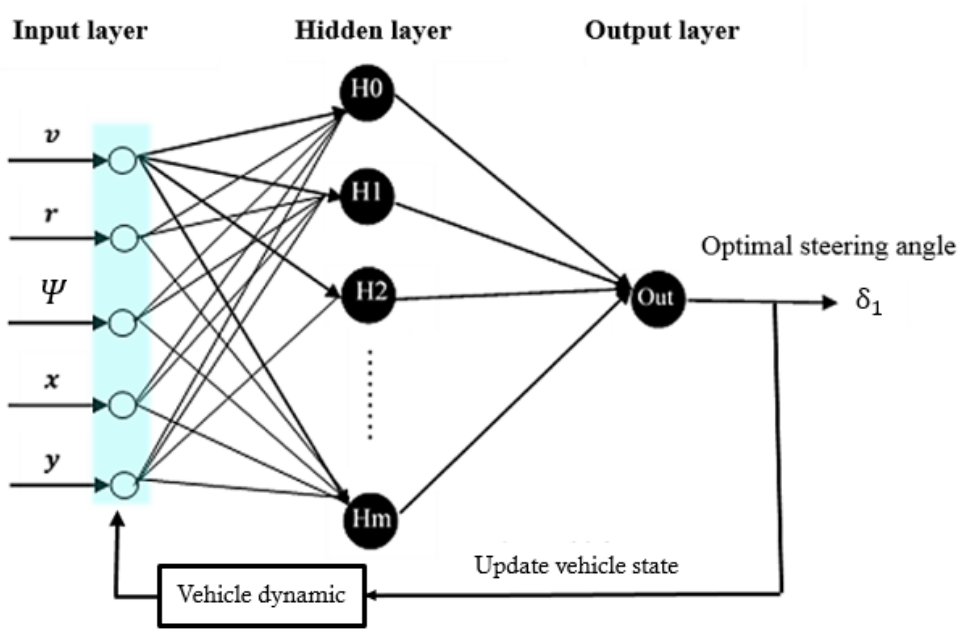


Figure **4-1 Architecture of the proposed ANN.** can be defined as follows:

NN = {I, O, T, W}, where I represents the set of input nodes, O represents the output node, T represents the topology of the network including the number of layers and the number of their neurons, and W represents the set of weights values.

I = {v, r, θ, x, y}

T = {Lin-5, Lh0-20, Lout-1}

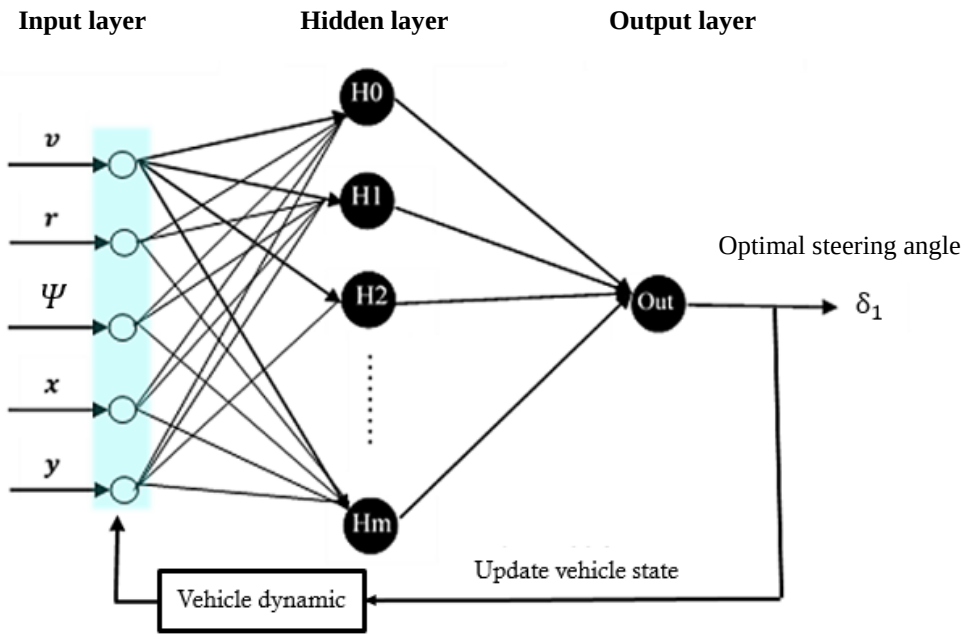


Figure 4-1 Architecture of the proposed ANN.

Before the ANN model can be trained, the generated optimal paths from chapter 3 should be collected. This is performed using different starting points in the workspace for the vehicle. After the paths data collection step is completed, the ANN is used to generate a learning model from the captured datasets through offline training.

4.3.1 Backpropagation algorithm

The proposed neural network model shown in

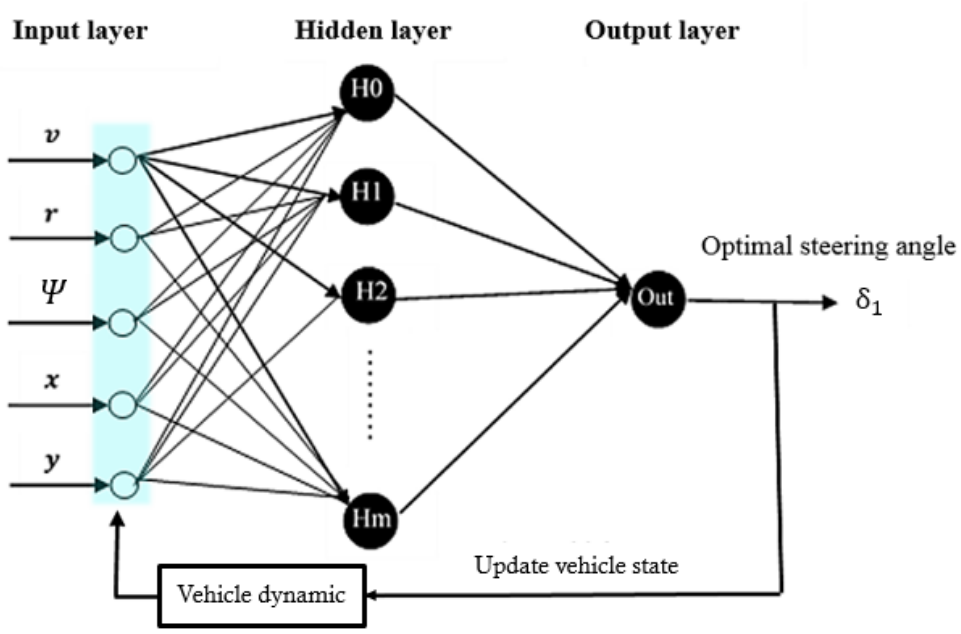


Figure 4-1 is trained through a supervised learning approach using the backpropagation algorithm [92]. The main purpose of using backpropagation algorithm is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs.

Backpropagation algorithm contains two passes in terms of the forward pass and backward pass. The forward pass propagates the input data in the forward direction from the input to output layers of the ANN, which ends by providing an output value. Afterword, it computes the error by subtracting the desired output from the actual output. If the error is within an acceptable range, the network is trained with a new set of input data; otherwise, the backward pass is executed. The backward pass propagates the error signal backward through the network layers in order to update the synaptic weights of the network.

Additionally, the developed ANN model is an iterative type as follows; first, it is initialized using the initial input data of the vehicle states. Subsequently, the corresponding steering wheel angle is calculated. The obtained steering wheel angle is used to update the vehicle states. The updated states are used to update the ANN model in each iteration to estimate

the new steering wheel angle of the moving vehicle for the next navigation step. The developed motion planning algorithm can be summarized as follows:

1. Specify the initial configurations of the vehicle states in terms of $V_i, r_i, \Psi_i, x_i, y_i$.
2. Initialize the ANN model using the initial input data in order to calculate the corresponding front steering wheel angle.
3. The obtained steering wheel angle from the ANN model is used to update the vehicle parameters, then inputs the updated parameter to the ANN model to return the new front steering wheel angle for the next vehicle movement.
4. Generate the optimal path for the multi-wheeled combat vehicle by repeating step 3 until the vehicle reaches its target position.

4.4 Simulation and Result

The proposed artificial neural network model is implemented using OpenNN. It is an open source class library written in C++ programming language, which implements neural networks models. The main advantage of this open source library is the high performance, which is developed in C++ for better memory management and higher processing speed. The ANN is compiled under the MS Visual Studio. It incorporates a training engine that can be fed with various static and dynamic parameters including initial weights, input data, learning rate, and the number of epochs to execute.

The developed ANN is tested and validated to generate the vehicle optimal path through the following three scenarios. The first scenario shows the obtained vehicle optimal path between two points compared with the training paths for testing. Afterwards, it is validated by choosing a different starting point in-between the training data. The second scenario examines the vehicle's optimal path from different starting points. The third scenario shows the effects of the vehicle longitudinal speed on the obtained paths.

4.4.1 First scenario

The main objective of this scenario is to check the capability of the proposed ANN model in terms of providing the optimal path for the vehicle from a given starting point to the goal point safely without hitting the destination border. Figure 4-2 shows the obtained optimal path using the ANN model represented in a dashed red line compared with the training data represented in blue for a starting point (0,14). The obtained result shows that the proposed ANN model has the capability to generate the optimal path in real time for the vehicle to reach its goal location through the safe and correct trajectory compared with the original optimal path without hitting the border.

Additionally, the time history of the vehicle parameter in terms of lateral velocity, heading and steering angles and yaw rate during this maneuver are shown in Figure 4-3 compared with the optimal path planning model.

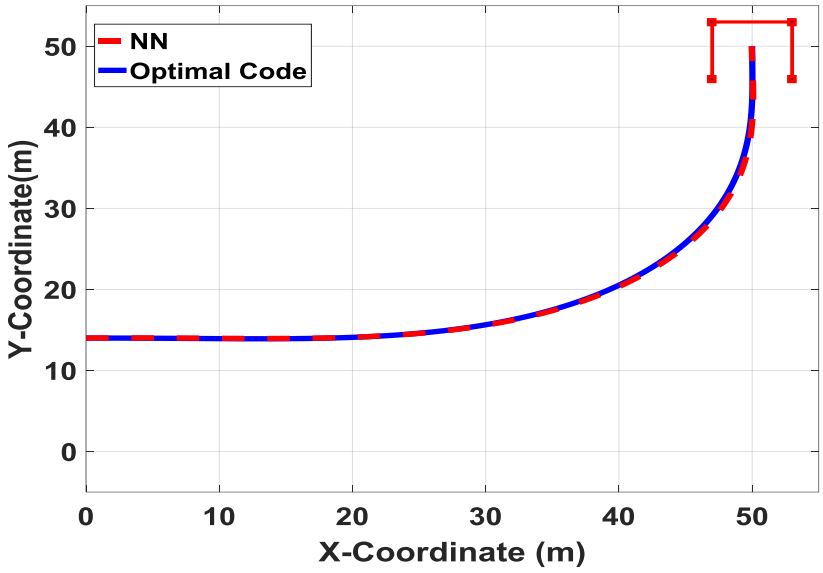


Figure 4-2 Obtained path for both NN model and optimal control for the same starting point

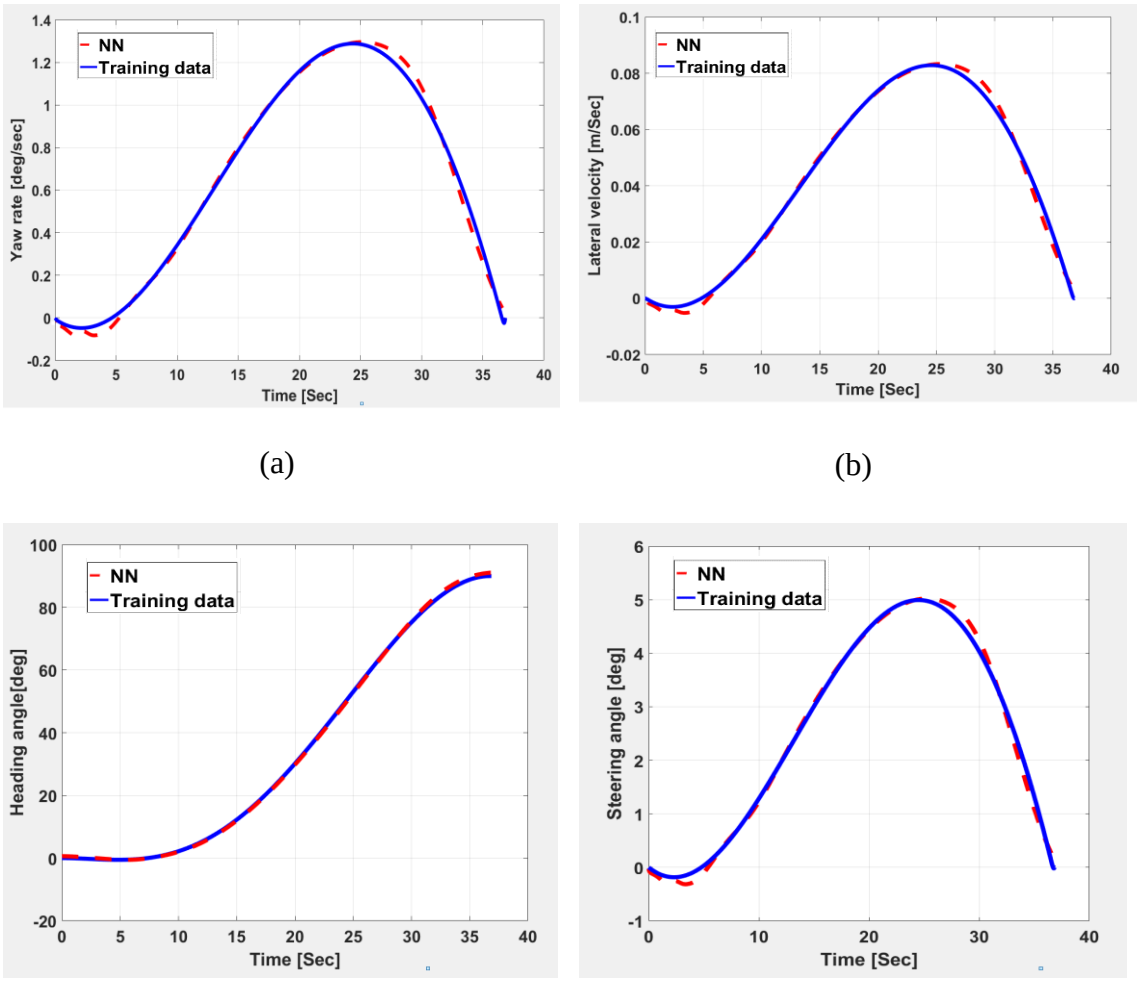


Figure 4-3 Comparison between time-history of the vehicle parameters obtained from NN model and Optimal control (trained data) for vehicle model: a) Yaw Rate (deg/s), b) lateral velocity (m/s), c) Heading angle (deg), d) Steering angle (deg).

The ANN is validated by the training data by generating multiple optimal paths in-between the training paths as shown in Figure 4-4. It has been noticed that the introduced ANN model has the capability to generate the vehicle optimal paths that behave the same as the training paths even choosing different starting points.

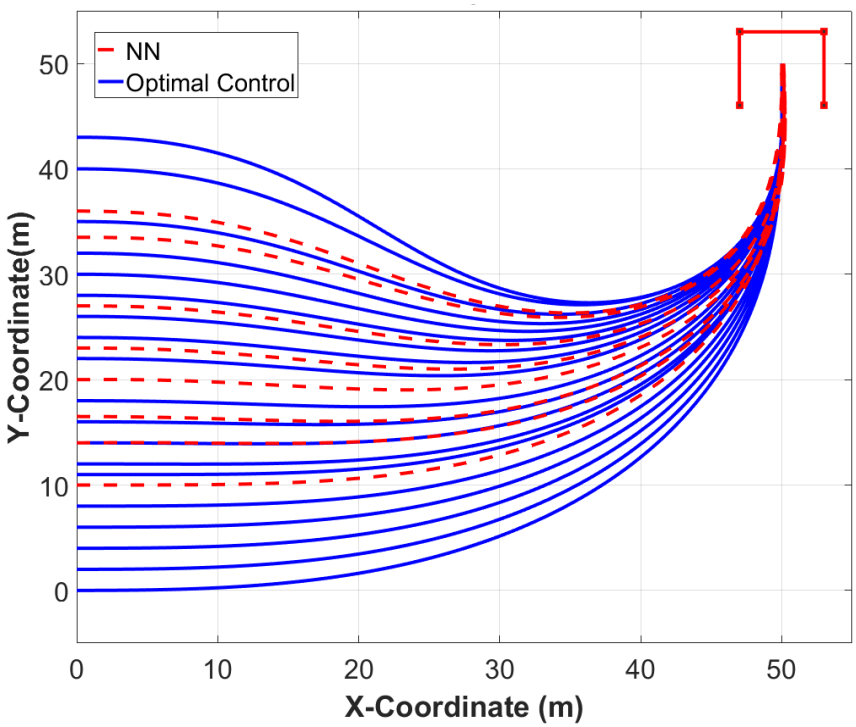


Figure 4-4 Optimal path between two points using NN and optimal control

4.4.2 Second scenario

In this scenario, the capability of the proposed ANN model is examined in terms of navigating the vehicle using various starting points in the workspace as shown in Figure 4-5. The obtained result shows that the vehicle is able to autonomously navigate from any point in the workspace to reach the goal point following the optimal path in red compared with optimal control theory result (training data) in blue.

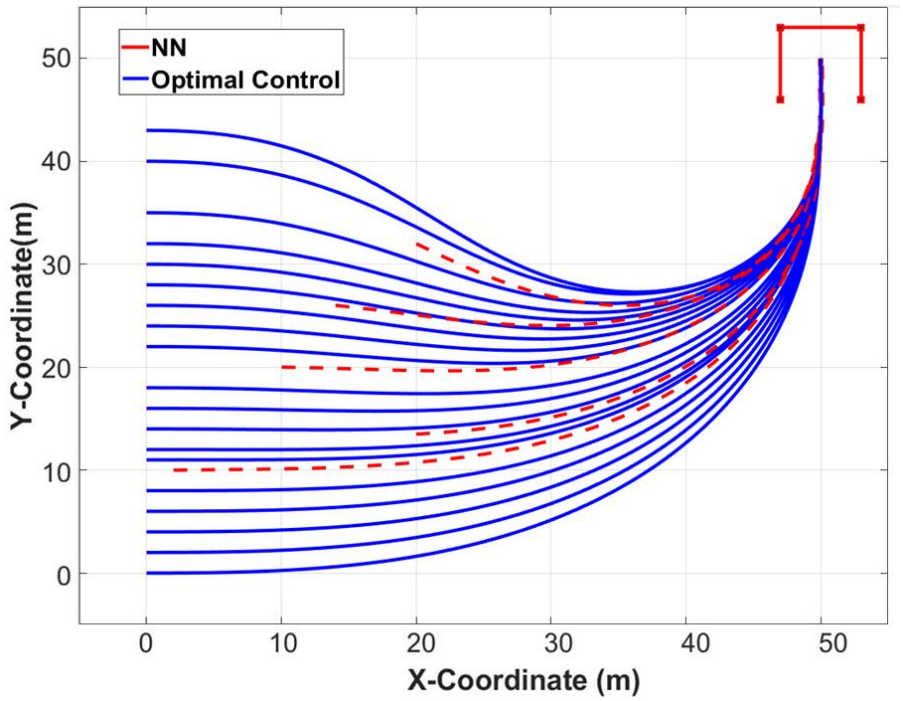


Figure 4-5 Generating optimal path from different starting point in the space using ANN and optimal control

4.4.3 Third scenario

This scenario aims to show the effect of the vehicle longitudinal speed on the obtained paths. For this purpose, the vehicle optimal path is generated at speeds 2,4,8 and,10 m/s using the MATLAB model discussed in Chapter 3. Afterword, it compared to the obtained results with the ANN path at speed 2 m/s as shown in Figure 4-6. The obtained result shows that the effect of increasing the longitudinal speed is very small within this range. Consequently, the obtained path using the ANN model is approximately satisfying the obtained paths at different speeds.

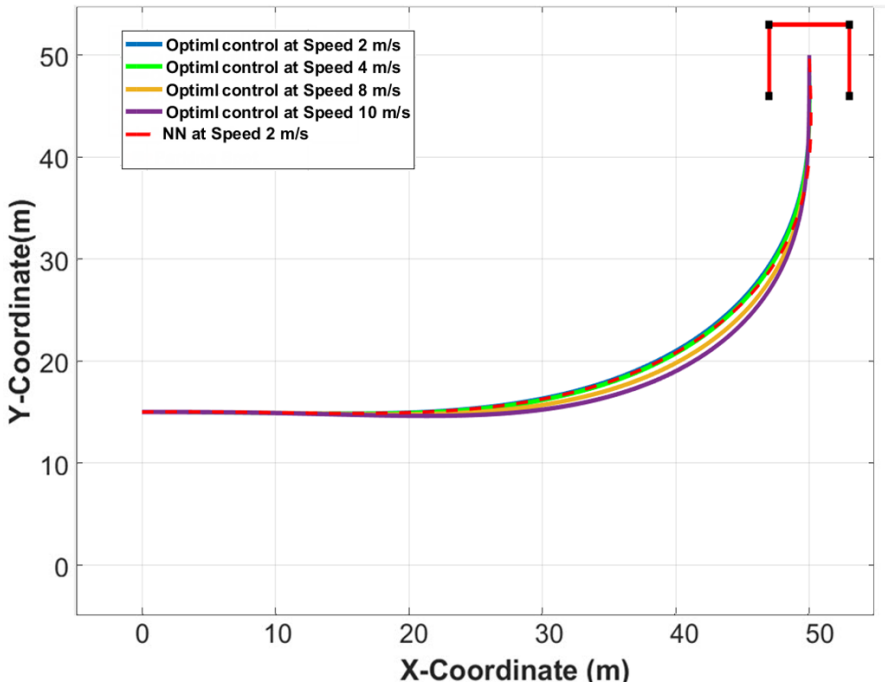


Figure 4-6 Generating optimal path at different longitudinal speed NN and optimal control

4.5 Chapter Summary

This chapter presented a motion planning algorithm for the multi-wheeled combat vehicle in real time based on an ANN , optimal control theory, and APF. The proposed ANN is a multilayer network, which is trained in offline mode using backpropagation learning algorithm. The training data is acquired based on optimal control theory discussed in the CHAPTER 3. The introduced ANN model is a significant improvement in the field of autonomous combat vehicle navigation and control, due to its capability to provide the vehicle optimal path in real time for this application.

The effectiveness of the ANN algorithm is validated through various numerical simulations considering different starting points in the workspace. The simulation results demonstrate that the proposed ANN model has the capability to optimally generate the vehicle path in real time to reach the goal location safely, while keeping the vehicle parameters including lateral velocity, heading and steering angles, and yaw rate identical to the training data.

CHAPTER 5

Comparative Study of Dynamic Programming and Pontryagin's Minimum Principle for Vehicle Path Planning

5.1 Introduction

This chapter presents a comparative study of two path planning algorithms based on the optimal control theory for the autonomous multi-wheeled combat vehicle. The developed optimal path planning algorithms use Pontryagin's Minimum Principle (PMP), which is discussed in Chapter 4 and the Dynamic Programming (DP) approach. PMP and DP are two major branches of the optimal control theory. Up to now, little work has been done to compare their performance in the application of navigating autonomous vehicles.

The simplified two DOF vehicle model is used to develop the differential equations of the multi-wheeled combat vehicle. The cost function associated with the path planning is minimized with the vehicle dynamics equations to satisfy vehicle dynamics and boundary conditions. The performance analysis of the obtained optimal paths is carried out considering various scenarios. The simulation results show that the obtained optimal solution using PMP is very close to the DP solution, which is the guaranteed global optimum. In addition, the initial and final condition parameters and the vehicle dynamics are satisfied. However, the computation time of the PMP is significantly less than that of the DP.

According to the review, there are two optimization approaches that seem best suited to autonomous vehicles path planning. The local and the global optimal path planning approaches include the Pontryagin's minimum principle and dynamic programming, respectively. Additionally, other approaches have been explored for obtaining the optimal path including methods such as Genetic Algorithms (GA) [93,94].

The Pontryagin's minimum principle was introduced in Chapter 3 as an optimal control solution [95-98], where the Hamiltonian is used as the mathematical function for generating the optimal collision-free path planning for the multi-wheeled combat vehicle.

The main advantage of using PMP is that the computation time is less when compared to other methods. On the other hand, DP algorithm is considered the most powerful numerical method for solving optimal control problems, which has been investigated in numerous research studies [99-103]. The main advantage of the DP approach compared with other methods is that the global optimality of the obtained outcome is guaranteed, regardless of the problem type. However, the computational effort increases exponentially with the number of state variables used as the required inputs associated with the underlying dynamic system. Furthermore, dealing with systems that have continuous state variable requires discretization of the time state space [104]. PMP and DP are two major branches of the optimal control theory. Little work has been done to compare their performance in the application of autonomous vehicles.

In this chapter, a DP algorithm is applied to generate the optimal paths for the multi-wheeled combat vehicle to autonomously navigate from its current location to the goal location. The simplified two DOF vehicle model was used to develop the differential equations of the vehicle. The cost function associated with the path planning problem is minimized to satisfy the initial, final conditions, and vehicle dynamics. A comparative study of the developed DP algorithm and PMP is carried out. The obtained path using the DP algorithm is the global optimum, which can be used as a reference or standard for other methods. The experiment is set up to autonomously navigate the multi-wheeled combat vehicle from the starting point to the goal point following the optimal path. The obtained results show that the generated path using the PMP algorithm is very close to the obtained path from DP. However, the computation time of the PMP method is significantly less than that of the DP method.

5.2 Global Optimal Path Planning Algorithm using DP Technique

Dynamic programming technique is a numerical optimization method that has the capability to solve the optimal control problem and provide the optimal solution based on the Bellman's principle of optimality [105]. Consequently, it guarantees the global optimum through an extensive search of all control and state grids [106-108]. This technique is based on the principle of optimality, which states the following: let u(k) be the

optimal control trajectory for the entire time range $\forall\in \left[t_0, t_f\right]$, and $k \in [\mathrm{k}_0, \mathrm{k}_n]$ such that the truncated control path $u(k), k \in [\mathrm{k}_1, \mathrm{k}_2]$, for $k_0 \leq \mathrm{k}_1 \leq \mathrm{k}_2 < \mathrm{k}_3 \leq \mathrm{k}_n$, is then the optimal control for this suboptimal problem [105]. Dynamic programming achieves this principle via three steps. First, dividing the optimal control problem into simpler discrete subproblems. Second, each suboptimal problem is locally optimized, finally obtaining the global optimal control path.

5.2.1 Dynamic programming algorithm

In the dynamic programming algorithm, the optimal path for a multistage decision can be defined [109]. Suppose that the first position is a and the next position is b. The results are in segment $a - b$ with cost J_{ab} and that the remaining position yields segment $b - z$ at a cost of J_{bz}. The minimum cost J_{az}^* from a to z is defined.

$$J_{az}^* = J_{ab} + J_{bz} \tag{5-1}$$

An nth-order time-invariant system described by the state equation will be as follows:

$$\dot{x}(t) = a(x(t), u(t)) \tag{5-2}$$

Subsequently, it is desired to determine the control law which minimizes the performance measure

$$J = h\left(x\left(t_f\right)\right) + \int_0^{t_f} g(x(t), u(t))dt \tag{5-3}$$

where t_f is the final time, $x\left(t_f\right)$ is the final state, h and g are scalare functions. The system is accomplished by considering N equally spaced time increments in the interval $0 \leq t \leq t_f$ and $\Delta t = \frac{t_f}{N}$, $t = k\,\Delta t$ which become for small Δt, and we shall denote by

$$\frac{x(t + \Delta t) - x(t)}{\Delta t} \approx a(x(t), u(t)) \tag{5-4}$$

$$x(t + \Delta t) = x(x) + \Delta t\, a\left(x(t), u(t)\right) = a_D(x(t), u(t)) \tag{5-5}$$

Then,

$$x(k+1) = x(k) + \Delta t\, a\big(x(k), u(k)\big) = a_D(x(k), u(k)) \tag{5-6}$$

Operating on the performance measure in a similar manner, we obtain

$$J = h\big(x(N\,\Delta t)\big) + \int_0^{\Delta t} g dt + \int_{\Delta t}^{2\Delta t} g dt + \ldots + \int_{(N-1)\Delta t}^{N\Delta t} g dt \tag{5-7}$$

For small Δt it becomes as follows:

$$J \approx h\big(x(N)\big) + \Delta t \sum_{k=0}^{N-1} g(x(k), u(k)) \tag{5-8}$$

Subsequently, denote by

$$J \approx h\big(x(N)\big) + \sum_{k=0}^{N-1} g_D(x(k), u(k)) \tag{5-9}$$

Define J_{NN} is the cost of reaching the final state value $x(N)$, and

$$J_{NN} = h(x(N)) \tag{5-10}$$

Then,

$$J_{N-1,N}\big(x(N-1), u(N-1)\big) = g_D(x(N-1), u(N-1)) + J_{N,N}\,(x(N)) \tag{5-11}$$

Consequently, the optimal cost of operation during the interval $(N-1)\Delta t \le t \le N\Delta t$

$$\begin{aligned} J^*_{N-1,N}\big(x(N-1)\big) &= \min_{u(N-1)} \{\, g_D(x(N-1), u(N-1)) \\ &\quad + J_{N,N}\,(a_D(x(N-1), u(N-1))\} \end{aligned} \tag{5-12}$$

Continuing backward in the same manner, the result for a k-stage process can be obtained as follows:

$$J^*_{N-k,N}\big(x(N-k)\big) = \min_{u(N-k)} \{\, g_D(x(N-\mathrm{k}), u(N-k)) + J^*_{N-(k-1),N}(a_D(x(N-k), u(N-k))\} \quad 5\text{-}13$$

The underlying system was earlier described by the dynamic Equations 3-13 with the following state constraints included:

$$\begin{aligned} &\Psi(t) \in [0, \pi] && \forall \in [0, t_f] \\ &x(t) \in [0{,}100] && \forall \in [0, t_f] \\ &y(t) \in [0{,}100] && \forall \in [0, t_f] \\ &\delta(t) \in [0{,}1] && \forall \in [0, t_f] \end{aligned} \quad 5\text{-}14$$

Consider the same initial and final conditions used in Pontryagin's minimum principle technique in order to generate the path under the same situations.

Consequently, from Equation 5-13 the recurrence relation can be obtained for the multiple optimal trajectories. The DP algorithm was implemented in MATLAB and applied in order to attain the optimal control solution.

5.2.2 Comparative analysis for the results from the DP and PMP techniques

5.2.2.1 First scenario

The generated optimal paths using both techniques are shown in Figure 5-1. The vehicle starts to navigate from (0,0) to reach the goal location at (50,50), and the final heading angle is 90 degrees. It can be noticed that the obtained two optimal paths using both techniques are very close to each other.

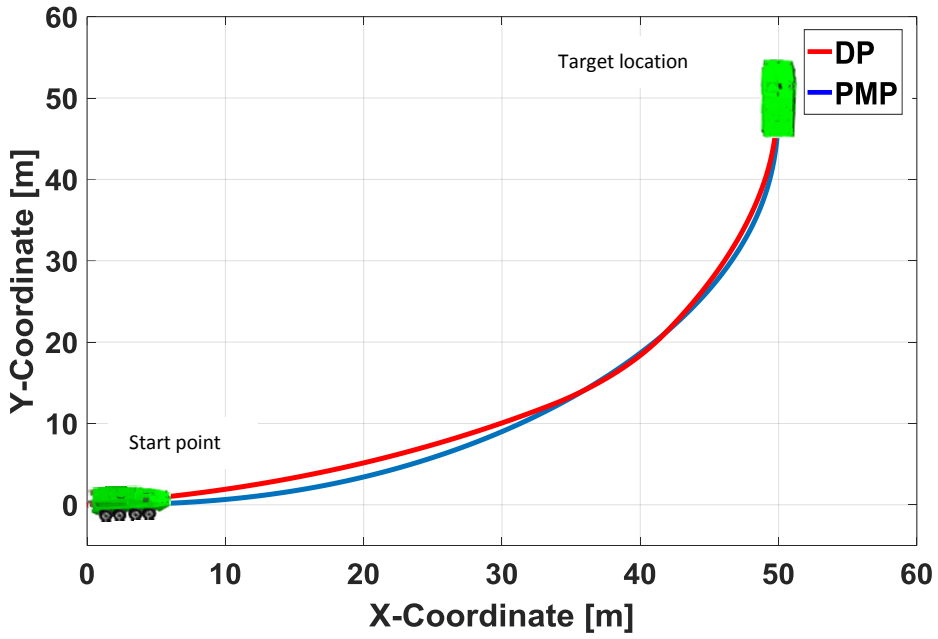


Figure 5-1 Generated optimal path between two points using DP and PMP

A comparative analysis of the time history of the vehicle states in terms of longitudinal and lateral displacements, and heading and average front wheels steering angle during this maneuver is shown in Figure 5-2. The longitudinal and lateral displacement time history for the two algorithms are close to each other as shown in Figure 5-2 a, b. In addition, the time history of the heading angle for the two algorithms is close to each other as shown in Figure 5-2 c. However, the minor difference determined was that the generated control effort using PMP was lower than DP, which makes the steering angle to be smoother as shown in Figure 5-2 d.

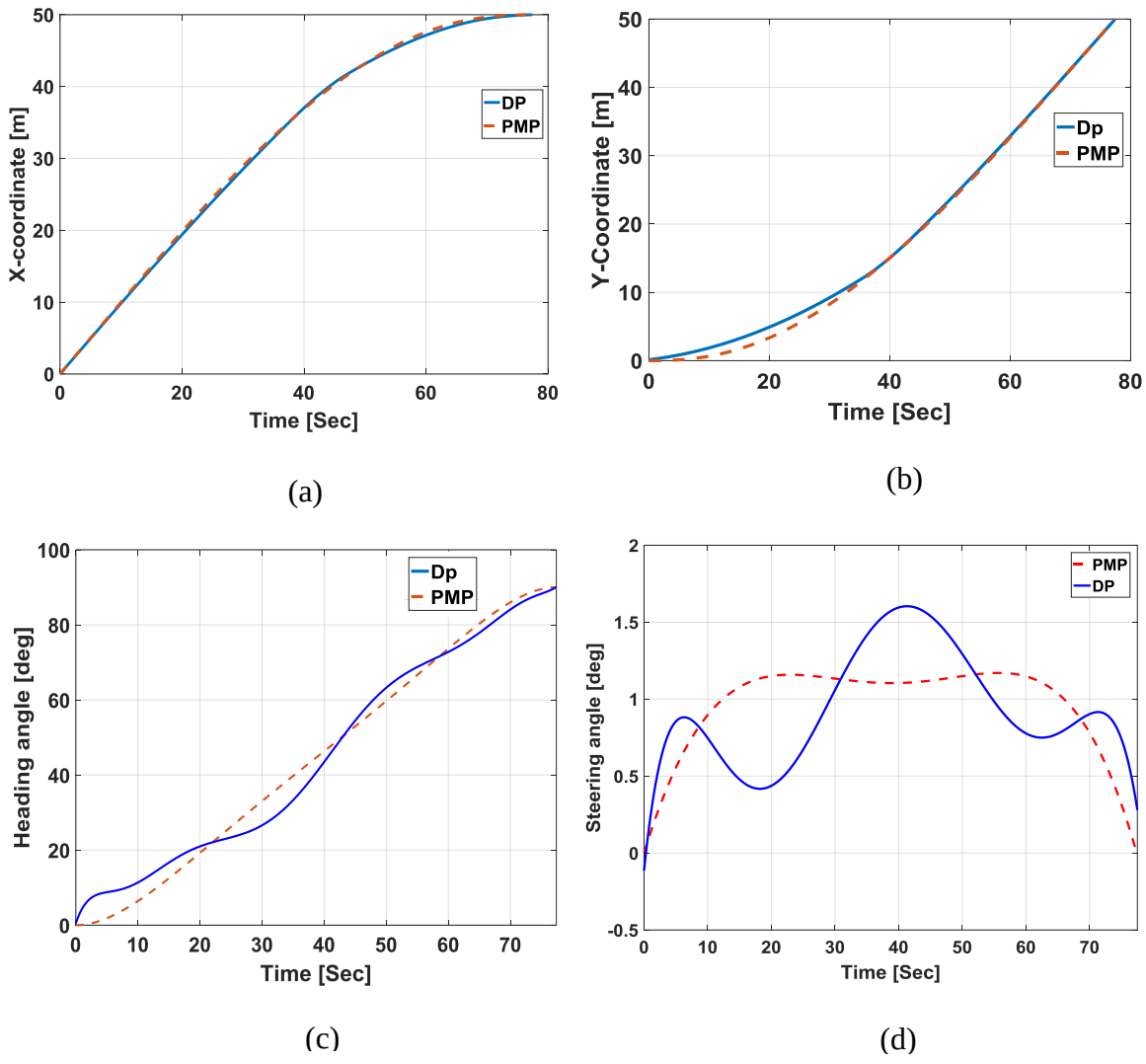


Figure 5-2 Vehicle parameter using DP and PMP starting at (0,0): a) longitudinal displacement (m), b) lateral displacement (m), c) heading angle (deg), d) steering angle (deg).

5.2.2.2 Second scenario

In this scenario, the vehicle starting location is (10,0). The obtained optimal paths using both techniques are shown in Figure 5-3. It can be noticed the generated path using PMP is still close to that in the DP at this maneuver. In addition, the time history of the vehicle states is shown in Figure 5-4.

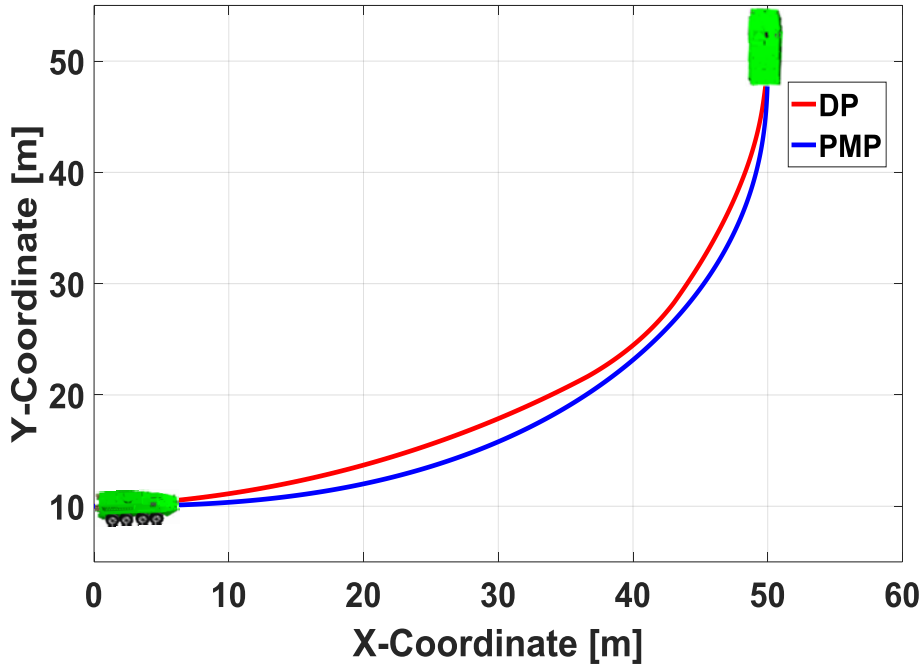
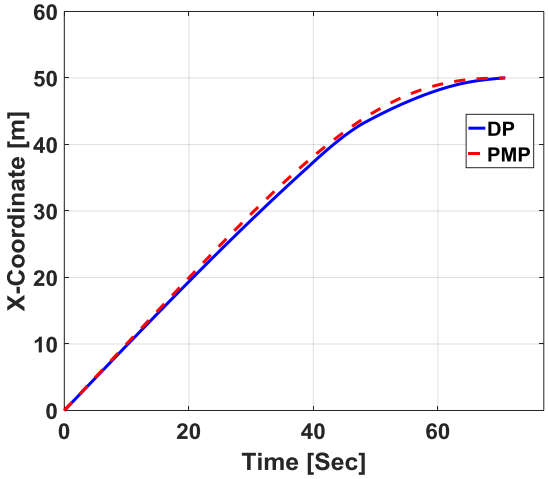
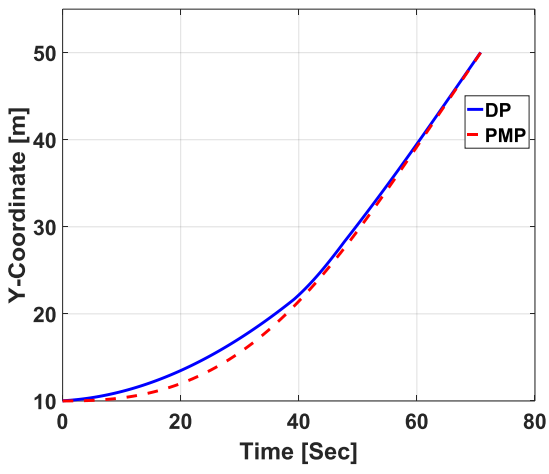


Figure 5-3 Generated optimal path between two points using DP and PMP

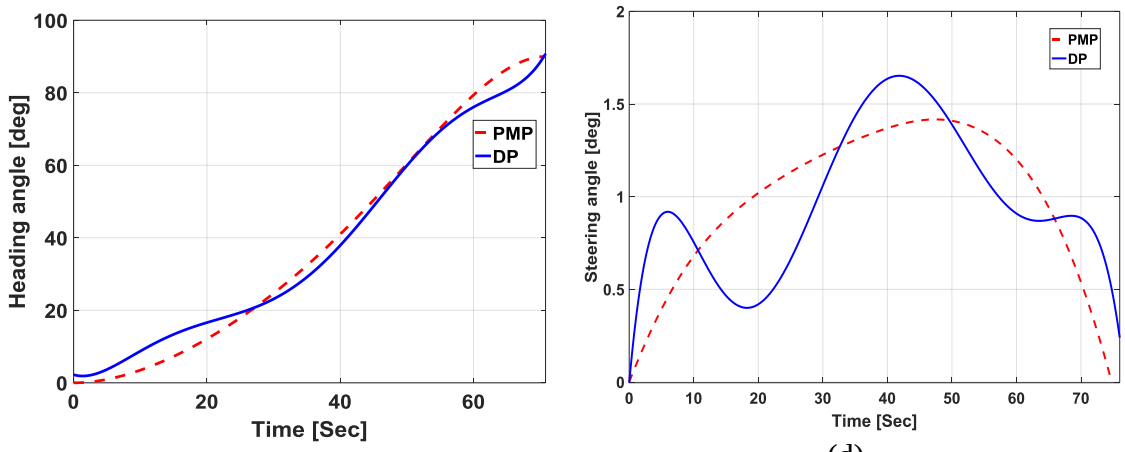
Comparing the time history of the vehicle states demonstrated that the longitudinal and lateral displacements using PMP and DP agree with each other as shown in Figure 5-4 a, b. Nonetheless, it was determined that the control effort generated by PMP approach is smaller than that of the DP. An illustration of the control effort differences can be shown in the steering angle in Figure 5-4 d.



(a)



(b)



(c) (d)

Figure 5-4 Vehicle parameter using DP and PMP starting at (0,10): a) longitudinal displacement (m), b) lateral displacement (m), c) heading angle (deg), d) steering angle (deg).

5.3 Chapter Summary

In this chapter, the problem of obtaining the optimal path for the multi-wheeled combat vehicle moving between two points is formulated and then dynamic programming approach is applied to solve this problem. The simulation results of the generated paths using both algorithms are compared, which demonstrates that the generated paths using PMP is close to the DP, while DP consumes more calculation time. In addition, the PMP is demonstrated to be effective in generating near-optimal results, close to those of DP that guarantee the global optimal solution, which is considered the major contribution in this chapter. Furthermore, the PMP can save up to 70% of the time of DP, including the iteration to approximate the co-states, while successfully satisfying the initial and final condition parameters.

CHAPTER 6

Hybrid Positioning Technique-Based Integration of GPS/INS for an Autonomous Vehicle Navigation

6.1 Introduction

This chapter presents a hybrid framework for the positioning technique combining both loosely and tightly coupled Kalman Filter (KF) algorithms. This framework fuses information from a Global Positioning System (GPS) and an Inertial Navigation System (INS) sensors and creates a navigation system for an autonomous multi-wheeled combat vehicle. The developed algorithm is able to provide accurate and reliable vehicle positioning information even if the number of visible satellites is less than four, due to the harsh vehicle operation environments. Therefore, two coupled algorithm modes were considered and combined as a hybrid framework; the first mode is loosely coupled and the second is tightly coupled. If the number of visible satellites degrades to be less than four, the first mode will not be able to provide a reliable measurement update to correct the drift of the INS raw data. Consequently, the developed hybrid framework will switch to the second mode, where the aiding of the INS with the raw GPS measurements is still possible. The proposed hybrid algorithm increases the reliability and the robustness of the position information by providing the GPS with KF aiding, particularly, when the effective number of satellites falls below the minimum number. The modelling strategies and the data fusion process for these GPS/INS integration scenarios are discussed with numerical analysis results, demonstrating the potential performance of the proposed integration. A simulation of the proposed hybrid positioning algorithm is performed, first using a Satellite Navigation (SatNav) toolbox in MATLAB and second using collected data from sensors mounted on a ground vehicle tracking a pre-defined route. The information from the GPS and INS data were fused. The experimental results validated the feasibility and effectiveness of the proposed algorithm.

The development of navigation systems is one of the challenges in autonomous vehicle design, where the GPS and INS are the most widely used navigation sensors. These sensors

can be used individually or integrated [111-113]. The main advantage of using an INS in autonomous ground vehicles is that the acceleration, angular rotation, and attitude data are provided at high update rates. Consequently, the vehicle velocity and position can be estimated easily. On the other hand, the GPS receiver is an absolute low frequency sensor which can provide the state information at low update rates [114]. The combination of these two sensors will increase the reliability of the position information for the autonomous vehicle.

Generally, the integration between data from several sensors is known as data fusion or sensor fusion. This integration improves the accuracy, which is not achievable when utilizing each sensor separately. Data fusion techniques have been widely applied to multisensory environments including cameras, GPS, and IMU. The goal of using data fusion is to reduce the possibility of detectable errors and to obtain a higher rate of reliability by using data from multiple distributed sources.

KFs are widely used in integrated navigation systems to combine the obtained data from the navigation sensors. Consequently, KF is considered a powerful mathematical tool for analyzing and solving localization estimation problems [115-119]. It was originally proposed by Kalman [44] and has been widely studied and applied since then. It is mainly employed to fuse low-level data.

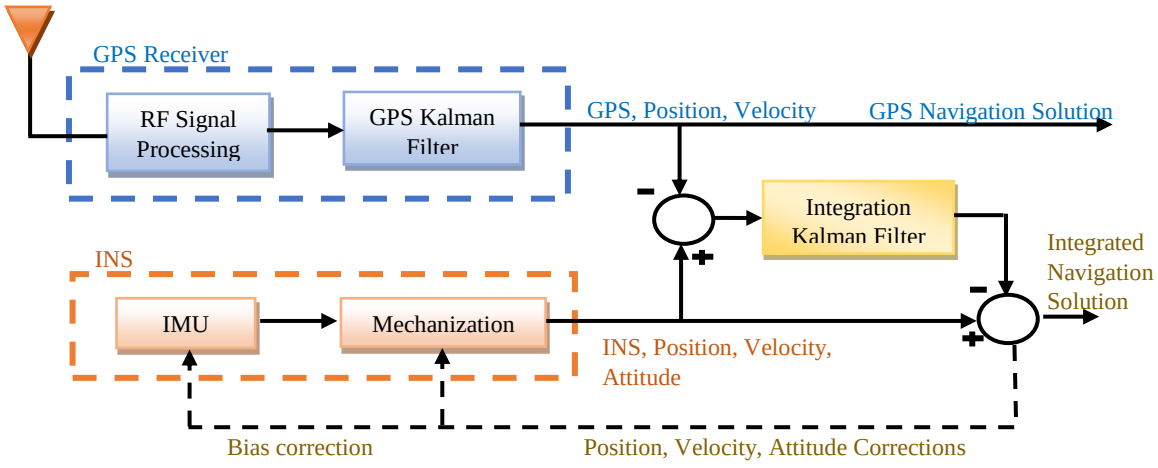
6.2 GPS/INS Hybrid Positioning Technique

In this section, the methodology of the proposed hybrid positioning technique is described. In order to achieve an accurate and robust positioning estimation for the autonomous multi-wheeled combat vehicle, a hybrid positioning technique based on a KF method is proposed. The developed technique fuses the data from the low-cost GPS and INS sensors. It has two modes of operation, which can provide an accurate navigation solution during the vehicle maneuver by updating the mode selection according to the GPS signal and the available number of the satellites as follows.

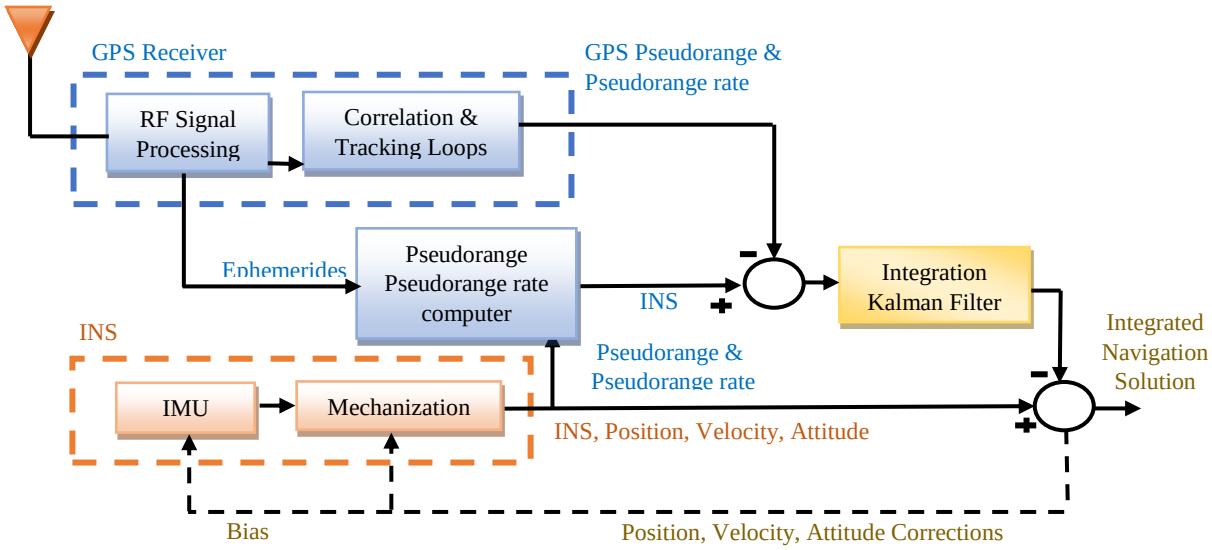
If the GPS signal is available, and the number of visible satellites is sufficient, the developed framework will be operated in the first mode based on loosely coupled KF

GPS/INS integration as shown in Figure 6-1 (a). In this case, the GPS and INS will operate independently to provide two independent navigation solutions for position, velocity, and attitude. In order to get the third solution, which is the best of both, the obtained data from the GPS and INS will be fed to KF (fused together). By taking the difference between the collected data and based on the error models, the INS errors can be estimated. Based on the estimated errors, the INS solution will be corrected and consequently will provide the integrated navigation solution. The main feature of using loosely coupled GPS/INS integration is using two separate KF, i.e. the GPS and the integration filter, which used in cascaded integration.

On the other hand, if the GPS signal lacks credibility and the effective number of satellites is less than four, the developed framework will switch to the second mode, which is based on tightly coupled KF, as shown in Figure 6-1 (b). Both GPS and INS work as a basic sensor providing their raw outputs of pseudorange and pseudorange rate measurements. The difference between the pseudorange and pseudorange rate measurements from the GPS and the INS are processed directly in the navigation KF to estimate the INS errors. Consequently, according to the estimated errors, the INS output can be corrected, and the integrated navigation solution can be obtained. In this way, aiding the INS with the raw GPS measurement is possible even if the number of visible satellites is below the minimum [120]. Consequently, using the proposed hybrid algorithm has the benefit for the combat vehicle navigation system when the satellite signals are difficult to receive.



(a)



(b)

Figure 6-1 Proposed hybrid positioning methodology. (a) The first, mode loosely coupled KF; (b) the second, mode tightly coupled KF mode

6.3 Construction of Kalman Filtering Algorithm

In this section, the construction of KF for the proposed hybrid framework is described.

6.3.1 Enhance GPS using Kalman filter

KF is proposed to improve and enhance the GPS data before fusing it with INS data. The continuous-time system state equation of KF can be expressed as follows:

$$\dot{X}(t) = F(t)X(t) + G(t)W(t) \qquad 6\text{-}1$$

where

$F(t)$ is called the dynamic matrix, which propagates the errors over time,

$X(t)$ is the state vector.

$G(t)$ is the noise distribution matrix,

$W(t)$ is the random forcing functions.

Consequently, based on the discretizing method in [121], the discrete time linear system equation can be expressed as follows:

$$X_k = \varphi_{k,k-1}\ X_{k-1} + G_{k-1}W_{k-1} \qquad 6\text{-}2$$

where

X_k is the state vector,

$\varphi_{k,k-1}$ is the state transition matrix,

G_{k-1} is the noise distribution matrix,

W_{k-1} is the process noise vector, k is the measurement epoch.

Then, the state vector $\boldsymbol{X}$ will be defined as follows:

$$\boldsymbol{X} = [\delta_N\ \delta_E\ \delta_U\ \ \delta_{VN}\ \ \delta_{VE}\ \ \delta_{VU}\ \delta_{Offset}\ \ \delta_{drift}\ ]^T$$

where

$[\delta_N \delta_E \delta_U]$ is denote the north, east and up positions respectively,

$[\delta_{VN}\ \delta_{VE}\ \delta_{VU}]$ is denote the north, east and up velocities respectively,

$\left[\delta_{Offset}\ \ \delta_{drift}\right]$ are the receiver clock offset and drift.

The initial state vector prediction is $\hat{X}_0$=[0 0 0 0 0 0 0 0 0]^T, and the initial prediction error covariance matrix P_0= $\begin{bmatrix} 10 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 10 \end{bmatrix}$

Considering Equation 6-1, the dynamic matrix will be considered as follows:

$F = \begin{bmatrix} 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{bmatrix}$, then, the overall system will be represented as follows:

$$\begin{bmatrix} \dot{\delta}_N \\ \dot{\delta}_E \\ \dot{\delta}_U \\ \dot{\delta}_{VN} \\ \dot{\delta}_{VE} \\ \dot{\delta}_{VU} \\ \dot{\delta}_{Offset} \\ \dot{\delta}_{drift} \end{bmatrix} = \begin{bmatrix} 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{bmatrix} \begin{bmatrix} \delta_N \\ \delta_E \\ \delta_U \\ \delta_{VN} \\ \delta_{VE} \\ \delta_{VU} \\ \delta_{Offset} \\ \delta_{drift} \end{bmatrix} + \begin{bmatrix} 1&0&0&1&0&0&0&0 \\ 0&1&0&0&1&0&0&0 \\ 0&0&1&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \begin{bmatrix} W_N \\ W_E \\ W_U \\ W_{VN} \\ W_{VE} \\ W_{VU} \\ W_{Offset} \\ W_{drift} \end{bmatrix} \quad 6\text{-}3$$

The state transition matrix in Equation 6-4 represents the known dynamic behavior of the system, which relates the state vector from epoch $k-1$ to epoch k.

$$\varphi = e^{F.\Delta t} \cong \mathrm{I} + F\,\Delta t \quad 6\text{-}4$$

where

I is the identity matrix.

Δt is the sampling interval

The discrete-time of the power spectral density can be expressed as follows:

$$Q_k = \int_0^{\Delta t} \varphi\, G \quad Q\, G^T \varphi^T\, dt \quad 6\text{-}5$$

Subsequently, the discrete time linear measurement equation of the system is as follows:

$$Z_k = \; H_k\, X_k + \xi_k, \quad 6\text{-}6$$

where Z_k is the measurement vector of the sensor output, ξ_k is the white gaussian observation noise for the sensor with zero mean with the obtained covariance matrix $R_k = E\left[\xi_k\, {\xi_k}^T\right]$, $H_k$is the measurement design matrix associated with the sensor.

From the model described in Equations 6-2, 6-5, and 6-6, the KF can be computed as a prediction stage and update stage [122-124] as follows.

- **The Prediction Stage.**

In this stage the initial state $\hat{X}_k$ is estimated from the epoch $k-1$ to k, then the covariance of the prediction state P_k is calculated. The prediction stage is defined by Equations 6-7 and 6-8.

$$\hat{X}_k = \varphi_{\boldsymbol{k,k-1}}\hat{X}_{\boldsymbol{k-1}} \quad 6\text{-}7$$

$$P_k = \varphi_{\boldsymbol{k,k-1}}P_{k-1}{\varphi_{\boldsymbol{k,k-1}}}^{\boldsymbol{T}}+Q_{\boldsymbol{k-1}} \quad 6\text{-}8$$

- **The Update Stage**

In this stage, the first step is to compute the Kalman gain K_k. This is depending on the a priori covariance P_k , the measurement noise covariance R_k, and the measurement design matrix H_k. If P_k is higher than R_k then the gain is higher. However, if R_k is higher, then the gain is lower, and vice versa. The second step is to correct the estimated state $\hat{X}_k$, whenever a measurement is received. This is based on the difference of the predicted measurements and the actual measurements as follows:

$$\hat{X}_k = \hat{X}_{k-1} + K_k[z_k - H_k\,\hat{X}_{k-1}] \quad 6\text{-}9$$

where the Kalman gain for the data associated to the sensor is expressed as follows:

$$K_k = P_k{H_k}^T(H_kP_k{H_k}^T + R_k)^{-1} \quad 6\text{-}10$$

and the corrected covariance matrix $P_k = [I - K_kH_k]P_{k-1}$.

6.3.2 GPS/INS integration

Considering the loosely coupled GPS/INS integration KF, the state vector consists of fifteen states in terms of three position states ($\delta_\varphi\delta_\lambda\delta_h$), three velocity states ($\delta_{VE}\delta_{VN}\delta_{VU}$), three attitude states ($\delta_p\delta_r\delta_A$) in east, north and up directions, three accelerometer biases ($\delta_{f_x}\delta_{f_y}\delta_{f_z}$), and three gyro drifts ($\delta_{\omega_x}\delta_{\omega_y}\delta_{\omega_z}$) as follows:

$$X = \left[\delta_\varphi\delta_\lambda\delta_h\delta_{VE}\delta_{VN}\delta_{VU}\delta_p\delta_r\delta_A\delta_{f_x}\delta_{f_y}\delta_{f_z}\,\delta_{\omega_x}\delta_{\omega_y}\delta_{\omega_z}\right]^T \quad 6\text{-}11$$

The INS accuracy is affected by several sources, which can include errors during the initial alignment, sensor errors, and the processing algorithm. These errors affect the navigational solution in terms of position, velocity, and attitude as follows:

- **Position Error for Local Level Frame**

The position error δ_r^l for the Local Level Frame (LLF) in simple form is can be obtained as follows:

$\dot{\delta}_r^l = D^{-1}\delta_v^l$ (applying the orientation of LLF w.r.t earth),

where D^{-1} transforms the velocity vector from rectangular coordinates into curvilinear coordinates in the Earth-Centered Earth-Fixed Frame (ECEF).

Consequently, the generalized form of the position error is as follows:

$$\dot{\delta}_r^l = \begin{bmatrix} \delta_{\dot{\varphi}} \\ \delta_{\dot{\lambda}} \\ \delta_{\dot{h}} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{R_M+h} & 0 \\ \frac{1}{(R_N+h)\mathrm{Cos}(\varphi)} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta_{v_n} \\ \delta_{v_e} \\ \delta_{v_u} \end{bmatrix} \quad \text{6-12}$$

where v_n, v_e, v_u are the velocity component, φ, λ, h are the velocity component R_N is the normal radius of curvature for the east-west direction, and R_M is the meridian radius of curvature for the north-south direction.

- **The Velocity Error for Local Level Frame**

The velocity error for Local Level Frame (LLF) can be obtained as follows:

$$\dot{\hat{V}}^l = R_b^l f^b - \left(2\Omega_{ie}^l + \Omega_{el}^l\right)\hat{V}^l + g^l \quad \text{6-13}$$

where

$\hat{V}^l$ is the velocity vector in the local level frame,

R_b^l is the transformation matrix from the body-frame to the inertial-frame,

f^b is the specific force measured by the accelerometers in the body-frame, Ω_{ie}^l and Ω_{el}^l are the skew-symmetric matrices corresponding to the rotation of the Earth about its spin axis and the change of orientation of the local-level frame with respect to the Earth respectively.

Taking into account the error δ in the measurements, where $\hat{\dot{V}}^l = \dot{V}^l - \delta\dot{V}^l$.

$$\delta\dot{V}^l = R_b^l\left(f^b + \delta f^b\right) - \left[2\left(\Omega_{ie}^l + \delta\Omega_{ie}^l\right) + \left(\Omega_{el}^l + \delta\Omega_{el}^l\right)\right](V^l + \delta V^l) + (g^l + \delta g^l) \qquad 6\text{-}14$$

After algebraic manipulation of Equation 6-14, the velocity error for Local Level Frame (LLF) is as follows in Equation 6-15:

$$\delta\dot{V}^l = -\boldsymbol{F}^l\varepsilon^l + R_b^l\delta f^b - 2\left(\Omega_{ie}^l + \Omega_{el}^l\right)\delta V^l + \boldsymbol{V}^l\left(2\delta\omega_{ie}^l + \delta\omega_{el}^l\right) + \delta g^l \qquad 6\text{-}15$$

where $\boldsymbol{F}^l$ is the skew-symmetric matrix corresponding to specific force, ω_{ie}^l is the angular velocity vector in the LLF obtained from the rotation of the earth about its spin axes. $[\omega^e = 15\ deg/h]$, then $\omega_{ie}^l = \begin{bmatrix} 0 \\ \omega_e.\cos(\varphi) \\ \omega_e.\sin(\varphi) \end{bmatrix}$. Ω_{ie}^l is the skew-symmetric matrix corresponding to ω_{ie}^l. $\delta\omega_{ie}^l$is the error in the rotation rate of the Earth. Ω_{el}^l is the skew-symmetric matrix corresponding to $\omega_{el}^l = \begin{bmatrix} -\dot{\varphi} \\ \dot{\lambda}.\cos(\varphi) \\ \dot{\lambda}.\sin(\varphi) \end{bmatrix}$ where $\dot{\lambda} = \frac{v_e}{(R_N+h)\cos(\varphi)}$. $\delta\omega_{el}^l$ is vector of the error in the angular velocity vector ω_{el}^l. $\boldsymbol{V}^l$ is the skew-symmetric matrix of the corresponding velocity vector $\hat{V}^l$.

The generalized form of each term of Equation 6-15 can be obtained as follows:

- First term $F^l\varepsilon^l$

$$\boldsymbol{F}^l\varepsilon^l = \begin{bmatrix} 0 & -f_u & f_n \\ f_u & 0 & -f_e \\ -f_n & f_e & 0 \end{bmatrix}\begin{bmatrix} \delta_p \\ \delta_r \\ \delta_a \end{bmatrix} \qquad 6\text{-}16$$

where, f_e, f_n, f_u are the body acceleration in east, north and up directions.

- Second term $(R_b^l\delta f^b)$

The accelerometer biases δf^b are transformed from the body frame to the LLF using the R_b^l matrix as follows:

$$R_b^l \delta f^b = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \delta_{fx} \\ \delta_{fy} \\ \delta_{fz} \end{bmatrix} \quad \text{6-17}$$

• Third term $(-2(\Omega_{ie}^l + \Omega_{ie}^l)\delta_v^l)$

$$-2(\Omega_{ie}^l + \Omega_{ie}^l)\delta V^l = -2\left\{ \begin{bmatrix} 0 & -\omega_e \sin(\varphi) & \omega_e \cos(\varphi) \\ \omega_e \sin(\varphi) & 0 & 0 \\ -\omega_e \cos(\varphi) & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\lambda} \sin(\varphi) & \dot{\lambda} \cos(\varphi) \\ \dot{\lambda} \sin(\varphi) & 0 & \dot{\varphi} \\ -\dot{\lambda} \cos(\varphi) & -\dot{\varphi} & 0 \end{bmatrix} \right\} \begin{bmatrix} \delta_{ve} \\ \delta_{vn} \\ \delta_{vu} \end{bmatrix} \quad \text{6-18}$$

Rearranging,

$$-2(\Omega_{ie}^l + \Omega_{ie}^l)\delta V^l \quad \text{6-19}$$
$$= \begin{bmatrix} 0 & (2\omega_e + \dot{\lambda})\sin(\varphi) & -(2\omega_e + \dot{\lambda}) \cos(\varphi) \\ -(2\omega_e + \dot{\lambda}) \sin(\varphi) & 0 & \dot{\varphi} \\ (2\omega_e + \dot{\lambda}) \cos(\varphi) & \dot{\varphi} & 0 \end{bmatrix} \begin{bmatrix} \delta_{ve} \\ \delta_{vn} \\ \delta_{vu} \end{bmatrix}$$

• Fourth term $(2\delta\omega_{ie}^l + \delta\omega_{el}^l)\boldsymbol{V}^l$

From the definitions of ω_{ie}^l, ω_{el}^l in Equation 6-15, the $\delta_{\omega_{ie}}^{\ l} + \delta_{\omega_{el}}^{\ l}$ can be obtained as follows:

$$\delta\omega_{ie}^l = \begin{bmatrix} 0 \\ -\omega_e \delta\varphi \sin(\varphi) \\ \omega_e \delta\varphi \cos(\varphi) \end{bmatrix} \delta\omega_{el}^l = \begin{bmatrix} -\dot{\delta\varphi} \\ -\dot{\lambda} \delta\varphi \sin(\varphi) + \delta\dot{\lambda} \cos(\varphi) \\ \dot{\lambda} \delta\varphi \cos(\varphi) + \delta\dot{\lambda} \sin(\varphi) \end{bmatrix} \quad \text{6-20}$$

Subsequently, the fourth term can be obtained by applying algebraic manipulation to be as follows:

$$\left(2\delta\omega_{ie}^{l}+\ \delta\omega_{el}^{l}\right)\boldsymbol{V}^{l}=\begin{bmatrix}2\omega_e(v_u sin(\varphi)+v_n cos(\varphi))+\frac{\dot{\lambda}v_u}{cos(\varphi)} & 0 & 0\\ -2\omega_e v_e\, cos(\varphi)-\frac{\dot{\lambda}v_e}{cos(\varphi)} & 0 & 0\\ -2\omega_e v_e\, sin(\varphi) & 0 & 0\end{bmatrix}\begin{bmatrix}\delta_\varphi\\ \delta_\lambda\\ \delta_h\end{bmatrix}+ \quad 6\text{-}21$$

$$\begin{bmatrix}\frac{-v_n}{R_N+h}+\frac{v_n\,tan(\varphi)}{R_N+h} & 0 & 0\\ \frac{-v_e\,tan(\varphi)}{R_N+h} & \frac{-v_n}{R_M+h} & 0\\ \frac{v_e}{R_N+h} & \frac{v_n}{R_M+h} & 0\end{bmatrix}\begin{bmatrix}\delta v_e\\ \delta v_n\\ \delta v_u\end{bmatrix}$$

- Fifth term

The term δg^l is the error in normal gravity due primarily to the error in the altitude

$$\delta g^l=\begin{bmatrix}-g/r\\ -g/r\\ 2g/(\mathrm{r}+h)\end{bmatrix}\begin{bmatrix}\delta_\varphi\\ \delta_\lambda\\ \delta_h\end{bmatrix} \quad 6\text{-}22$$

where g represents the normal component of gravity, r is the mean radius of the Earth and h is the height.

Finally, the third and the fourth terms can be neglected due to the multiplication of the velocities by the earth rotation rate and/or divided the velocities by the earth radius and Equation 6-15 can be rewritten as follows:

$$\begin{bmatrix}\delta\dot{v}_e\\ \delta\dot{v}_n\\ \delta\dot{v}_u\end{bmatrix}=-\begin{bmatrix}0 & -f_u & f_n\\ f_u & 0 & -f_e\\ -f_n & f_e & 0\end{bmatrix}\begin{bmatrix}\delta_p\\ \delta_r\\ \delta_a\end{bmatrix}+\begin{bmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{bmatrix}\begin{bmatrix}\delta_{fx}\\ \delta_{fy}\\ \delta_{fz}\end{bmatrix} \quad 6\text{-}23$$

- **<u>The Attitude Error for Local Level Frame</u>**

The attitude error for Local Level Frame (LLF) can be obtained

$$\dot{\varepsilon}^l=-\Omega_{il}^{l}\varepsilon^l-\delta\omega_{il}^{l}+R_b^l\delta\omega_{ib}^{b} \quad 6\text{-}24$$

where the vector ε^l is the attitude errors $[\delta p,\delta r,\delta A]^T$. ω_{il}^{l} is the angular velocity of the LLF w.r.t the inertial frame. ω_{ib}^{b} is the angular velocity of the body frame to the LLF. Ω_{il}^{l}

is the skew-symmetric matrix corresponding to the vector ω_{il}^{l}. R_b^l is the transformation matrix from the body frame and the LLF.

After algebraic manipulation and neglect some terms due to divided by earth radius the simplified time rate of change of the attitude errors can be written as follows:

$$\dot{\varepsilon}^l = \begin{bmatrix} \dot{\delta}p \\ \dot{\delta}r \\ \dot{\delta}A \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{R_M + h} & 0 \\ \frac{-1}{R_M + h} & 0 & 0 \\ \frac{-tan(\varphi)}{R_M + h} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{ve} \\ \delta_{vn} \\ \delta_{vu} \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \delta\omega_x \\ \delta\omega_y \\ \delta\omega_z \end{bmatrix} \quad 6\text{-}25$$

- **Accelerometer bias error**

For the accelerometer, the time rate of change of the accelerometer bias errors can be expressed as follows:

$$\begin{bmatrix} \dot{\delta}_{fx} \\ \dot{\delta}_{fy} \\ \dot{\delta}_{fz} \end{bmatrix} = \begin{bmatrix} -\beta_{fx} & 0 & 0 \\ 0 & -\beta_{fy} & 0 \\ 0 & 0 & -\beta_{fz} \end{bmatrix} \begin{bmatrix} \delta_{fx} \\ \delta_{fy} \\ \delta_{fz} \end{bmatrix} + \begin{bmatrix} \sqrt{2\beta_{fx}\sigma_{fx}^2} \\ \sqrt{2\beta_{fy}\sigma_{fy}^2} \\ \sqrt{2\beta_{fz}\sigma_{fz}^2} \end{bmatrix} w(\mathrm{t}) \quad 6\text{-}26$$

where,

$\beta_{fx}, \beta_{fy}, \beta_{fz}$ are the reciprocals of the correlation times associated with the autocorrelation sequence of $\delta_{fx}, \delta_{fy}, \delta_{fz}$,

$\sigma_{fx}^2, \sigma_{fy}^2, \sigma_{fz}^2$ are the variances associated with the accelerometer errors,

$w(\mathrm{t})$ is white Gaussian noise.

- **Gyro bias error**

For gyroscope, the time rate of change of the bias errors can be can be expressed as follows:

$$\begin{bmatrix}\dot{\delta}_{\omega x}\\ \dot{\delta}_{\omega y}\\ \dot{\delta}_{\omega z}\end{bmatrix}=\begin{bmatrix}-\beta_{wx} & 0 & 0\\ 0 & -\beta_{wy} & 0\\ 0 & 0 & -\beta_{wz}\end{bmatrix}\begin{bmatrix}\delta_{\omega x}\\ \delta_{\omega y}\\ \delta_{\omega z}\end{bmatrix}+\begin{bmatrix}\sqrt{2\beta_{\omega x}\sigma_{\omega x}^2}\\ \sqrt{2\beta_{\omega y}\sigma_{\omega y}^2}\\ \sqrt{2\beta_{\omega z}\sigma_{\omega z}^2}\end{bmatrix}w(t) \quad \text{6-27}$$

where

$\beta_{\omega x}, \beta_{\omega y}, \beta_{\omega z}$ are the reciprocals of the correlation times associated with the autocorrelation sequence of $\delta_{\omega x}, \delta_{\omega y}, \delta_{\omega z}$.

$\sigma_{\omega x}^2, \sigma_{\omega y}^2, \sigma_{\omega z}^2$ are the variances associated with the gyroscope errors.

Based on the above equations, consequently, the state equation in Equation 6-1 can be written as follows:

$$\begin{bmatrix}\dot{\delta}_{\varphi}\\ \dot{\delta}_{\lambda}\\ \dot{\delta}_{h}\\ \dot{\delta}_{ve}\\ \dot{\delta}_{vn}\\ \dot{\delta}_{vu}\\ \dot{\delta}_{p}\\ \dot{\delta}_{r}\\ \dot{\delta}_{A}\\ \dot{\delta}_{f_x}\\ \dot{\delta}_{f_y}\\ \dot{\delta}_{f_z}\\ \dot{\delta}_{\omega_x}\\ \dot{\delta}_{\omega_y}\\ \dot{\delta}_{\omega_z}\end{bmatrix}=\begin{bmatrix}A11 & A12 & A13 & A14 & A15\\ A21 & A22 & A23 & A24 & A25\\ A31 & A32 & A33 & A34 & A35\\ A41 & A42 & A43 & A44 & A45\\ A51 & A52 & A53 & A54 & A55\end{bmatrix}\begin{bmatrix}\delta_{\varphi}\\ \delta_{\lambda}\\ \delta_{h}\\ \delta_{ve}\\ \delta_{vn}\\ \delta_{vu}\\ \delta_{p}\\ \delta_{r}\\ \delta_{A}\\ \delta_{f_x}\\ \delta_{f_y}\\ \delta_{f_z}\\ \delta_{\omega_x}\\ \delta_{\omega_y}\\ \delta_{\omega_z}\end{bmatrix}+\begin{bmatrix}0&0&0&0&0&0&0&0&0&\sqrt{2\beta_{fx}\sigma_{fx}^2}&\sqrt{2\beta_{fy}\sigma_{fy}^2}&\sqrt{2\beta_{fz}\sigma_{fz}^2}&\sqrt{2\beta_{\omega x}\sigma_{\omega x}^2}&\sqrt{2\beta_{\omega y}\sigma_{\omega y}^2}&\sqrt{2\beta_{\omega z}\sigma_{\omega z}^2}\end{bmatrix}\begin{bmatrix}w_{\varphi}\\ w_{\lambda}\\ w_{h}\\ w_{ve}\\ w_{vn}\\ w_{vu}\\ w_{p}\\ w_{r}\\ w_{A}\\ w_{f_x}\\ w_{f_y}\\ w_{f_z}\\ w_{\omega_x}\\ w_{\omega_y}\\ w_{\omega_z}\end{bmatrix} \quad \text{6-28}$$

where F(t) is 15 X 15 matrix where $A11$ to $A55$ are 3 X 3 matrices (details in Appendix 1).

Subsequently, the measurement equation according to Equation 6-6 will be as follows:

$$Z(t)=[H_1\ H_1\ \ldots.H_{15}\ ]\left[\delta_{\varphi}\delta_{\lambda}\delta_{h}\delta_{VE}\delta_{VN}\delta_{VU}\delta_{p}\delta_{r}\delta_{A}\delta_{f_x}\delta_{f_y}\delta_{f_z}\,\delta_{\omega_x}\delta_{\omega_y}\delta_{\omega_z}\right]^T+\ \xi \quad \text{6-29}$$

where $H_{1\ldots15}$ is the measurement matrix associated with the sensor and can be determined as follows:

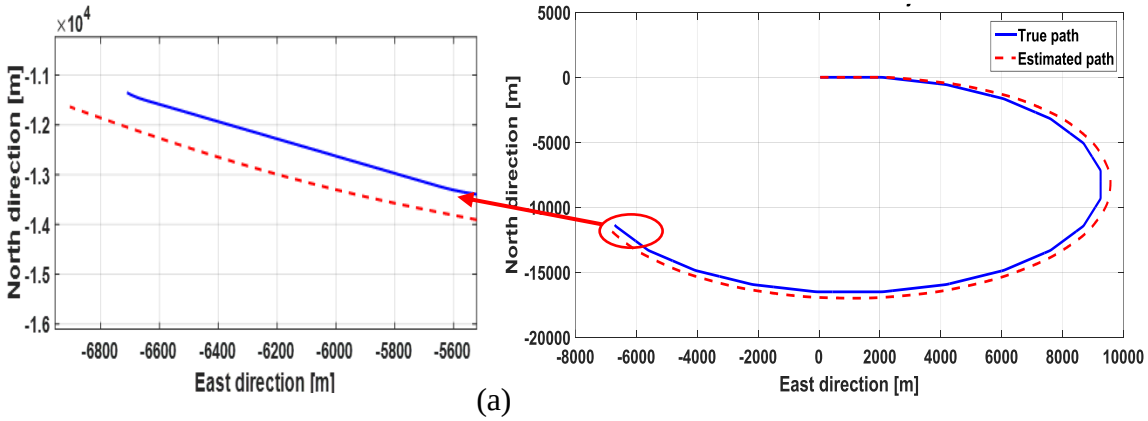
$$\begin{aligned}H=[&(\varphi_{INS}-\varphi_{GPS})\quad(\lambda_{INS}-\lambda_{GPS})\quad(h_{INS}-h_{GPS})\quad(VE_{INS}-VE_{GPS})\ (VN_{INS}\\ &-VN_{GPS})\ (VU_{INS}-VU_{GPS})\ (p_{INS}-p_{GPS})\ (r_{INS}-r_{GPS})\ (A_{INS}\\ &-A_{GPS})\ (fx_{INS}-fx_{GPS})\ (fy_{INS}-fy_{GPS})\ (fz_{INS}\\ &-fz_{GPS})\ (\omega x_{INS}-\omega x_{GPS})\ (\omega y_{INS}-\omega y_{GPS})\ (\omega z_{INS}-\omega z_{GPS})\ ]^T\end{aligned} \quad \text{6-30}$$

6.4 Tests and Results

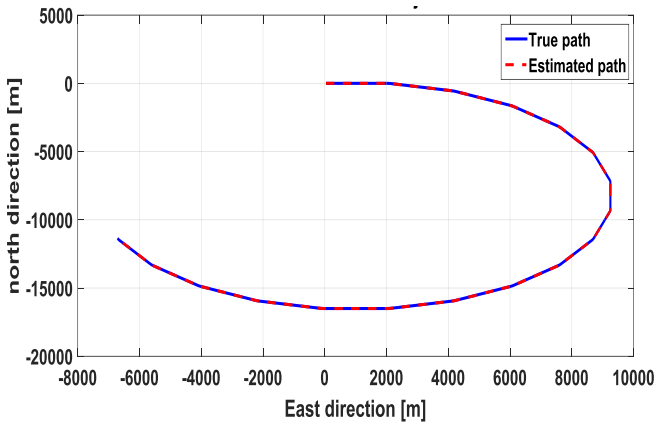
In this section, the numerical simulation and field test for the proposed technique are presented. The performance evaluation in the field test is conducted using real data from sensors fitted with a ground vehicle. This vehicle is equipped with 48 all-in-view tracking channels BU-353-S4 waterproof SiRFIV GPS receiver with 1Hz update rate. In addition, the attitude and heading reference system UM6 ultra-miniature orientation sensor is used, which has rate gyros, accelerometers, and magnetic sensors to compute sensor orientation at 500 times per second.

6.4.1 Simulation test

The simulation test is carried out using the SatNav toolbox in MATLAB. SatNav toolbox is mainly designed for the navigation purpose, which simulates the satellites and receivers in addition to the standalone positioning algorithms. In this work, the SatNav is used to simulate the obtained raw data from the GPS. First the route that will be used as a reference is prepared for the evaluation using SatNav, then the proposed technique is applied with and without KF. The trajectory of the obtained navigation solution without KF is shown in Figure 6-2 a, where the division from the true route is illustrated. On the other hand, Figure 6-2 b shows the estimated path after adding the designed KF to the GPS, where the obtained result is almost identical to the true route.



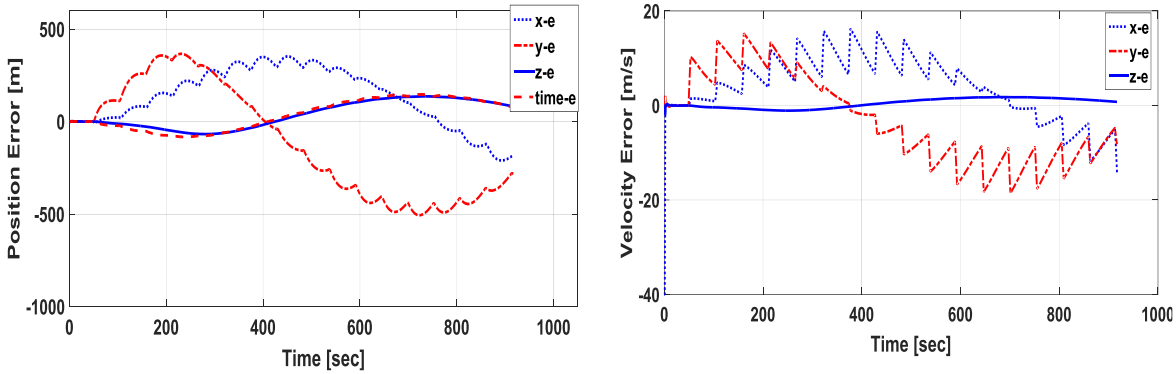
(a)



(b)

Figure 6-2 Comparing the reference route to the obtained trajectories: (a) GPS trajectory without KF, (b) GPS trajectory with KF

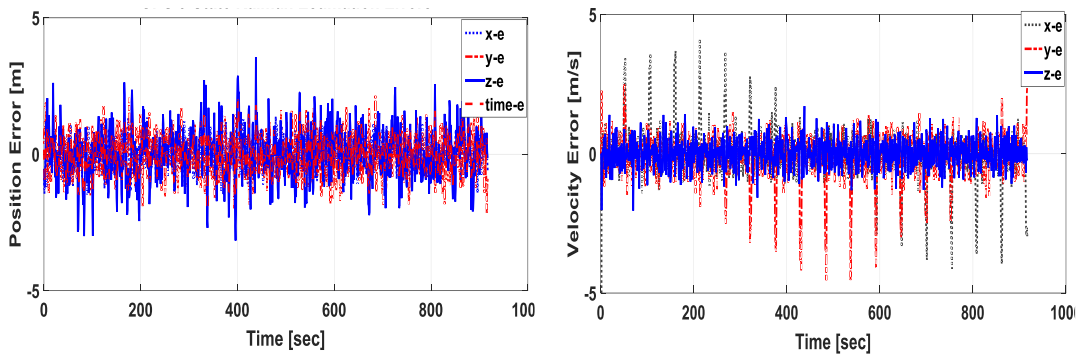
In addition, the average error in position without KF is approximately 100 m as shown in Figure 6-3 a, which cause the large division from the true path. Moreover, the error in velocity is shown in Figure 6-3 b, where the average error in the velocity is approximately 5 m/s.



(a) (b)

Figure 6-3 (a) GPS error in position [m], (b) GPS error in velocity [m/s]

However, the average error in position using GPS with KF is approximately 1.5 m as shown in Figure 6-4a. In addition, the average error in the velocity is approximately 0.8 m/s as shown in Figure 6-4b.



(a) (b)

Figure 6-4 GPS using KF (a) Error in position [m], (b) Error in velocity [m/s]

Consequently, from the above results, it can be concluded that the developed KF for the GPS is able to provide more accurate positioning solution. Consequently, it will increase the accuracy ofter the integration with INS.

6.4.2 Field test

The proposed hybrid positioning technique is validated in a field test, which is carried out along a prescribed route as shown in Figure 6-5. This validation is done through two scenarios. The first scenario shows the obtained navigation solution for the field test in case of the number of visible satellites is sufficient. The second scenario shows the obtained navigation solution in case of an insufficient number of visible satellites. The obtained data from the sensors is recorded to be evaluated.

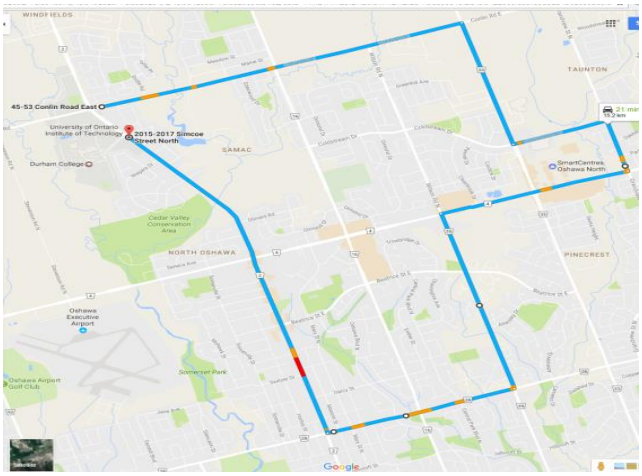


Figure 6-5 The proposed route for the field test

It has been noticed that, if the number of visible satellites is more than four, the first mode has the capability to provide an accurate navigation solution. On the other hand, if the number of visible satellites is less than four, the algorithm is switched to the second mode, where aiding the INS with GPS raw data still possible for accurate position.

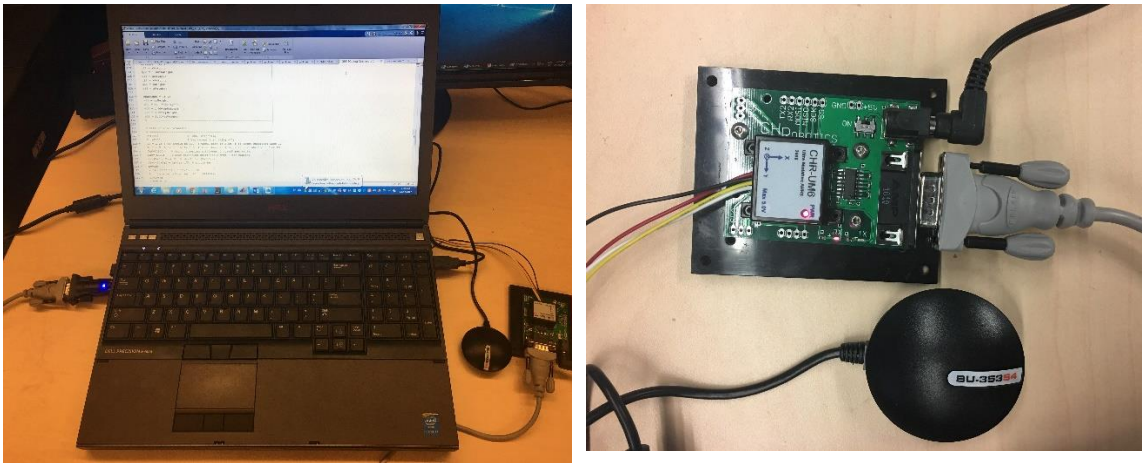


Figure 6-6 Experimental integrated navigation system.

The experimental setup for the integrated navigation system is shown in Figure 6-6. The system consists of a GPS receiver BU-353-S4, UM6 orientation sensor, power supply, and PC, which is used as a navigation computer in order to run the navigation algorithms and log the navigation sensor data for offline processing.

6.4.2.1 First Scenario

In this scenario, the outdoor experiment is performed to evaluate the performance of the proposed technique in case of a sufficient number of satellites. The obtained result of the vehicle route during the field test is shown in Figure 6-7.

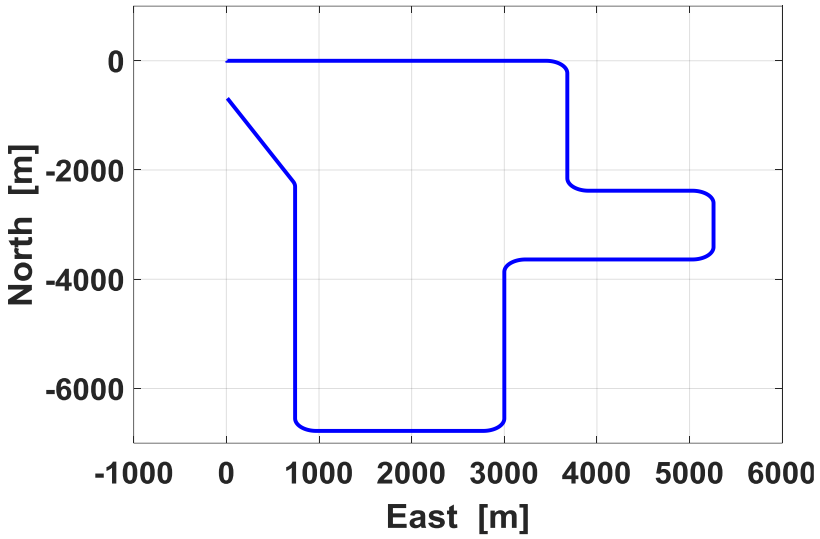


Figure 6-7 Ground vehicle route during the field test

The obtained results from fusing the GPS and INS are analyzed in terms of the horizontal position error as shown in Figure 6-8. The obtained result clarifies that the average value of the horizontal error is approximately 0.9 m within 14 min. In addition, the error in position is shown in Figure 6-9, where the error for aided latitude and longitude is almost zero compared with the unaided one. Furthermore, the overall accumulation of the error in the horizontal position is shown in Figure 6-10, where the error reaches 400 m within 14 min.

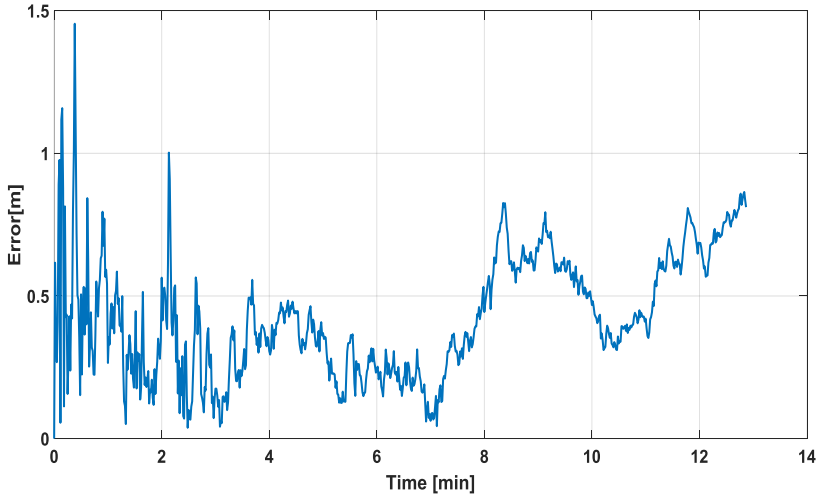


Figure 6-8 Horizontal position error using KF

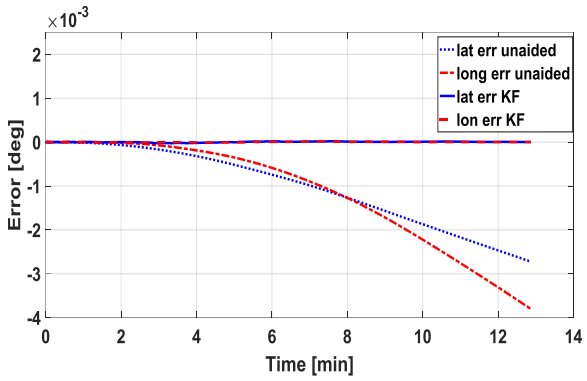


Figure 6-9 Lat, Long error aided and unaided KF

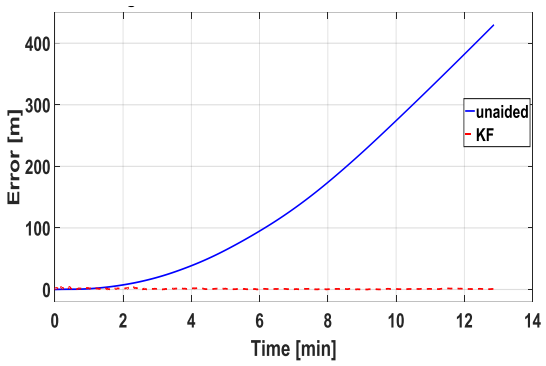


Figure 6-10 Aided and unaided horizontal position error

In addition, the velocity errors in the north and east frame are shown in Figure 6-11, which shows the difference between the aided and the unaided INS outputs. Furthermore, the attitude angles errors are shown in Figure 6-12, where all errors converge to zeros

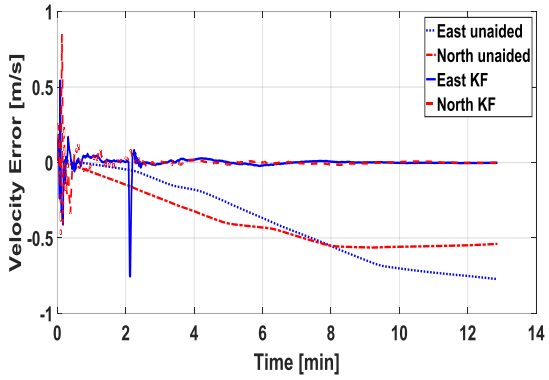


Figure 6-11 The velocity error

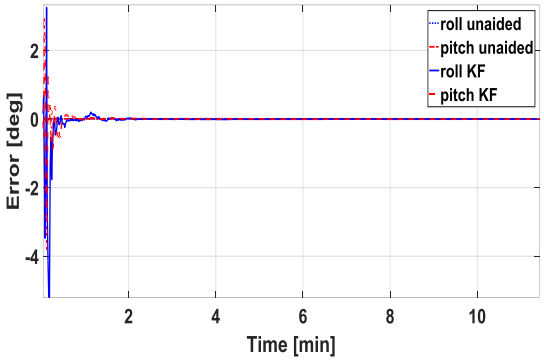


Figure 6-12 Attitude angles error

Consequently, based on the obtained results, the navigation solution of the proposed algorithm provides better accuracy in terms of the RMS and the position error.

6.4.2.2 Second Scenario

In this scenario, the number of visible satellites is decreased over time as shown in Figure 6-13, where the number of satellites starts at 8 satellites and after 5 min it decreased to 4 satellites, then at 6 min it becomes 3 satellites. The estimated error in horizontal position is shown in Figure 6-14, which shows the effect of the decreases in the number of the

satellites on the horizontal error. The average value of the error is approximately 2 m, which reflects an accurate performance while the number of satellites is decreased. The velocity errors in north and east are shown in Figure 6-15, which shows the difference between the aided and unaided INS where the error is accumulated. In addition, the errors in the attitude angles for both aided and unaided are shown in Figure 6-16, where all errors converge to zeros.

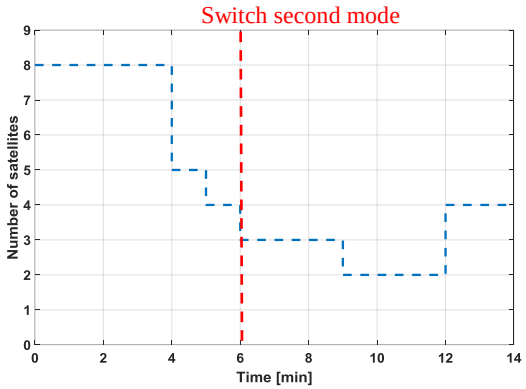


Figure 6-13 GPS satellites degradation over the time

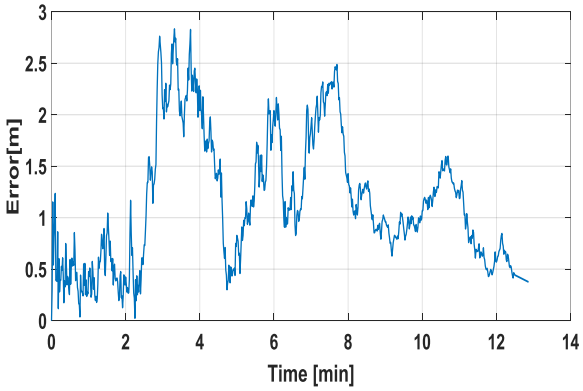


Figure 6-14 The horizontal position error for state KF

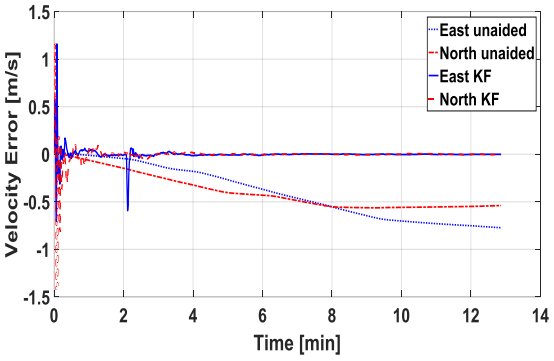


Figure 6-15 The velocity error

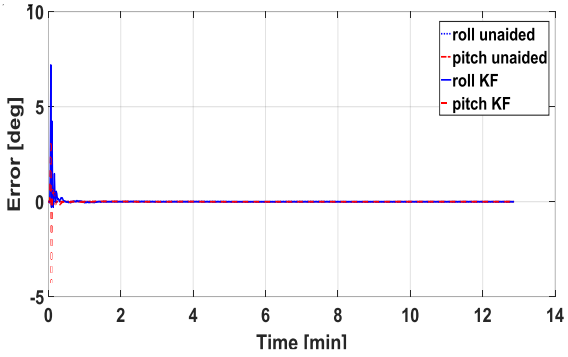


Figure 6-16 Attitude angles error

The position error is shown in Figure 6-17, which shows that the errors in aided lateral and longitudinal are almost zero compared with the unaided one. This scenario clearly shows that the proposed technique has the capability to provide an accurate positioning solution even if the number of satellites falls below the minimum.

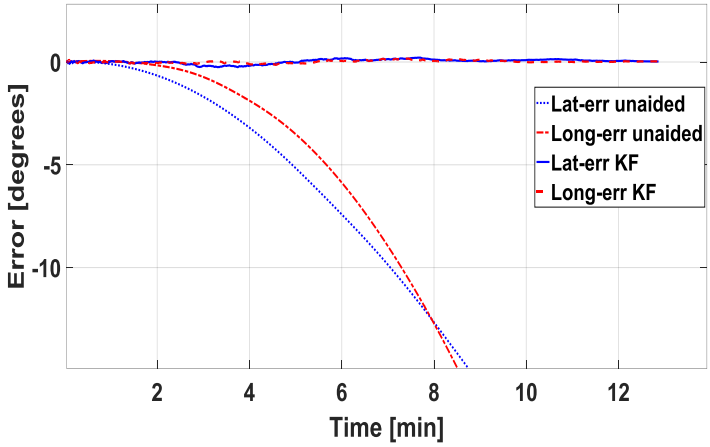


Figure 6-17 Lat, Long error aided and unaided KF

6.5 Chapter Summary

In this chapter, a hybrid framework for a positioning technique based on the integration of GPS/INS has been successfully developed and has achieved a robust positioning performance for autonomous combat vehicles. The proposed algorithm has the capability to adapt itself for solving the problem when the number of visible satellites is less than four by switching between the introduced two modes. In addition, it is able to fuse the obtained data from low-cost sensors such as the GPS and the INS. For this purpose, two KFs are developed; the first to improve the GPS information, and the second to enhance the INS position, velocity and attitude with the aid of GPS position and velocity. The simulation and field test are conducted to evaluate the performance of the developed hybrid framework. The simulation was performed using a SatNav toolbox in MATLAB. The field test is carried out by collecting data from sensors mounted on a ground vehicle. The simulation and experimental results show that the proposed framework has the capability to improve the positioning accuracy by switching between the two modes, even if the available number of visible satellites falls below the minimum.

CHAPTER 7

Modeling of Scaled Remotely Operated Multi-Wheeled Combat Vehicle Using System Identification Technique

7.1 Introduction

This chapter focuses on the modeling of a scaled Remotely Operated Multi-Wheeled Combat Vehicle (ROMWCV) using system identification methodology. The scaled vehicle was developed at the Vehicle Dynamics and Crash Research (VDCR) lab at University of Ontario Institute of Technology (UOIT) to analyze the characteristics of the full-size model. In this chapter, the vehicle input/output signals are recorded and analyzed in an open loop system through an experimental test, which is considered as Multiple-Input-Single-Output (MISO) system. The experiment test shows that it is practically feasible to represent the dynamic characters of the vehicle using the system identification techniques. Subsequently, different system identification methods are considered to solve and identify the ROMWCV model. The identified model is validated using several statistical tests and the results are compared. The estimation and validation results demonstrated that the obtained identified model was able to explain 88.44% of the output.

System identification methodology is introduced to study the performance of a developed dynamic system and estimate its mathematical model by observing input/output signals [125]. These signals contain the mathematical expression that precisely defines the system input and output relation. System identification has a long history in solving such significant problems in the field of autonomous systems and robotics. For example; system identification has been used in the following kinematic problems: modeling and calibration of robotic manipulators [126-131], parameter identification and nonlinear modeling [132-134], adaptive control and neural network-based system identification [135-139], estimation of inertial parameters [140-142], and the prediction of the environment [143].

7.2 Scaled Remotely Operated Multi-Wheeled Combat Vehicle.

The complete scaled remotely operated multi-wheeled combat vehicle system is shown in Figure 7-1. This vehicle was developed at the Vehicle Dynamics and Crash Research (VDCR) lab at University of Ontario Institute of Technology (UOIT) [147]. The developed vehicle is a 1:6 scale model of an 8x8 electric combat vehicle that can perform multiple steering modes to meet situational needs. In addition, all eight wheels are powered individually. The vehicle is equipped with four axles, which can be operated in either 4WD or 2WD.

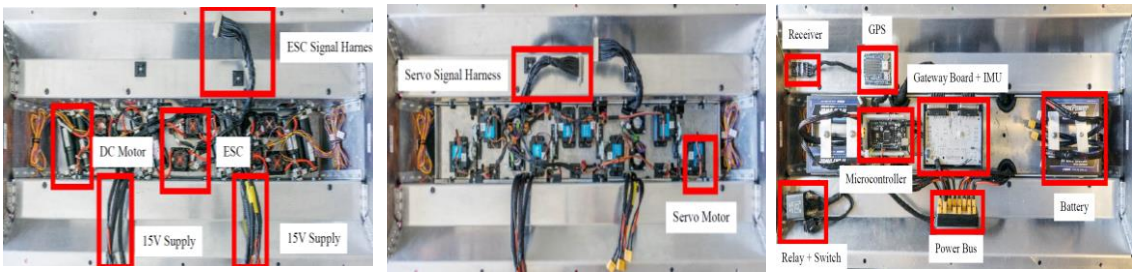
Figure 7-1 Scaled remotely operated multi-wheeled combat vehicle [147]

The four front wheels R1, R2, L1, and L2 are individually controlled, which are responsible to navigate the vehicle in the desired direction. Consequently, the developed remotely operated scaled multi-wheeled combat vehicle can be navigated by generating four separate voltages responsible for controlling the direction of the four front wheels.

7.2.1 Brief description of the MWCV

The vehicle consists of three levels; the first level contains the eight DC motors and the Electronic Speed Controllers (ESCs) as shown in Figure 7-2a. The middle level is the steering layer as shown in Figure 7-2b. This level contains eight steering servos that are connected by a DB25 connector. At the top as shown in Figure 7-2c, there are 4 batteries

that are connected to the scalable power bus where all ESCs are connected. In addition, a microcontroller is used to connect the gateway board to the two harnesses from the servos and the ESCs. The Inertial Measurement Unit (IMU) is placed under the gateway board enclosure as it should to be as centered as possible.



(a) (b) (c)

Figure 7-2 a) Chassis first level, b) Chassis middle level, c) Chassis top level [147]

In addition, the designed hardware architecture for the electronic powertrains is shown in Figure 7-3. Tow microcontrollers are introduced in this design. The primary one is used to control the vehicle by receiving the driver's request via the radio receiver, then calculate the corresponding outputs for the steering servo motors as well as the driving DC motors. The connection between the microcontroller and the motors is established using PWM channel. Each DC motor is individually controlled by an ESC that draws its power from the power system. The second microcontroller is used to connect the GPS and IMU with the primary controller and handles the logging and processing obtained data.

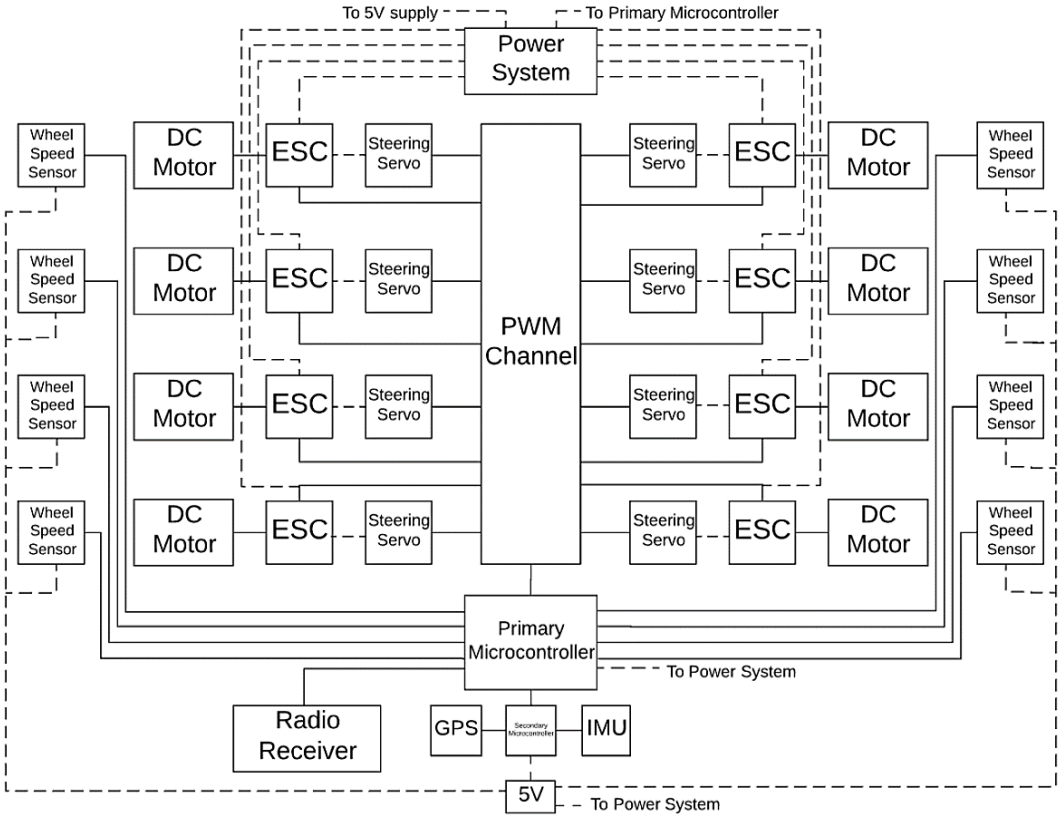


Figure 7-3 Hardware architecture [147]

The complete MWCV system is represented as shown in

Figure 7-4. This figure provides an overview of the vehicle control system, starting by sending the command signal from the remote control to the radio receiver ends by the corresponding vehicle action. It can be noticed that the microprocessor receives the command signal, then calculates the corresponding output for the four front wheels in order to control and navigate the vehicle in the desired direction.

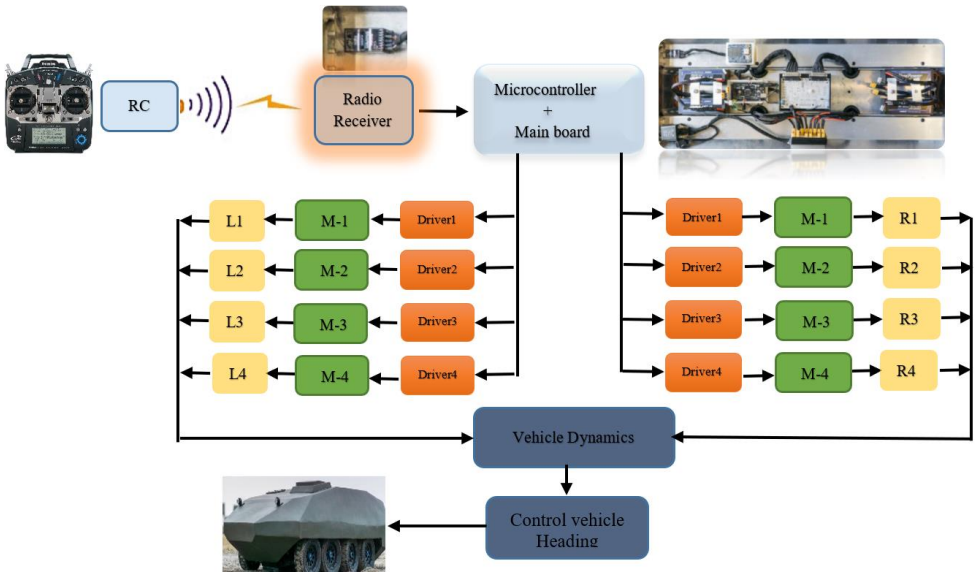


Figure 7-4 ROMWCV's system

7.2.2 Vehicle features

The features of the remotely operated scaled multi-wheeled combat vehicle are as follows [147]:

- Capable of forward and backwards movement, where all eight wheels are powered individually.
- Alternate between 3 unique steering configurations.
- Capability to operates under extreme conditions.
- Sufficient ground clearance to overcome obstacles like large object.
- Ability to steer on multiple surfaces, such as sand, dirt, snow, mud, and pavement.
- High maneuverability to operate in confined areas where moving space is limited.
- Able to climb over steep terrain and reach moderate speeds similar to that of the full model.

7.3 System Identification (SI)

Generally, SI is the process of modelling systems dynamics based on the measured input/output signals via an experimental test. It has the capability to provide an accurate mathematical model of the system dynamics. SI approach goes through five steps [148] as shown in Figure 7-5; (1) experiment design, (2) data collection, (3) parameter estimation algorithm and system identification model selection, (4) model validation (5) model implementation. If the model validation is not good enough to represent the actual model of the system dynamics, the first three steps will be repeated again until the model validation becomes good enough to represent the actual model of the system dynamics.

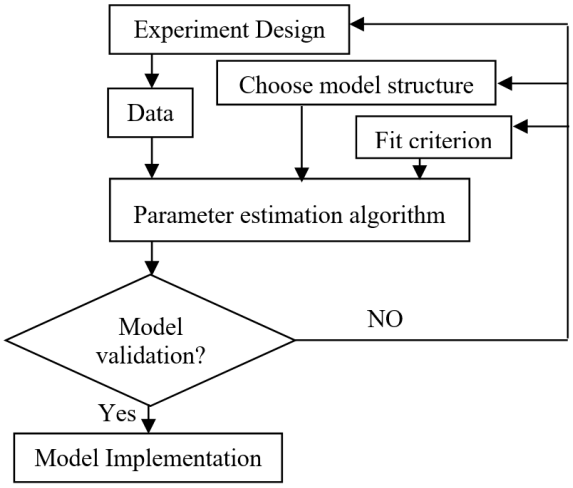


Figure 7-5 System identification procedure

In this chapter, the ROMWCV can be considered a MISO system due to the four input signals sent to the four motors and the output is the vehicle heading angle. Consequently, the identification of the vehicle model was carried out using parametric identification techniques.

SI using parametric identification techniques has a specific model structure. The parameters are estimated using the observation of the input/output data. In addition, it is providing a large variety and possibilities regarding different ways of describing the system, where the output of system $Y(Z)$ can be defined as follows.

$$Y(z) = G(z)X(z) + M(z) \quad 7\text{-}1$$

Equation 7-1 can be rewritten as follows:

$$Y(z) = G(z)X(z) + H(z)E(z) = \frac{N(z)}{D(z)}X(z) + \frac{A(z)}{B(z)}E(z) \quad 7\text{-}2$$

where;

$Y(z)$ is the n_y output;

$X(z)$ is the n_x input;

$E(z)$ is the transform of a white noise, $\in (t)$;

$G(z)$ is the transfer function of the system;

$\boldsymbol{H(z)}$ is the stochastic behavior of noise;

$G(z)$, $H(z)$ are rational functions whose numerator and denominator are parameter polynomials to be estimated. Consequently, the relationship between both functions defines several model structures of describing equation 7-2.

The modeling of ROMWCV model will be considered by applying AutoRegressive eXogeneous (ARX), AutoRegressive Moving Average eXogeneous (ARMAX), and Transfer function (TF) models. The characteristics of each model were studied in [149]:

The ARX model in Equation 7-3 is considered the simplest estimation model. The ARX model block diagram is shown in Figure 7-6. The main weakness is the disturbance model $(1/N(z))$ that comes with the system's poles. Consequently, an incorrect estimation of the system dynamics can be accrued due to the term A in Equation 7-2. Accordingly, this issue can be avoided by the requirement of higher orders coefficients of terms A, B in Equation 7-2, where the signal to noise ratio is acceptable.

$$D(z)Y(z) = N(z)X(z) + E(z) \quad 7\text{-}3$$

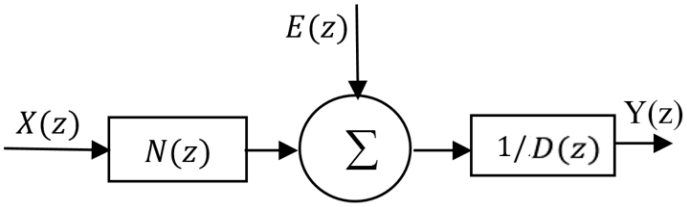


Figure 7-6 ARX model

The ARMAX model in Equation 7-4 has the capability to handle the disturbance modeling compared with ARX model. For this purpose, ARMAX has become a standard tool in control for system description. Consequently, is considered the most popular model that can be used in many applications. The block diagram of the ARMAX model is shown in Figure 7-7.

$$D(z)Y(z) = N(z)X(z) + A(z)E(z) \qquad 7\text{-}4$$

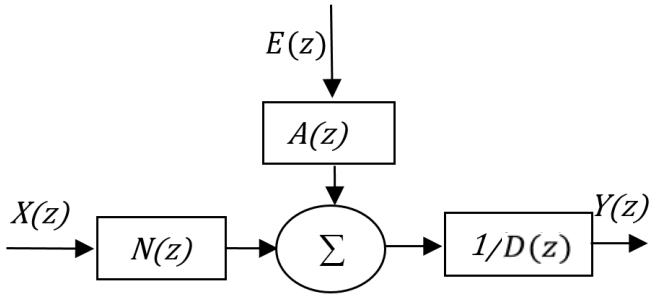


Figure 7-7 ARMAX model

In a TF model, the output of the system Y(z) for an input X(z) can be obtained as shown in Equation 7-5, where the mathematical representation of the relation between the input and output of a system can be obtained.

$$Y(z) = G(z)X(z) \qquad 7\text{-}5$$

Consequently, the transfer function of a system can be defined as the ratio of the output and the input. This function is rational with numerator and denominator polynomials of the complex variable z:

$$G(z) = \frac{Y(z)}{X(z)} = \frac{N(z)}{D(z)} = \frac{b_0 + \cdots + b_m Z^{-m}}{a_0 + \cdots + a_n Z^{-n}} \qquad 7\text{-}6$$

A useful format for the transfer function is to describe it in terms of Z^{-1} because this is the unit delay operator. Roots of $N(z)$ are called zeros and roots of $D(z)$ are called poles of the system

7.3.1 System identification algorithm

System identification starts by selecting a model structure followed by the computation of an appropriate model in the structure. The selected model will be evaluated afterward. Figure 7-8 shows this process that is summarized as follows:

Step1: Record the input/output signals from the vehicle.

Step2: Examine the data and select useful portions of the original data.

Step3: Select and define the appropriate identification model structure within which the model of the system can be obtained.

Step4: Choose the best model structure corresponding to the input/output data and the given fit to estimation criterion.

Step5: Examine the obtained model's properties (pole-zero configurations).

Step6: If the selected model is good enough to represent the identified system, then stop; otherwise go back to the fourth step to try another model set. Possibly also try another estimation method in the fifth step or work further on the input-output data obtained in first and second steps.

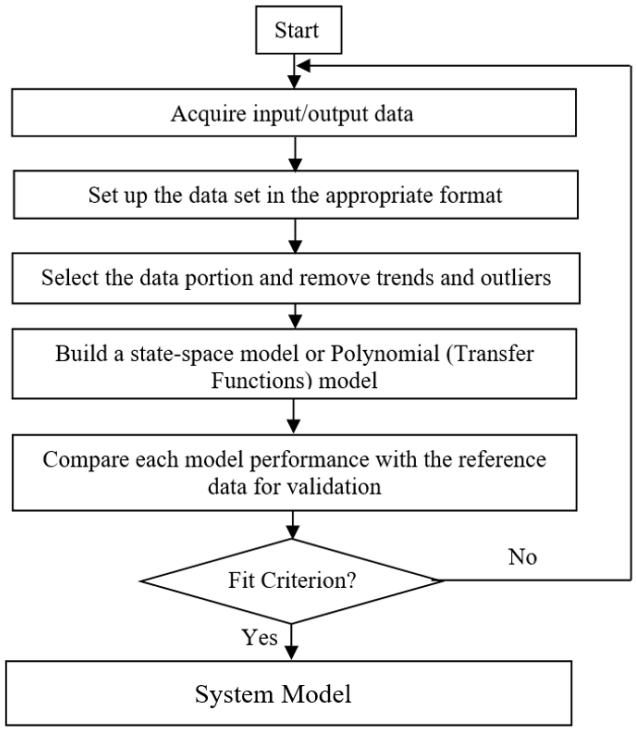


Figure 7-8 System identification procedure flowchart

7.4 Experimental Set-up

In this section, the experimental setup for recording and analyzing the input/output signals is discussed. A tilt Compensated Magnetic Compass (CMPS11) is used to provide the vehicle Euler angles as shown in Figure 7-9. It has three magnetometers, three gyros, and three accelerometers. The main advantage of this sensor is the use of the KF. This KF is used to integrate the gyro and accelerometer in order to avoid the errors that may be accrued by PCB tilting.

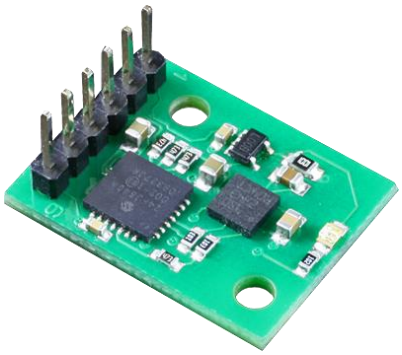
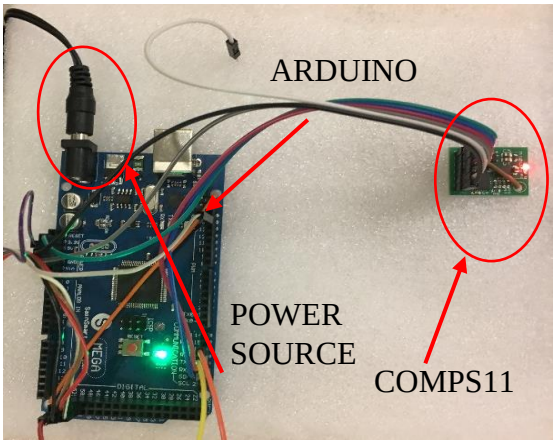
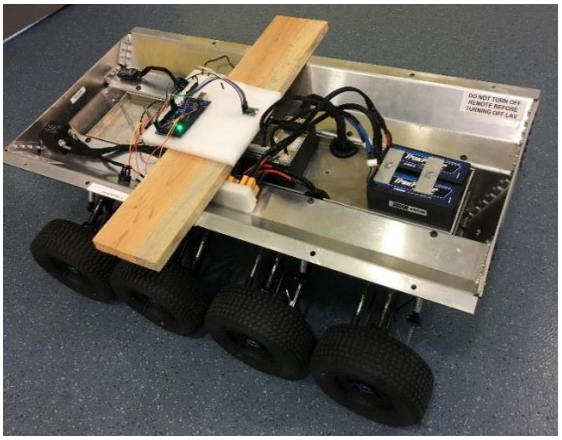


Figure 7-9 Tilt Compensated Magnetic Compass (CMPS11)

The CMPS11 will be interfaced with Arduino to save the obtained input/output data on an SD card during the test as shown in Figure 7-9. Figure 7-10 a, b showed the experimental setup while attached to the vehicle.



(a)



(b)

Figure 7-10 Experiment test a) COMP11 and Arduino, b) Sensors attached to the vehicle

The road test of the ROMWCV is shown in Figure 7-11, where the vehicle heading angle is controlled in an open loop test via the remote control. In order to obtain reliable data, the vehicle heading angle should be changed continuously during the maneuver. This test was repeated five times to make sure that the recorded input/output signals are accurate when applying a system identification technique.

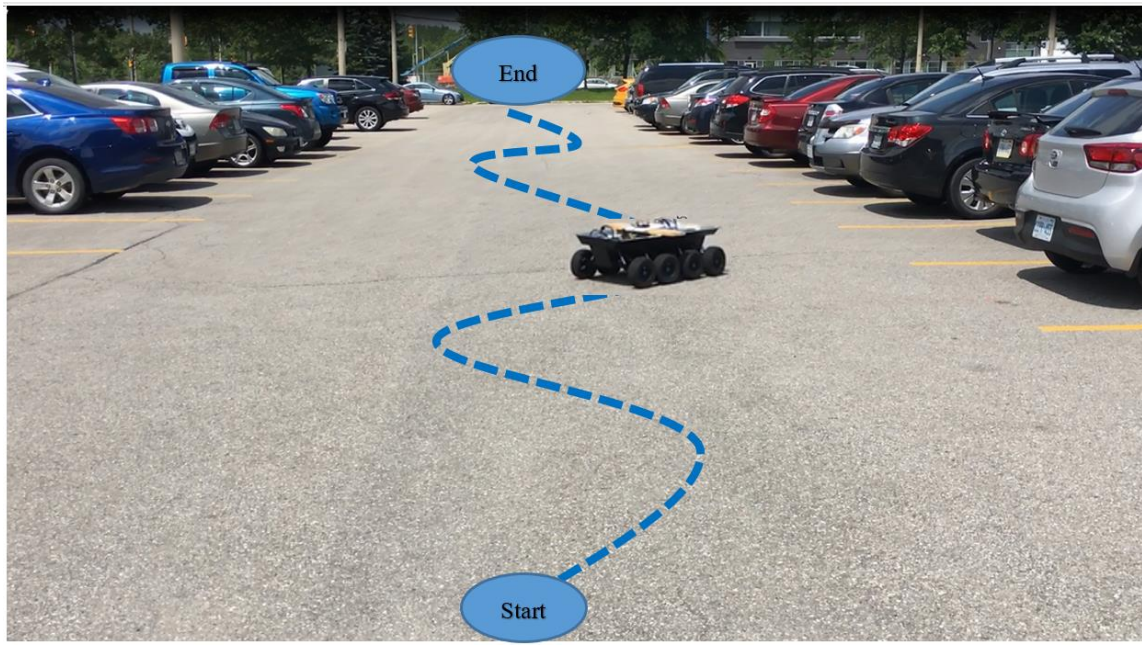


Figure 7-11 Road Test

7.5 Simulation and Experimental Results

In this section, the ROMWCV input/output data is recorded and analyzed to deduce a model as shown in Figure 7-12. The right and left wheels of the first axle are represented by R1 and L1, while the right and left wheels of the second axle are represented by R2 and L2.

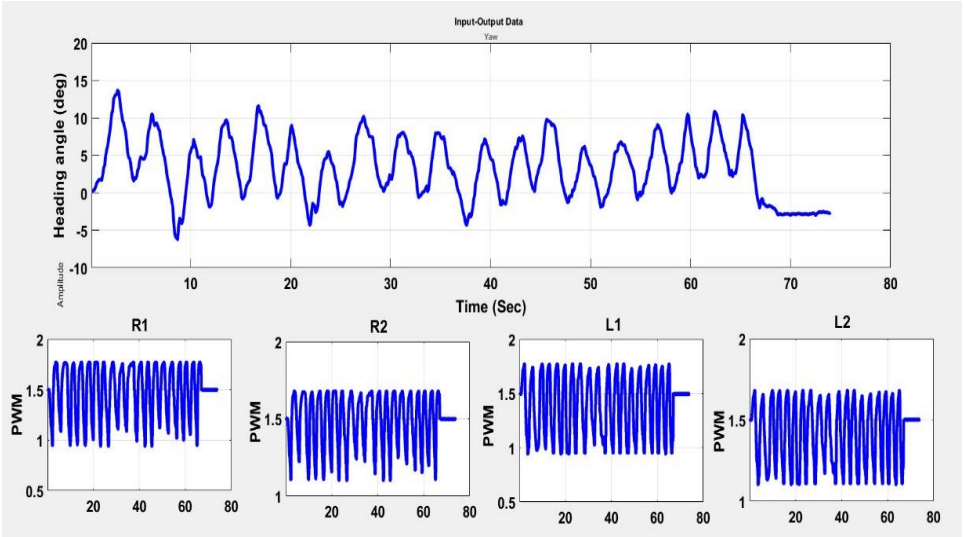


Figure 7-12 Experiment results for the recorded input and output signals

Subsequently, based on the measured input/output data the system identification toolbox using MATLAB software will be used to develop the vehicle model. First, the data is loaded on MATLAB command window, the recorded inputs and output data are set, then the command "ident " opens the system identification toolbox interface as shown in Figure 7-13.

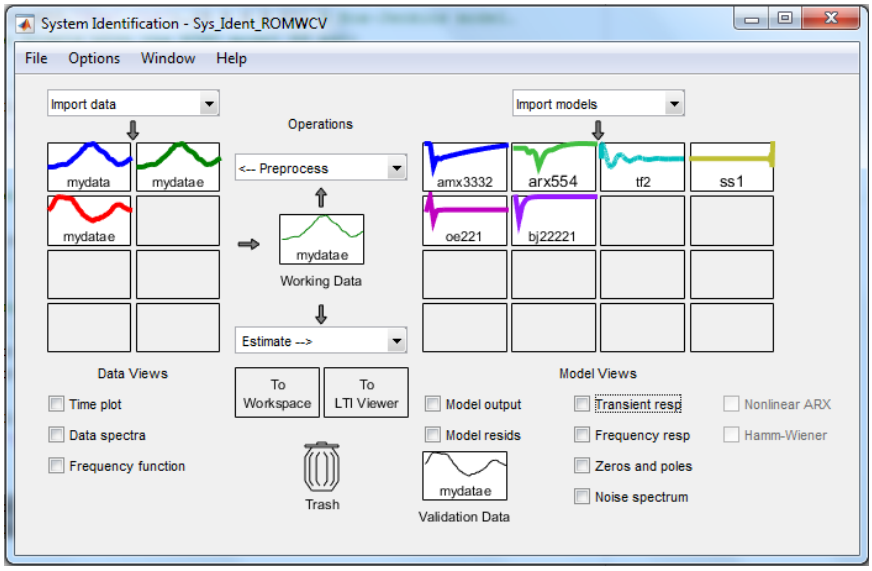


Figure 7-13 System identification toolbox

The obtained step response of the ROMWCV is shown in Figure 7-14, which represents the inputs/output relation.

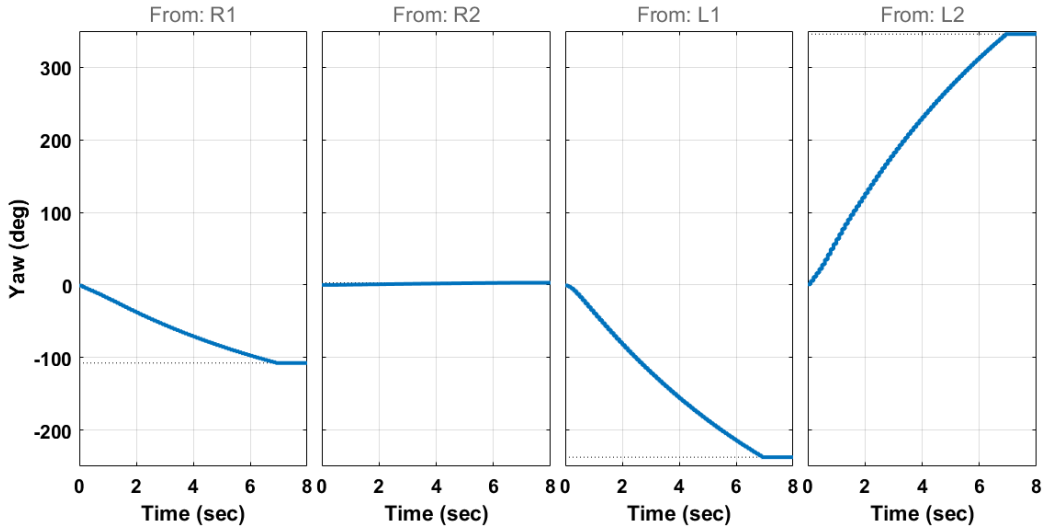


Figure 7-14 Step responses for estimated dynamics for each channel

In order to estimate the appropriate model that can describe the ROMWCV system, the ARX, ARMAX, and TF identification models are applied as Single Input Single Output (SISO) for each wheel individually. Subsequently, each model will be evaluated to choose the best one that can provide the accurate vehicle model, then applied as MISO for the four wheels of the vehicle in order to provide the whole vehicle model. The obtained validation results with the unseen data for each wheel are shown in Figure 7-15 to Figure 7-18.

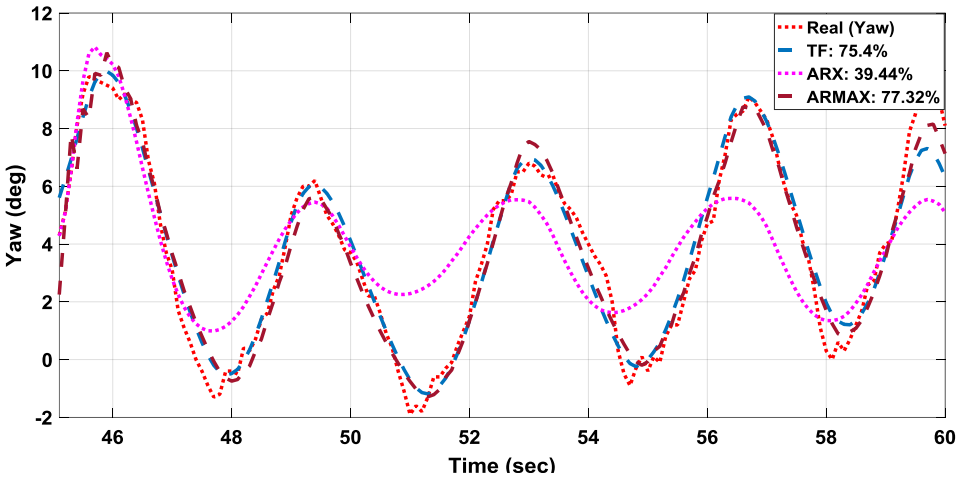


Figure 7-15 Fit to estimation results for the SI models for R1

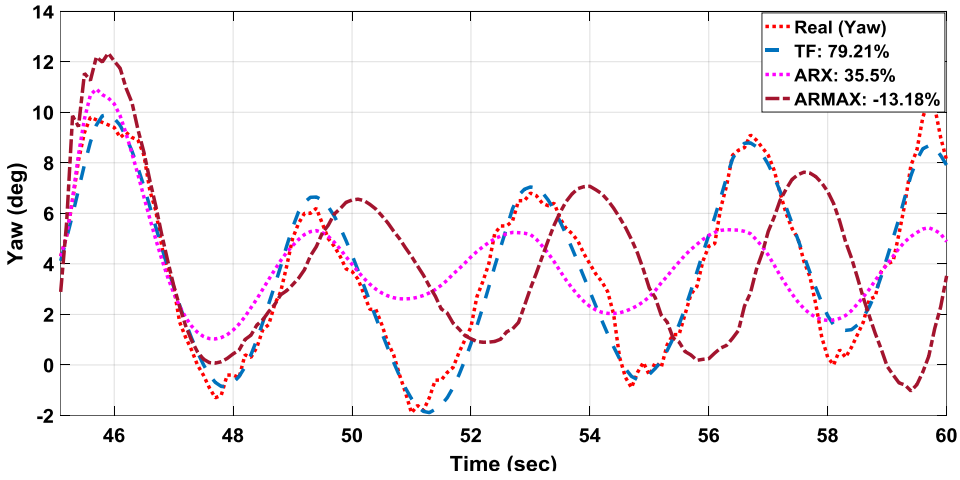


Figure 7-16 Fit to estimation results for the SI models for R2

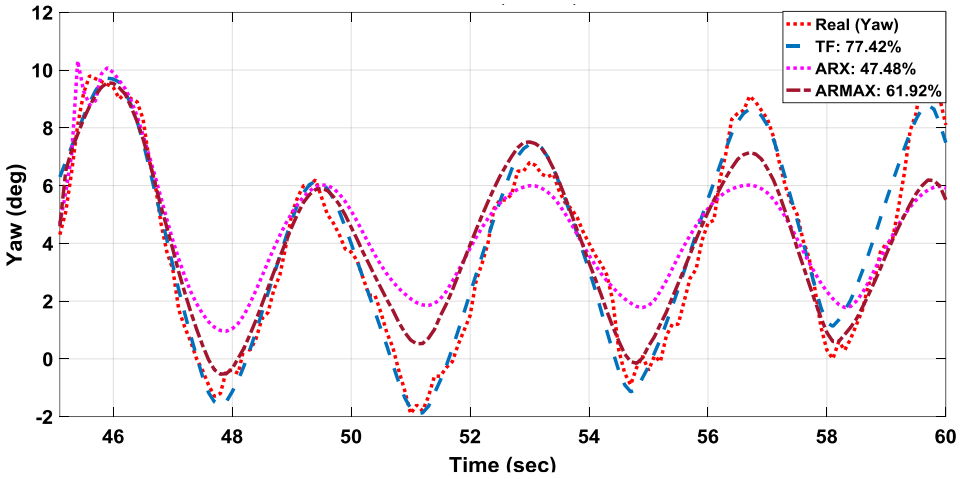


Figure 7-17 Fit to estimation results for the SI models for L1

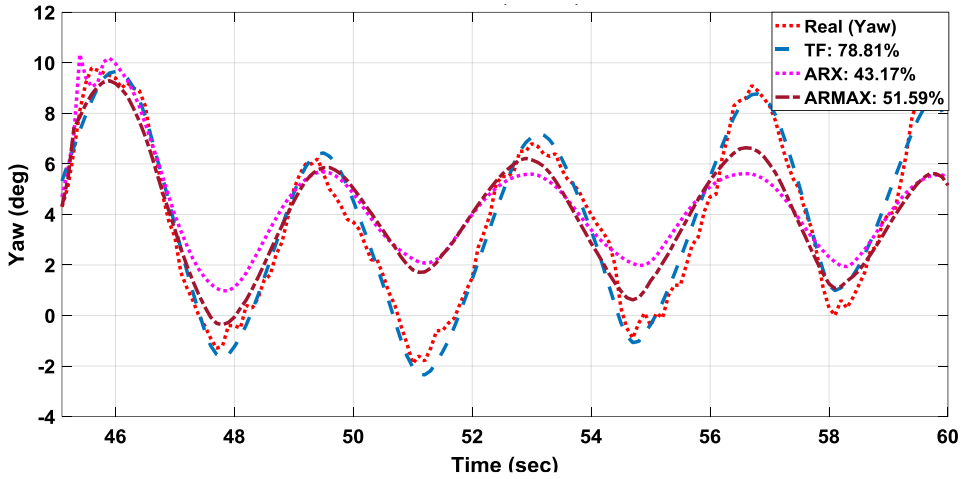


Figure 7-18 Fit to estimation results for the SI models for L2

Based on the obtained validation result, a comparison between each model is proposed in Table 7-1 to 7-3.

Table 7-1: ARX Model results

ARX Model	Validation data %
R1	39.44%
R2	35.3%
L1	47.48%
L2	43.17%

Table 7-2: ARMAX Model results

ARMAX Model	Validation data %
R1	77.32%
R2	13.18%
L1	61.92%
L2	51.59%

Table 7-3: TF Model results

TF Model	Validation data %
R1	75.4%
R2	79.21%
L1	77.42%
L2	78.81%

The above tables concluded that the TF model has the capability to achieve the best validation result for the unseen data compared with other models. Consequently, it will be applied to the vehicle system as MISO. By observing the pole-zero configuration, the TF model can be tuned. The obtained results demonstrate that the model has the capability to achieve 88.44% of the output data as shown in Figure 7-19, which is considered good enough to identify the vehicle model.

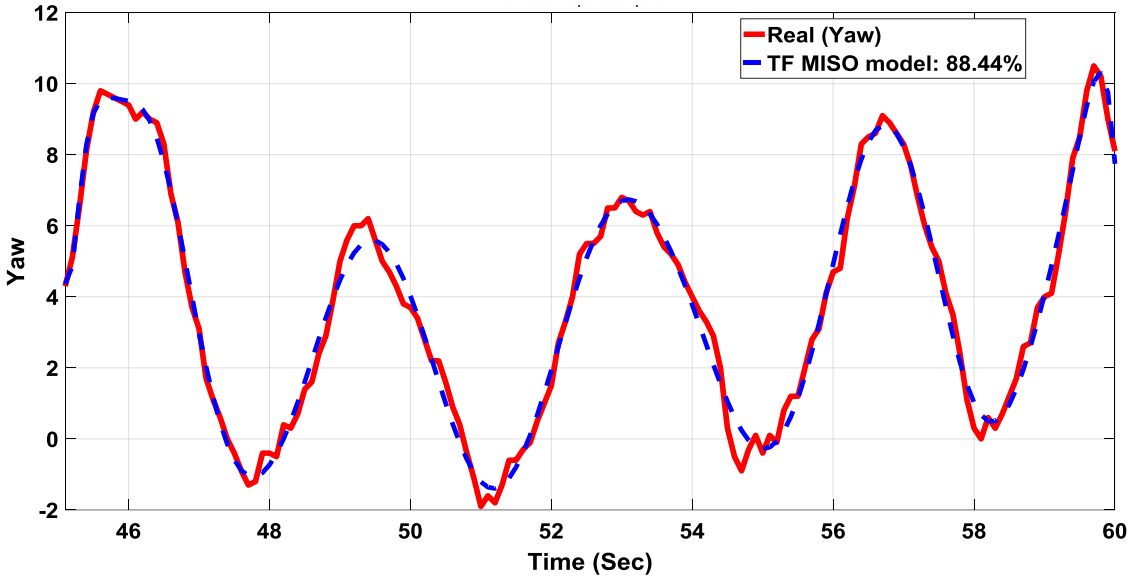


Figure 7-19 Validation result for the TF model with the actual yaw angle

Consequently, the obtained model will be as follows:

$$G(s)_{Total} = G_{R1}(s)\, u_{R1} \; + \; G_{R2}(s)\, u_{R2} + G_{L1}(s)\, u_{L1} + G_{L2}(s)\, u_{L2}$$

where

$$G_{R1}(s) = \frac{-3.453\, s \; - \; 0.6946}{s^2 + \; 0.1762\, s \; + \; 0.1323} \qquad G_{R2}(s) = \frac{-7.118\, s \; + \; 1.158}{s^2 + \; 0.388\, s \; + \; 0.3099}$$

$$G_{L1}(s) = \frac{5.76\, \mathrm{s} \; + \; 0.1182}{s^2 \; + \; 0.2714\, \mathrm{s} \; + \; 0.1083} \qquad G_{L2}(s) = \frac{7.382\, s \; + \; 0.8258}{s^2 + \; 0.5481\, \mathrm{s} \; + \; 0.2887}$$

7.6 Chapter Summary

This chapter discussed the development and modeling of the scaled remotely operated multi-wheeled combat vehicle using system identification techniques. The vehicle was developed to simulate and analyze the characteristics of the full-size model. It is an electric-powered 1:6 scale model of an 8x8 combat vehicle model. It has eight wheels which are independently driven by electric motors. The vehicle has the capability to steer on multiple surfaces such as sand, dirt, snow, mud, and pavement.

An experimental test was carried out on an open loop system for measuring the input/output signals using a tilt compensated unit CMPS11. The input/output signals are recorded and analyzed. Subsequently, several system identification models were applied in terms of ARMAX, ARX, and TF models to provide an accurate vehicle model. Several statistical analyses were applied, and the results are compared. The validation results revealed that the TF model provided an accurate ROMWCV model. The validation process was verified using real data which achieved 88.44% of the output data.

CHAPTER 8
Processor-in-the-Loop Co-Simulations and Control Design for MWCV

8.1 Introduction

This chapter describes the design and the implementation of PID and fuzzy logic algorithms to track the desired heading angle for a scaled Autonomous Multi-Wheeled Combat Vehicle. The main challenge of designing such control systems is to individually regulate the steering of the four front wheels in the same order for obtaining the predefined heading angle of the vehicle. The performance of the developed controllers is validated in the presence of noise and disturbance in order to evaluate their robustness. Subsequently, a Processor-In-The-Loop (PIL) co-simulation is conducted to permit and achieve a more realistic situation where the developed control algorithms are evaluated while running on a dedicated processor. The obtained results from both simulation and PIL are compared, which demonstrate the performances of the developed controllers and their effectiveness in achieving the desired heading angle.

Controller design and testing for autonomous combat vehicles have been widely studied by researchers around the world. Applying autonomy to such vehicles is a complicated process due to their large dimensions, heavy weight, and complex geometry. In the literature, many control algorithms were applied to autonomous vehicles and mobile robots, which have a long history in solving significant problems in this field. For example, these control algorithms can solve the following, autonomous path tracking [150-153], obstacle avoidance [154-156], control for mobile robots [157-159], and steering control [160-163].

Fuzzy logic (FL) is a powerful soft computing technique, which has the capability to control complex systems based on human expert knowledge. FL can be used to control autonomous systems in many engineering fields [164-167]. Furthermore, the main advantages of using FL are easy implementation, efficient computation, and better in performance [168]. On the other hand, a Proportional-Integral-Derivative (PID) controller

is the prevalent control loop feedback mechanism that is widely used in industrial control systems. This classical controller attempts to eliminate the error between the actual and the desired output by calculating the appropriate control signal. This control signal adjusts the process accordingly based on smoothing the output movement and minimizing the rise time and peak error.

Processor-In-The-Loop co-simulation is introduced to test and validate controller performance [169-171]. PIL provides some hardware features that allow the achievement of more realistic situations where the control algorithm is running on a processor. Consequently, the performance of the control algorithm can be easily evaluated and compared with the simulation results.

There has been some work done dealing with PIL, for example, a PIL and software-in-the-loop (SIL) simulations for microsatellite Attitude Determination and Control Subsystem (ADCS) were carried out by Juang [172]. The ADCS algorithm was implemented using a PIC microcontroller, while the SIL simulation is developed in MATLAB environment. Chen [173] developed a local-loop based robot action control module using independent microprocessors. The action command is transmitted from the PC via serial port to the microprocessor, which permits the simultaneously and rapidly accomplishment of different actions of the mobile robot. Seelaender [174] introduced a co-simulation approach to control satellite dynamic model simulation using a Field Programmable Gate Array processor (FPGA). The aim of his work is to control the wheel reaction in the Simulink environment using the designed controller. The created code is implemented on the FPGA.

8.2 Design of the Fuzzy Logic Controller (FLC)

The developed fuzzy logic controller consists of three processes as shown in Figure 8-1. The first process in designing the FLC is fuzzification. In this step the inputs and outputs are transformed from real value into grades of membership. The second step is the fuzzy inference process. In this step the input data using fuzzification is mapped to conduct the fuzzy reasoning process. The third step is the defuzzification process, which transforms the membership degree of fuzzy sets into a crisp set that can be applied to the real system.

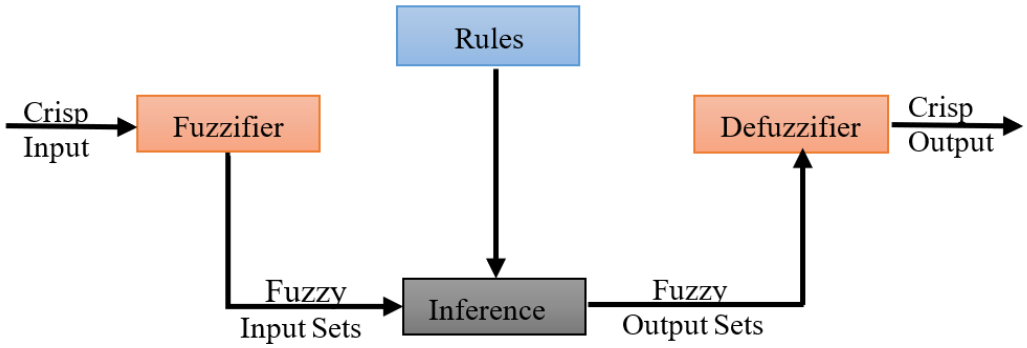
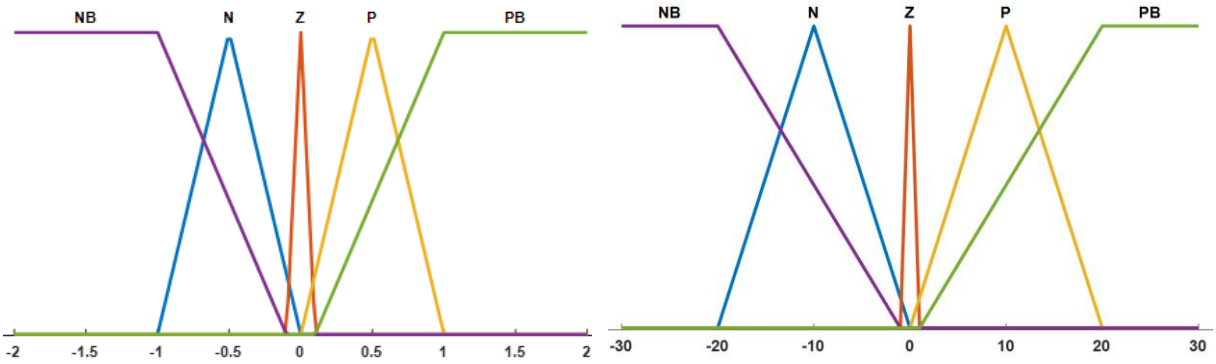


Figure 8-1 Block diagram of fuzzy logic controller

The main challenge in designing the FLC is to obtain the appropriate fuzzy rules and membership functions due to the MWCV behavior to fulfill the objective of the heading angle control. For this purpose, four fuzzy controllers are designed to individually control the steering of the four front wheels of the vehicle. Each controller takes one input; the heading angle error, which is the difference between the desired and the actual vehicle heading angle. The output of the controller is the PWM for each wheel. In addition, the previously measured inputs/outputs data of the vehicle in Chapter 7, Figure 7-12 is very helpful for defining the appropriate rules in the designing process for the fuzzy controller.

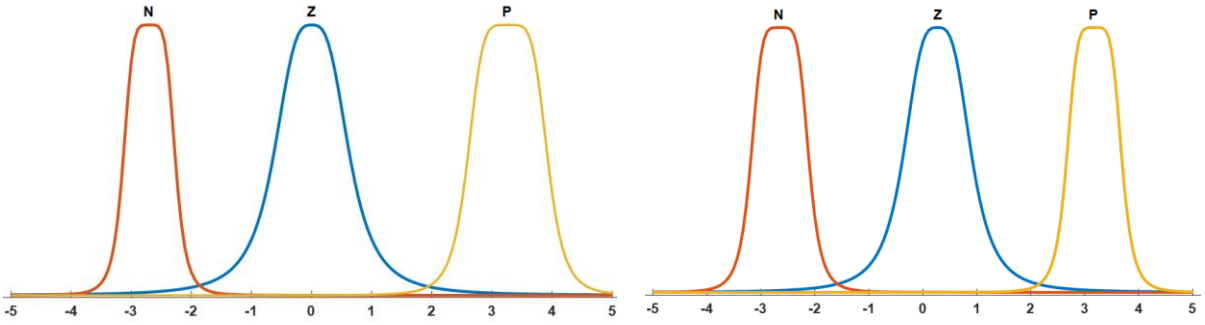
Subsequently, triangular and trapezoidal membership functions are defined for the input and output variables. The input/output variables for the first controller are defined by five membership functions NB, N, Z, P and PB as shown in Figure 8-2a.



(a) (b)

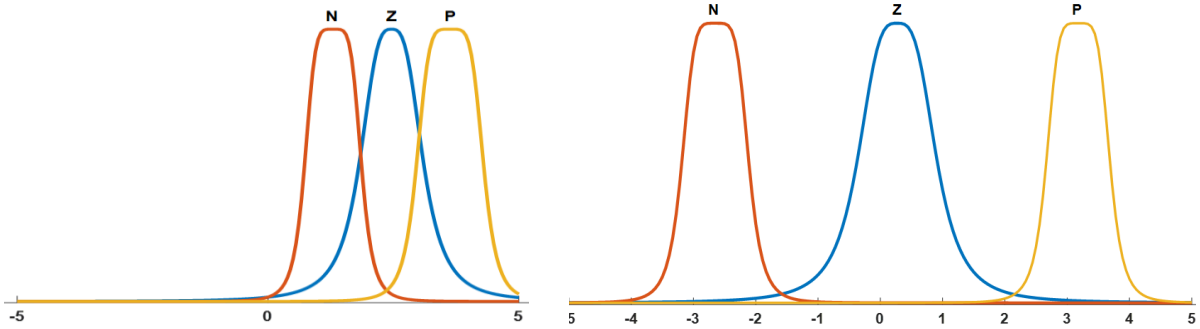
Figure 8-2 Fuzzy 1 (a) Input membership function for the heading angle (b) Output membership function for PWM

Consequently, the input and output variables for the second, third and fourth fuzzy controllers are defined by three membership functions N, Z and P as shown in Figure 8-3-Figure 8-5.



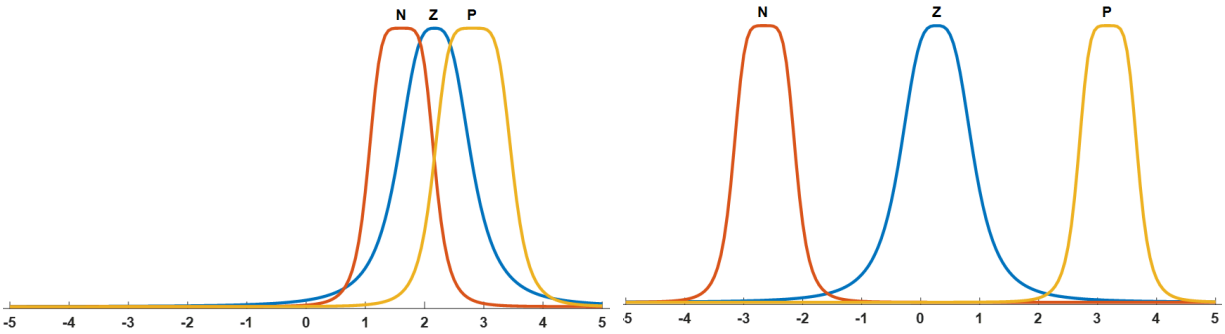
(a) (b)

Figure 8-3 Fuzzy 2 (a) Input membership function for the heading angle (b) Output membership function for PWM



(a) (b)

Figure 8-4 Fuzzy 3 (a) Input membership function for the heading angle (b) Output membership function for PWM



(a) (b)

Figure 8-5 Fuzzy 4 (a) Input membership function for the heading angle (b) Output membership function for PWM

The membership functions parameters have been tuned by successive iterations based on the obtained heading angle values. Figure 8-6 shows the developed block diagram control

loop in Simulink for MWCV using four fuzzy controllers to reach the desired heading angle.

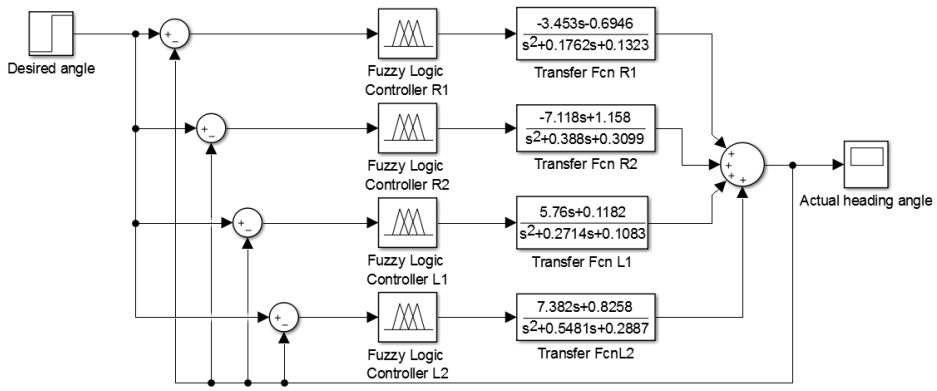


Figure 8-6 Fuzzy heading angle controller Simulink diagram for the R1, R2, L1, L2

8.3 Design of the Proportional Integral and Derivative Controller (PID)

PID controller is widely used in industrial control systems as shown in Figure 8-7. Development of PID controller according to the conventional strategies requires previous knowledge of the MWCV nature (input/output see Figure 7-12) to be controlled.

$$U(s) = K_p\left(1 + \frac{1}{K_i s} + K_d s\right)E(s) \qquad 8\text{-}1$$

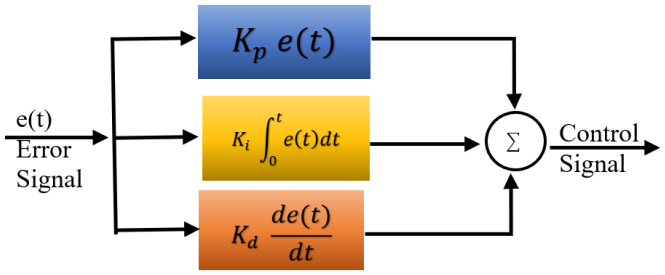


Figure 8-7 Block diagram of PID controller

The proposed PID algorithm contains three separate parameters in terms of the Proportional K_p, the Integral K_i and the Derivative K_d as explained in Equation 8-1. The weighted sum K_p, K_i, and K_d is used to control the MWCV heading angle, then by make a fine tuning for these parameters, the PID controller is capable to provide a much better result. Based on Ziegler/Nichols tuning method [175,176] the proposed PID controllers'

parameters are tuned. The developed block diagram control loop in Simulink for MWCV using the developed four PID controllers to reach the desired heading angle is shown in Figure 8-8.

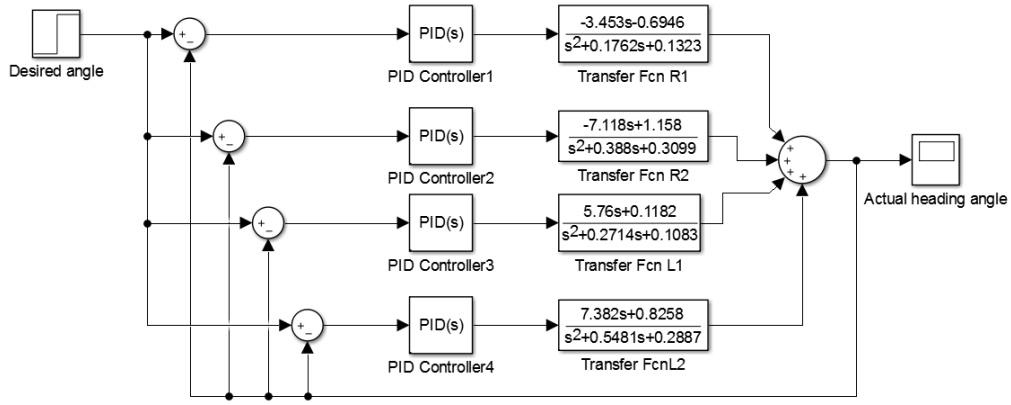


Figure 8-8 PID heading angle controller configuration for the R1, R2, L1, L2

It can be noticed that tuning the PID controller using Ziegler-Nichols rule is able to provide the best values for the three PID gain parameters as shown in Table 8-1. Consequently, the PID controller attempts to correct the error between a measured process variable and the desired set points by calculating and outputting a corrective action that can adjust the process accordingly.

Table 8-1 Controller parameters

Parameter	PID1	PID2	PID3	PID4
$\mathbf{K_p}$	-2.672	0.0801	1.626	1.268
$\mathbf{K_i}$	-4.449	0.016	1.488	1.129
$\mathbf{K_D}$	-0.0515	-0.945	-0.031	-0.066

8.4 Processor-In-The-Loop Co-simulation (PIL)

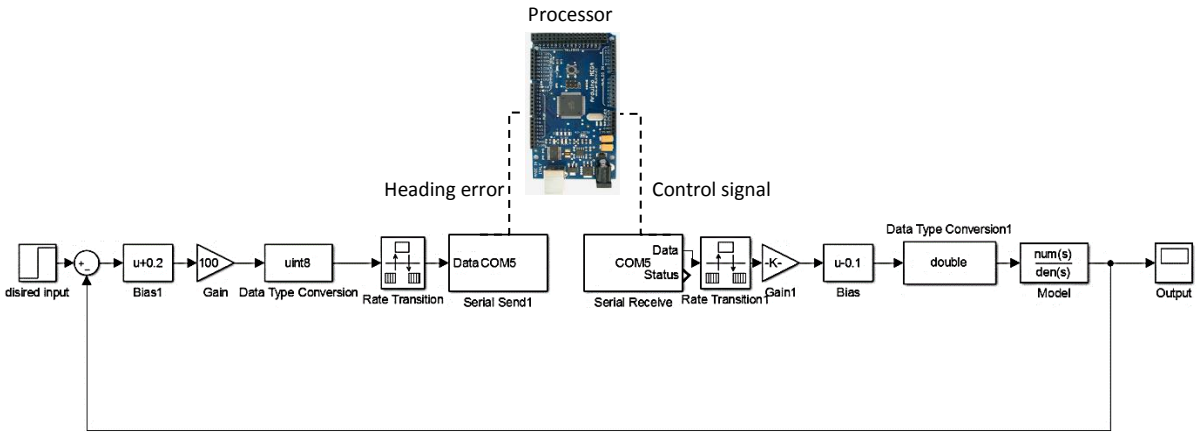
In this section, the PIL co-simulation of the proposed two control algorithms is discussed. The PIL permits and achieves a more realistic environment where the developed control algorithms will run on a processor and consequently, the performance can be compared and verified with simulation results. In addition, PIL allows debugging the controller and correct any error that may affect the system performance. For this purpose, a MATLAB/Simulink environment is used to implement and perform a PIL co-simulation.

The developed algorithms are implemented using Arduino board to perform PIL co-simulation.

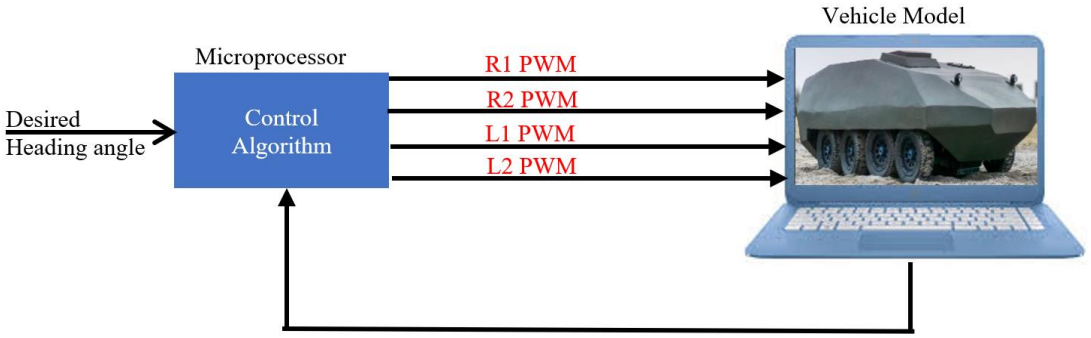
8.4.1 PIL validation

The validation of the proposed heading angle controllers is carried through the scheme of co-simulation, where the PC performs the vehicle model simulation using MATLAB/Simulink. On the other hand, the Arduino board processes the developed PID and fuzzy controllers. Afterwards, a comparison between the developed two algorithms for tracking the desired heading angle is considered under the same condition that assumed in the simulation section.

The co-simulation scheme using MATLAB/Simulink is shown in Figure 8-9. The Simulink send the heading angle error signal to the processor via the serial port. Subsequently, the processor calculates the control signal and sends it to the vehicle model to take the action.



(a)



(b)

Figure 8-9 PIL Co-simulation Simulink block diagram

It can be noticed that, when sending and receiving data through the serial port, these data must be unsigned integers. For this purpose, the heading error signal is biased with a certain value to be positive, then multiplied by a gain to make sure that all sent data to the processor are an integer as shown in Figure 8-9a. On the other hand, before the processor receives these data it should be biased and multiplied by opposite values to return it to their original values. The same will be done for the processor output (make sure all data are unsigned integers) before sending it back to PC via the serial port.

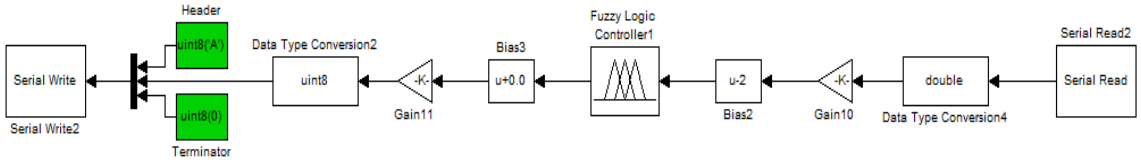


Figure 8-10 PIL Block diagram in Simulink environment

The developed PIL block diagram using MATLAB/Simulink environment is shown in Figure 8-10, which implemented on the processor using the generate and built code command. In this way, the communication with the processor can be easily established in order to send the heading angle error to the processor, then calculate the appropriate control signal that will be sent to the vehicle model in Simulink to take action. Consequently, based on the obtained results using the proposed two control algorithms in simulation and PIL, a comparative analysis is proposed in the next section.

8.5 Simulation and Experimental Results

In this section, the PIL and the simulation results of the developed controllers for the heading angle control will be discussed.

8.5.1 Fuzzy logic controller

The closed-loop system step response performance using the developed fuzzy logic controller compared with the uncontrolled system is shown in Figure 8-11. It has been noticed that the steady state error for the closed loop performance of the uncontrolled system is high. On the other hand, after applying the developed fuzzy controller the steady state error reaches zero, while the settling time is 0.03 sec.

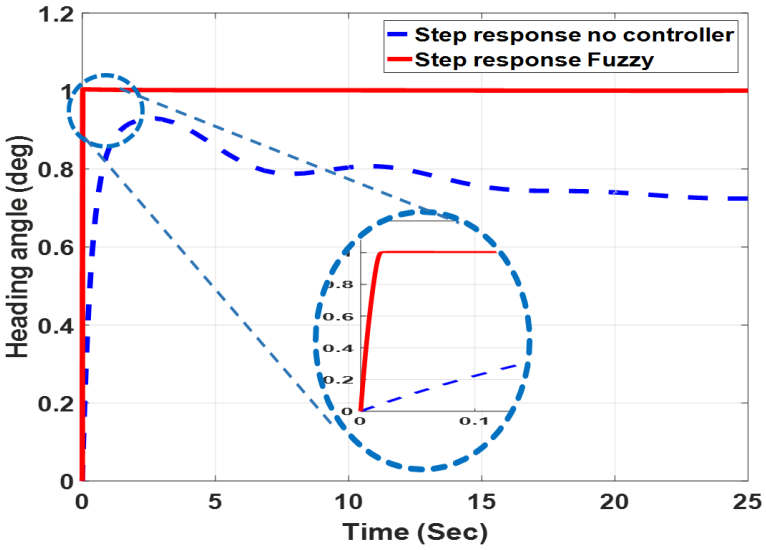


Figure 8-11 Comparison of transient response of the heading angle for fuzzy controller and uncontrolled system

8.5.2 PID controller

The closed-loop system step response performance using PID controller compared with uncontrolled system is shown in Figure 8-12. Based on the obtained results, it can be noticed that the steady state error of the PID controller reaches zero, while the settling time is 0.1 sec compared with the uncontrolled system. As such, the steady state error of the uncontrolled system is high.

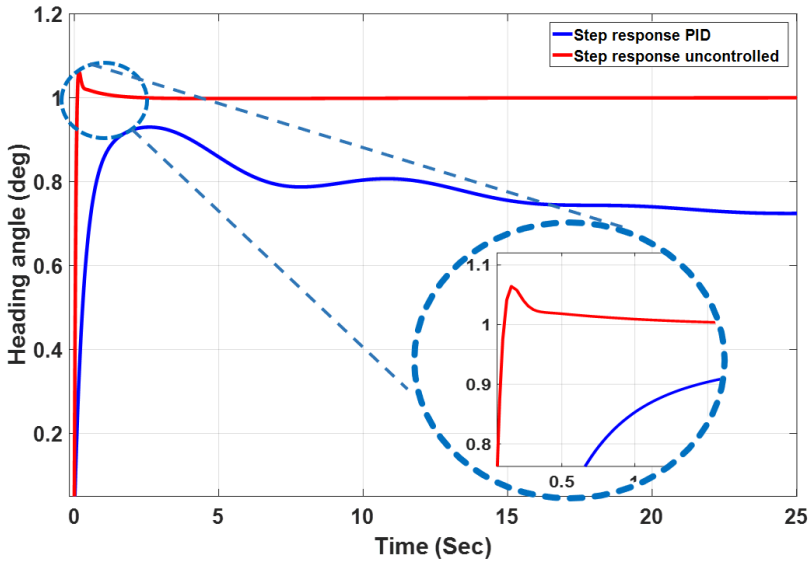


Figure 8-12 Comparison of transient response of the heading angle for PID controller and uncontrolled system

8.5.3 Controllers evaluation

The evaluation of the proposed controller algorithms for tracking the predefined heading angle is experienced through two scenarios. The first scenario shows the MWCV tracking the heading angle to reach 25 degrees as shown in Figure 8-13. The obtained result clarifies that both controllers have the capability to track and follow the desired heading angle. However, the error between the desired and the actual heading angle using the fuzzy controller is smaller compared with the PID controller, especially when the direction is changed.

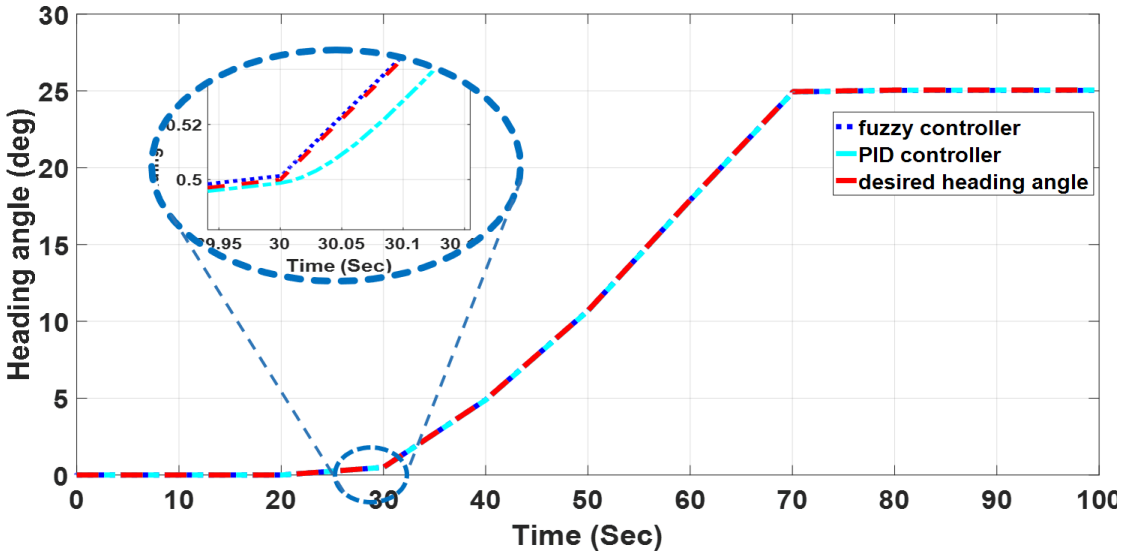


Figure 8-13 Comparison of tracking heading angle using fuzzy and PID controllers to reach the desired 25 degree heading angle.

In the second scenario, the reference heading angle is exported from TruckSim software for a circle maneuver. The comparison between the proposed two controllers is shown in Figure 8-14. It can be noticed that both controllers are able to track and follow the desired heading angle. However, the fuzzy controller is more accurate in tracking the desired heading angle compared with the PID controller.

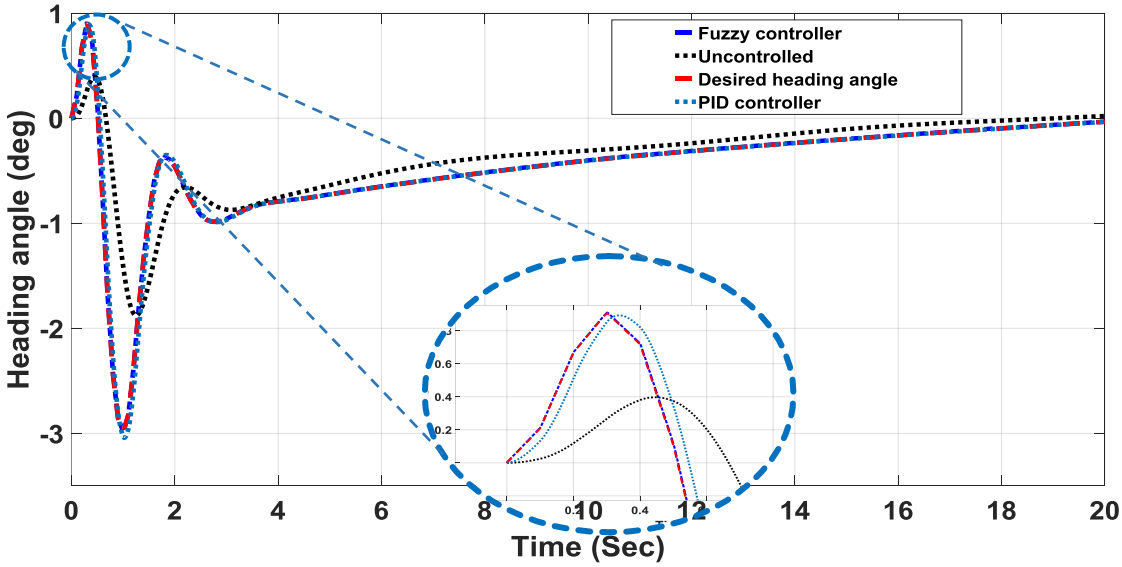


Figure 8-14 Comparison of tracking heading angle using fuzzy and PID controllers to reach the desired heading angle.

The animation for this scenario using TruckSim for this maneuver is shown in Figure 8-15, where the blue and yellow vehicles represent the fuzzy and PID controlled respectively, and the red one is the uncontrolled vehicle.

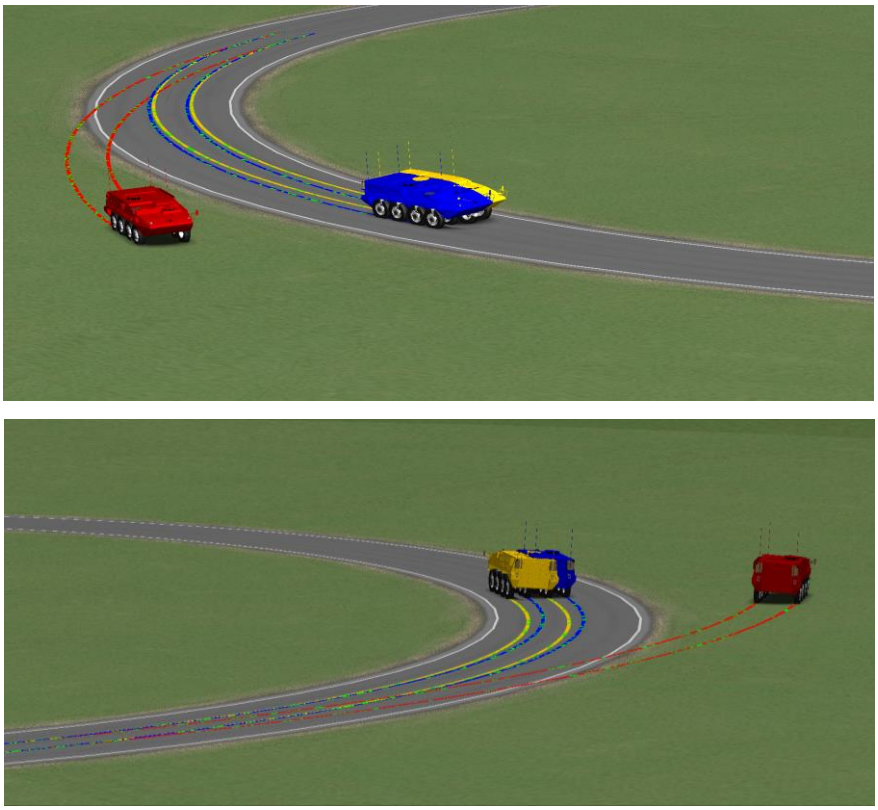


Figure 8-15 TruckSim animation of tracking heading angle using fuzzy and PID controllers with uncontrolled one to reach the desired heading angle.

8.6 Validation in Presence of Disturbance and Noise

a) Disturbance rejection

In this section, the performance evaluation of the developed controllers to show the capability to reject the disturbance during the maneuver is introduced. For this purpose, an impulse disturbance is applied to the system output and the obtained result is shown in Figure 8-16. It can be noticed that, the settling time of the fuzzy controller is very low compared with the PID controller. Consequently, the convergence using the fuzzy controller is better than the PID controller as it rejects

50% of the disturbance within 0.03 sec and rejects 95% of the disturbance within 0.05 sec. On the other hand, the PID controller rejects 50% of the disturbance within 0.06 sec and reject 95% of the disturbance within 0.1 sec.

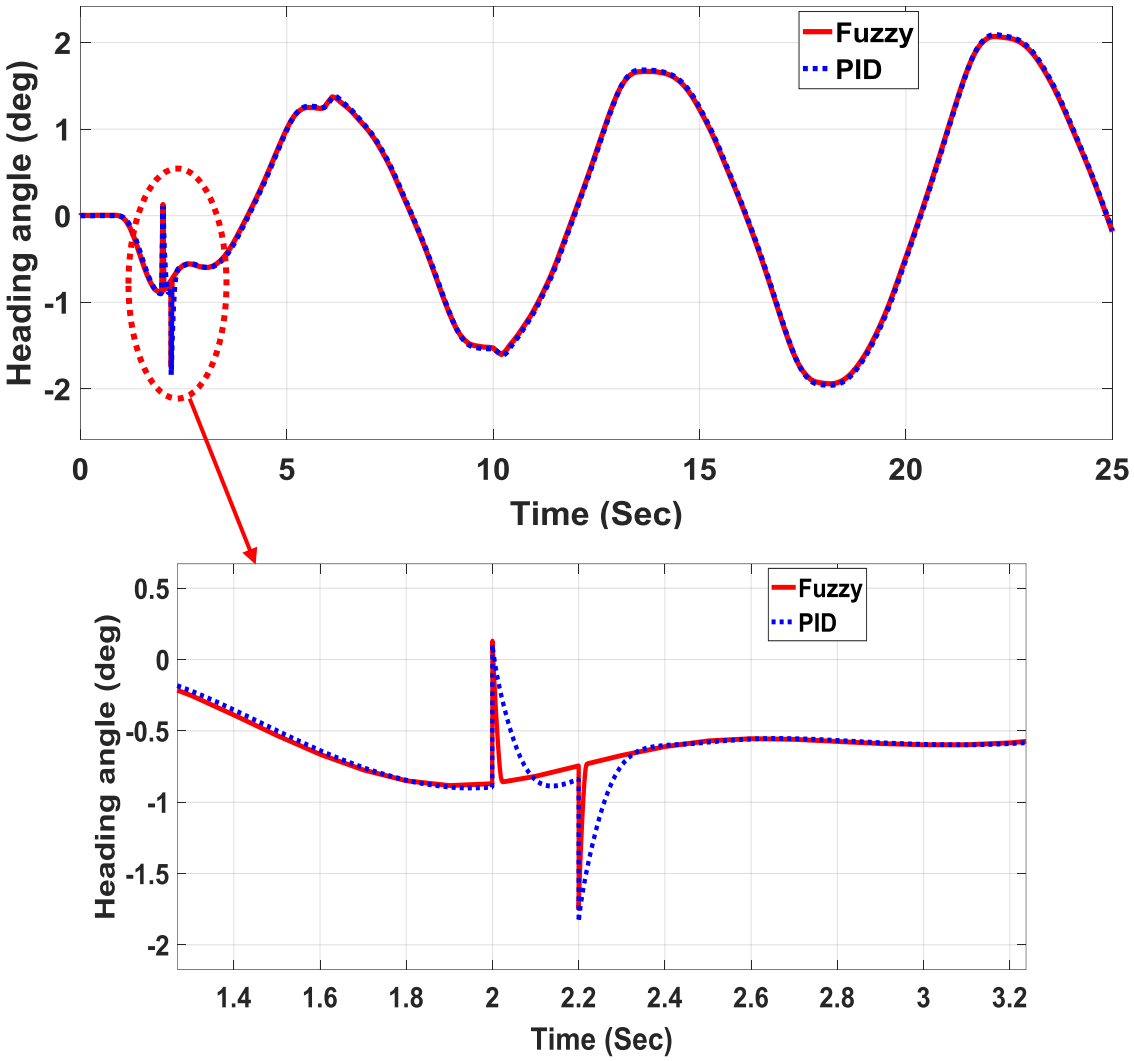


Figure 8-16 Comparison of disturbance rejection using PID and fuzzy logic controller

b) Noise sensitivity

In this section, the performance of the proposed algorithms is evaluated in the presence of noise. For this purpose, a white Gaussian noise is applied at the output of the system and the obtained step response is shown in Figure 8-17. Based on the obtained result, it can be found that the fuzzy logic controller is sensitive to additive

noise compared with the PID controller. Moreover, by comparing the control efforts generated from both controllers, the fuzzy logic controller is higher than the PID controller in control efforts with a ratio of 2:1, which explains its sensitivity to the applied noise.

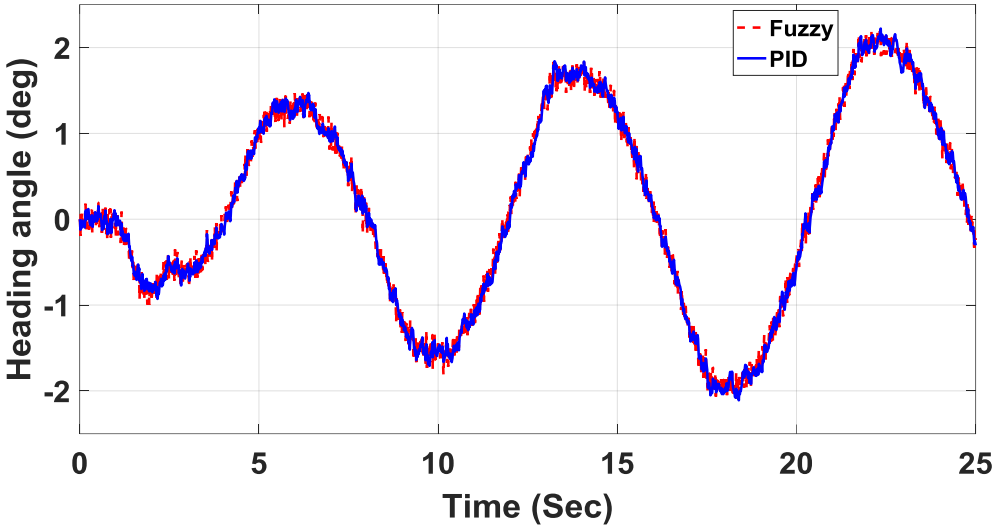


Figure 8-17 Comparison of noise sensitivity using PID, fuzzy logic controller

8.7 Comparative Analysis of PIL and Simulation

In this section, a comparative analysis of the obtained results using the developed control algorithms in the case of PIL (implemented controller on the processor) and the Simulation (simulation only using MATLAB/ Simulink environment).

a) Fuzzy logic controller

The obtained closed-loop responses for PIL and the simulation are shown in Figure 8-18. It can be noticed that the PIL result is close to the simulation result, which means that the developed controller has the capability to achieve a proper implementation on the processor environment.

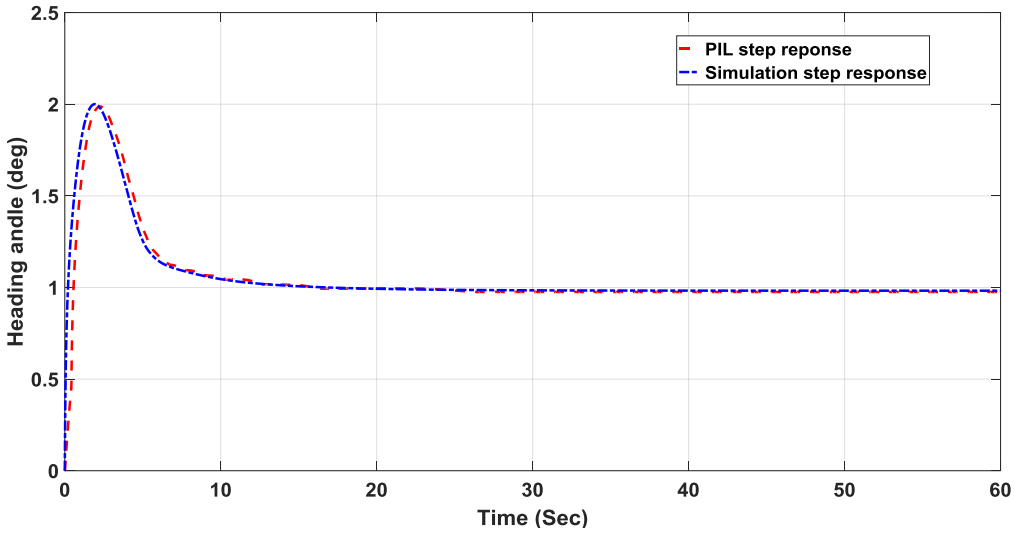


Figure 8-18 Comparison of transient response of the heading angle for fuzzy controller in simulation and PIL

a) PID controller

The obtained closed loop response for both PIL and the simulation is shown in Figure 8-19. It can be noticed that the obtained results using PIL are close to the simulation results and the controller is proper implementation in the processor environment.

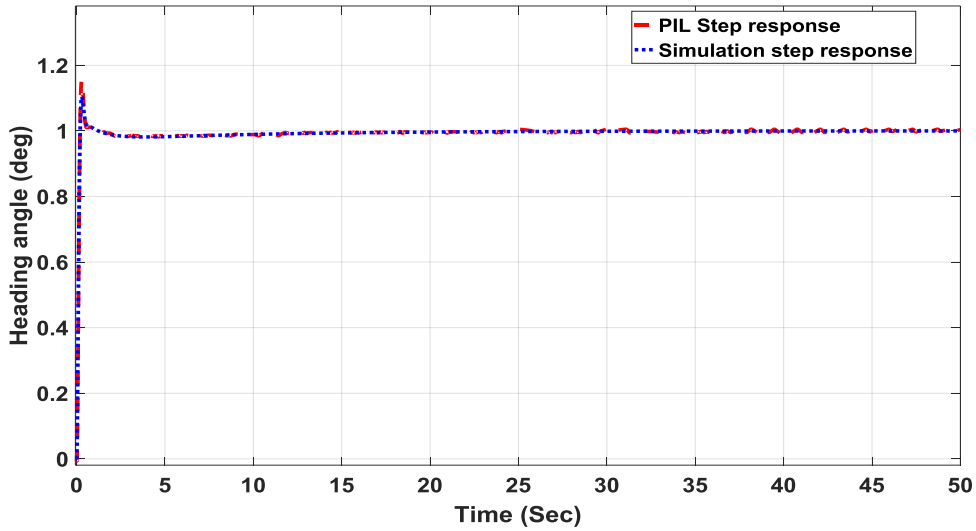


Figure 8-19 Comparison of transient response of the heading angle for PID controller in simulation and PIL

8.8 Chapter Summary

This chapter presented the design, tuning and implementation of heading angle tracking controllers for a scaled autonomous multi-wheeled combat vehicle. For this purpose, a fuzzy logic and PID controllers have been proposed. The developed control algorithms are implemented in a processor to be tested together with the vehicle model running on PC for heading control through the PIL. The PIL experimental results have been presented to validate the implementation in a processor with the simulation results. The study of all obtained simulation results and PIL co-simulation led one to believe that there is a trade-off in any chosen control strategy. Consequently, based on the performance of the systems using both controllers, the fuzzy logic controller showed the best results, which can follow the desired heading angle with very low steady state error and settling time compared with the PID controller. In addition, it has the capability to reject the disturbance within a very short time. However, the fuzzy logic controller is less sensitive to applied noise compared with the PID controller.

CHAPTER 9

Design and Performance Analysis of Robust H_∞ Controller for Scaled Autonomous Multi-Wheeled Combat vehicle

9.1 Introduction

This chapter describes the design of a robust controller to track the desired heading angle for a scaled multi-wheeled combat vehicle considering the presence of uncertainties that are difficult to estimate such as noise and disturbance. For this purpose, a robust H_∞ heading angle control algorithm is developed for the vehicle to tracking the desired heading. The augmented plant is structured by adjusting the weighting functions taking into account the presence of noise and disturbance. The developed H_∞ framework is well suited for controlling the steering of the vehicle's four front wheels individually to obtain the correct heading angle. The vehicle simulations in the presence of disturbances and noise are carried out and the robustness of the proposed heading angle tracking algorithm is incorporated. Subsequently, a Processor-In-The-Loop co-simulation is conducted in order to evaluate the controller while running on a dedicated processor. The simulation and PIL results are compared, which shows the effectiveness of the H_∞ controller and its capability to achieve the desired heading angle changes with better performances and efficiency.

By increasing demands and progressing technologies, the development of autonomous ground combat vehicles become an emerging research focus. Applying advanced control techniques for an autonomous vehicle became challenging to handle and control. One of the important issues facing such vehicles is heading angle tracking, which is required to design the appropriate controller that has the capability to force and control the vehicle dynamics to reach and track the predefined directions precisely. Recent research starts to pay attention to applying autonomy to multi-wheeled combat vehicles due to the expected outcomes such as increasing combat capability and soldier safety in different battlefield scenarios [177-180].

Various control schemes and analysis have been proposed to control and improve autonomous vehicle performance, which includes Wildcat autonomous ground vehicle

construction and implementation [181], autonomous ground vehicle path tracking [182-186], and navigation [187,188]. In addition, a heading angle tracking controller was developed and implemented by Sahoo [24] for unmanned ground vehicles considering a linearized dynamic bicycle model. The authors applied a Point-to-Point navigation algorithm in order to control both the steering and heading angles, while considering the limits on rotation of the steering wheel and steering motor rate. Dadras [189] developed a Fractional Order Extremum Seeking Controller (FO-ESC) in order to control an autonomous ground vehicle to track a predefined reference path. The authors claim that the robustness and higher performance fractional order operators lead ESC to perform better and more efficiently. Eski [190] developed a model based on an artificial neural network PID controller for trajectory control of the unmanned agricultural vehicle. The developed neural network model has the capability to learn the PID structure with a high level of performance.

Robust control has been proven to be a powerful tool when applied to various control problems [191-193]. Consequently, robust H_∞controller has played an important role in the study and analysis of control theory since its original formulation in an input–output setting [194,195]. It has the capability to provide the required system response in the presence of noise and disturbance.

In this chapter, a heading angle tracking algorithm based on robust H_∞ approach for a scaled autonomous multi-wheeled combat vehicle was developed. The introduced technique has the capability to track the desired heading angle by controlling the steering of the front four wheels of the vehicle individually. Several statistical analyses are applied to validate the performance and the robustness of the introduced controller in the presences of the disturbance and noise. The simulation results demonstrated that the proposed H_∞ controller successfully tracked the desired heading angle in a reliable and smooth way, while it is robust against the disturbance and noise.

9.2 H_∞ Controller

9.2.1 Feedback control

The feedback control system must satisfy certain performance specifications as shown in Figure 9-1. It has three components: the plant, the sensors to measure the outputs, and the controller to provide the appropriate control signal. Generally, this system has three inputs in terms of $r(t), d(t), n(t)$ and three outputs in terms of $y(t), u(t), e(t)$ which are described by the following matrix in Equation 9-4.

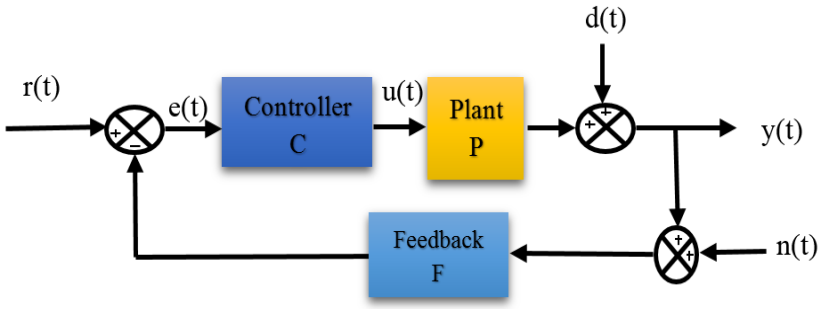


Figure 9-1. Feedback control

In case of unity feedback system, the sensitivity function (S), the complementary sensitivity function (T) and the control sensitivity function (R) are defined as follows:

$$S = \frac{1}{1+L} \tag{9-1}$$

$$T = \frac{L}{1+L} \tag{9-2}$$

$$R = \frac{C}{1+L} \tag{9-3}$$

where, L represents the loop transfer function ($L = PC$) and ($T = 1 - S$).

$$\begin{bmatrix} y \\ u \\ e \end{bmatrix} = \begin{bmatrix} T & S & -T \\ R & -R & -R \\ S & -S & -S \end{bmatrix} \begin{bmatrix} r \\ d \\ n \end{bmatrix} \tag{9-4}$$

where,

$y(t)$ is the actual output;

$e(t)$ is the tracking error;

$u(t)$ is the controller signal;

$r(t)$ is the desired signal;

$d(t)$ is the disturbance input;

$n(t)$ is the measurement noise;

S is the sensitivity function;

T is the complementary sensitivity function;

R is the control sensitivity function.

9.2.2 Weighting function selection

In order to facilitate the robust H_∞ control problem, the vehicle model of an augmented plant P(s) is considered with the weighting functions $W_p(s), W_u(s), and\ W_t(s)$, where represent the error signal, control signal, and output signal respectively as shown in Figure 9-2.

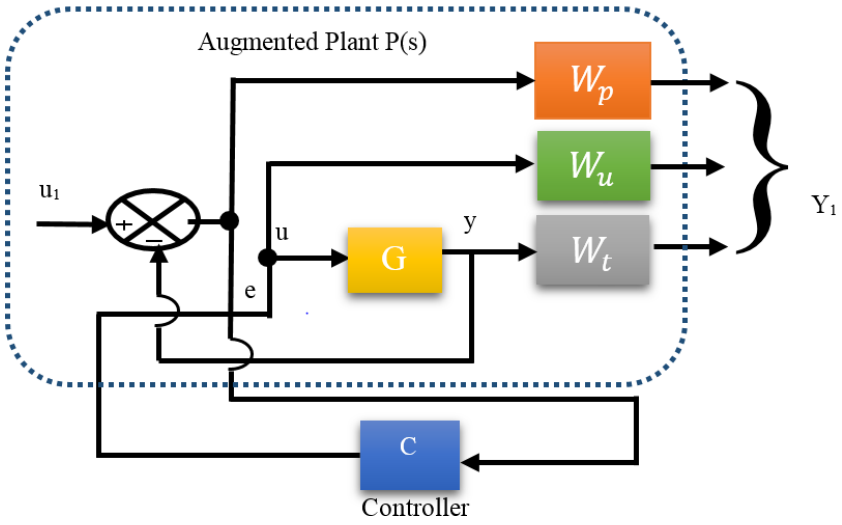


Figure 9-2. Augmented plant P(s)

Consequently, the closed-loop transfer function matrix is the weighted mixed sensitivity Equation 9-5.

$$T_{y1u1} = \begin{bmatrix} W_p\ S \\ W_u\ R \\ W_t\ T \end{bmatrix} \qquad 9\text{-}5$$

Before attempting the controller design, the control and error weighting functions must be selected in a way that reflects the frequency and time domain requirements. These weight

functions play an important role in providing a good feedback design for the system. The selection of the of these weights should satisfy the following constraints: (1) at low frequency the system should be insensitive to external disturbances, (2) at high frequency it should be able to filter unwanted noise signals.

Consequently, the methodology of weight function selection is to compromise between sensitivity and complementary sensitivity over the whole pattern of frequencies. The following consideration based on Equations 9-1, 9-2, and 9-3 should be taken into account to satisfy this compromise as follows:

- The sensitivity function S should be small to reduce the external disturbance effect, which can be achieved by increasing the loop gain of L. Subsequently, as long as the sensitivity function is small, the tracking errors $e(t)$ will be small.
- The complementary sensitivity function T should be kept small to minimize the noise effect applied on the system output, which can be attained by reducing the loop gain of L.
- The control sensitivity function R must be limited in order to make sure that the acting control signal on the system will not exceed the system tolerances.

Subsequently, it can be concluded that the disturbance rejection and errors $e(t)$ minimization necessitates a small sensitivity S. However, noise suppression requires small complementary sensitivity T. It has been noticed that it is not reasonable to decrease S and T to zero simultaneously since the summation of both transfer functions is unity. Consequently, in order to solve this problem, the disturbance and command input are considered low frequency signals and the noise is considered a high frequency signal. Afterwards, by maintaining the sensitivity function as small in the low frequency range and the complementary sensitivity function as small in the high frequency range, this problem can be solved [196,197].

9.2.3 H_∞ controller design

In this section, the developed H_∞ controller for heading angle tracking is discussed. This algorithm has the capability to individually control the steering of four front wheels of the

vehicle in order to tracking the desired heading angle. For this purpose, four H_∞ controllers are developed in the following four sections.

9.2.3.1 R1 Robust controller

The developed H_∞ controller for R1 is designed by selecting the following weight functions:

$$W_p = \frac{s + 105}{1.5\,s + 0.0105}, \quad W_t = \frac{0.03\,s + 1.2}{10\,s + 2}, W_t = 1 \qquad 9\text{-}6$$

Consequently, the obtained H_∞ controller will be as follows:

$$C_{R1} = \frac{-3.58e04\,s^2 - 6307\,s - 4736}{s^3 + 8006\,s^2 + 1664\,s + 11.26} \qquad 9\text{-}7$$

The frequency response of the obtained weight sensitivities is shown in Figure 9-3. Subsequently, the closed loop response is shown in Figure 9-4a, which clarifies that the rise time is 0.5 sec and the settling time is 0.95 sec. On the other hand, the closed loop response for the uncontrolled system is shown in Figure 9-4b.

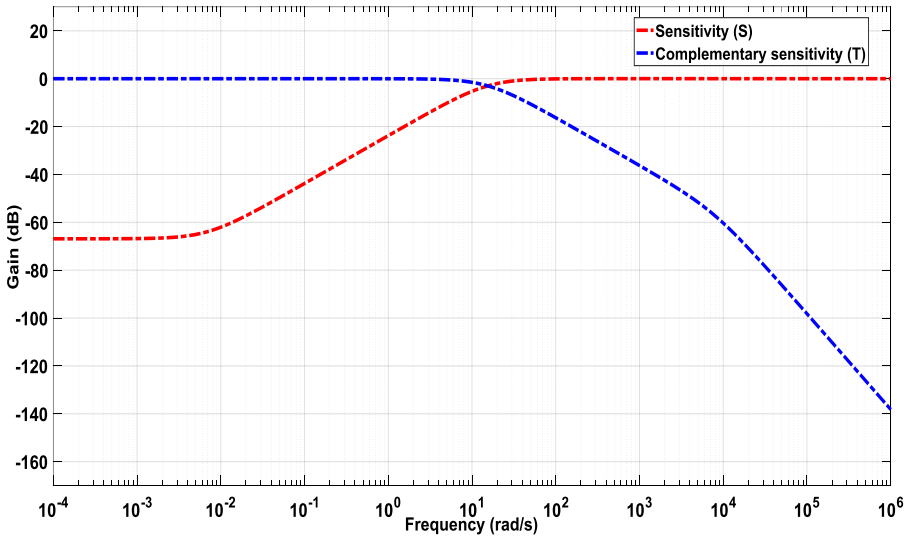
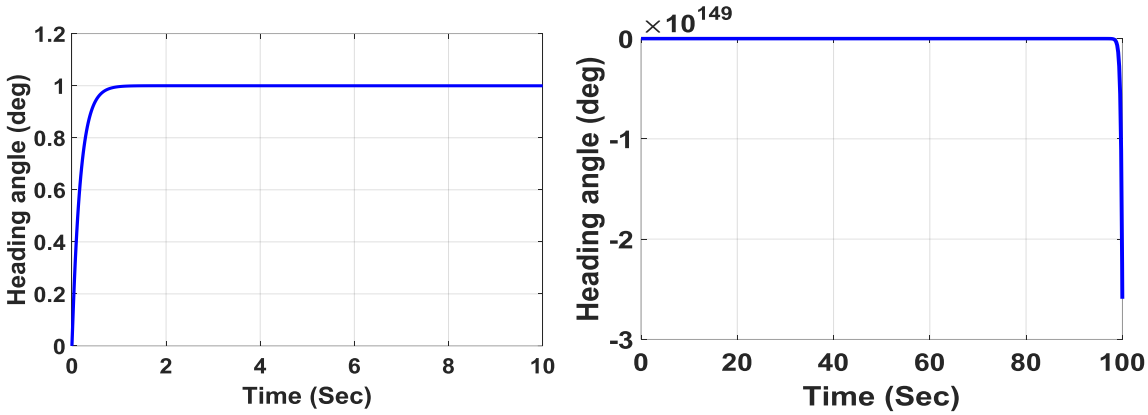


Figure 9-3. Sensitivity and complementary sensitivity weighting functions



(a) (b)

Figure 9-4. (a) Step response of robust closed loop system (b) Step response of uncontrolled closed loop system

9.2.3.2 R2 Robust controller

The developed H_∞ controller for R2 is designed by selecting the following weight functions:

$$W_p = \frac{s + 7.5}{1.5\,s + 0.00075}, W_t = \frac{0.05\,s + 1}{7\,s + 2},\ W_t = 1 \qquad 9\text{-}8$$

Consequently, the obtained H_∞ controller will be as follows:

$$C_{R2} = \frac{6.986e04\,s^2 + 2.71e04\,s + 2.165e04}{s^3 + 2195\,s^2 + 5.189e05\,s + 259.5} \qquad 9\text{-}9$$

The frequency response of the obtained weight sensitivities is shown in Figure 9-5. The closed loop response is shown in Figure 9-6 a. These results clarify that the rise time is 20 sec and the settling time is 60 sec. On the other hand, the closed loop response for the uncontrolled system is shown in Figure 9-6b.

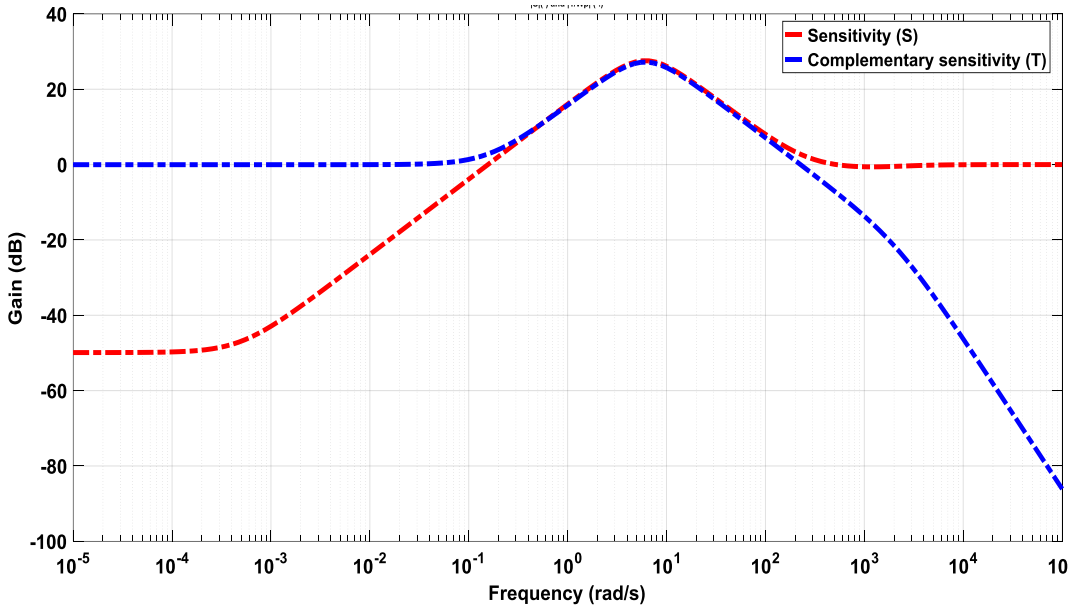
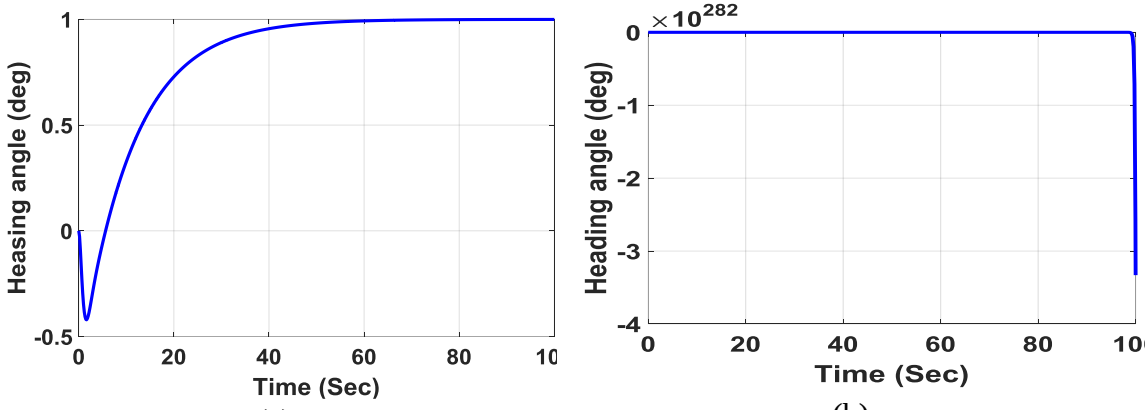


Figure 9-5. Sensitivity and complementary sensitivity weighting functions



(a) (b)

Figure 9-6. (a) Step response of robust closed loop system (b) Step response of uncontrolled closed loop system

9.2.3.3 L1 Robust controller

The developed H_∞ controller for L1 is designed by selecting the following weight functions:

$$W_p = \frac{s + 150}{1.5\,s + 0.015}, W_t = \frac{0.2\,s + 3.25}{8.35\,s + 2.5}, W_t = 1 \qquad 9\text{-}10$$

Consequently, the obtained H_∞ controller will be as follows:

$$C_{l1} = \frac{4.707e04\ s^2 + 1.277e04\ s + 5097}{s^3 + 1.138e04\ s^2 + 341.8\ s + 2.279} \qquad 9\text{-}11$$

The frequency response of the obtained weight sensitivities is shown in Figure 9-7. The closed loop response is shown in Figure 9-8a. It can be noticed that the rise time is 0.33 sec and the settling time is 0.48 sec. On the other hand, the closed loop response for the uncontrolled system is shown in Figure 9-8b.

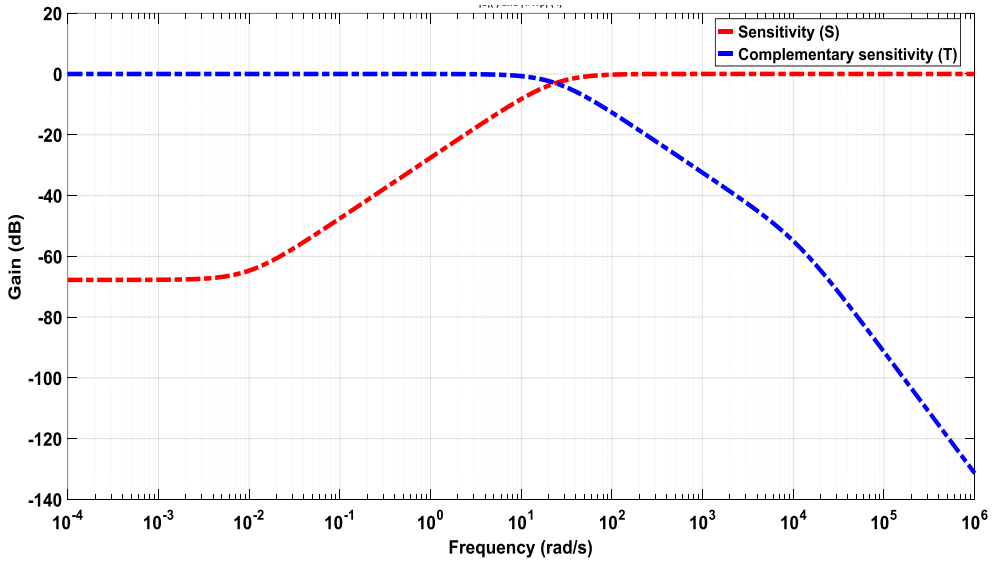
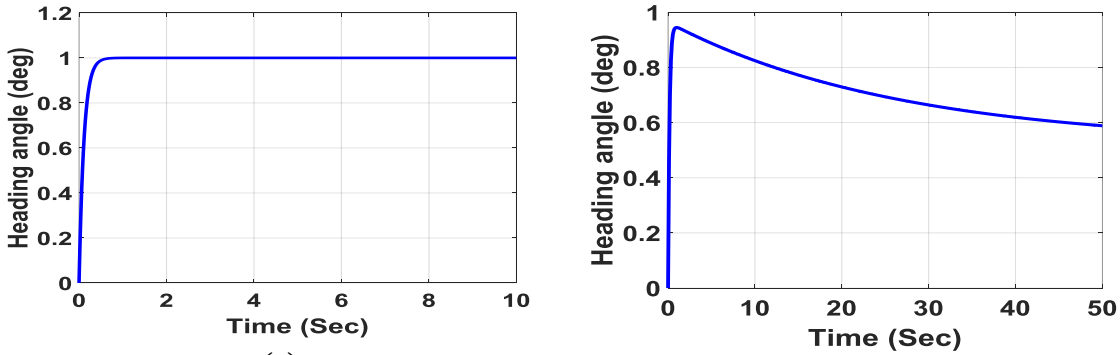


Figure 9-7. Sensitivity and complementary sensitivity weighting functions



(a) (b)

Figure 9-8. (a) Step response of robust closed loop system (b) Step response of uncontrolled closed loop system

9.2.3.4 L2 Robust controller

The developed H_∞ controller for L2 is designed by selecting the following weight functions:

$$W_p = \frac{s + 1100}{11\,s + 0.0011}, W_t = \frac{s + 1.75}{10.21\,s + 1}, W_t = 1 \qquad 9\text{-}12$$

Consequently, the obtained H_∞ controller will be as follows:

$$C_{l2} = \frac{4.798e04\,s^2 + 2.63e04\,s + 1.385e04}{s^3 + 1.305e04\,s^2 + 1455\,s + 0.1453} \qquad 9\text{-}13$$

The frequency response of the obtained weighted sensitivities is shown in Figure 9-9, and the closed loop response is shown in Figure 9-10a. The obtained result clarifies that the rise time is 0.35 sec and the settling time is 0.5 sec. On the other hand, the closed loop response for the uncontrolled system is shown in Figure 9-10b.

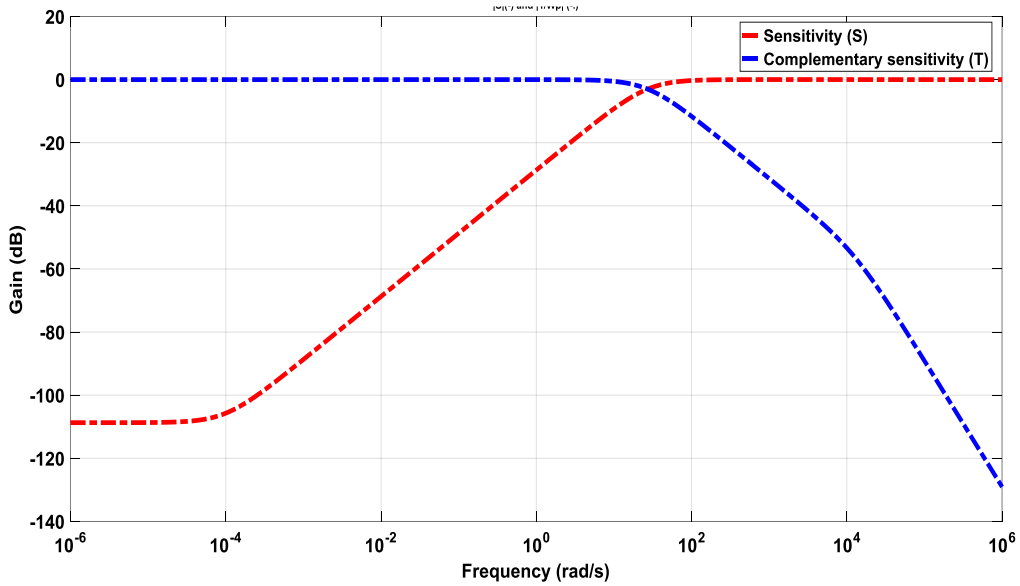
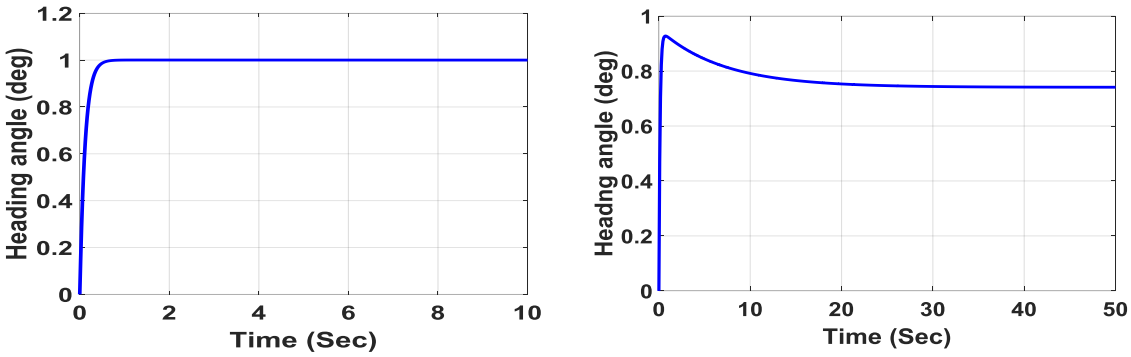


Figure 9-9. Sensitivity and complementary sensitivity weighting functions



(a) (b)

Figure 9-10 (a) Step response of robust closed loop system (b) Step response of uncontrolled closed loop system

9.3 Simulation and Results

In this section, the simulation results of the developed H_∞ controller are discussed. The closed loop step response performance of the system for the developed H_∞ controller compared with the uncontrolled system is shown in Figure 9-11. It has been noticed that the steady state error of H_∞controller reaches zero while the settling time is 0.12 sec compared with the uncontrolled system. As such, the steady state error of the uncontrolled system is high.

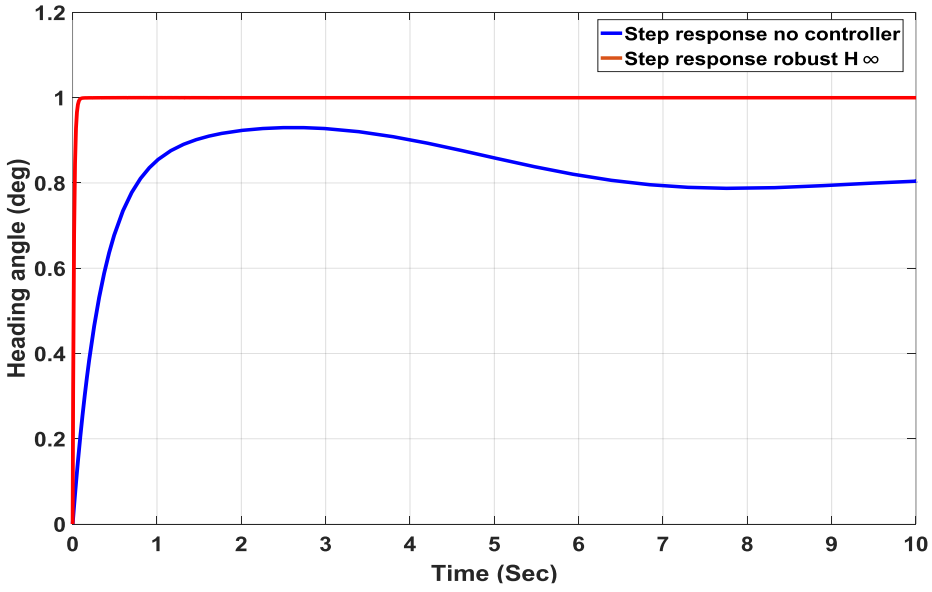


Figure 9-11 Transient response of the heading angle for $\boldsymbol{H}_\infty$ controller compared with uncontrolled system

Several scenarios are proposed considering prescribed heading angles to evaluate the performance of the developed heading angle tracking algorithm.

The first scenario is developed to achieve the desired heading angle of 25 degrees as shown in Figure 9-12, where the desired angle is represented by the blue line and the actual response is red. The obtained result demonstrated that the proposed H_∞ controller has the capability to track the desired heading angle with very low steady state error and steeling time.

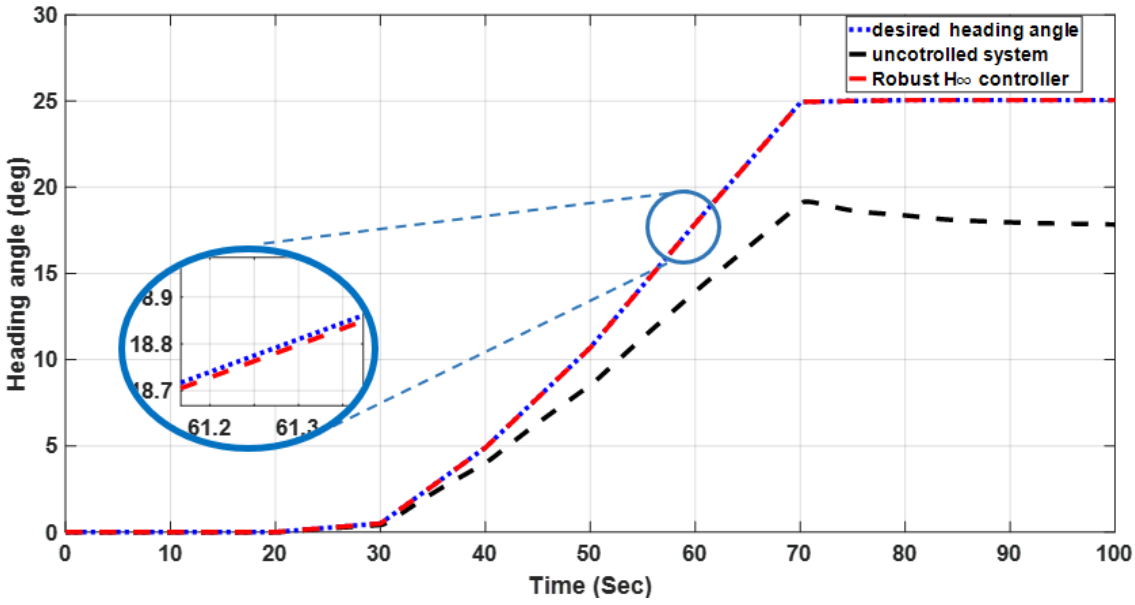


Figure 9-12. Comparison of robust controller and uncontrolled to achieve the desired heading angle of 25 degrees

In the second scenario, a sinusoidal heading angle is considered as a reference, which varies between 35 to -35 degrees as shown in Figure 9-13. The tracking results show that the vehicle is able to track the desired heading angle even with such high angle variations.

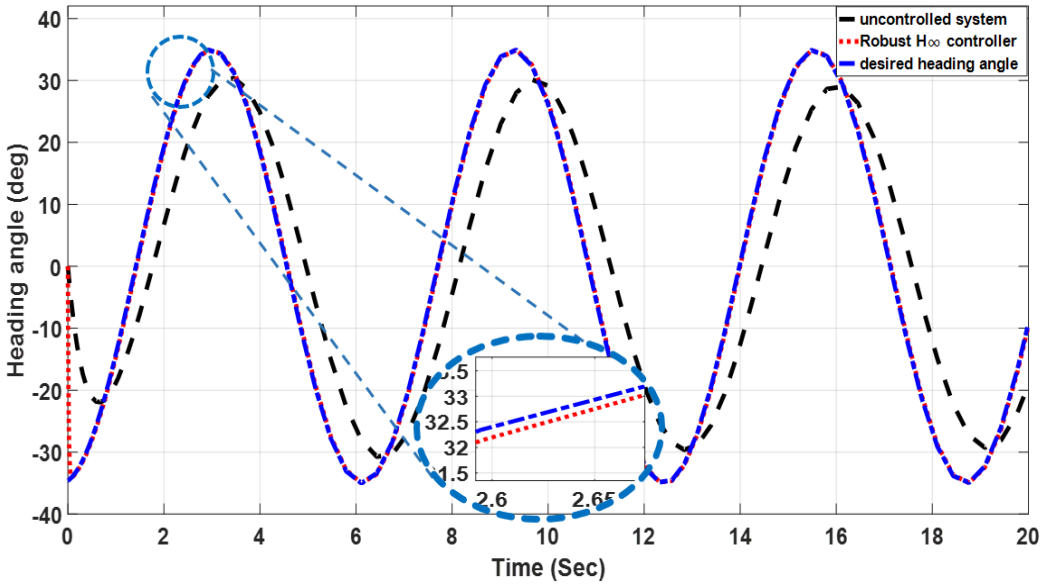


Figure 9-13. Comparison of robust controller and uncontrolled to achieve a sinusoidal desired heading angle varying 35: -35 degrees

The third scenario is more realistic due to the consideration of using a measured heading angle from a real test of the vehicle as a reference. This is the desired heading in this scenario as shown in Figure 9-14. The developed H_∞ controller shows good tracking for the desired heading angle compared with the uncontrolled one.

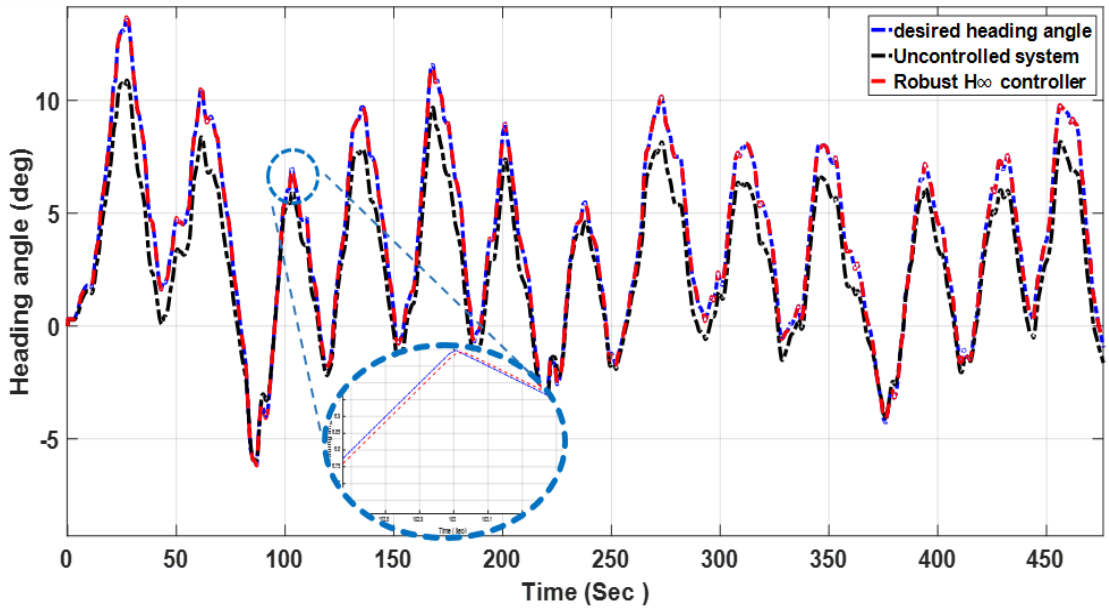


Figure 9-14. Comparison of robust controller and uncontrolled to achieve the desired angle

9.3.1 Validation in presence of disturbance and noise

a) Disturbance rejection

The performance evaluation of the proposed robust algorithms in the presence of disturbance is considered. For this purpose, an impulse disturbance is applied at the system output. The system step response in presence of the applied disturbance is shown in Figure 9-15. It can be noticed that the system rejects 50% of the disturbance within 0.04 sec and rejects 95% of the disturbance within 0.076 sec in addition to the lowest control effort. Consequently, the developed algorithm has the capability to reject the applied disturbance within a short time.

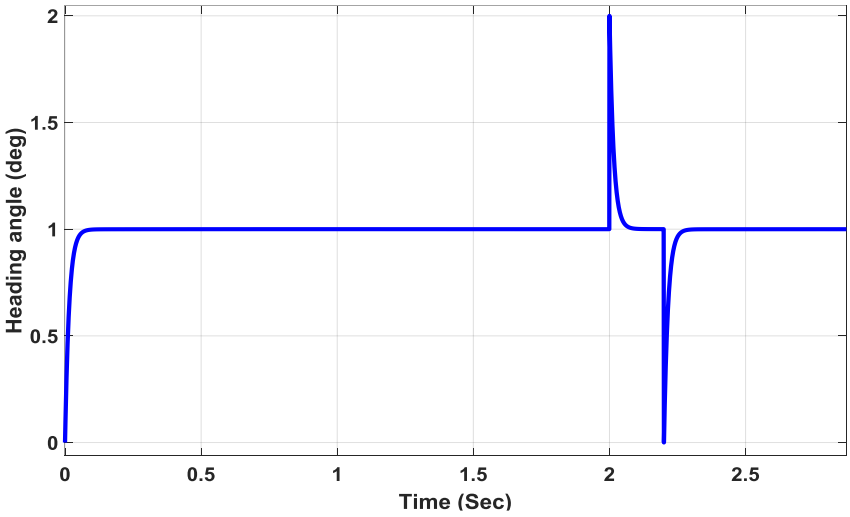


Figure 9-15. Disturbance rejection using $\boldsymbol{H}_{\infty}$ controller

b) Noise sensitivity

The performance evaluation of the developed H_{∞} algorithms in the presence of noise is considered. For this purpose, a white Gaussian noise is applied at the output of the system. The control effort in the presence of noise is shown in Figure 9-16, which shows the capability of the H_{∞}controller to track the desired heading in the presence of noise, which means that the system is less sensitive to additive noise.

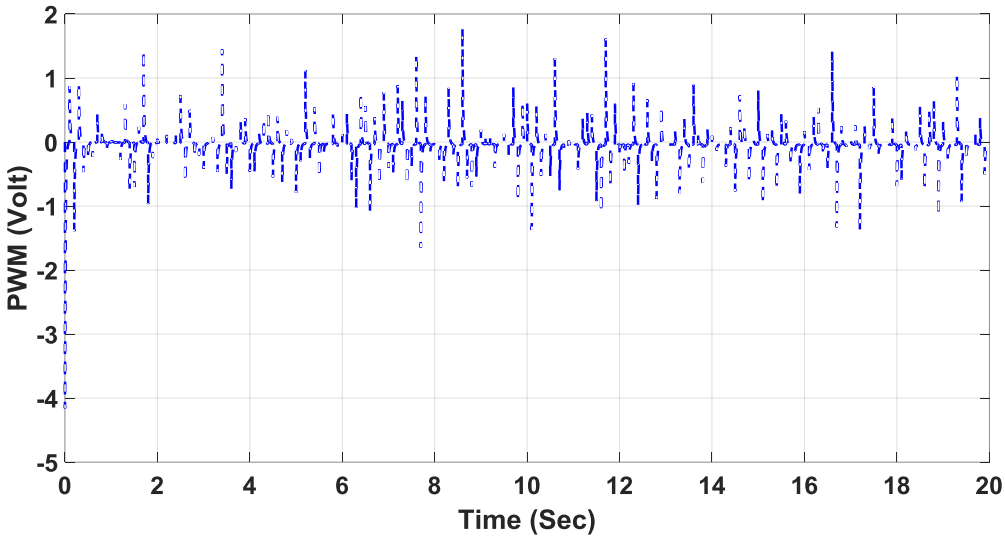


Figure 9-16. Control effort in the presence of noise

9.4 Processor-In-The-Loop (PIL) Co-simulation

In this section, the Processor-In-The-Loop (PIL) co-simulation of the proposed robust H_∞ controller is applied. The same procedures were used in Chapter 8 for the PIL experiment will be applied to the developed H_∞ controller in this section. In this way, using the PIL can achieves a more realistic environment where the developed control algorithm will run on a processor. Therefore, the performance of the PIL and simulation results can be compared.

The comparative analysis of the developed control algorithm using the simulation in MATLAB/ Simulink environment and the PIL is presented. The obtained closed-loop responses for PIL and the simulation are shown in Figure 9-17. It can be noticed that the PIL result is close to the simulation result, which means that the developed controller has the capability to achieve a good implementation on the processor environment.

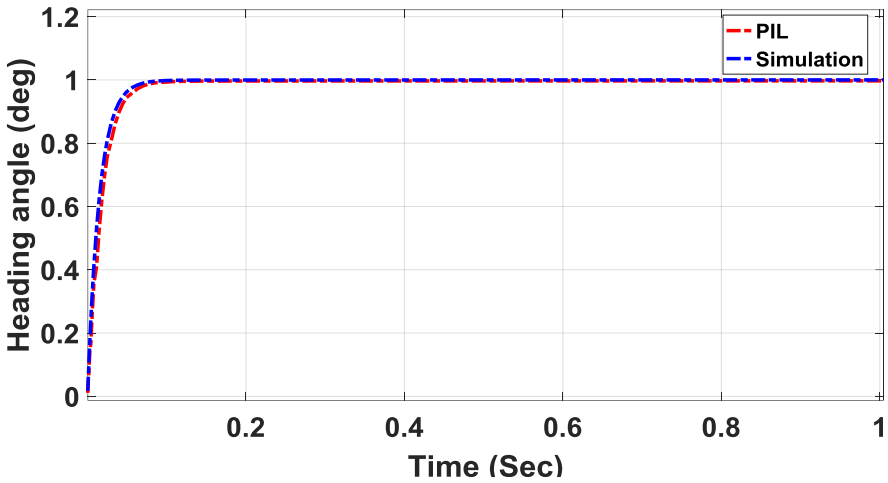


Figure 9-17 Comparison of transient response of the heading angle for H_∞ controller in simulation and PIL

9.5 Comparative Analysis of the Developed Control algorithms

The developed heading angle tracking controllers in chapters 8 and 9 provide a very close tracking response, with minimum settling time, to the application of heading angle tracking. The PID controller is designed first, using the Ziegler–Nichols method and subsequently, a fine tuning through simulation for better results. In addition, the FL heading angle tracking controller is designed. Once more, triangular and trapezoidal membership functions with overlap are used. Finally, a robust H_∞ algorithm is introduced, where the augmented plant is structured by adjusting the weighting functions. On the other hand, the PIL co-simulation for the three developed controllers shows good implementation on the processor environment. Consequently, a comparative study of the developed controllers is carried out to stop by the best controller that has the capability to track the desired heading, while it is less affected by the noise and disturbance.

In this section a comparative evaluation to assess the feasibility of the developed control algorithms is discussed. It is appropriate at this stage to compare the performance of PID, FLC, and robust H_∞ algorithms for tracking the desired heading to emphasize the merits and demerits of each controller. The performance of the three controllers is experienced through two scenarios, which were designed to regulate the steering of the four front wheels of the vehicle to follow the desired heading. In addition, the disturbance and noise are applied to the three controllers, so that any comparison is justifiable.

9.5.1 First scenario

In this scenario, the performance evaluation of the proposed control algorithms for tracking and reaching the predefined heading angle is shown in Figure 9-18.

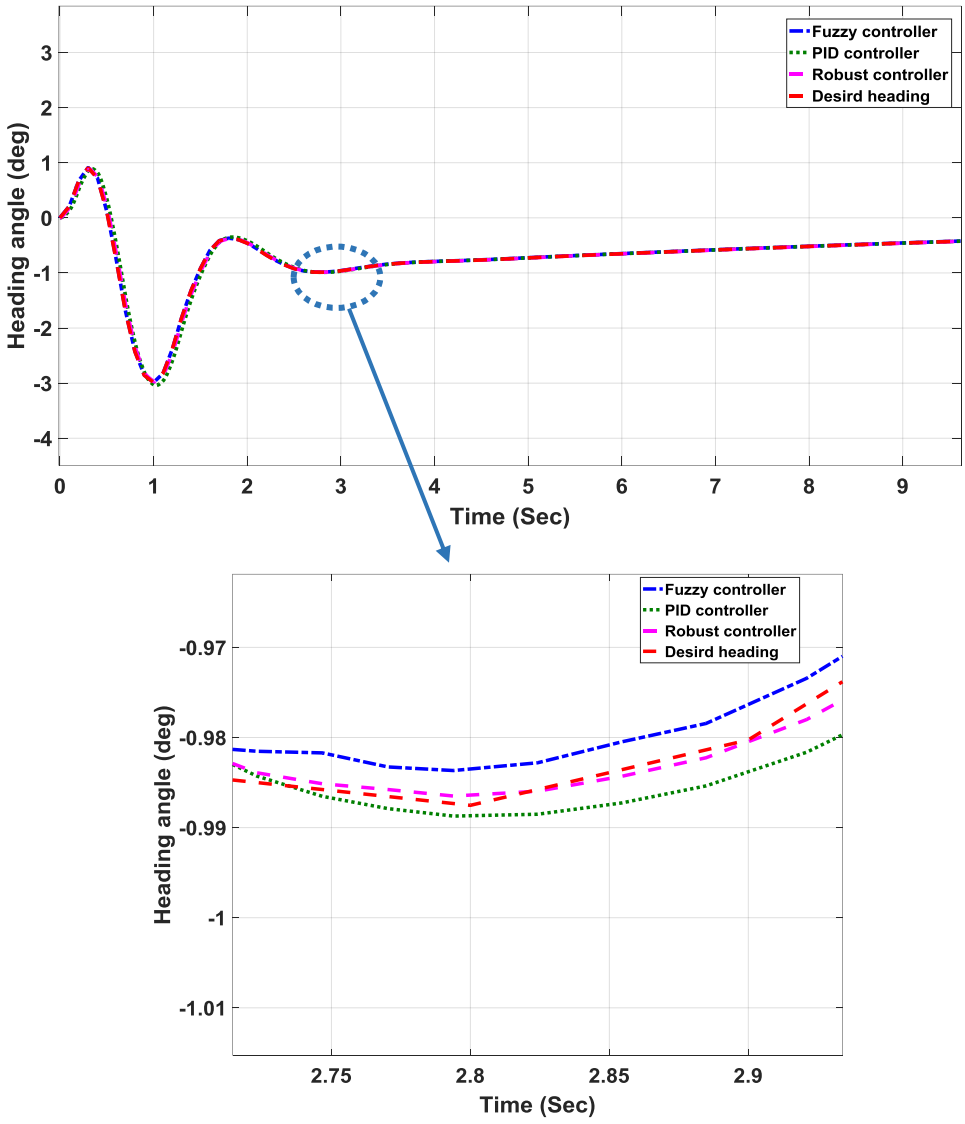


Figure 9-18. Fi-rst scenario: Comparison of heading angle tracking using fuzzy and PID, robust $\boldsymbol{H}_{\infty}$ controllers

It has been noticed that the three controllers are able to track and follow the desired heading angle, where both FLC and robust H_{∞} controller have very close trajectories to the desired heading compared with the PID controller. However, the error between the desired and the obtained heading using a robust controller is the smallest, which means more accurate in tracking.

9.5.2 Second scenario

In this scenario, the desired heading angle is predefined while showing a continuous change over time in order to be more challenging for the controllers to track. The obtained tracks performance of the developed controllers is shown in Figure 9-19, which clarifies that the robust H_∞ controller successfully tracking the desired heading angle accurately compared with the PID and FL.

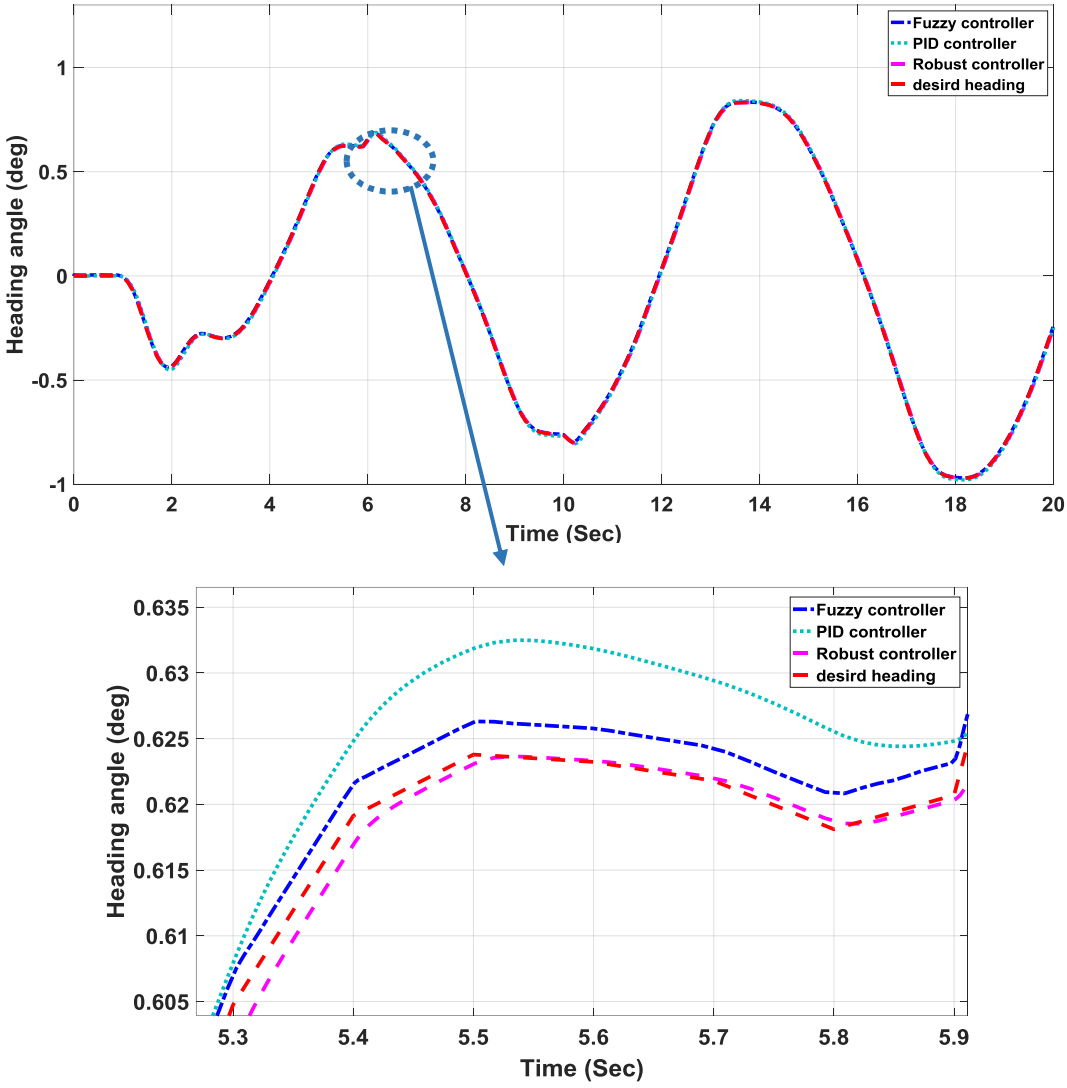


Figure 9-19. Second scenario: Comparison of heading angle tracking using fuzzy and PID, robust H_∞ controllers

9.5.3 Validation in presence of disturbance and noise

a) Disturbance rejection

In this section, the performance evaluation of the proposed algorithms for disturbance rejection during a maneuver is introduced. An impulse disturbance is applied to the system output. The obtained result in the presence of disturbance is shown in Figure 9-20.

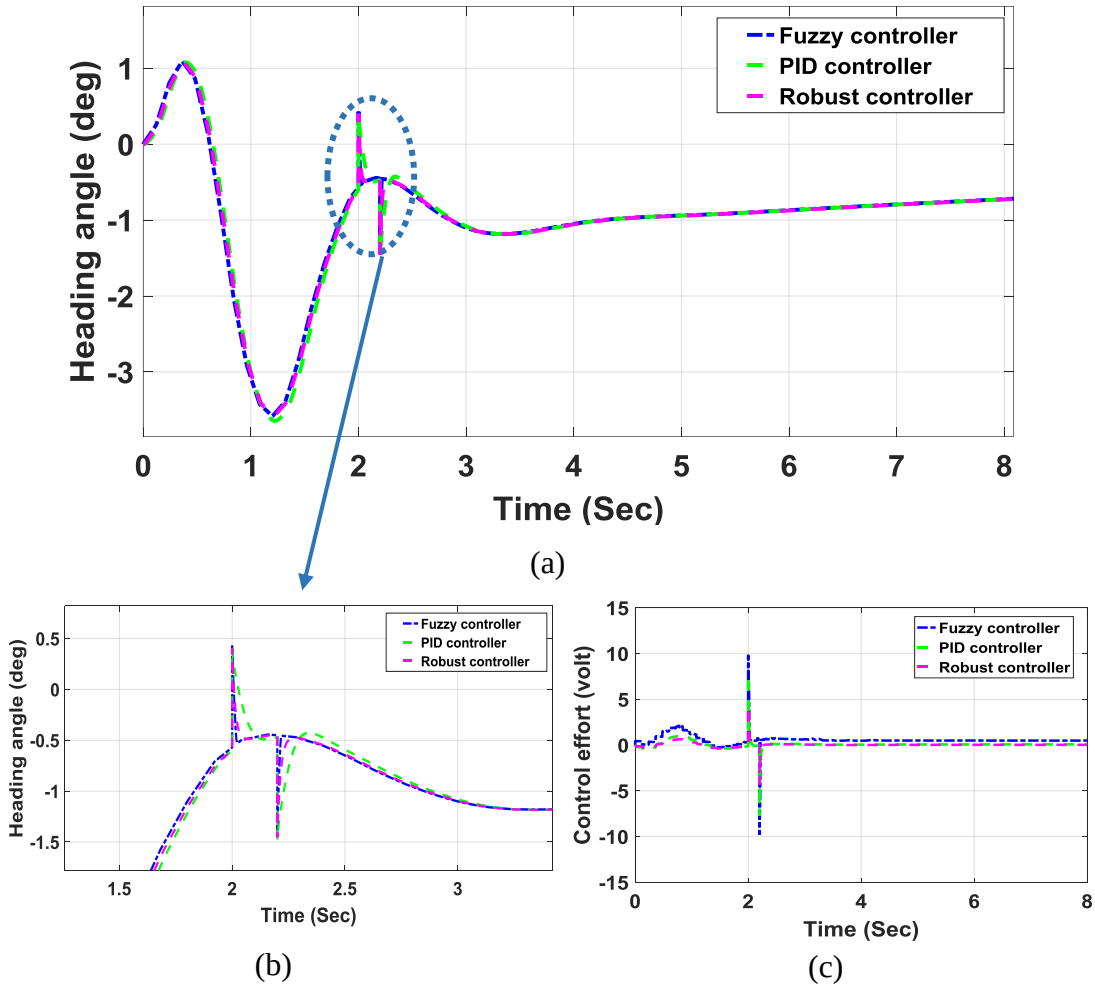


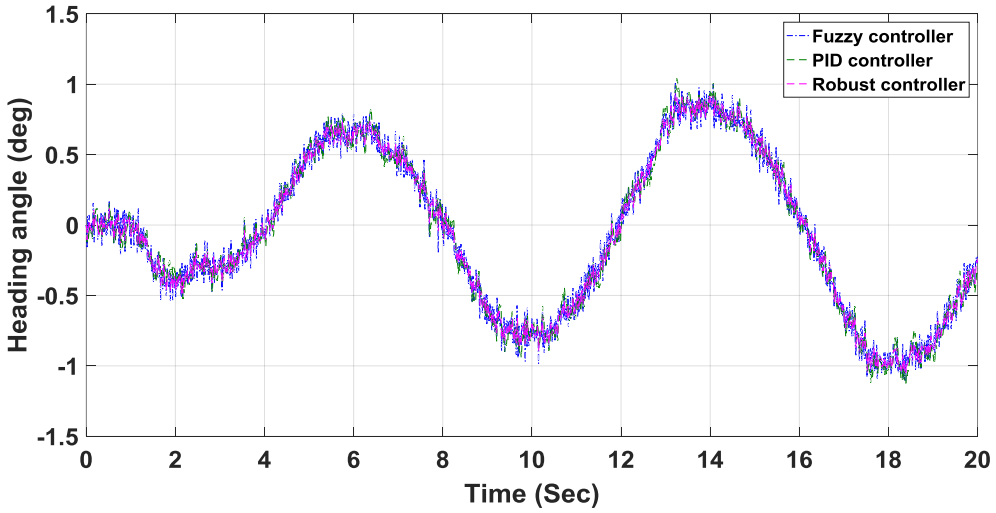
Figure 9-20. (a) Comparison of disturbance rejection capabilities for PID, fuzzy logic Robust controller, (b) Disturbance rejection, (c) Comparison of control effort

It has been noticed that the convergence of the fuzzy controller is better than the PID and robust H_∞ controllers as it rejects 50% of the disturbance within 0.01 sec and rejects 95% of the disturbance within 0.018 sec. On the other hand, the PID controller

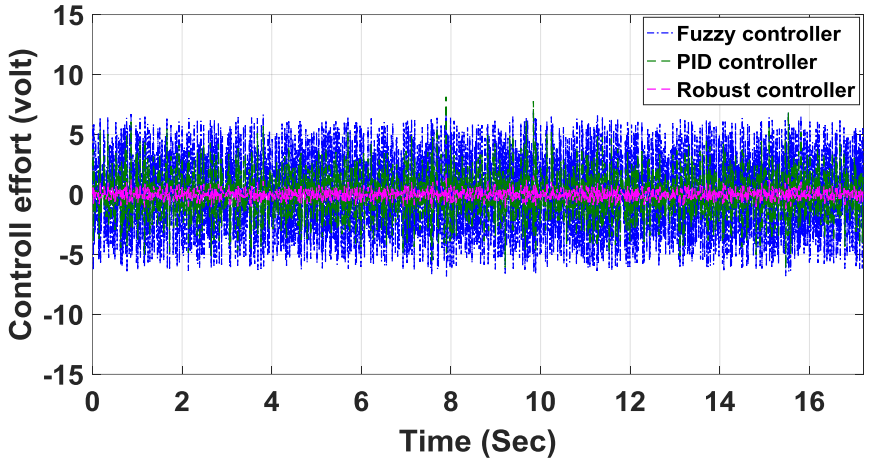
rejects 50% of the disturbance within 0.05 sec and rejects 95% of the disturbance within 0.1 sec. In addition, the robust controller system rejects 50% of the disturbance within 0.016 sec and rejects 95% of the disturbance within 0.04 sec. However, the control effort variance of the robust H_∞ controller is 7.0430 which is considered the lowest control effort compared with the PID and fuzzy logic which is 18.8013 and 193.4385 respectively. These results clarify that the amount of energy or power necessary for the robust H_∞ controller to control the vehicle heading angle is small.

b) Noise sensitivity

In this section, the performance of the proposed algorithms will be evaluated in the presence of noise. For this purpose, a white Gaussian noise was applied at the output of the system. The obtained response in the presence of noise is shown in Figure 9-21. Based on the obtained result the fuzzy logic controller is sensitive to additive noise compared with the PID and H_∞ controllers. Moreover, by comparing the control effort of the developed controllers, the control effort variance of the robust controller is 0.2123 which is considered the lowest compared with the PID and fuzzy which is 4.8129 and 57.4326 respectively. These results explain the sensitivity of the fuzzy and PID controllers to the applied noise.



(a)



(b)

Figure 9-21. (a) Comparison of noise sensitivity using PID, fuzzy logic, Robust controllers, (b) Comparison of control effort

9.6 Chapter Summary

In this paper, a robust H_∞ heading angle tracking algorithm was developed for an autonomous scaled multi-wheeled combat vehicle. The proposed algorithm enables the vehicle to track the desired heading angle autonomously by individually controlling the steering of the four front wheels. The methodology of the H_∞ controller weight function selection was discussed, which is based on the compromise between sensitivity and complementary sensitivity. In addition, the system is validated in presence of disturbance and noise, which shows a good disturbance rejection. Moreover, the controller is less sensitive to the applied white Gaussian noise. Several scenarios are considered for tracking the predefined heading angles.

In addition, the developed controller is implemented in a processor to be tested together with the vehicle model running on PC for heading control through the PIL. The PIL experimental results show a good implementation in the processor environment compared with simulation result. The obtained results demonstrated that the proposed H_∞ controller has the capability to regulate the steering of the four front wheels to enable the vehicle to follow the desired heading angle in a reliable and smooth way in addition to the robustness of the system in presence of disturbance and noise.

Finally, the comparative study of the developed three controllers shows their capability to regulate the four front wheels of the vehicle for tracking the desired heading angle. However, the robust controller is closer and smoother in tracking to the heading angle compared with the PID and FLC. In addition, in the presence of disturbance the robust controller has achieved a good disturbance rejection with low control effort compared with the FLC that has a high control effort. Subsequently, in the presence of noise the robust controller shows a good result, which is less sensitive to the applied noise compared with the PID and FL controllers, furthermore, is low in the control effort.

CHAPTER 10

Conclusions and Future Works

In this chapter the conclusions obtained from the thesis work are described then the major contributions are listed. The future work and recommendations are then presented and finally the publications done during the course of PhD study are listed.

10.1 Conclusions

The most important findings of the current work are concluded as follows:

- The theory of optimal control has been successfully employed to generate an optimal path for a multi-wheeled combat vehicle to move from a given starting point and safely reach its target point while avoiding the destination border and any obstacles imposed on the vehicle path.

- The artificial potential field method is used to represent both the target location and different obstacles which are placed in different locations.

- Based on the combination of artificial potential filed and optimal control theory, the proposed algorithm is able to avoid obstacles and achieve global optimization simultaneously.

- The artificial neural network has been successfully employed to solve the real time problem, and it is able to generate the vehicle optimal path, based on the combination between the APF and optimal control theory.

- The Pontryagin's minimum principle technique has been successfully guaranteed the global optimal solution for the generated paths compared with dynamic programming, while can save approximately 70% in computing time.

- The hybrid positing technique has been successfully combined both tightly and loosely coupled GPS/INS integration using KF and achieve reliable positioning

performance even in adverse GPS environments where the number of satellites falls below the minimum.

- The identified model of the scaled multi-wheeled combat vehicle was succeeded in providing an accurate vehicle model, which validated with real data and achieved 88.44% of the output.

- The developed PID, fuzzy, and robust H_∞ controllers successfully regulated the four front wheels of the vehicle in order to achieve and track the desired vehicle heading angle. However, the H_∞ is more accurate in the tracking compared with PID and fuzzy controllers.

- The developed PID, fuzzy, and robust H_∞ controllers have been successfully achieved a proper implementation in the processor environment using processor-in-the-loop co-simulation.

- The developed H_∞ controller has successfully achieved a good disturbance rejection with low control effort, which rejects 50% of the disturbance within 0.016 sec, and rejects 95% of the disturbance within 0.04 sec compared with the PID and fuzzy.

10.2 Current Research Contributions

- The current research addresses the research gap in the literature which relates to applying autonomy feature to the scaled 8x8 combat vehicle when moving between two locations.

- The current research provides a clear understanding of the generated optimal path where the optimal control theory is integrated with APF providing the optimal collision free path planning.

- Introduced a comparative performance analysis of the Pontryagin's Minimum Principle and Dynamic Programming techniques that showed the capability of the PMP to generate the global optimal solution in our case.

- Developed a hybrid positioning framework based on the combination of tightly and loosely coupled GPS/INS integration using Kalman Filter which has the capability to achieve reliable positioning performance where the number of satellites falls below the minimum.

- Developed a model of scaled multi-wheeled vehicle using system identification techniques and successfully identifying and achieving 88.44% of the output.

- Conducted a comparative performance analysis of the developed control algorithms and proved the accuracy of the H_∞ controller to track the desired heading angle and robustness in the presence of disturbance and noise.

10.3 Future Work and Recommendation

- In this research, an optimal path planning technique has been employed to generate the vehicle optimal path and avoiding the imposed static obstacles. Therefore, it is recommended in the future work to change the static obstacles to dynamic obstacles. This should be done in both the simulation environment and field testing through various scenarios.

- It is necessary to improve the mechanical parts of the scaled multi wheeled combat vehicle, which will help to identify a more accurate vehicle model. Consequently, the implementation of the designed control system can achieve better results.

- It is also recommended to implement and examine the developed control systems in the real vehicle in order to examine their robustness against disturbances and noise. This should be done through various field testing.

- Enhancing the vehicle positioning estimation by integrating a stereo camera, INS, and GPS, which will provide an accurate location estimation for the vehicle. By this way, the advantage of the inertial sensor's fast response and visual sensor's slow drift can be considered.

10.4 Publication

10.4.1 Referred Journal Papers

Amr Mohamed., El-Gindy, M. and Ren, J., (2018). Advanced control techniques for unmanned ground vehicle: literature survey. International Journal of Vehicle Performance, vol.4, no.1, pp.46-73.

Amr Mohamed., Ren, J., Sharaf, A.M. and EI-Gindy, M., (2018). Optimal path planning for unmanned ground vehicles using potential field method and optimal control method. International Journal of Vehicle Performance, vol. 4, no. 1, pp.1-14.

Amr Mohamed, Moustafa EI-Gindy, Xishi Huang, Jing Ren, Haoxiang Lang, (2018). Real Time Optimal Control of Multi-Wheeled Combat Vehicles using Artificial Neural Network and Potential Fields. Advancied in Military Technology, AIMT, vol. 13 no. 2, pp. 193-207.

Amr Mohamed, Jing Ren, Moustafa EI-Gindy, Haoxiang Lang, A.N Ouda, (2018). "Literature Survey for Autonomous Vehicles: Sensor Fusion, Computer Vision, System Identification and Fault Tolerance". International Journal of Automation and Control (IJAAC), vol. 12, no. 4, pp.555-581.

Amr Mohamed, Jing Ren, Moustafa EI-Gindy, and Haoxiang Lang. (2017). Optimal path planning for an autonomous articulated vehicle with two trailers. International Journal of Automation and Control (IJAAC), (in press).

Amr Mohamed, A.N. Ouda, Jing Ren, Moustafa El-Gindy. (2018). Processor-in-the-Loop Co-Simulations and Control System Design for a Scaled Autonomous Multi-Wheeled Combat Vehicle. International Journal of Automation and Control (IJAAC). (in press).

A.N Ouda, Amr Mohamed, Jing Ren, Moustafa EI-Gindy, Haoxiang Lang. (2018). Hybrid Positioning Technique Based Integration of GPS/INS for an Autonomous Vehicle Navigation. Submitted control engineering practice, (under review).

A.N Ouda, Amr Mohamed, Jing Ren, Moustafa EI-Gindy, Haoxiang Lang, (2018) "Development and Modeling of Remotely Operated Scaled Multi-Wheeled Combat Vehicle Using System Identification and Model Validation. (accepted).

10.4.2 Referred Conference Papers

Amr Mohamed, Jing Ren, Haoxiang Lang and Moustafa EI-Gindy, (2017). Optimal collision free path planning for an autonomous articulated vehicle with two trailers. IEEE International Conference on Industrial Technology (ICIT 2017) March 22-25, Toronto, ON, pp. 860-865.

Amr Mohamed, Jing Ren, Haoxiang Lang and Moustafa EI-Gindy, (2017). Optimal collision free path planning for an autonomous multi wheeled combat vehicle. ASME (2017). International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 3: 19th, Cleveland, August 26-29, Ohio, USA.

Amr Mohamed, Moustafa El-Gindy, Jing Ren, (2018). Design and Performance Analysis of Robust H_∞ controller for a Scaled Autonomous Multi-Wheeled Combat Vehicle Heading Control" ASME (2018). International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, August 26-29, Quebec City, QC, Canada.

Appendix 1

$A11$ to $A66$ in above equation are 3 X 3 matrices are described as follows:

$$O_{3x3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A11 = \begin{bmatrix} 0 & \dot{\lambda}\sin(\varphi) & -\dot{\lambda}\cos(\varphi) \\ -\dot{\lambda}\sin(\varphi) & 0 & -\dot{\varphi} \\ \dot{\lambda}\cos(\varphi) & \dot{\varphi} & 0 \end{bmatrix}, A12 = I_3, A13 = I_3 \mathrm{X} O_{3x3},$$

$$A14 = I_3 \mathrm{X} O_{3x3},$$

$$A15 = I_3 \mathrm{X} O_{3x3},\ A16 = I_3 \mathrm{X} O_{3x3}$$

$$A21 = \begin{bmatrix} -g/r & 0 & 0 \\ 0 & -g/r & 0 \\ 0 & 0 & \frac{2g}{r+h} \end{bmatrix}$$

$$A22 = \begin{bmatrix} 0 & \left(2\omega_e + \dot{\lambda}\right)\sin(\varphi) & -\left(2\omega_e + \dot{\lambda}\right)\cos(\varphi) \\ -\left(2\omega_e + \dot{\lambda}\right)\sin(\varphi) & 0 & \dot{\varphi} \\ \left(2\omega_e + \dot{\lambda}\right)\cos(\varphi) & \dot{\varphi} & 0 \end{bmatrix}$$

$$A23 = \begin{bmatrix} 0 & -f_u & f_n \\ f_u & 0 & -f_e \\ -f_n & f_e & 0 \end{bmatrix}, A24 = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{21} & R_{32} & R_{33} \end{bmatrix}, A25 = I_3 \mathrm{X} O_{3x3}, A26 = I_3 \mathrm{X} O_{3x3}$$

$$A31 = I_3 \mathrm{X} O_{3x3}\ , A32 = I_3 \mathrm{X} O_{3x3},$$

$$A33 = \begin{bmatrix} 0 & \left(\omega_e + \dot{\lambda}\right)\sin(\varphi) & -\left(\omega_e + \dot{\lambda}\right)\cos(\varphi) \\ -\left(\omega_e + \dot{\lambda}\right)sin(\varphi) & 0 & -\dot{\varphi} \\ \left(\omega_e + \dot{\lambda}\right)cos(\varphi) & \dot{\varphi} & 0 \end{bmatrix}$$

$$A34 = I_3 \mathrm{X} O_{3x3}\ ,\ \ A35 = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{21} & R_{32} & R_{33} \end{bmatrix}, A36 = I_3 \mathrm{X} O_{3x3}$$

$A41$=$\mathrm{I_3XO_{3x3}}$,$A42 = \mathrm{I_3XO_{3x3}}$, $A43 = \mathrm{I_3XO_{3x3}}$

$$A44 = \begin{bmatrix} -\beta_{fx} & 0 & 0 \\ 0 & -\beta_{fy} & 0 \\ 0 & 0 & -\beta_{fz} \end{bmatrix}, A45 = \mathrm{I_3XO_{3x3}}$$

$$A46 = \mathrm{I_3XO_{3x3}}$$

$A51 = \mathrm{I_3XO_{3x3}}$, $A52 = \mathrm{I_3XO_{3x3}}$, $A53 = \mathrm{I_3XO_{3x3}}$, $A54 = \mathrm{I_3XO_{3x3}}$

$$A55 = \begin{bmatrix} -\beta_{\omega x} & 0 & 0 \\ 0 & -\beta_{\omega y} & 0 \\ 0 & 0 & -\beta_{\omega z} \end{bmatrix}$$

$A56 = \mathrm{I_3XO_{3x3}}$ $A61 = \mathrm{I_3XO_{3x3}}$, $A62 = \mathrm{I_3XO_{3x3}}$, $A63 = \mathrm{I_3XO_{3x3}}$, $A64 = \mathrm{I_3XO_{3x3}}$, $A65 = \mathrm{I_3XO_{3x3}}$,

$$A66 = \begin{bmatrix} \delta_{GPS_x} & 0 & 0 \\ 0 & \delta_{GPS_y} & 0 \\ 0 & 0 & \delta_{GPS_z} \end{bmatrix}$$

REFERENCES

[1] Peeie, M.H.B., Ogino, H. and Oshinoya, Y., (2014). Skid control of small electric vehicles (Direct yaw moment control using tire steer angle). Proceedings of the School of Engineering of Tokai University, vol.39, pp.73-80.

[2] Brahim Gasbaoui, Abdelfatah Nasri, (2012). Artificial Intelligence Application's for 4WD Electric Vehicle Control System. Intelligent Control and Automation, vol.3 No.3,pp.236-242

[3] Sun, Z., Cai, W.C., Liao, X.H., Dong, T. and Song, Y.D., (2006). Adaptive path control of unmanned ground vehicles (UGVs). In Proceedings of the 38th Southeastern Symposium on System Theory Tennessee Technological University Cookeville., TN, USA March 5-7, pp. 508-511.

[4] Lei Yuan, Hong Chen , Bingtao Ren , Haiyan Zhao, (2015). Model predictive slip control for electric vehicle with four in-wheel motors. Proceedings of the 34th Chinese Control Conference July 28-30, Hangzhou, China. pp. 7895-7900

[5] Zhang Y, Yan Y, Yan N, Tian P, (2014). Research and simulation of trajectory tracking control algorithm for multi-wheel independent drive skid steering unmanned ground vehicle. Advances in Intelligent Systems and Computing. vol. 279, pp. 439-448

[6] Kanayama Y, Kimura Y, Miyazaki F, Noguchi T, (1990). A stable tracking control method for an autonomous mobile robot. Proceedings., IEEE International Conference on Robotics and Automation. pp.384-389.

[7] Noor MZH, Zain, S. A. S. M, Mazalan L, (2013). Design and development of remote-operated multi-direction Unmanned Ground Vehicle (UGV). IEEE 3rd International Conference on System Engineering and Technology. IEEE; pp.188-192

[8] Kim, J. and Kim, H., (2007). Electric Vehicle Yaw Rate Control using Independent In-Wheel Motor. Power Conversion Conference – Nagoya, April IEEE 2007. PCC '07 , pp. 705-710.

[9] Matsumura S, Omatu S, Higasa H, (1993). Speed control of an electric vehicle system using PID type neurocontroller. Proceedings of International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), vol,1,pp.:661-664

[10] Yin, G., Li, J., Jin, X., Bian, C., & Chen, N., (2015). Integration of motion planning and model-predictive-control-based control system for autonomous electric vehicles. Taylor & Francis, VILNIUS.Transport, vol. 30, No. 3, pp. 353-360.

[11] Sakai S, Sado H, Hori Y, (1999). Motion control in an electric vehicle with four independently driven in-wheel motors. IEEE/ASME Transactions on Mechatronics, New York. , vol. 4, No. 1, pp. 9-16.

[12] Yin G, Wang R, Wang J, (2015). Robust control for four wheel independently-actuated electric ground vehicles by external yaw-moment generation. International Journal of Automotive Technology, vol. 16, No. 5, pp. 839-847.

[13] Elshazly O, Abo-Ismail A, Abbas HS, Zyada Z, (2014). Skid steering mobile robot modeling and control. UKACC International Conference on Control (CONTROL), 2014. IEEE,pp.:62-67.

[14] Lucet E, Grand C, Bidaud P., (2010). Sliding-mode velocity and yaw control of a 4WD skid-steering mobile robot. In Brain, body and machine, Berlin Heidelberg, Springer; vol 83, pp.247-258.

[15] Kang J, Kim W, Lee J, Yi K., (2010). Design, implementation, and test of skid steering-based autonomous driving controller for a robotic vehicle with articulated suspension. Journal of Mechanical Science and Technology., Korean Society of Mechanical Engineers, Heidelberg, vol. 24, No. 3, pp. 793-800.

[16] Aslam J, Qin S, Alvi MA, (2014). Fuzzy sliding mode control algorithm for a four-wheel skid steer vehicle. Journal of Mechanical Science and Technology, vol. 28, No. 8, pp. 3301-3310.

[17] Kececi EF, Tao G, (2006). Adaptive vehicle skid control. Mechatronics, Elsevier Ltd, OXFORD, vol. 16, No. 5, pp. 291-301.

[18] Al-Mayyahi A, Wang W, Birch P, (2014). Adaptive Neuro-Fuzzy Technique for Autonomous Ground Vehicle Navigation. Robotics, vol. 3, No. 4, pp. 349-370.

[19] Mohammad ,E, Naraghi M, (2011). Robust Adaptive Stabilization of Skid Steer Wheeled Mobile Robots Considering Slipping Effects. Advanced Robotics, vol.25, No.1/2, pp.205-227.

[20] Mostafa Salama, Vladimir V. Vantsevich, (2014). A Parallel Control of Four Independently Driven Wheels to Maintain UGV Inverse Dynamics. Proceedings of the ASME International Mechanical Engineering Congress and Exposition IMECE 2014, ASME, Montreal, Quebec, vol. 12

[21] Tu C, Qi G, van Wyk BJ, Du S, (2009). Motion control and stabilization of a Skid-Steering Mobile Robot. 2nd International Conference on Adaptive Science & Technology (ICAST 2009), IEEE. pp.325-330

[22] Jiménez F, Naranjo JE, Gómez Ó, (2015). Autonomous collision avoidance system based on accurate knowledge of the vehicle surroundings. IET Intelligent Transport Systems, vol. 9, No. 1, pp. 105-117.

[23] Arslan S, Temeltas H, (2011). Robust motion control of a four wheel drive skid-steered mobile robot. In 7th International Conference on Electrical and Electronics Engineering (ELECO 2011), IEEE.pp.415-419.

[24] Sahoo S, Subramanian SC, Mahale N, Srivastava S, (2015). Design and development of a heading angle controller for an unmanned ground vehicle. International Journal of Automotive Technology. vol.16, pp.27-37.

[25] T. Yasuno, D. Tanaka and A. Kuwahara, (2014). Autonomous navigation system based on collision danger-degree for unmanned ground vehicle. International Power Electronics Conference (IPEC-Hiroshima 2014 - ECCE ASIA), Hiroshima, Japan. pp. 3179-3184.

[26] Ullah, I., Ullah, Q., Ullah, F. and Shin, S., (2012). Mobile robot navigation with distance control. In Robotics and Artificial Intelligence (ICRAI 2012), Rawalpindi, International Conference on IEEE, pp. 61-67.

[27] L. H. Prayudhi, A. Widyotriatmo and K. S. Hong, (2015). Wall following control algorithm for a car-like wheeled-mobile robot with differential-wheels drive. Control, Automation and Systems (ICCAS 2015), Busan, 15th International Conference on, pp. 779-783.

[28] Thomaz, C.E., Pacheco, M.A.C. and Vellasco, M.M.B., (1999) Mobile Robot Path Planning Using Genetic Algorithms In International Work-Conference on Artificial Neural Networks Springer, Berlin, Heidelberg, vol. 1606, pp. 671-679.

[29] A. Chakravarthy and D. Ghose, (1998). Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 28, No. 5, pp. 562-574.

[30] Mashadi B, Majidi M. (2014). Global optimal path planning of an autonomous vehicle for overtaking a moving obstacle. Latin American Journal of Solids and Structures. vol. 11, no. 14, pp.:2555-2572

[31] Benamati L, Cosma C, Fiorini P, (2005). Path planning using flat potential field approach. In Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on, pp.103-108.

[32] Luh G-, Liu W, (2007). Motion planning for mobile robots in dynamic environments using a potential field immune network. Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering. pp.1033-1046.

[33] Conkur ES, (2005). Path planning using potential fields for highly redundant manipulators. Robotics and Autonomous Systems,vol.52, pp.209-228.

[34] Ge, S.S. and Cui, Y.J., (2002). Dynamic Motion Planning for Mobile Robots Using Potential Field Method. Autonomous Robots, , Kluwer Academic Publishers, Boston, vol. 13, No. 3, pp. 207-222.

[35] Panagiotis G. Zavlangas & Spyros G. Tzafestas, (2003) Motion control for mobile robot obstacle avoidance and navigation: a fuzzy logic-based approach. Systems Analysis Modelling Simulation, vol.43, no12, pp. 1625-1637,

[36] Jaradat, M.A.K., Garibeh, M.H. and Feilat, E.A., (2012) "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field.". Soft Computing, Springer-, vol. 16, no. 1, pp. 153-164.

[37] Tarokh M, (2008). Hybrid intelligent path planning for articulated rovers in rough terrain. Fuzzy Sets and Systems. Elsevier B.V, Amsterdam, vol. 159, no. 21, pp. 2927–2937

[38] Sezer V, Ercan Z, Heceoglu H, Bogosyan S, Gokasan M, (2012). A new fuzzy speed planning method for safe navigation. IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012), pp.381-386.

[39] Lu Yin and Yixin Yin, (2008). An improved potential field method for mobile robot path planning in dynamic environments. Intelligent Control and Automation, WCICA . 7th World Congress on, Chongqing, pp. 4847-4852.

[40] Brown. C, H. Durrant-Whyte, J. Leonard, B. Rao, and B. Steer, (1992) “Distributed data fusion using Kalman filtering: a robotics application,” A robotics application. In Data Fusion in Robotics and Machine Intelligence, Academic Press, Inc., San Diego, CA,, pp. 267–309.

[41] Okello, N., & Ristic, B., (2003). Maximum likelihood registration for multiple dissimilar sensors. IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 3, pp. 1074-1083.

[42] Chen, S., Guo, Q., Leung, H., & Bossé, E., (2011). A maximum likelihood approach to joint image registration and fusion. IEEE Transactions on Image Processing, vol.20, no. 5, pp. 1363-1372.

[43] Kok, M., Schon, T. B., (2016). Linköpings universitet, Tekniska högskolan, Reglerteknik, & Institutionen for systemteknik,. Magnetometer calibration using inertial sensors. IEEE Sensors Journal, vol.16, no. 14,pp. 5679-5689.

[44] Kalman, R.E., (1960). A new approach to linear filtering and prediction problems. Journal of basic Engineering, vol. 82, no. 1, pp. 35–45.

[45] Luo, R.C. and Kay, M.G., (1992). Data fusion and sensor integration: State-of-the-art 1990s. Data Fusion in Robotics and Machine Intelligence, Academic Press, Inc., San Diego, CA,, pp.7-135.

[46] Welch and G. Bishop, (2001). An Introduction to the Kalman Filter, ACM SIC-CRAPH, Course Notes, 2001.

[47] Caron, F., Duflos, E., Pomorski, D., & Vanheeghe, P., (2006). GPS/IMU data fusion using multisensor kalman filtering: Introduction of contextual aspects. Information Fusion, vol.7, no. 2, pp. 221-230.

[48] Li, X., Xu, Q., Chan, C., Li, B., Chen, W., & Song, X., (2015). A hybrid intelligent multisensor positioning methodology for reliable vehicle navigation. Mathematical Problems in Engineering, vol.2015, pp. 1-13.

[49] Ryu, J. H., Gankhuyag, G., & Chong, K. T., (2016). Navigation system heading and position accuracy improvement through GPS and INS data fusion. Journal of Sensors, vol. 2016,pp. 1-6.

[50] Xian, Z., Hu, X., & Lian, J., (2015). Fusing stereo camera and low-cost inertial measurement unit for autonomous navigation in a tightly-coupled approach. Journal of Navigation, vol. 68, no. 3, pp. 434-452.

[51] Wang, J., Garratt, M., Lambert, A., Wang, J.J., Han, S., Sinclair, D., (2008). Integration of GPS/INS/vision sensors to navigate unmanned aerial vehicles. ISPRS – Int. Arch. Photogramm. Remote Sens. Spatial Inform, vol.37, pp. 963-970.

[52] D. Crisan and A. Doucet, (2002). “A survey of convergence results on particle filtering methods for practitioners,” IEEE Transactions on Signal Processing, vol. 50, no. 3, pp. 736–746.

[53] Kurashiki, K., Fukao, T., Ishiyama, K., Kamiya, T. and Murakami, N., (2010). Orchard traveling UGV using particle filter based localization and inverse optimal control. In System Integration (SII), December, 2010 IEEE/SICE International Symposium on pp. 31-36.

[54] Rigatos, G. G., (2013). Sensor fusion-based dynamic positioning of ships using extended Kalman and particle filtering. Robotica, vol.31, no.3,pp. 389-403.

[55] Hiremath, S., van der Heijden, G., van Evert, F., Stein, A., & ter Braak, C., (2014). Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. Computers and Electronics in Agriculture, vol.100, pp. 41-50.

[56] Mon, Y.J., (2013). Vision-Based Obstacle Avoidance Controller Design for Mobile Robot by Using Single Camera. International Journal of Computer Science Issues (IJCSI), vol.10, no.2 Part 2, p.248.

[57] Crassidis, J.L., Junkins, J.L., (2011). Optimal Estimation of Dynamic Systems, 2nd edn. CRC Press, Boca Raton.

[58] P. Nourizadeh, M. Ayati and A. Yousefi-Koma, (2015). "System identification and model validation of nonholonomic wheeled mobile robots,". 3rd RSI International Conference on Robotics and Mechatronics (ICROM 2015), Tehran, pp. 586-592.

[59] Aras, M., Shahrieel, M., Rashid, A., Zamzuri, M., & Azhan, A., (2013). Development and modeling of unmanned underwater remotely operated vehicle using system identification for depth control. Journal of Theoretical and Applied Information Technology, vol.56, no.1, pp.136-145.

[60] Hashimoto, S., Ogawa, T., Adachi, S., Tan, A., & Miyamori, G., (2000). System identification experiments on a large-scale unmanned helicopter for autonomous flight. In Control Applications. Proceedings of the 2000 IEEE International Conference on pp. 850-855.

[61] G. C. Karras et al., (2013), "On-line identification of autonomous underwater vehicles through global derivative-free optimization,". IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 3859-3864.

[62] Y. H. Eng, K. M. Teo, M. Chitre and K. M. Ng, (2016), "Online System Identification of an Autonomous Underwater Vehicle Via In-Field Experiments,". In IEEE Journal of Oceanic Engineering, vol. 41, no. 1, pp. 5-17.

[63] Samal, M. K., Anavatti, S., & Garratt, M., (2008). Neural network based system identification for autonomous flight of an eagle helicopter. IFAC Proceedings Volumes, vol.41, no.2, pp.7421-7426.

[64] Jian'an, X., Mingjun, Z., & Jian, Z., (2005). Kinematic model identification of autonomous mobile robot using dynamical recurrent neural networks. In Mechatronics and Automation, 2005 IEEE International Conference, vol. 3, pp. 1447-1450.

[65] Fisher, A., (2014). Inside Google's Quest to Popularize Self-Driving Cars. Popular Science. Retrieved from http://www.popsci.com/cars/article/2013-09/google-self-driving-car

[66] Aaron M. Kessler, (2015). "Elon Musk Says Self-Driving Tesla Cars Will Be in the U.S. by Summer," The New York Times, March 19, 2015, https://www.nytimes.com/2015/03/20/business/elon-musk-says-self-driving-tesla-cars-will-be-in-the-us-by-summer.html?_r=0

[67] Viale, M, T. Tsubochi and S. Yuta, (1997). "A practical path and motion planner for a tractor-trailer robot,". Intelligent Robots and Systems. IROS '97, Proceedings of the 1997 IEEE/RSJ International Conference on, Grenoble, vol.2, pp. 989-996.

[68] Beyersdorfer, S., & Wagner, S., (2013). Novel model based path planning for multi-axle steered heavy load vehicles. Paper presented at the 424-429.

[69] L. E. Kavraki, P. Svestka, J. -C. Latombe, and M. H. Overmars, (1996) ."Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 566–580.

[70] LaValle, S., and Kuffner, J., (2001). Rapidly-exploring Random Trees: Progress and prospects. Algorithmic and Computational Robotics: New Directions pp.293–308.

[71] Y. Hwang and N. Ahuja, (1992). "A potential field approach to path planning,", IEEE Transactions on Robotics and Automation, vol. 8, no. 1, pp. 23– 32.

[72] J.R. Andrews and N. Hogan, (1983). Impedence control as a framework for implementing obstacle avoidance in a manipulator. In Control of

Manufacturing Processes and Robot Systems presented at the ASME Winter Annual Meeting, Boston, pp. 243- 251,.

[73] Khatib, O., (1986). Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research, vol.5, no.1, pp.90-98.

[74] Berry, A., Howitt, J., Postlethwaite, I. and Gu, D.W., (2010). Enabling the operation of multiple micro-air-vehicles in increasingly complex obstacle-rich environments. Atlanta, Georgia. Reston,American Institute of Aeronautics and Astronautics Inc. In AIAA Infotech@ Aerospace 2010, p. 3340

[75] Chen, Y.-B., LUO, G.-C., MEI, Y.-S., YU, J.-Q. & SU, X.-L., (2016). UAV path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science, vol.47, no.6, pp. 1407-1420.

[76] Adeli.H, M.H.N. Tabrizi, Alborz Mazloomian, Ehsan Hajipour, Mehran Jahed., (2011). Path Planning for Mobile Robots using Iterative Artificial Potential Field Method. IJCSI International Journal of Computer Science Issues, Vol. 8, no.4, pp 28-32.

[77] Ross, I.M., (2009). A Primer on Pontryagin's Principle in Optimal Control; Collegiate Publishers: San Francisco, CA, USA, 2009.

[78] Stryker,https://en.wikipedia.org/wiki/Stryker, viewed 10 February 2016.

[79] Ren, J. and McIsaac, K.A., (2003). A hybrid-systems approach to potential field navigation for a multi-robot team. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on (Vol. 3, pp. 3875-3880). IEEE.

[80] Zhang, Y., Li, S. and Guo, H., (2017). A type of biased consensus-based distributed neural network for path planning. Nonlinear Dynamics, vol.89, no.3, pp.1803-1815.

[81] Kodgule, S.A., Das, A. and Selvakumar, A.D.A., (2017). A Comparative Study Between Hopfield Neural Network and A* Path Planning Algorithms for Mobile Robot. In Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, Singapore, pp. 33-48.

[82] C. Luo, J. Zhang, G. E. Jan, and H. Mo, (2016). "A new hopfield-type neural network approach to multi-goal vehicle navigation in unknown environments," in IEEE International Conference on Vehicular Electronics and Safety (ICVES 2016), pp. 1–6.

[83] Ni, J., Wu, L., Shi, P. and Yang, S.X., (2017). A Dynamic Bioinspired Neural Network Based Real-Time Path Planning Method for Autonomous Underwater Vehicles. Computational intelligence and neuroscience, vol.2017.

[84] Rehder, E., Quehl, J. and Stiller, C., (2017). Driving Like a Human: Imitation Learning for Path Planning using Convolutional Neural Networks. In International Conference on Robotics and Automation Workshops.

[85] Nazemi, A., Omidi, F., (2013) An efficient dynamic model for solving the shortest path problem. Transportation Research Part C: Emerging Technologies, vol.26, pp.1-19.

[86] Zhang, Y., Wu, L., Wei, G. and Wang, S., (2011). A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network. Digital Signal Processing, vol.21, no.4, pp.517-521.

[87] Sang, Y., Lv, J., Qu, H., Yi, Z, (2016). Shortest path computation using pulse-coupled neural networks with restricted autowave. Knowledge-Based Systems, vol.114, pp.1-11.

[88] Li, X., Ma, Y., Feng, X., (2013). Self-adaptive autowave pulsecoupled neural network for shortest-path problem. Neurocomputing, vol.115, pp. 63–71.

[89] Simon Haykin, (1994). Neural Networks: A Comprehensive Foundation, Prentice Hall, PTR, 2nd edtion.

[90] Haykin, S.S., Haykin, S.S., Haykin, S.S. and Haykin, S.S., (2009). Neural networks and learning machines, Prentice Hall. Upper Saddle River, NJ, USA: Pearson.

[91] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas, (2007). "Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering vol.160, pp. 3-24.

[92] Kaur T., (2012) Implementation of Back Propagation algorithm: A Neural Network approach for pattern recognition. International Journal of Engineering Research and Development.; vol.1, no.5, pp.30–7.

[93] Onieva, E., Naranjo, J. E., Milanés, V., Alonso, J., García, R., & Pérez, J., (2011). Automatic lateral control for unmanned vehicles via genetic algorithms. Applied Soft Computing Journal, vol.11, no.1, pp. 1303-1309.

[94] Ali, M.Z., Awad, N.H., Suganthan, P.N., Shatnawi, A.M. and Reynolds, R.G., (2018). An improved class of real-coded Genetic Algorithms for numerical optimization. Neurocomputing, vol.275, pp.155-166.

[95] G. Rousseau, D. Sinoquet, and P. Rouchon, (2007) "Constrained optimization of energy management for a mild-hybrid vehicle," Oil-Gas Sci. Technol., IFP, vol. 62, no. 4, pp. 623–624.

[96] X. Wei, L. Guzzella, V. I. Utkin, and G. Rizzoni, (2007). "Model-Based fuel optimal control of hybrid electric vehicle using of hybrid electric vehicle using variable structure control systems," ASME J. Dyn. Syst., Meas., Control, vol. 129, no. 1, pp. 13–19.

[97] Serrao, L. and Rizzoni, G., (2008). Optimal control of power split for a hybrid electric refuse vehicle. In American Control Conference, Seattle,WA 2008 pp. 4498-4503.

[98] Serrao, L., Onori, S. and Rizzoni, G., (2009). ECMS as a realization of Pontryagin's minimum principle for HEV control. In American Control Conference, St. Louis, MO 2009. ACC'09, pp. 3964–3969.

[99] Bellman, R., (1962). Dynamic programming treatment of the travelling salesman problem. Journal of the ACM (JACM), vol.9, no.1, pp.61-63.

[100] Bertsekas, D.P., (1996). Dynamic programming and optimal control. Vol. 1. Belmont, Massachusetts: Athena Scientific.

[101] C. C. Lin, H. Peng, J. W. Grizzle, and J. Kang, (2003). "Power management strategy for a parallel hybrid electric truck," IEEE Trans. Control Syst. Technol., vol. 11, no. 6, pp. 839–849.

[102] Liu, J. and Peng, H., (2006). Control optimization for a power-split hybrid vehicle. In Proc American Control Conference 2006, Minneapolis, MN, pp. 466–471.

[103] Z. Han, Z. Yuan, T. Guangyu, C. Quanshi, and C. Yaobin, (2004). "Optimal energy management strategy for hybrid electric vehicles," SAE International, Warrendale, PA, 2004-01-0576, 2004.

[104] P. Elbert, S. Ebbesen and L. Guzzella, (2013). "Implementation of Dynamic Programming for 2-Dimensional Optimal Control Problems with Final State Constraints," in IEEE Transactions on Control Systems Technology, vol. 21, no. 3, pp. 924-931.

[105] Bellman, R., (1957). Dynamic Programming. Princeton: Princeton University Press.

[106] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, (1962), The Mathematical Theory of Optimal Processes. John Wiley & Sons, New York, 1962.

[107] H. P. Geering, (2007). Optimal Control with Engineering Applications. Springer-Verlag: Berlin Heidelberg, Germany, 2007

[108] Donald K., (1970). Optimal Control Theory: An Introduction. Mineola, NY: Dover Publications, Inc

[109] D. E. Kirk, (2004). Optimal Control Theory: An Introduction. Mineola, New York: Dover Publications, 2004.

[110] Kaya CY, Martínez JM., (2007). Euler Discretization and Inexact Restoration for Optimal Control. Journal of Optimization Theory and Applications.vol.134, no.2, pp.191-206.

[111] Z. Wu, M. Yao, H. Ma, and W. Jia, (2013). "Improving accuracy of the vehicle attitude estimation for low-cost INS/GPS integration aided by the GPS-measured course angle," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 553–564.

[112] Santos, M.C., Santana, L.V., Brandão, A.S., Sarcinelli-Filho, M. and Carelli, R., (2017). Indoor low-cost localization system for controlling aerial robots. Control Engineering Practice, vol.61, pp.93-111.

[113] Han, X., Kim, H.J., Jeon, C.W., Moon, H.C. and Kim, J.H., (2017). Development of a low-cost GPS/INS integrated system for tractor automatic navigation. International Journal of Agricultural and Biological Engineering, vol.10, no.2, pp.123-131.

[114] Sukkarieh, S., Nebot, E.M. and Durrant-Whyte, H.F., (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. IEEE Transactions on Robotics and Automation, vol.15, no.3, pp.572-578.

[115] Adam Werries, and John M. Dolan, (2016). Adaptive Kalman filtering methods for low-cost gps/ins localization for autonomous vehicles, Robot. Commons.

[116] Liu, Y., Fan, X., Lv, C., Wu, J., Li, L. and Ding, D., (2018). An innovative information fusion method with adaptive Kalman filter for integrated INS/GPS

navigation of autonomous vehicles. Mechanical Systems and Signal Processing, vol.100, pp.605-616.

[117] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. de Vicuna, and M. Castilla, (2011). "Hierarchical control of droop-controlled AC and DC microgrids Ageneral approach toward standardization," IEEE Trans. Ind. Electron, vol. 58, no. 1, pp. 158–172.

[118] Moore, T. and Stouch, D., (2016). A generalized extended kalman filter implementation for the robot operating system. in Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13). vol.13, pp. 335-348.

[119] Tiano, A., Sutton, R., Lozowicki, A. and Naeem, W., (2007). Observer Kalman filter identification of an autonomous underwater vehicle. Control engineering practice, vol.15, no.6, pp.727-739.

[120] Scherzinger, B.M., (2000). Precise robust positioning with inertial/GPS RTK. In Proceedings of the 13th International Technical Meeting for the Satellite Division of the Institute of Navigation (ION GPS), Salt LakeCity, UT, USA, 2000, pp. 155–162.

[121] J. A. Farrell, T. D. Givargis, and M. J. Barth, (2000). "Real-time differential carrier phase GPS-aided INS," IEEE Transactions on Control Systems Technology, vol. 8, no. 4, pp. 709 721.

[122] Shalom BY, L., (2001). Kirubarajan T. Estimation with Applications to Tracking and navigation: Theory, Algorithm and software, New York, NY, USA: Wiley, 2001.

[123] B. Grocholsky, (2002). Information-theoretic control of multiple sensor platforms, PhD Thesis, PhD, U. of Sydney.

[124] Durrant-Whyte, H., Stevens, M. and Nettleton, E., (2001). Data fusion in decentralised sensing networks. In Proceedings of the 4th International Conference on Information Fusion, Montreal, Canada,2001, pp. 302-307.

[125] Åström, K.J. and Eykhoff, P., (1971). System identificationa survey. Automatica, vol.7, no.2, pp.123-162.

[126] An CH, Atkeson CG, Hollerbach JM, (1988). Model-based control of a robot manipulator. The MIT Press. Cambridge, MA.

[127] Ranó, I. and Iglesias, R., (2016). Application of systems identification to the implementation of motion camouflage in mobile robots. Autonomous Robots, vol.40, no.2, pp.229-244.

[128] Raptis, I.A., Valavanis, K.P. and Moreno, W.A., (2009). System identification and discrete nonlinear control of miniature helicopters using backstepping. Journal of intelligent & robotic systems, vol.55, no.2, pp.223-243.

[129] West, C., Montazeri, A., Monk, S.D., Duda, D. and Taylor, C.J., (2017). A new approach to improve the parameter estimation accuracy in robotic manipulators using a multi-objective output error identification technique. In Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on pp. 1406-1411.

[130] Li, M., Wu, H., Handroos, H., Wang, Y., Loving, A., Crofts, O., Coleman, M., Skilton, R., Burroughes, G., Keep, J. and Team, R.R.C., (2017). Dynamic model identification method of manipulators for inside DEMO engineering. Fusion Engineering and Design, vol.124, pp.638-644

[131] Park, K. and Choi, Y., (2016). System identification method for robotic manipulator based on dynamic momentum regressor. In Control and Automation (ICCA), 2016 12th IEEE International Conference on pp. 755-760.

[132] Huang, J.T., (2003). A new approach to parametric identification of a single-link flexible-joint manipulator. Journal of Intelligent & Robotic Systems, vol.37, no.3, pp.273-284.

[133] Robert Wilensky, (2016). An Extensive Review on Generator Excitation System Modeling, Design, and Parameter Identification.

[134] Ding, F., (2014). Combined state and least squares parameter estimation algorithms for dynamic systems. Applied Mathematical Modelling, vol.38, no.1, pp.403-412.

[135] Mousavian, S.H. and Koofigar, H.R., (2017). Identification-Based Robust Motion Control of an AUV: Optimized by Particle Swarm Optimization Algorithm. Journal of Intelligent & Robotic Systems, vol.85, no.2, pp.331-352.

[136] Middletone, R.H. and Goodwin, G.C., (1986). Adaptive computed torque control for rigid link manipulators. In Decision and Control, 1986 25th IEEE Conference, Athens, Greece on (Vol. 25, pp. 68-73).

[137] Ortega R, Spong MW, (1989). Adaptive motion control of rigid robots: a tutorial. Automatica; vol.25, no.6, pp.877-887.

[138] Moradi, M. and Malekizade, H., (2013). Neural network identification based multivariable feedback linearization robust control for a two-link manipulator. Journal of Intelligent & Robotic Systems, vol.72, no.2, pp.167-178.

[139] Noël, J.P. and Kerschen, G., (2017). Nonlinear system identification in structural dynamics: 10 more years of progress. Mechanical Systems and Signal Processing, vol.83, pp.2-35.

[140] Richter, H., Simon, D., Smith, W.A. and Samorezov, S., (2015). Dynamic modeling, parameter estimation and control of a leg prosthesis test robot. Applied Mathematical Modelling, vol.39, no.2, pp.559-573.

[141] Gautier M, Khalil W, (1992). Exciting trajectories for the identication of base inertial parameters of robots. International journal of robotics research, vol.11, no.4, pp.362-375.

[142] Bisheban, M. and Lee, T., (2017). Computational geometric system identification for the attitude dynamics on SO (3). International Journal of Control, Automation and Systems, vol.15, no.6, pp.2776-2785.

[143] Haverkamp, B.R.J., Chou, C.T., Verhaegen, M.H. and Johansson, R., (1996). Identification of continuous-time MIMO state space models from sampled data, in the presence of process and measurement noise. In Decision and Control, 1996., Proceedings of the 35th IEEE Conference on vol. 2, pp. 1539-1544.

[144] Eng, Y.H., Teo, K.M., Chitre, M. and Ng, K.M., (2016). Online system identification of an autonomous underwater vehicle via in-field experiments. IEEE Journal of Oceanic Engineering, vol.41, no.1, pp.5-17.

[145] Garg, A., Tai, K. and Panda, B.N., (2017). System Identification: Survey on Modeling Methods and Models. In Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, Singapore 2017, pp. 607-615.

[146] Lai, Y.C. and Le Tri, Q., (2017). System identification and control of a small unmanned helicopter at hover mode. In Control and Robotics Engineering (ICCRE 2017), 2nd International Conference on pp. 92-96.

[147] Hao Tan,Joseph Iacobellis, Alexander Levenko, Ryla Mutiger, Kelvin Ng, and Jay Averill., (2017). Design and Development of 8x8 Electric Combat Vehicle. Capstone Systems poroject final Engineering Report, University of Ontario Institute of Technology.

[148] Mohd Shahrieel Mohd Aras, Shahrum Shah Abdullah, Azhan Ab Rahman, Muhammad Azhar Abd Aziz, (2012). Thruster modeling for underwater vehicle using system identification method, International Journal on Robotics and Control System,vol 10 ,pp 1-12.

[149] Ljung, L. (1987). System Identification:Theory for the User. Englewood Cliffs, NJ: Prentice Hall.

[150] Chiddarwar, S.S. and Babu, N.R., (2012). Optimal trajectory planning for industrial robot along a specified path with payload constraint using trigonometric splines. International Journal of Automation and Control, vol.6, no,1, pp.39-65.

[151] Nguyen, A.D., Ngo, V.T., Ha, Q.P. and Dissanayake, G., (2008). Robotic formation: initialisation, trajectory planning and decentralised control. International Journal of Automation and Control, vol.2, no.1, pp.22-45.

[152] Al-Sagban, M., & Dhaouadi, R. (2016). Neural based autonomous navigation of wheeled mobile robots. Journal of Automation, Mobile Robotics and Intelligent Systems, vol.10, no.2, pp. 64-72.

[153] Amer, N.H., Zamzuri, H., Hudha, K. and Kadir, Z.A., (2017). Modelling and Control Strategies in Path Tracking Control for Autonomous Ground Vehicles: A Review of State of the Art and Challenges. Journal of Intelligent & Robotic Systems, vol.86, no.2, pp.225-254.

[154] Ji, J., Khajepour, A., Melek, W.W. and Huang, Y., (2017). Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints. IEEE Transactions on Vehicular Technology, vol.66, no.2, pp.952-964.

[155] Fedele, G., D'Alfonso, L., Chiaravalloti, F. and D'Aquila, G., (2018). Obstacles Avoidance Based on Switching Potential Functions. Journal of Intelligent & Robotic Systems, pp.1-19.

[156] Valls Miro, J., Taha, T., Wang, D. and Dissanayake, G., (2008). An adaptive manoeuvring strategy for mobile robots in cluttered dynamic environments. International Journal of Automation and Control, vol.2, no.2/3, pp.178-194.

[157] Gopinathan, L., Datta, K.B. and Ray, G., (2012). Design of a set of controllers for an n-link robot manipulator. International Journal of Automation and Control, vol.6, no.3/4, pp.277-290.

[158] Dong, Y., Zhang, Y. and Ai, J., (2017). Experimental test of unmanned ground vehicle delivering goods using RRT path planning algorithm. Unmanned Systems, vol.5, no.1, pp.45-57.

[159] Jaroszek, P. (2016). Any-angle global path planning for skid-steered mobile robots on heterogeneous terrain. Journal of Automation, Mobile Robotics and Intelligent Systems, vol.10, no.2,pp. 50-55.

[160] Park, M., Lee, S. and Han, W., (2015). Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm. ETRI Journal, vol.37, no.3, pp.617-625.

[161] Harasim, P., & Trojnacki, M., (2016). State of the art in predictive control of wheeled mobile robots. Journal of Automation, Mobile Robotics and Intelligent Systems, vol.10, no.1,pp. 34-42.

[162] Du , X. and Tan, K.K., (2016). Autonomous vehicle velocity and steering control through nonlinear model predictive control scheme. In Transportation Electrification Asia-Pacific (ITEC Asia-Pacific), 2016 IEEE Conference and Expo pp. 1-6.

[163] Cox, I.W., (2017). Steering control in an aircraft equipped with a wheel drive system. Borealis Technical Limited, U.S. Patent 9,751,621, issued September 5, 2017.

[164] Faddel, S., Mohamed, A.A. and Mohammed, O.A., (2017). Fuzzy logic-based autonomous controller for electric vehicles charging under different conditions in residential distribution systems. Electric Power Systems Research, vol.148, pp.48-58.

[165] Kuo, C.H., (2016). Trajectory and heading tracking of a mecanum wheeled robot using fuzzy logic control. In Instrumentation, Control and Automation (ICA 2016), International Conference on pp. 54-59.

[166] Cao, Z., Shen, F., Zhou, C., Gu, N., Nahavandi, S. and Xu, D., (2016). Heading control for a robotic dolphin based on a self-tuning fuzzy strategy. International Journal of Advanced Robotic Systems, vol.13, no.28, pp. 1-8.

[167] Jasour, A.M. and Farrokhi, M., (2010). Fuzzy improved adaptive neuro-NMPC for online path tracking and obstacle avoidance of redundant robotic manipulators. International Journal of Automation and Control, vol.4, no.2, pp.177-200.

[168] Pedrycz, W., (1995). Fuzzy sets engineering. CRC press.

[169] Martínez, R., Castillo, O., & Aguilar, L. T. (2009). Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Information Sciences, vol.179, no.13, pp. 2158-2174.

[170] J. Mina, Z. Flores, E. López, A. Pérez and J. H. Calleja, (2016). "Processor-in-the-loop and hardware-in-the-loop simulation of electric systems based in FPGA". 2016 13th International Conference on Power Electronics (CIEP), Guanajuato, pp. 172-177.

[171] Hu, M., Zeng, G., Yao, H. and Tang, Y., (2010). Processor-in-the-loop demonstration of coordination control algorithms for distributed spacecraft. In Information and Automation (ICIA 2010), IEEE International Conference on pp. 1008-1011.

[172] Juang, J.C., Chong, C.Y., Tsai, Y.F., Miau, J.J., Tsai, J.R. and Pan, H.P., (2011). Design, Implementation and Verification of Microsatellite Attitude Determination and Control Subsystem Based on Processor-in-the-loop. In The 3rd Nano-Satellite Symposium: conference proceedings, Kitakyushu, Japan,2011.

[173] Chen, C. H., Lin, H. C., Liu, Y. C. and Hsu, W. C., (2010), Local-loop based robot action control module using independent microprocessors. Computer Applications in Engineering Education,, vol.18, no.4, pp. 593–606.

[174] Seelaender G., (2009). Emulation and co-simulation of attitude control system for the PMM satellite and electro-hydraulic system for an aircraft using FPGAs. MSc. Thesis. Brazilian Institute for Space Researches.

[175] Ziegler, J.G. and Nichols, N.B., (1942). Optimum settings for automatic controllers. trans. American Society of Mechanical Engineers (ASME), vol.64, no.11, pp. 759-768.

[176] Åström, K.J. and Hägglund, T., (1988). Automatic tuning of PID controllers. Instrument Society of America (ISA). Chapter 52, *The Control Handbook*, IEEE/CRC Press, William S. Levine.

[177] Mohamed A, El-Gindy M, Ren J and Lang H, (2017). Optimal Collision-Free Path Planning for an Autonomous Multi-Wheeled Combat Vehicle. Proc ASME. 58158; Volume 3: 19th International Conference on Advanced Vehicle Technologies; 14th International Conference on Design Education.

[178] Mohamed A, Ren J, Lang H, El-Gindy M., (2017). Optimal collision free path planning for an autonomous articulated vehicle with two trailers. IEEE

International Conference on Industrial Technology (ICIT 2017), IEEE, pp.860-865.

[179] Ali, M.A., Kim, C., Kim, S., Khan, A.M., Iqbal, J., Khalil, M.Z., Lim, D. and Han, C., (2017). Lateral acceleration potential field function control for rollover safety of multi-wheel military vehicle with in-wheel-motors. International Journal of Control, Automation and Systems, vol.15, no.2, pp.837-847.

[180] Wu, Z. and Chen, C., (2017). On Optimizing Steering Performance of Multi-axle Vehicle Based on Driving Force Control. In MATEC Web of Conferences EDP Sciences, vol. 124, pp. 07005.

[181] Rusere, K., Dahlin, M., Mo, Y. and Guo, C., (2016). Autonomous Ground Vehicle. Department of Physics, Linfield College, McMinnville OR

[182] Shen, C., Guo, H., Liu, F. and Chen, H., (2017). MPC-based path tracking controller design for autonomous ground vehicles. In Control Conference (CCC 2017), 36th Chinese, pp. 9584-9589.

[183] Hwang, C.L., Yang, C.C. and Hung, J.Y., (2017). Path Tracking of an Automatic Ground Vehicle with Different Payloads by Hierarchical Improved Fuzzy Dynamic Sliding-Mode Control. IEEE Transactions on Fuzzy Systems, vol.26, no.2, pp.899-914.

[184] Wang, R., Hu, C., Yan, F. and Chadli, M., (2016). Composite nonlinear feedback control for path following of four-wheel independently actuated autonomous ground vehicles. IEEE Transactions on Intelligent Transportation Systems, vol.17, no.7, pp.2063-2074.

[185] Wang, R., Jing, H., Hu, C., Yan, F. and Chen, N., (2016). Robust H_infinity Path Following Control for Autonomous Ground Vehicles with Delay and Data Dropout. IEEE Transactions on Intelligent Transportation Systems, vol.17, no.7, pp.2042-2050.

[186] Kayacan, E., Ramon, H. and Saeys, W., (2016). Robust trajectory tracking error model-based predictive control for unmanned ground vehicles. IEEE/ASME Transactions on Mechatronics, vol.21, no.2, pp.806-814.

[187] Mohammed, I., Erol, B.A., Mohammed, I.H., Benavidez, P.J. and Jamshidi, M., (2017). Improved route optimization for autonomous ground vehicle navigation. In 12th System of Systems Engineering Conference (SOSE 2017), pp. 1-6.

[188] Goodin, C., Carrillo, J.T., McInnis, D.P., Cummins, C.L., Durst, P.J., Gates, B.Q. and Newell, B.S., (2017). Unmanned ground vehicle simulation with the Virtual Autonomous Navigation Environment. In Military Technologies (ICMT 2017), International Conference on pp. 160-165.

[189] Dadras, S., (2017). Path Tracking Using Fractional Order Extremum Seeking Controller for Autonomous Ground Vehicle. SAE Technical Paper No. 2017-01-0094.

[190] Eski, İ. and Kuş, Z.A., (2017. Control of unmanned agricultural vehicles using neural network-based control system. Neural Computing and Applications, pp.1-13.

[191] Verhaegen, A. and Żbikowski, R., (2017). Aeroservoelastic modelling and control of a slender anti-air missile for active damping of longitudinal bending vibrations. Aerospace Science and Technology, vol.66, pp.20-27.

[192] Kun, Y., Chen, X. and Li, C., (2017). Robust control for snake maneuver design of missile. In Seventh International Conference on Electronics and Information Engineering (Vol. 10322, p. 103223H). International Society for Optics and Photonics.

[193] Xu, B., Zhou, D., Liang, Z. and Zhou, G., (2017). Robust adaptive sliding sector control and control allocation of a missile with aerodynamic control surfaces and reaction jets. Proceedings of theInstitution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol.231, no.3, pp.397-406.

[194] Zames, G., (1981), Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. IEEE Transactions on automatic control, vol.26, no.2, pp.301-320.

[195] Doyle, J.C., Glover, K., Khargonekar, P.P. and Francis, B.A., (1989). State-space solutions to standard H/sub 2/and H/sub infinity/control problems. IEEE Transactions on Automatic control, vol.34, no.8, pp.831-847.

[196] Stefani, R.T., Savant, C.J., Shahian, B. and Hostetter, G.H., (1994). "Design of Feedback Control System", Third Edition, Saunders College Publishing.

[197] EL-Sheikh, G.A., (1994). "On H_∞ self-tunning control and its aerospace application" Ph.D. Thesis, Industrial control center, University of Strathclyde, U.K.