FLATNESS BASED COOPERATIVE CONTROL OF MULTIPLE MOBILE ROBOTIC SYSTEMS

LINTLE TSIU

Dissertation submitted in fulfilment of the requirements for the degree

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

at the

Central University of Technology, Free State

Supervisor: Prof ED Markus

Bloemfontein

2022

Declaration

I, Lintle Magdalina Tsiu (Student Number ________) hereby declare that this research project which has been submitted to the Central University of Technology, Free State for the degree MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING, is my own independent work, complies with the Code of Academic Integrity, as well as other relevant policies, procedures, rules and regulations of the Central University of Technology, Free State, and has not been submitted before by any person in fulfilment (or partial fulfilment) of the requirements for the attainment of any qualification.

Student Signature:

Date:18/06/22

Central University of Technology, Free State

Acknowledgements

My gratitude goes beyond words to my supervisor, Prof. Elisha Didam Markus for his invaluable supervision, advice, continuous support, and patience during this research study. Throughout my academic journey, I have been encouraged by his dedication, tremendous knowledge, and abundant experience. I would also like to thank the National Research Foundation and the Research Grants and Scholarship Committee of the Central University of Technology, Free State for funding this research. As part of my appreciation, I would like to acknowledge my family and friends who have always encouraged and supported me during my academic journey. Lastly, despite the COVID-19 pandemic, the Central University of Technology provided a conducive environment for conducting this research and for that I am deeply grateful.

Abstract

Cooperative multi-robotic systems can be more useful in numerous applications when compared to single-robotic systems. However, to ensure that cooperative systems execute tasks accurately, highly effective control architectures are vital. This study aims to improve the coordination control of a model based cooperative multiple mobile robotic system using differential flatness theory. To achieve this, a comprehensive analysis of literature on fundamentals of wheeled mobile robots and their cooperative systems was conducted. Then, mathematical modelling is performed for a cooperative robotic system composed of differential drive wheeled robots. First the kinematic model is presented to describe the motion of a robot without consideration of the forces causing it. Following that, the dynamic model is presented to describe the robot's motion in relation to forces applied to it. These models allowed for better design of a control system and paved way for the differential flatness characterisation of the mobile robotic system. Next, the robot differential flatness characterisation and analysis were performed, and the flatness properties were exploited to design a flatness-based control algorithm for motion planning and to generate trajectories and track them. Thereafter, the formation model was derived using a leader-follower formation. In this approach, three similar robots were modeled. Among the robots, one is selected as leader, and it is followed by two follower robots. Only the leader robot has access to information about the desired tracking path and all follower robots rely on it to coordinate their motion. Also, each follower must maintain a constant defined distance and orientation from the leader. Lastly, a flatness-based formation controller was developed. Tests were conducted to compare the flatness-based controller with the conventional PID controller. According to the results, Flatness-based controllers significantly reduce tracking errors of the cooperative system, whereas PID controllers have slightly higher tracking errors. This is because increasing the gains of the PID controller beyond a certain threshold causes it to get saturated, whereas the Flatness controller can be adjusted without any concern for saturation. The key findings of this study was that differential flatness allows for the whole system to be represented by a reduced number of variables and thus the computational cost is significantly reduced especially

when dealing with multiple robots that could otherwise entail solving large robotic model differential equations. This significantly simplifies the motion planning problem of the cooperative system. Also, a differential flatness characterisation of the robotic formation enabled the linearisation of the system to a stable linear equivalent system. Furthermore, in flat output space a simple polynomial-based trajectory planning can be used, that is simplifying the trajectory generation problem. Additionally, trajectories are solved without integrating robot model differential equations. Thus, it is concluded from this study that differential flatness theory improves coordination control of cooperative multiple mobile robotic systems.

Abbreviations

4IR	Fourth Industrial Revolution
ADRC	Active Disturbance Rejection Control
AI	Artificial Intelligence
AUV	Autonomous underwater vehicle
COMRADE	Cooperative Multi-Robot Automated Detection
FOPID	Fractional Order PID controllers
PID	Proportional–Integral–Derivative
RZNN	Robust Zeroing Neural Network
SAC-PID	Soft Actor-Critic PID
SMC	Sliding Mode Control
WMR	Wheeled Mobile Robots

Symbols

q	Generalised coordinates
$(x\ y)$	Robot Position
θ	Orientation of the robot
θ_r and θ_l	Wheel velocities
b	the length between a driven wheel and the centre of the robot chassis
d	Distance between Cartesian coordinates (x, y) and the centre of mass (CM)
r	Wheel radius
τ_r and τ_l	the input left and right driving motor torques
CM	Robot the centre of mass
m_o	Robot mass about its CM
I_o	Total equivalent moment of inertia
$A(q)$	Motion constraints
v	Forward velocity
v_y	Lateral velocity
ω	Angular velocity
$S(q)$	Transformational matrix

Central University of Technology, Free State

Table of Contents

List of Figures

Central University of Technology, Free State

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background

As the world moves into the Fourth Industrial Revolution (4IR), robotic researchers are increasingly paying more attention to the concept of cooperative multiple robot formation [1-7]. This is because they have realised that using a group of robots could significantly increase efficiency, flexibility, redundancy, and maneuverability when compared to using single robots. As a result, cooperative multiple robot formations have a wide range of applications. For example, [2] developed a multi-robot system called Cooperative Multi-Robot Automated Detection (COMRADE) to detect and locate landmines in post-conflict areas to ensure human safety and life. Other applications include search and rescue assignments [3], security and surveillance [4] and surgical/medical applications [5] to name a few. To execute tasks accurately (as illustrated in Figure 1), however, an autonomous cooperative multi-robot system needs a proper coordination strategy [6, 7]. Thus, efforts were made by numerous researchers to develop highly effective control architectures that can be resilient to environmental uncertainties, malfunctions and disturbances [8].

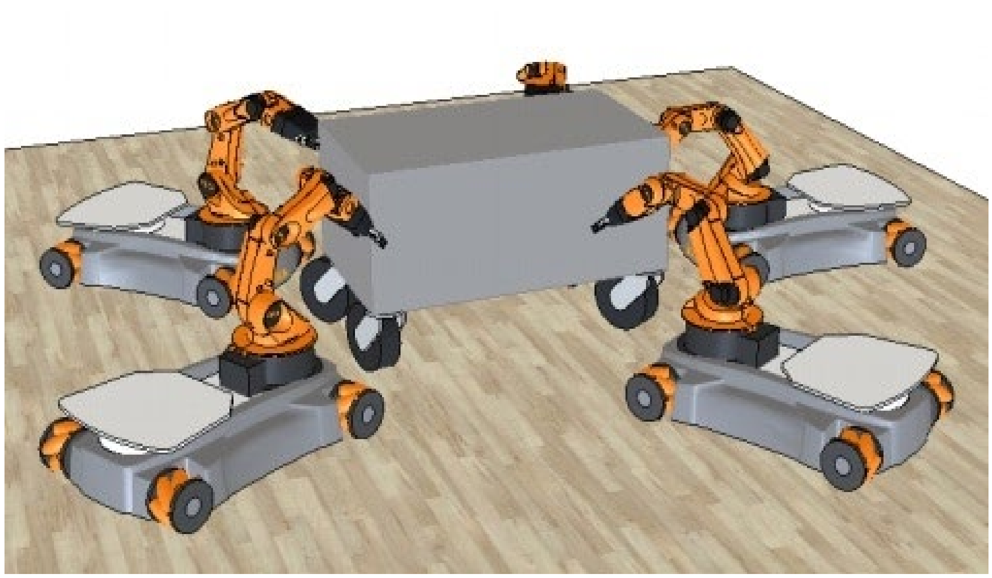


Figure 1: Cooperative Multiple Mobile Robotic System [9]

Typically, the control of cooperative multi-robot system is inundated with problems. Thus, control system designers have to overcome a variety of challenges which include high energy consumption, non-holonomic constraints, cross-coupling uncertainty and partial observability of the robot workspace, resolution of communication protocol and collision avoidance [6,8]. Consequently, researchers use various control techniques to solve these problems.

Furthermore, there exist quite a few control schemes that have been explored in the field of multi-robotic cooperation control, namely: synchronization control in which individual robot follows its desired path as well as maintain a synchronized formation with adjacent robots [10]. Another control scheme is the Cooperative control where the robots communicate and exchange information for task execution [11]. Also, in consensus control the robots communicate constantly to reach a common value [12]. Lastly, coordination control includes upholding a particular kinematic relationship between the robots [13].

Moreover, to successfully carry out the above control schemes, an appropriate motion planning in conjunction with an effective control architecture should be implemented for the wheeled mobile robotic system. Hence, it is fundamental to accurately capture the kinematic and dynamic properties of the robot. Several researchers have used nonlinear control approaches to control mobile robot. These include improved sliding mode variable structure [14], sliding mode control [15], fuzzy logic-based control [16], proportional–integral–derivative (PID) and fractional order PID (FOPID) controllers [17], and back-stepping control [18], among others.

However, most of the earlier studies estimated the dynamics of the system therefore failing to represent the nonlinearity of the system completely. As a result, modelling complex robotic scenarios with more complex coordination tasks becomes intricate. Also, for a multiple robotic system, an increasing number of robots in the system, means an increase in the computational control time, hence the less accurate the controller task implementation is. Thus, an alternative method is obligatory. Hence, this dissertation concentrates on the motion control of a team of wheeled mobile robots cooperating with each other to achieve a common goal.

1.2 Motivation

Recent advancements in robotics have resulted in systems that focus on flexibility, responsiveness, durability, and cost-effectiveness, and are therefore applicable in a wide range of applications. For instance, a cooperative multi-robotic system can be used for space exploration. In its mission called Mars 2020, NASA launched two cooperating mobile explorer robots, a rover (*Perseverance)* and an aerial robot (*Ingenuity*). While Perseverance searches for signs of ancient microbial life, Ingenuity, the small robotic coaxial helicopter scouts for locations of interest [19]. Also, cooperative multi robot system is useful in security and surveillance. For example, in October 2016 the Strategic Capabilities Office of the United States Department of Defence launched cooperative swarm of over a hundred autonomous micro-drones, called Perdix drones. These are to be used for unmanned aerial surveillance [20]. In addition, a cooperative multi robot system can be used for agricultural purposes. For instance, among the projects underway at Wyss Institute is the development of RoboBees, which are artificial bees that aid in plant pollination as well as accurate weather forecasting [21].

Likewise, it is imperative for South Africa to make use of such technologies as it has committed itself to the attainment of the 2030 Agenda for Sustainable Development Goals (SDG). One of the Sustainable Development goals, SDG 2, is dedicated to eradicating hunger and attaining food security by the year 2030 [22]. This necessitates improvement of the production of food. The use of a cooperative multi robot system can help in achieving this goal as they can be utilised in agriculture to plant and fertilise seeds at a speed beyond human abilities resulting in higher crop yields. Also, a cooperative multi-robotic system can also be utilised in the achievement of SDG 8, which aims to promote sustainable economic growth [22]. Cooperative multi robot system increases efficiency and reduces turnaround time in industrial processes. This will lower the cost of production while increasing productivity, thus boosting the economy.

Evidently, it is imperative that a good controller is designed to enable the member robots to generate dynamically feasible motion for accurate task execution. Hence, it is becoming increasingly popular to adopt new and more effective control strategies. One theory that has drastically trivialised dynamical systems control strategy is Differential Flatness. In Differential Flatness, a defined group of variables known as flat outputs, together with their time derivatives are used to parameterise the state variables and control inputs. Also, since there are as many flat outputs as there are control inputs, nonlinear differential equations are converted into systems of algebraic equations which are typically easier to solve. Differential flatness can be helpful in trajectory planning since it allows a trajectory to be planned using algebra with interpolations and a polynomial that is of proper order in flat output space. Dynamic equations automatically align the trajectories planned in the flat output space. The system can also be modeled as a series of integrators in flat output space for the development of controllers that are able to exponentially stabilise the system. These features are very appealing because in state space of non-holonomic systems, the generation of feasible trajectories is quite challenging. As a result, this study focuses on developing a controller that uses differential flatness to synchronise multiple wheeled mobile robotic systems.

1.3 Problem statement

The use of multiple robots is beneficial in numerous ways. To ensure the accurate performance of complex tasks, a high-performance formation control architecture is therefore needed. Ideally, formation control architecture should be scalable, robust, flexible, and able to switch topologies, avoid collisions at the group level, and be stable. Thus, for a cooperative control strategy to be effective, the team ought to have an ability to counteract unanticipated situations or environmental changes sensed while the cooperative task is being carried out. Also, an agreement should be reached on the appropriate action to be carried out with minimum computational time and energy consumption. In robotics, one critical drawback of model-based formation control is that the dynamics taken into account are sometimes approximations or are generally too simplified to adequately represent complex robotic situations. To improve coordination control of model-based multi-robotic Wheeled Mobile Robots (WMR), this research applies differential flatness theory. This theory allows nonlinear dynamical systems to be linearised globally, which is useful for solving complex nonlinear control problems.

The primary research problem of this study is addressed by introducing three sub-problems:

1.3.1 Sub-Problem 1

The first sub-problem is to conduct a differential flatness characterisation and analysis of the WMR so as to exploit its flatness properties and thus generate effective control architecture. To achieve this, a dynamic model of a differentially driven wheeled mobile robot is derived using the Euler-Lagrange technique. Then, a motion controller for the model is designed through the utilisation of its differential flatness properties.

1.3.2 Sub-Problem 2

The second sub-problem is to formulate feasible trajectories and design motion and trajectory tracking controllers to facilitate the cooperative robotic system to successfully complete a predetermine tasks. The use of flatness to the systems simplifies trajectory generation problem to simple algebra. Thus, it is entirely possible to define trajectory without solving complex differential equations. This is particularly appealing because it is challenging to originate feasible trajectories by non-holonomic robotic systems in the state space.

1.3.3 Sub-Problem 3

Finally, the last challenge is to develop a controller to successfully coordinate cooperative system for accurate task execution. A multi-robot system has more dimensions in state-space, thus, the computation required for planning and controlling them will continue to increase as the robots increase in number within the system. Hence, new competent coordination schemes are in demand to simplify coordination problems and reduce the computational cost.

1.4 Aim and objectives

This study aims to improve coordination control of a model based cooperative multiple mobile robotic system using differential flatness theory. In this study, the objectives are:

- To determine a kinematic and dynamic model of cooperative multiple mobile robotic system so as to facilitate accurate motion controller design.
- To perform a differential flatness characterisation and analysis on the robots involved in the cooperative control so as to take advantage of the flatness properties of the system and improve coordination control of model.
- To design flatness-based motion controller that will facilitate easy synchronised robot motion.
- To generate trajectories of motion for the robotic system and control them such that they track those trajectories

1.5 Research methodology

To achieve the above objectives, the following steps were taken:

Literature Survey

A comprehensive analysis of literature on fundamentals of wheeled mobile robotics was conducted. This was done to understand the mechanical behaviour (kinematical and dynamical characteristics) of the robot and its constraints, in order to design appropriate controllers to control it. The survey also includes path planning, motion planning and motion control. Furthermore, the cooperative formation behaviour of a group of these wheeled mobile robots was also investigated. Then, numerous different existing formation control architectures were studied and analysed.

System Modelling and Simulation

An analysis of the dynamic and kinematic behavior of a differential drive wheeled robot and the cooperative formation of these robots yielded mathematical models. Based on these models, a flatness analysis of the mobile robotic system could be performed, and a control system could be designed more efficiently. These models were further used to simulate the system on a simulation software program. As a result of the simulations, an understanding of the behaviour of the system was gained.

System Flatness Analysis

Using differential drive robot kinematic and dynamic models, flatness characters are investigated. Flat outputs are chosen and a one-to-one mapping between the flat output space and the state space is proven (referred to as diffeomorphism). That is, the state and input variables are written in terms of the flat outputs and a finite number of its successive derivatives. Conversely, the flat outputs are written in terms of the states and the inputs (and their time derivatives). Consequently, full-state controllability is now possible in flat output space, thus simplifying the trajectory planning and control challenge significantly.

Motion Planning and Control

In this study, an approach to generating desired trajectories that meet a specific terminal condition was presented, based on a simple polynomial model in flat output space. To accurately track these generated trajectories, a differential flatness-based controller was developed. Also, in the cooperative formation there is only one leader that has knowledge of the desired trajectories. There are also two follower robots that rely on this leader robot to coordinate their movements and maintain a constant desired relative distance and orientation to it. Thus, a flatness-controller was also designed to maintain this formation.

Simulation Tests and Results Analysis

MATLAB was used to implement a simulation to demonstrate the effectiveness of differential flatness-based controllers. A comparative study was made between the developed differential flatness-based controller and the existing PID controller. A comprehensive simulation analysis was conducted, and the results were discussed.

1.6 Hypotheses

Following the identification of the sub-problems described above, the following hypotheses appear to be feasible:

- A mathematical model of the robot systems kinematics and dynamics can provide an appropriate platform for flatness analysis to design an effective flatness-based controller for the cooperative robotic system.
- Wheeled mobile robots are flat systems [23]. In other words, there are a predetermined number of differentially independent variables are known as flat outputs, allowing all system variables to be completely parameterised. Therefore, in addition to the flat outputs and their time derivatives, the states can be determined by considering the inputs and their derivatives. Thus, there exists an observability advantage in flat systems.
- There are as many flat outputs as there are control inputs. The result is that it becomes possible to convert nonlinear differential equations into a set of algebraic equations that are generally easier to solve than nonlinear differential equations. Hence, for trajectory planning, differential flatness can be useful since the trajectories of states and inputs are deduced from the desired flat output trajectory without solving differential equations. In other words, it is possible to plan the

desired trajectory using interpolating functions in flat output space to ascertain terminal conditions. As a result, the computational cost is reduced and hence useful in simplifying motion control and synchronisation of a robot formation.

1.7 Delimitation

- A limited scope of research was established as follows: The internal and external disturbances of the individual robots will not be included in the study.
- Trajectory tracking will not include obstacle and collision avoidance or driving on uneven terrain.
- All member robots are assumed to have the same physical parameters.
- No prototype will be built in this research, only software simulation will be utilised to evaluate the effectiveness of the control technique.

1.8 Contribution to knowledge

The key contribution of this study was the application of differential flatness theory to improve the formation control of a team of wheeled mobile robots. The study is relevant as the world is advancing into the Fourth Industrial Revolution (4IR), during which the cooperation of mobile robots can be used in numerous applications. The application of differential flatness therefore simplifies the trajectory tracking of the member robots and thus they can easily maintain the desired formation. The simulation results attained from the flatness-based controller are analysed and a comparison is made to that of the classic PID controller. Thus, the flatness-based controller becomes a better alternative to the widely used PID controller.

1.9 Research output

The following papers were published as part of the study:

- Tsiu, L. and Markus, E.D., 2020. A Survey of Formation Control for Multiple Mobile Robotic Systems. *International Journal of Mechanical Engineering and Robotics Research*, *9*(11).
- Tsiu, L. and Markus, E.D., 2022. Multiple mobile robotic formation control based on differential flatness. Accepted for publication in Springer EAI Endorsed Transactions

1.10 Outline of the dissertation

A brief overview of the dissertation chapters is presented in this section.

Chapter 1 introduces the dissertation, by first providing a research background on wheeled mobile control and formation control. The problem statement, research objectives, research methodology, hypothesis, delimitation, and research outputs are also explained.

Chapter 2 presents a comprehensive literature review of recent wheeled mobile robot control techniques and cooperative multi-robotic system control architectures.

In Chapter 3, the mathematical model for the nonlinear wheeled mobile robot and the cooperative multi-robotic system is presented.

Chapter 4 introduces the concept of differential flatness. Thereafter, the flatness analysis of a differential drive robot is presented. Using differential flatness, trajectory generation is presented and finally, formation controllers are designed using PID and the flatness theory.

Chapter 5 presents the simulation test results on effectiveness of the formation controllers designed in the previous chapter. The two controllers are also compared with each other,

Chapter 6 concludes the research study. A summary of the initial objective and findings, contributions, conclusions, and future study suggestions are presented.

1.11 Summary

An overview of the research is provided in this chapter. Since using a team of cooperating robots has much higher flexibility and efficiency than a single robot, there has been increased interest in the concept of cooperative multiple-robotic formation control. Using differential flatness theory, this study aims to improve the formation control architecture of cooperative multiple mobile robotic systems. Also, the primary problems of the study is to conduct a differential flatness characterisation of the cooperative system, so as to simplify the trajectory generation problem and design a formation controller that effectively exploits the flatness properties of the cooperative system. After the research problems were stated, hypotheses were formulated. Also, a detailed description of the research method is provided in the chapter, together with the limitations of the study. Finally, a brief description of each chapter is presented.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Robotics and artificial intelligence (AI) are revolutionising business, society, and social life in recent years. Modern society is heavily reliant on robotics, and it is for this reason that numerous researchers are devoted to developing new, better control approaches for robots to increase their effectiveness. This chapter introduces several recent studies in the field of multi-robot systems coordination, as well as some other recent work that has been conducted in the field. The chapter first reviews the literature on the control of a wheeled mobile robot before considering the control of multiple wheeled mobile robots in coordination with each other.

2.2 Control of Wheeled Mobile Robots

Wheeled mobile robots (WMRs) are capable of moving about in their surroundings and are not physically tied to one place. They move from one place to another by using motorised wheels propelled by motors. The use of wheels typically reduces the amount of energy required for a robot to move and allows it to move more quickly. As a result, they are commonly used in mechanically simple and energy-efficient applications. It is for this reason that WMRs are becoming increasingly common in numerous sectors, especially when flexible motion is required. They find application in environments with or without obstacles, which includes exploration, transport, inspection, and surveillance.

Moreover, to perform predefined tasks accurately, a WMR should have a controller. Therefore, many researchers are involved in the challenging process of creating WMR controllers [24-43]. The design of these controllers is made challenging by the fact that WMRs, whether omnidirectional or nonholonomic, are inherently nonlinear. Thus, highly nonlinear control techniques had to be developed. Some frequently employed nonlinear control techniques include fuzzy logic, neural network, PID, sliding mode, flatness control, genetic algorithm, input/output linearisation, backstepping, receding horizon control and model predictive, among others. Also, these techniques may be combined to form hybrid systems, for example neuro-fuzzy, fuzzy-PID, etc. This section investigates some of this WMR control techniques that are used by numerous researchers.

Moreover, the fuzzy logic approach has been effectively used by numerous research studies to control the orientation and position of a WMR. The fundamental problems in mobile robotics that most researchers aim to solve is the issue of navigation and obstacle avoidance. To address this problem, Singh and Thongam [24] implemented an algorithm for navigating a WMR in a static environment using fuzzy logic. As a result, the robot avoids obstacles in static environment, and thus it is able to reach the target faster and with less bending energy. To improve the performance of fuzzy logic controller, Patle et al. in [25] created a duality technique by incorporating probability. As a result, the robot is able to choose a safe and optimal path in an environment containing moving obstacles and moving goals. Similarly, to further improve the fuzzy logic controller the genetic algorithm was incorporated in study [26]. This resulted in a significant energy consumption reduction.

Another mobile robot navigation technique that has gained popularity among researchers is the neural network controller. It is one of the important techniques for the mobile robot navigation because it allows the robot to autonomously explore unfamiliar and inhospitable environments. Singh and Thongam in [27] designed an artificial neural network-based controller for a robot to travel collision-free in an environment with moving and stationary obstacles. The robot autonomously generates near-optimal path and speed and successfully reaches its destination. To track its trajectory, a robot needs sensors. In study [28], the authors demonstrate how a robot with poor binary sensors can successfully track its trajectory by use of an online neural network controller. To improve neural network controller, [29] introduces the robust zeroing neural network (RZNN) model which makes it possible for the robot to track its trajectory within a fixed time while also cancelling noise.

Additionally, one of the oldest vastly used controllers is the proportional–integral–derivative (PID) controller. One of the benefits of PIDs is that they are simple and robust, and they are incredibly familiar to the control community. This has led to a great deal of research to determine the best parameters for PID for different models of processes. In [30] PID is used for agricultural purposes. It is used to control a smart irrigation mobile robot used to water crops and therefore reduce water wastage. The robot successfully tracks its trajectory despite any disturbances. In some applications the traditional PID control may be insufficient. It is for this reason that Wang and Chang in [31] used a hybrid Fuzzy PID controller to improve the task execution ability of a WMR. The result was a decrease in the overshoot and settling time which resulted in a better dynamic response. The fuzzy PID controller was further improved in [32] by introducing the self-adaptive model-free soft actor-critic (SAC)-PID control which increased robustness and real-time performance.

Sliding mode control (SMC) has garnered a lot of attention because of its fast response, simplicity, and robustness. The study in [33] improves the transient as well as the steady state responses while the robot tracks its desired trajectory in the presence of disturbances and uncertainties, by introducing a sliding mode controller. Further improvement of the sliding mode control was done by Moudoud et al. in [34]. In this study a fuzzy adaptive sliding mode controller was developed for trajectory tracking task of an electrically driven wheeled mobile robot. The incorporation of the fuzzy logic results in a smooth computing voltage and reduction of velocity and pose errors. The fuzzy adaptive sliding mode controller was also proposed in [35] for a WMR to track a desired trajectory and reach a target position. The combination of the fuzzy logic controller and sliding mode controller increase the response time of the WMR thus increasing its efficiency.

Another technique that is gaining popularity is centred on Differential flatness theory [36-43]. The theory was initially introduced in the early 90s by Fliess et al. in [37]. Differential flatness provides a new technique to design and implement an advanced control architecture for nonlinear systems [38]. In their book [23], Sira-Ramirez and Agrawal have shown WMR to be differentially flat, and thus substantial progress in trajectory generation and tracking has been made by different researchers in this field. Therefore, the authors in [39] use the flatness properties of WMR to design a robust motion controller which can withstand disturbances and uncertainties. The trajectories are defined in flat output space and the sliding manifold is imposed. This results in accurate and robust trajectories. Moreover, Khesrani et al. [40] combine the fuzzy logic and the differential flatness controller to increase controller efficiency. The nonholonomic robot is able to track its desired trajectories despite its constraints and uncertainties. The study in [41] aims to design a controller for a mobile robot and generate optimal trajectories. Differential flatness properties are exploited to solve an optimisation problem to accurately calculate the desire trajectories. A flatness-based controller is then designed to aid in trajectory tracking. In addition, Abadi et al. in [42] deigns a controller that is based on the differential flatness theory and the interval observer. This controller aids the robot in tracking the desired trajectories accurately despite disturbances, bounded uncertain slip and error in measurements. Similarly, to solve the trajectory tracking problem of WMR, the study in [43] used a flatness-based Active Disturbance Rejection Control (ADRC). Also, the linearised form of the nonlinear WMR system has been obtained using differential flatness theory. Variables perturbing the system are identified and compensated, resulting in a more robust controller.

Basically, differential flatness entails the parameterisation of the states and inputs by a finite set of independent variables, called the flat outputs, and their time derivatives. Additionally, since the number of flat outputs equals the number of control inputs, nonlinear differential equations can be converted to algebraic equations, which are often easier to solve. Hence, differential flatness can be useful for planning trajectory because differential flatness trivialises the trajectory planning task [23]. Also, implementation of exponential stabilising controllers is made possible since the system has the representation of a chain of integrators in the flat output space. Thus, differential flatness is important in WMR because it simplifies the generation of feasible trajectories.

2.3 Formation Control for Multiple Mobile Robotic Systems

Recently, numerous robotics researchers have been interested in multiple robot formations [44-65]. This popularity of multi-robotic systems can be attributed to the fact that besides improving time and quality efficiency of the assignment, they can also complete tasks that are not feasible for a single robot to accomplish or be flexible when executing tasks as well as being highly adaptive, low-cost, and easy to maintain. Therefore, multi-robotic systems are suited for many different applications, including exploration [44], mining support [45], agriculture [46], and so on. Additionally, since there are so many advantages to using multiple robots, many researchers are working on highly effective control architectures to accomplish increasingly complex tasks with greater accuracy [47-65].

A literature analysis was made on the different techniques used by researchers to control a group of cooperative mobile robots. Thus, Table 1 shows a summary of these techniques. A good formation control architecture should ideally have: Scalability, robustness, flexibility, topology switching ability, collision avoidance at group level and stability. As a result, for a cooperative control technique to work, the team must be able to deal with unplanned conditions or environmental changes that arise throughout the cooperative assignment. A consensus on the proper action to be taken with the least amount of computational time should be obtained. Additionally, the control of multirobot cooperative systems can be approached from a variety of angles, according to the literature, namely: cooperation, consensus synchronisation, and coordination control.

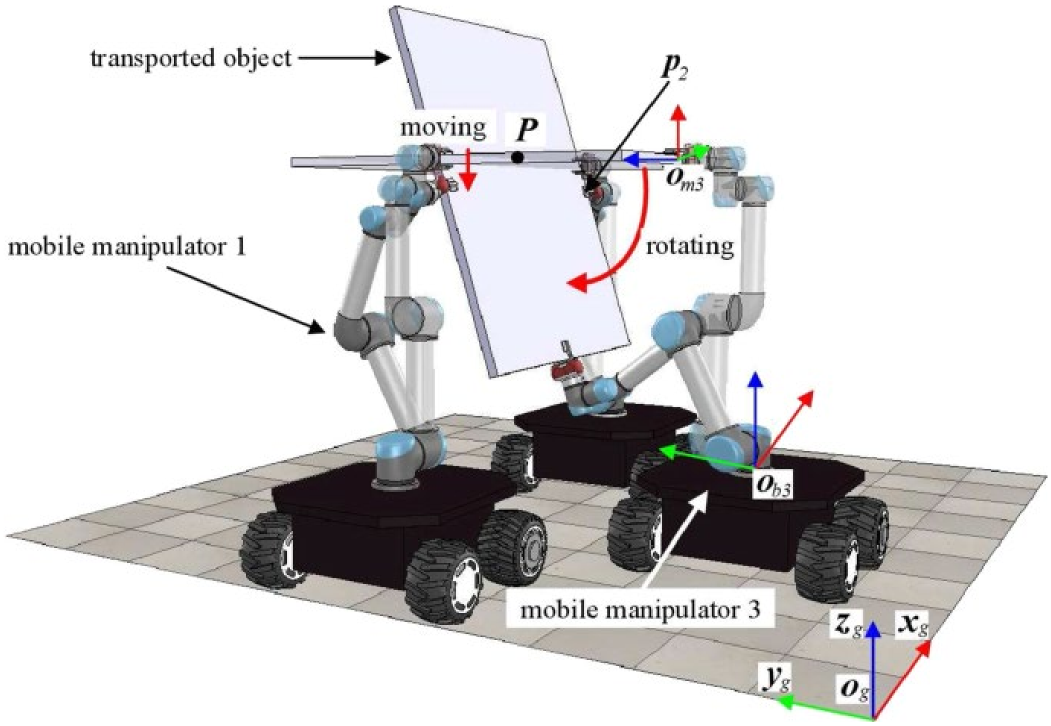


Figure 2: Wheeled mobile robot cooperative transportation [50]

Cooperative control entails sharing of information of member robot so as to achieve a common goal (Figure 2). These robotic systems can be used to perform numerous tasks namely: exploration, security surveillance, search and rescue operations, and mapping unfamiliar environments. For robots with limited sensing, processing, and communication capabilities, [47] used a probabilistic localisation and control method that considered the motion and sensing capabilities of the individual robots. Thus, providing the robots with an ability to adjust their sensing topologies depending on their limited sensing abilities and the target's motion. Additionally, method in [48,49] guaranteed a safe navigation of multi-robot formation in known cluttered environments, using minimal vision-based data, and minimising the number of times the cooperative robotic vehicles communicate.

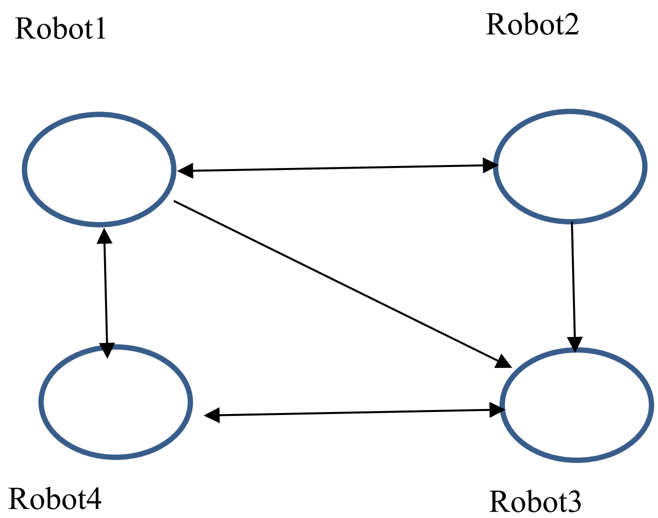


Figure 3: Local neighbour-to-neighbour information-exchange topology among four robots

Moreover, in consensus control the robots update their data so that they all come to the same conclusion (Figure 3). However, under conditions of numerous time delays, communication delays and noises, consensus might never be reached. [51,52] applied frequency domain analysis technique to convert the specific system equations into quadratic polynomials of pure imaginary eigenvalues and thus acquiring the maximum time delay's critical stability state during noise disturbances. Also, [53] used a distributed controllers designed by using a robot's individual information and its neighbours' information to converge to the similar value irrespective of whether there is a delay of communication or not. Robustness of the robotic system is equally as important, that is why some authors opted to use Lyapunov-based controller [54,55] and knowledge-oriented task and motion planning method [56] to enable the robotic system to create a feasible obstacle free path.

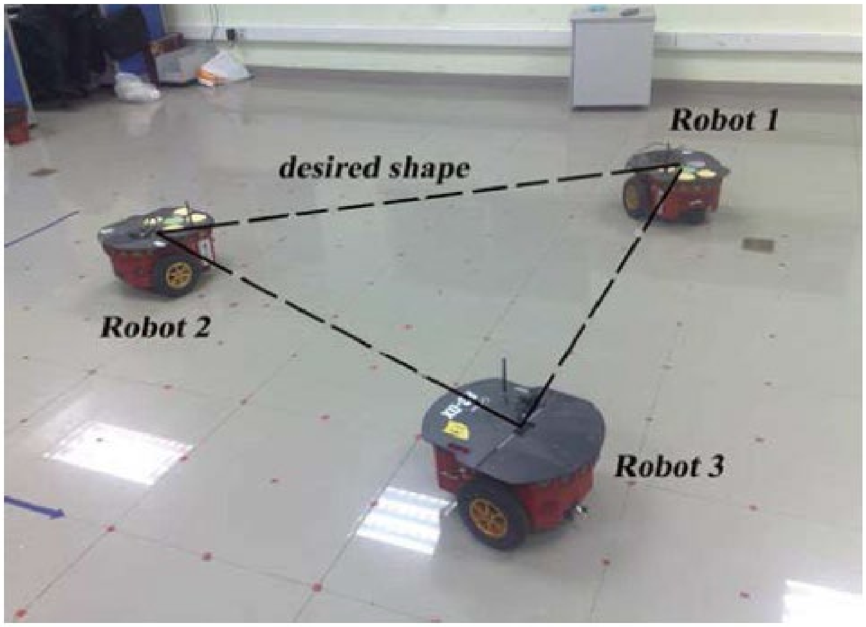


Figure 4: Three mobile robots synchronized in a triangular formation [59]

Additionally, some researchers opted to use synchronisation control (Figure 4). In synchronisation control each robot tracks its desired path while maintaining a synchronised formation with two neighbouring robots. Synchronous formation control is mostly decentralised. The author in [57] developed a nonlinear synchronisation controller that considers the nonholonomic constraint of the unicycle robots and allows directed and undirected information flow amongst the robots. The controller offered formation robustness during disturbances and accurate trajectory tracking. In addition, the study in [58] the author used Lyapunov theory and backstepping techniques to develop geometric path following, while introducing helmsman behaviour to each path following control. In addition, path parameters were synchronised by using a combination of tools from linear algebra, graph theory and nonlinear control theory. Synchronous formation control has a simpler control structure than most formation control methods. It also has high motion coordination performance and strong robustness. However, communication constraints including time-varying delays and data sampling renders this method ineffective. To lessen the formation error, the cross-coupling control can be incorporated.

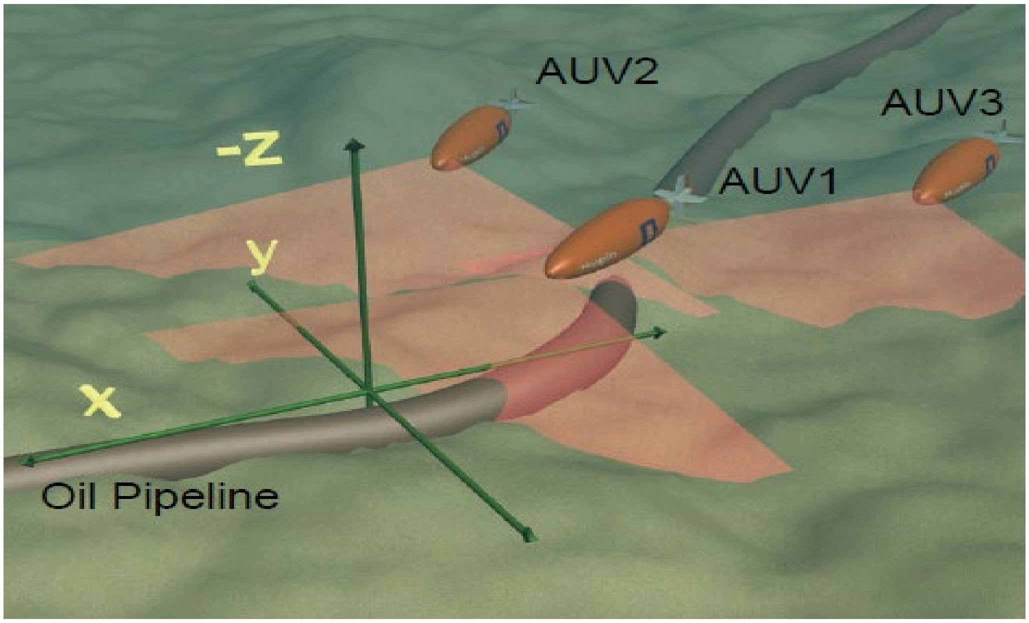


Figure 5: Coordinated multiple AUVs to examine an underwater oil pipeline [57]

Finally, coordination control involves maintaining certain kinematic relationship amongst the robots (Figure 5). Usually, a centralised control architecture approach is used. Other authors incorporated a leader-follower coordinated formation controller. That is, the leader robot has sensors and thus more information than the robots following it. The follower robots blindly execute the leader's motion commands. This approach increases accuracy and robustness but unfortunately costs the leader robot. To solve this problem, author [60] designed an integrated algorithm which merges the three degrees (the homodromous degree, district-difference degree, and the dispersion degree) into the potential field function of the surface-water environment. This approach resulted in no pre-learning procedure, good real-time and an increase in the coordination of the multi-AUV system thus reducing computational cost on the leader.

Table 1: Summary of multi-robotic formation control techniques [61]

Ref.	Type of Robot formation	Control Approach	Key objective	Strength of approach	Gap/future works
[62]	Inter-robot graph using Leader-follower	Lyapunov-based decentralized control	To solve formation maneuvering for unicycle robot	Robot formation can globally acquire and track whole tracks of desired trajectory	Collision avoidance strategy is not included in the proposed formation controller
[47]	Troop formation with sensing topology switching	Centralised cooperative control with probabilistic localisation and local optimisation	To develop a multi-robot cooperative control to determine the position of a target by use of on-board sensors	The approach provides the robots with an ability to adjust their sensing topologies depending on their limited sensing abilities and the target's motion	To use the approach for topology switching method that reserves scalability
[54]	Adaptive consensus-based formation	Lyapunov-based adaptive tracking controller with incorporation of the backstepping technique as well as the sliding mode approach	To develop a cooperative adaptive consensus tracking for multiple non-holonomic mobile robots	Robot formation can navigate desired trajectory thus effectively execute cooperative tasks.	Address cooperative robotic systems control challenges: robustness, resilience, scalability, and flexibility

[48]	Leader–Follower formation	Leader uses tractor–trailer system while Follower uses Vision-based sensing and localization of Leader and inter-vehicle collision avoidance	To develop motion synchronization and control approach for non-holonomic vehicles in a Leader–Follower formation with limited sight and communication	Offers safe routing of a multi-robotic formation in chaotic surroundings, using minimal sight and without sharing or approximating velocities online	Incorporating recovery modes to the designed hybrid system, that will get activated when a new leader needs to be selected and thus restore coordination
[49]	Variable geometric shape formations	Decentralised event-based cooperative controller	To develop an efficient formation control solutions for path tracking of a cooperative system by minimizing the number of times the cooperative robotic vehicles communicate	The rate of communication in the network is drastically decreased without affecting the stability and synchronisation of the system improving its performance	Investigates communication losses and delays so as to enable the testing of the vehicles underwater, using the acoustic communication channels
[53]	Leaderless consensus and leader-following consensus formation	Distributed consensus controller	To resolve the cooperative control challenges faced by distributed heterogenous multiple robotic system	Each robot is able to reach consensus despite communication delay by using its information and neighbors' information as feedback	Utilisation of visual servoing to resolve the consensus challenges
[63]	Hierarchical formation: commander, virtual	Cooperative control	To design a control strategy to enable multiple unmanned aerial	UAVs are able to achieve autonomous formation with a	N/A

	leaders and followers		vehicles (UAVs) to attain autonomous formation and reconfiguration	stable trend of positions and velocities	
[56]	Consensus-based formation	knowledge-oriented task and motion planning method called κ-TMP	To incorporate task planning into motion planning to enable mobile robot to drive in environments with moving obstacles among unfixed obstacles	Robot system is able to create a feasible obstacle free path of motions	Usage of a contingency-based task planner to handle uncertainty originating from the motion level to the task level
[52]	Consensus-based formation	Incorporated frequency domain analysis method in the consensus building process to convert the system's characteristic equations into quadratic polynomials of pure imaginary eigenvalues to solve them	To solve the consensus issues of second-order multi-robot systems that are under conditions of numerous time delays, communication delays and noises	The consensus of the second-order multi-robot system under delay and noise interference is attained	Application of the approach to higher-order systems consensus analysis that are under conditions of time delays and noises
[55]	Consensus control of a high-order chained structure	Lyapunov-based finite-time cooperative controller	To solve the finite-time consensus challenge for high order chained structured non-holonomic mobile robots	The multi-robotic system is able to achieve states consensus in finite time	N/A

[64]	Troop formation	Integrated coordination algorithm founded on the enhanced potential field (IPF) following the combination of three degrees (the homodromous degree, district-difference degree and the dispersion degree) into the potential field function of the surface-water environment	To design coordination mechanism and motion paths for cooperative target hunting Multiple Autonomous Underwater Vehicles (multi-AUV) which detect and surround an intelligent target (with unpredictable motion) in a surface-water environment	The approach results in no pre-learning procedure, good real-time and an increase in the coordination of the multi-AUV system as well as ability to conquer local minimum problem	Design techniques to help the AUVs withstand ocean surface-water currents while hunting Use the approach in the 3D surface-water environment
[65]	Time-varying virtual structure-based swarm formation	Lyapunov-based Synchronous controller	To design a coordination algorithm that maintains a predefined formation and trajectory tracking for cooperative multiple unicycle robots as well as to determine the most effective controller topology	Increased robustness of non-holonomic robots' formations during disturbances and accurate formation trajectory tracking	N/A
[58]	Behavior-based formation	Synchronised helmsman behavior-based control developed	Aimed to address the challenge of synchronization problem of	Robots are able to adjust their speed with very little	Incorporation of simultaneous paths following

		on Lyapunov theory and backstepping techniques	multiple underactuated homogenous autonomous AUVs	communication to the member robots	and obstacles avoidance
[57]	Leader–Follower based inter-vehicle geometric formation	Coordinated formation control	To resolve the challenge of coordinating movement of multiple autonomous underwater vehicles	The team of AUVs can track desired paths and thus form a desired inter-vehicle geometric formation	Incorporating proposed approach to a leaderless team to construct a desired formation shape in a decentralised coordination

2.4 Conclusion

A comprehensive literature review of different control techniques of wheeled mobile robots is covered in this chapter. The strengths and weaknesses of these techniques were identified. Studies of authors who solved some of the technique's weaknesses by introducing hybrid controllers were also identified. Furthermore, a survey of the current literature on formation control of multiple mobile robotic systems was conducted. Focus was also given to the differential flatness-based technique. This concept is seen to simplify the challenges associated with trajectory tracking. The system is represented by flat outputs which allows for the robot to be controlled without solving any differential equations.

CHAPTER 3: SYSTEM MODELING

3.1 Introduction

To better understand the system, mathematical models of the cooperative robotic system and individual member robots must be derived. Therefore, this chapter presents the kinematic and dynamic models of a two-wheeled robot that make up the robotic system. A kinematic model is a mathematical description of the motion of a robot without any consideration of the applied forces causing it. This model deals with the geometric relationships in the system. Conversely, the dynamics model defines the mathematical correlation between the applied forces/torques and the robot's resulting motion. These models will allow for more effective control system designs and paves way for the flatness characterisation of the mobile robotic system.

The mobile robot used in the cooperative system is a differential drive wheeled robot, that is, it is made up of two drive wheels attached on one axis, and each wheel is able to be independently driven forward or backward. Also, the robot is non-holonomic, which basically means it has constraints that cause a reduction in the local mobility of the robot. Finally, the model of the cooperative multi-robotic system will be derived and explored.

3.2 Modelling of a Wheeled Mobile Robot

Figure 6 represents a schematic diagram of a differentially driven wheeled mobile robot. The robot has two separately actuated wheels that have a radius r and are secured on a common horizontal axis of length $2b$. The robot is driven by the difference in velocity between the two driving wheels.

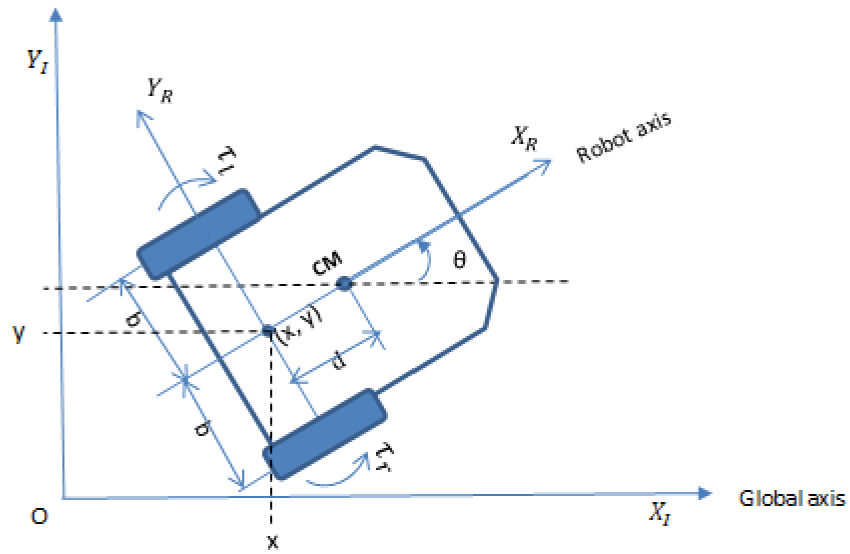


Figure 6: Differentially driven wheeled mobile robot (DDWMR)

Furthermore, there are three main situations that can occur with differentially driven wheeled mobile robots (Figure 7). Firstly, the robot will rotate about its vertical axis when the two wheels propelled at equal speed and in the same direction (contraclockwise or clockwise). Additionally, the robot will maintain a linear trajectory when the two wheels are driven at equal speed but in opposing direction. Furthermore, the robot is able to turn when the wheels drive at unequal speeds but in either the same or opposing direction. Lastly, the robot will turn 90° when one wheel rotates and the other is stationary. It is by this mechanism that the robot is able to change its direction. That is, individual wheel motions sum up to result in the overall movement of the robot and hence no additional steering motion is required.

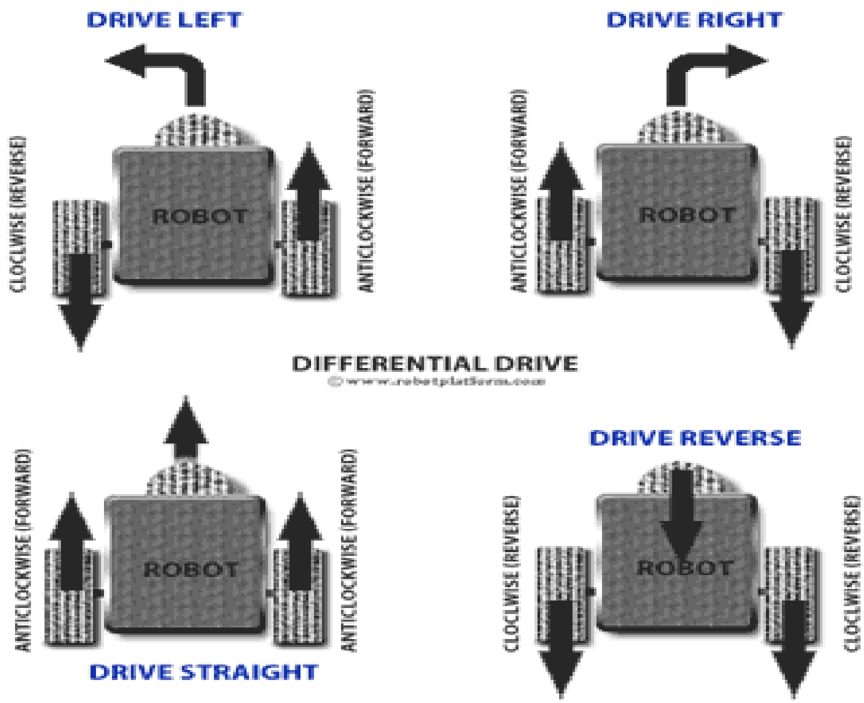


Figure 7: The driving mechanism of a Differentially driven wheeled mobile robot [66]

3.2.1 Kinematic Model of a Wheeled Mobile Robot

Tzafestas in [67] defines robot kinematics as a study involving the configuration of robots in their workspaces, the relationships between their geometric parameters, and the constraints imposed on their movements. The robot configuration is represented by its position and heading orientation $q = [x\ y\ \theta]^T$ is expressed in cartesian co-ordinate system of inertial frame. The symbols in Figure 6 are summarised in Table 2.

Table 2: The symbols and parameters of the DDMR

Symbols	Description
b	the length between a driven wheel and the centre of the robot chassis
d	Distance between cartesian coordinates (*x, y*) and the centre of mass (CM)
r	the wheel radius
$\boldsymbol{\tau_r}$ **and** $\boldsymbol{\tau_l}$	the input left and right driving motor torques that provide the robot motion
CM	the centre of mass

The separation distance between the cartesian coordinates (*x, y*) and the centre of mass (CM) is d. In the kinematic model, the system inputs are the wheel velocities θ_r and θ_l, where *r* and *l* represent quantities of right and left wheels respectively. All the robot parameters are assumed to be the same for all robots in the multirobotic system. Table 3 below summarises the physical parameters of mobiles as well as the design parameters.

Table 3: Physical parameters of the nonholonomic differential drive mobiles robot

Parameter Description	Symbol	Value	Unit
Robot mass about its CM	m_o	11	kg
Total equivalent moment of inertia	I_o	0.4136	kgm^2
Half of distance in between the two driving wheels	b	0.1	m
Distance between point of (x,y) and Centre of mass	d	0.04	m
Radius the of driven wheel	r	0.025	m

The robot is nonholonomic, thus its wheels impose constraints on its motion. This study assumes that there is no side sliding of the wheels (there is no velocity component for the contact point perpendicular to the plane where the wheels drive).

This is expressed as:

$$\dot{x}\sin\theta - \dot{y}\,cos\,\theta = 0 \tag{1}$$

Or it can be expressed as follows in matrix form:

$$A(q) \; = [sin\,\theta\,, -\cos\theta\,, 0] \tag{2}$$

Generally, the motion of the robot is defined by the equations:

$$\dot{x} = v\cos\theta - v_y\sin\theta \tag{3}$$

$$\dot{y} = v\,sin\,\theta + v_y\,cos\,\theta \tag{4}$$

$$\dot{\theta} = \omega \tag{5}$$

Where:

v is the forward(longitudinal) velocity

v_y is the linear (lateral) velocity

ω is the angular velocity

However, due to the wheel constrains shown in equation (1), v_y is zero and therefore the motion equations are simplified to:

$$\dot{x} = v\cos\theta \tag{6}$$

$$\dot{y} = v\,sin\,\theta \tag{7}$$

$$\dot{\theta} = \omega \tag{8}$$

These can be written in matrix form as:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S(q)\overline{v}(t) \qquad (9)$$

Equation (9) is the kinematic model formulation of a differential drive mobile robot. Where $q = [x, \quad y, \quad \theta]^T$ is the state vector and $\overline{v} = [v, \quad \omega]^T$ is the control input vector.

S(q) is the null space of the non-holonomic constraint A(q). That is:

$$A(q).S(q) = 0 \qquad (10)$$

Furthermore, the equivalent wheel inputs are then given by:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \frac{1}{r}\begin{bmatrix} 1 & b \\ 1 & -b \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix} \qquad (11)$$

3.2.2 Dynamic Model of a Wheeled Mobile Robot

To obtain the dynamic equations of the mobile robot, the Euler-Lagrange dynamic equation of motion for a non-holonomic differential drive mobile robot is used. However, viscous and friction forces will be ignored. The dynamic model is therefore given as:

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} = E(q)\tau - A^T(q)\lambda \qquad (12)$$

If the mobile robot has n degree-of-freedom (DOF) and it is subjected to p inputs and m constraints, then:

$q(t) \in \mathbb{R}^{n\times 1}$ is the generalised coordinates, $M(q) \in \mathbb{R}^{n\times n}$ is the symmetric positive-definite inertia matrix, $V(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the centripetal and Coriolis force matrix, $E(q) \in \mathbb{R}^{n\times p}$ is the actuation matrix, $\tau \in \mathbb{R}^{p\times 1}$ is the input vector, $A(q) \in \mathbb{R}^{m\times n}$ is the kinematic constraint matrix associated with the nonholonomic constraint equation, and $\lambda \in \mathbb{R}^{m\times 1}$ is the vector of constraint forces. The input driving motor torques τ_r and τ_l provide the robot motion hence the position of the robot (output) changes.

The dynamic equation of motion (12) can be written as [68]:

$$\begin{bmatrix} m_0 & 0 & -dm_0 \sin\theta \\ 0 & m_0 & dm_0 \cos\theta \\ -dm_0 \sin\theta & dm_0 \cos\theta & d^2 m_0 + I_0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 & -dm_0\dot{\theta}\cos\theta \\ 0 & 0 & -dm_0\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$
$$= \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ b & -b \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} - \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix}^T \lambda \tag{13}$$

Where m_0 and I_0 are respectively the mass and moment of inertia of the WMR about its centre of mass.

Equation (12) is then converted to unconstrained form. This is achieved by eliminating λ by multiplying both sides by $S(q)^T$ and taking $\overline{v}(t) = [v, \omega]^T$ as the minimal projected coordinate. Recall from equation (10) in the kinematic section, S(q) is the null space of the nonholonomic constraint $A(q)$. Thus equation (12) becomes:

$$S^T M(q)\ddot{q} + S^T V(q,\dot{q})\dot{q} = S^T E(q)\tau \tag{14}$$

$$\overline{\overline{M}}(q)\ddot{q} + \check{V}(q,\dot{q})\dot{q} = \overline{\overline{E}}(q)\tau \tag{15}$$

From the Forward kinematics of the robot

$$\dot{q} = S\overline{v} \tag{16}$$

$$\therefore \ddot{q} = S\dot{\overline{v}} + \dot{S}\overline{v} \tag{17}$$

Substituting (16) and (17) into (14)

$$S^T M(S\dot{\overline{v}} + \dot{S}\overline{v}) + S^T V(S\overline{v}) = S^T E\tau$$

$$S^T M S\dot{\overline{v}} + S^T(M\dot{S} + VS\overline{v}) = S^T E\tau$$

$$\bar{M}(q)\dot{\bar{v}} + \bar{V}(q,v) = \bar{E}(q)\tau \quad (18)$$

This can also be written as:

$$\begin{bmatrix} m_o & 0 \\ 0 & d^2 m_o + I_o \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} -dm_o\dot{\theta}^2 \\ dm_o v\dot{\theta} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 \\ b & -b \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (19)$$

The state space reduced model can be found by solving for $\dot{q}$ and $\dot{\bar{v}}$

$$\dot{q} = S(q)\bar{v}(t)$$
$$\dot{\bar{v}} = -\bar{M}^{-1}(q)\bar{V}(q,v) + \bar{M}^{-1}(q)\bar{E}(q)\tau \quad (20)$$

Equation (20) is a state-space representation of the robot and define its nonlinear control characteristics. Therefore, Equation (20) can be rewritten as:

$$\dot{x} = f(x) + g(x)u \quad (21)$$

By choosing the following state variables:

$x_1 = x$

$x_2 = y$

$x_3 = \theta$

$x_4 = v$

$x_5 = \omega \quad (22)$

Where $x = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T = [x \quad y \quad \theta \quad v \quad \omega]^T$ is the state vector and $u = [u_1 \quad u_2]^T = [\tau_r \quad \tau_l]^T$ is the control input vector.

The first order differentiation of equation (22) is given as:

$\dot{x}_1 = \dot{x} = v\cos\theta = x_4 \cos x_3$

$\dot{x}_2 = \dot{y} = v\sin\theta = x_4 \sin x_3$

$$\dot{x}_3 = \dot{\theta} = \omega = x_5$$

$$\dot{x}_4 = \dot{v} = d\dot{\theta}^2 - \frac{1}{Rm_0}(\tau_{r+}\tau_l) = d{x_5}^2 - \frac{1}{Rm_0}(u_1 \quad +u_2)$$

$$\dot{x}_5 = \dot{\omega} \ = \frac{1}{d^2m_o+I_o}\left[-dm_o v\dot{\theta} - \frac{b}{R}(\tau_r - \tau_l)\right] = \frac{1}{d^2m_o+I_o}\left[-dm_o x_4 x_5 - \frac{b}{R}(u_1 - u_2)\right] \quad (23)$$

Thus, the system can now be written in the form shown in equation (21). Finally, the dynamics of the differential drive robot in State space is:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cos x_3 & 0 \\ 0 & 0 & 0 & \sin x_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & dx_5 \\ 0 & 0 & 0 & \frac{-dm_o x_5}{d^2m_o+I_o} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \frac{1}{R} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{m_0} & -\frac{1}{m_0} \\ \frac{-b}{d^2m_o+I_o} & \frac{b}{d^2m_o+I_o} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (24)$$

3.3 Modelling the Cooperative System

The dynamics and the kinematic models of the robot that make up the cooperative system have been shown in the two previous sections. The cooperative robotic system is made up of robots which are similar (physical parameters and dynamics) to each other, thus the system is said to be homogenous. In this study, the leader-follower approach is used (Figure 8). Leader-follower approaches are designed such that one robot, designated the leader, follows a predetermined trajectory and the following robots maintain a constant specified orientation and distance away from the leader. This is known as the distance-orientation control. Ultimately, the major objective of the study is to create a controller to maintain cooperative system formation until the goal destination is reached. However, in this section the type of formation modeled is the leader-follower formation.

Let the leader and a follower robot be denoted as R_j and R_i, respectively. In this study, the subscript "j" and "i" denotes leader and follower, respectively. Thus, the states and the inputs of the leader and a follower are then represented as:

$$\dot{q}_j = \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{\theta}_j \end{bmatrix} = \begin{bmatrix} \cos\theta_j & 0 \\ \sin\theta_j & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_j \\ \omega_j \end{bmatrix} \tag{25}$$

$$\dot{q}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} \tag{26}$$

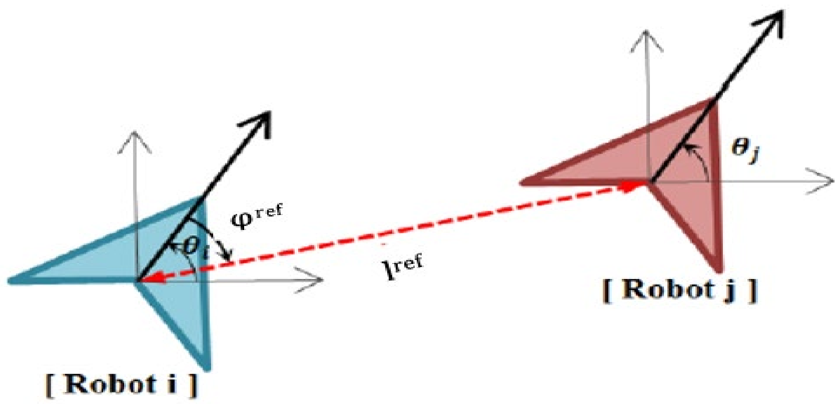


Figure 8: Leader-follower formation [69]

Moreover, in order for the robots to be in a formation, it is a requirement that followers keep a constant distance and separation bearing angle to the leader. The separation distance is then denoted as l^{ref} and the separation bearing angle is ϕ^{ref}. These are defined as:

$$l^{ref} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$\phi^{ref} = \pi - arctan2(y_i - y_{j,}x_i - x_j) - \theta_j \tag{27}$$

In order to maintain the desired l^{ref} and ϕ^{ref} a reference robot for the follower should exist (Figure 9) and it is given by [70,71]:

$$q_{ref} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ \theta_{ref} \end{bmatrix} = \begin{bmatrix} x_j - l^{ref} cos(\phi^{ref} + \theta_i) \\ y_j - l^{ref} sin(\phi^{ref} + \theta_i) \\ \theta_i \end{bmatrix} \quad (28)$$

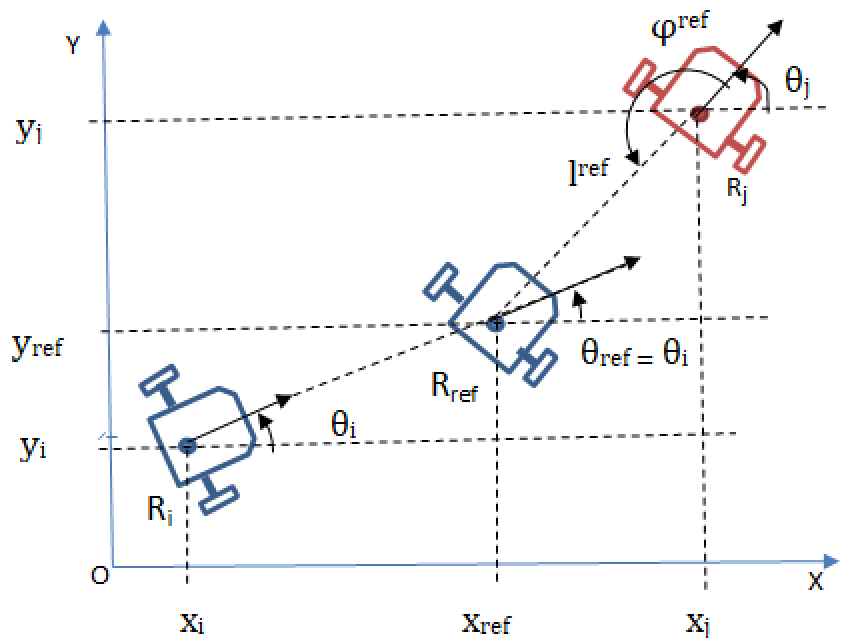


Figure 9: Leader- reference - follower robotic formation

It is necessary for the follower robot's velocity to match the reference velocity in order to maintain accurate separation distances and bearing angles, otherwise, formation will not be achieved. Calculated by differentiating equation 28, equation 29 represents the reference follower velocity and it is given by:

$$\dot{q}_{ref} = \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{\theta}_{ref} \end{bmatrix} = \begin{bmatrix} \dot{x}_j + l^{ref} sin\,(\phi^{ref} + \theta_i)\dot{\theta}_i \\ \dot{y}_j - l^{ref} cos(\phi^{ref} + \theta_i)\dot{\theta}_i \\ \dot{\theta}_i \end{bmatrix} \quad (29)$$

Finding the second derivative of equation 28 yield the reference acceleration. For the follower robot to successfully maintain formation, the reference acceleration should be maintained. The acceleration equation of the reference robot is given by:

$$\ddot{q}_{ref} = \begin{bmatrix} \ddot{x}_{ref} \\ \ddot{y}_{ref} \\ \ddot{\theta}_{ref} \end{bmatrix} = \begin{bmatrix} \ddot{x}_j + l^{ref}[\ddot{\theta}_i \sin(\phi^{ref} + \theta_i) + \dot{\theta}_i^{\,2} \cos(\phi^{ref} + \theta_i)] \\ \ddot{y}_j - \ l^{ref}[\ddot{\theta}_i \cos(\phi^{ref} + \theta_i) - \dot{\theta}_i^{\,2} \sin(\phi^{ref} + \theta_i)] \\ \ddot{\theta}_i \end{bmatrix} \quad (30)$$

3.4 Conclusion

The kinematic and dynamic models used in the cooperative system have been described in this chapter. The purpose of these models is to better understand the behaviour of the robot in the formation. This can then be used to design an appropriate controller and tune its performance. Most importantly, the models allow the simulation of robots and their flatness characterisation. The type of a mobile robot chosen for this research is the differentially driven wheeled mobile robot which are nonholonomic. Furthermore, the cooperative system was modelled, and the leader-follower approach was chosen. The leader-follower formations reduce the computational load of the system by only generating reference trajectories for the leader. Also, the system is simplified because the trajectories of followers are defined by the leader only and the control laws of the followers generate the system's stability. In the next chapter, a differential flatness analysis of the robots will be done, and controllers will then be designed.

CHAPTER 4: FLATNESS CHARACTERIZATION, TRAJECTORY GENERATION AND CONTROL DESIGN

4.1 Introduction

In this chapter, the differential flatness characterisation of the robots in the cooperative system is presented. Firstly, a brief overview of the concept of differential flatness theory is presented. Then, using the theory, the desired trajectories of the leader robot will be generated. To track these desired trajectories, a controller is needed. The purpose of the controller is to eliminate the error which is the difference between the anticipated and actual position of the robot, thus aid in the accurate tracking of the desired trajectories. Flatness theory will be used again to design a control to help the robot track its trajectories. Finally, a formation model will be designed, as well as a controller to help the team robots to stay in the predefined formation.

4.2 Flatness Characterisation of the Differentially Driven Mobile Robot

Differential flatness has become a significant concept in the field of control systems. In this section, a brief introduction to differential flatness theory is presented and the differential flatness characterisation of a differentially driven mobile robot will follow.

4.2.1 Brief Introduction to Differential Flatness Theory

The Differential Flatness theory states [38]:

For a non-linear system

$$\dot{x} = f(x, u) \tag{31}$$

Where x is the state vector in n dimensions ($x \epsilon R^n$), u is the control input vector in m dimensions ($u \epsilon R^m$) and m is less than or equal to n ($m \leq n$).

The system is differently flat provided there exists an output y of the same dimensions as the control input such that y is locally a function of x, u and a finite number of successive derivatives of the component of u. That is:

$$(y_1 \dots \dots \dots . y_m) = h(x, u, \dot{u}, \ddot{u} \dots \dots u^{(l)}) \quad (32)$$

Conversely, x and u should be able to be expressed as functions of y up to a finite number of its successive derivative, that is:

$$x = \alpha(y, \dot{y}, \ddot{y} \dots \dots y^{(h)}), u = \beta(y, \dot{y}, \ddot{y} \dots \dots y^{(h+1)}) \quad (33)$$

If x and u are then substituted in the non-linear system equation, it gets identically satisfied:

$$\dot{\alpha} = f(\alpha, \beta) \quad (34)$$

Thus, 'y' is a complete parametrisation of the trajectories of the system, and hence 'y' is what is called a flat output [38].

4.2.1 Flatness Analysis of a Differential Drive robot

The flatness analysis of the kinematic model of the Differential drive mobile robot is presented below. Equation (9) in chapter 3 clearly shows that the model has two control inputs in which $q = [x, \quad y, \quad \theta]^T$ is the state vector and $\overline{v} = [v, \quad \omega]^T$ is the control input vector. In their book [55], Sira-Ramirez and Agrawal stated that the number of flat outputs should be equal to the number of control inputs. Therefore, two flat outputs have been selected, x and y. Thus:

$$F = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{35}$$

When the flat outputs are differentiated with respect to time we get:

$$\dot{F} = \begin{bmatrix} \dot{F}_1 \\ \dot{F}_2 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{36}$$

To establish a diffeomorphism between the initial states and flat outputs as well as their derivatives, input prolongation is used [68,72]. This is done because in equation (36) the mapping between the robot's inputs to flat outputs is singular. Therefore, through a change of variables the system is transformed into an input-output linearised form. Thus, v is extended as a state.

$$\dot{x} = v\cos\theta$$

$$\dot{y} = v\,sin\,\theta$$

$$\dot{v} = \bar{u}$$

$$\dot{\theta} = \omega \tag{37}$$

Where $\bar{u}$ is the new input for the new extended system. Therefore, the new state vector becomes $[x, \quad y, \quad \theta, \; v]^T$ and the control input vector becomes $[\bar{u}, \quad \omega]^T$. To linearize the system a second derivative of the system is derived as:

$$\ddot{F} = \begin{bmatrix} \ddot{F}_1 \\ \ddot{F}_2 \end{bmatrix} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -v\,sin\,\theta \\ \sin\theta & v\,cos\,\theta \end{bmatrix} \begin{bmatrix} \bar{u} \\ \omega \end{bmatrix} \tag{38}$$

That is, $\ddot{F}$ is linear with respect to the new input $\bar{u}$ and ω if and only if v is not equal to zero. From the theory of differential flatness, it can be recalled that for a system to be considered flat, all the state variables ought to be written in terms of the flat outputs and their derivatives.

$$x = F_1$$

$$y = F_2$$

$$\theta = a\tan2\left(\frac{\dot{F}_2}{\dot{F}_1}\right)$$

$$v = \sqrt{\dot{F}_1{}^2 + \dot{F}_2{}^2}$$

$$\bar{u} = \dot{v} = \frac{\dot{F}_1\ddot{F}_1 + \dot{F}_2\ddot{F}_2}{\sqrt{\dot{F}_1{}^2 + \dot{F}_2{}^2}}$$

$$\omega = \dot{\theta} = \frac{\dot{F}_1\ddot{F}_2 - \dot{F}_2\ddot{F}_1}{\dot{F}_1{}^2 + \dot{F}_2{}^2} \qquad (39)$$

Furthermore, the flat outputs and their derivatives can completely be expressed in terms of all the state variable and their derivatives:

$$F_1 = x$$

$$F_2 = y$$

$$\dot{F}_1 = \dot{x} = v\cos\theta$$

$$\dot{F}_2 = \dot{y} = v\, sin\,\theta$$

$$\ddot{F}_1 = \bar{u}\cos\theta - v\omega\, sin\,\theta$$

$$\ddot{F}_2 = \bar{u}\, sin\,\theta - v\omega\cos\theta \qquad (40)$$

Equations (38) and (39) show a diffeomorphic relationship which proves that there exists a one-to-one mapping between the mobile robot's state space and the flat output space. This makes it possible to obtain full-state controllability in flat output space and thus trivialising motion planning and control problem as seen in this study.

4.3 Trajectory Generation

This section presents a polynomial-based trajectory planning that satisfies a specific set of terminal conditions at a time interval of $t \in [t_0 \quad T]$. The terminal conditions of the states are given as:

$$x(t_0), y(t_0), \theta(t_0), v(t_0), \bar{u}(t_0), \omega(t_0)$$

$$x(T), y(T), \theta(T), v(T), \bar{u}(T), \omega(T) \tag{41}$$

Also, the corresponding terminal conditions in flat output space are given as:

$$\begin{gathered} F_1(t_0), \dot{F}_1(t_0), \ddot{F}_1(t_0), F_2(t_0), \dot{F}_2(t_0), \ddot{F}_2(t_0) \\ F_1(T), \dot{F}_1(T), \ddot{F}_1(T), F_2(T), \dot{F}_2(T), \ddot{F}_2(T) \end{gathered} \tag{42}$$

With consideration of the terminal conditions, trajectories of the flat outputs can be constructed. Since there are six terminal conditions, a fifth-order polynomial is used[73]:

$$\begin{gathered} F_1(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \\ F_2(t) = b_5t^5 + b_4t^4 + b_3t^3 + b_2t^2 + b_1t + b_0 \end{gathered} \tag{43}$$

With consideration of the limits of the robot, the terminal conditions are set and used to determine the coefficients a_k and b_k where k = [1, … ,5].

Using fifth-degree polynomial together with terminal conditions shown in Table 4, the desired trajectories can then be derived. The destination should be reached in fifteen seconds, thus $t=15$ seconds. The initial time $t_o=0s$ and final time $T=15s$. The robot is driven from rest-to-rest positions.

Table 4: Desired Terminal Conditions

State	Initial condition	Final condition
x(m)	0.0	1.5
y(m)	0.0	0.5
θ(rad)	0.0	0.0
v(m/s)	0.0	0.0
ω(rad/s)	0.0	0.0
$\bar{u}$(m/s²)	0.0	0.0

$$F_1(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \qquad t \epsilon [0 \quad T]$$

$$\dot{F}_1(t) = 5a_5t^4 + 4a_4t^3 + 3a_3t^2 + 2a_2t + a_1$$

$$\ddot{F}_1(t) = 20a_5t^3 + 12a_4t^2 + 6a_3t + 2a_2 \tag{44}$$

For the initial time t = 0

$$F_1(0) = a_0$$

$$\dot{F}_1(0) = a_1$$

$$\ddot{F}_1(0) = 2a_2 \tag{45}$$

For the final time t = T

$$F_1(T) = a_5T^5 + a_4T^4 + a_3T^3 + a_2T^2 + a_1T + a_0$$

$$\dot{F}_1(T) = 5a_5T^4 + 4a_4T^3 + 3a_3T^2 + 2a_2T + a_1$$

$$\ddot{F}_1(T) = 20a_5T^3 + 12a_4T^2 + 6a_3T + 2a_2 \tag{46}$$

Using matrices, the coefficients can be obtained

$$\begin{bmatrix} F_1(0) \\ F_1(T) \\ \dot{F}_1(0) \\ \dot{F}_1(T) \\ \ddot{F}_1(0) \\ \ddot{F}_1(T) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \tag{47}$$

from (47) above, the corresponding leader's desired trajectories were calculated as:

$$F_1{}^d(t) = 0.00001185t^5 - 0.0004444t^4 + 0.004444t^3$$

$$F_2{}^d(t) = 0.000003951t^5 - 0.0001481t^4 + 0.001481t^3 \tag{48}$$

By differentiating these expressions, the velocity can be obtained, and differentiating them twice yields the acceleration. The trajectories were generated on MATLAB software and the results are shown in Figures (10-12). Also, appendix 1 shows the Simulink diagrams of the leader robot with a flatness controller.

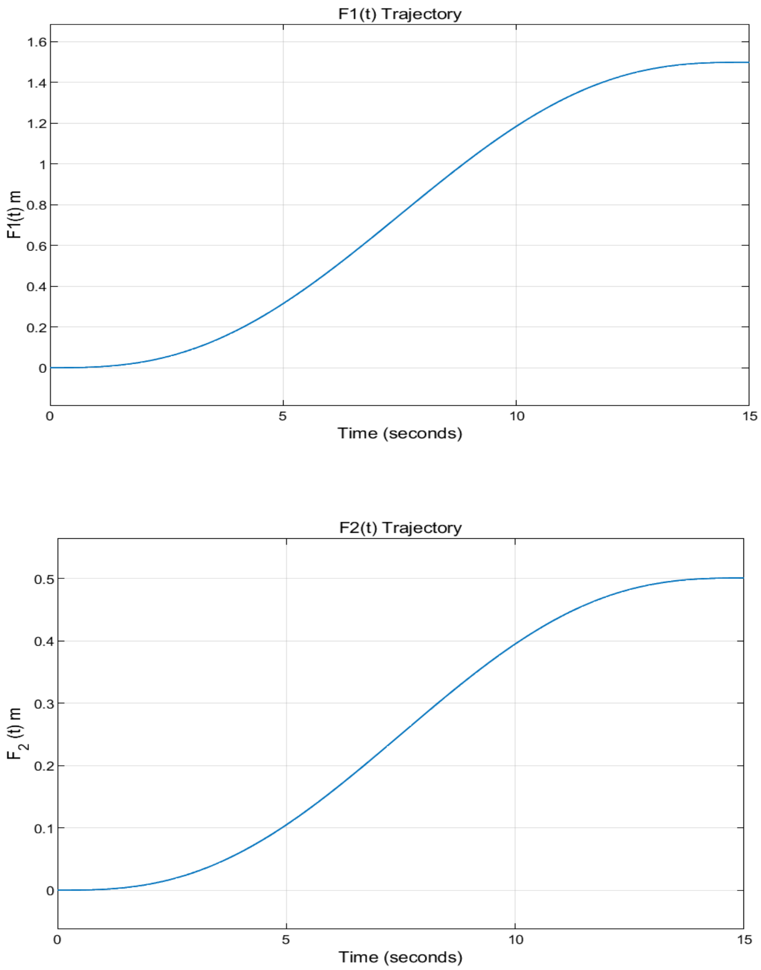


Figure 10: Flat Outputs Trajectories

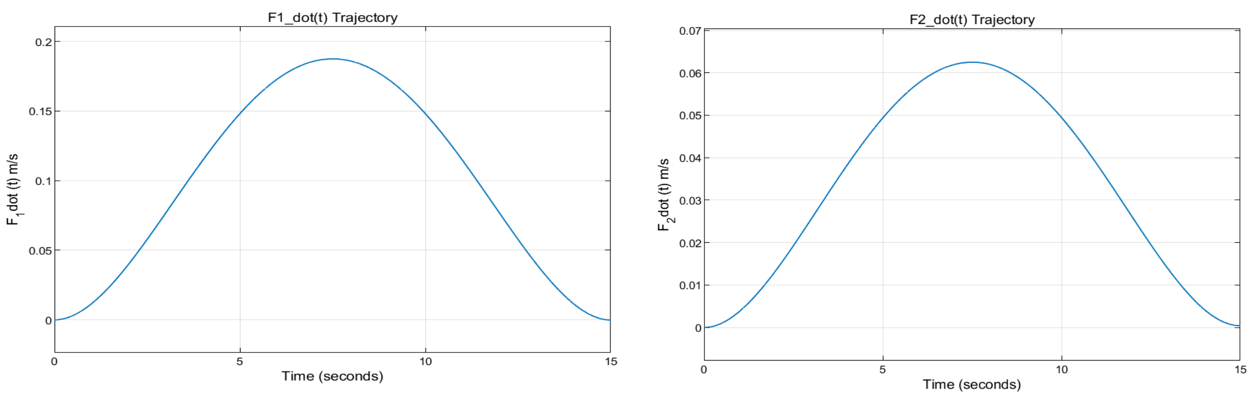
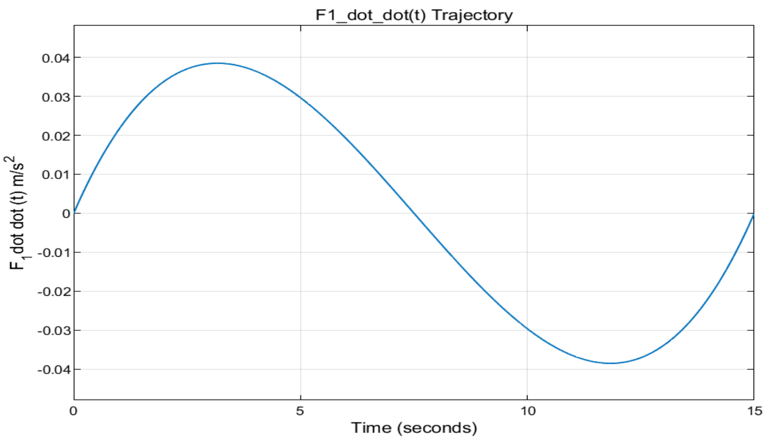


Figure 11: Velocity trajectories for the flat outputs



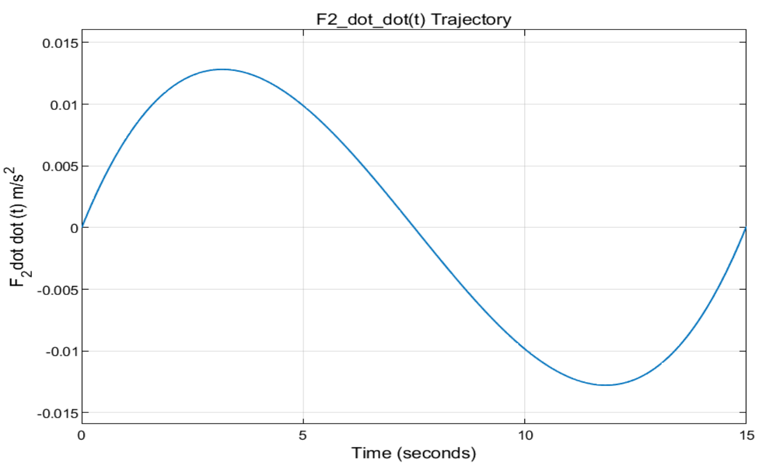


Figure 12: Accelerations trajectories for the flat outputs

4.4 Flatness-Based Controller Design

Having constructed the desired trajectories, the next step is the development of a flatness-based controller to facilitate an accurate tracking of the desired trajectories. Given that:

$F_1(t)$ and $F_2(t)$ are the actual flat output trajectories, and also $F_1^d(t)$ and $F_2^d(t)$ are the desired flat output trajectories, then error is defined as:

$$e_1 = F_1^d - F_1 \tag{49}$$

$$e_2 = F_2^d - F_2 \tag{50}$$

Let $\begin{bmatrix} \ddot{F}_1 \\ \ddot{F}_2 \end{bmatrix} = \begin{vmatrix} \delta_1 \\ \delta_2 \end{vmatrix}$ then the feedback control laws to these new inputs are presented as:

$$\delta_1 = \ddot{F}_1^d + \rho_1 \dot{e}_1 + \rho_0 e_1 \tag{51}$$

$$\delta_2 = \ddot{F}_2^d + \tilde{\rho}_1 \dot{e}_2 + \tilde{\rho}_0 e_2 \tag{52}$$

Where $\tilde{\rho}_1, \tilde{\rho}_0, \rho_1, \rho_0$ are the control gains. Next the error dynamics of the system in flat output space is determined.

$$\ddot{F}_1^d - \delta_1 = \ddot{F}_1^d - \ddot{F}_1 = \ddot{e}_1 \tag{53}$$

$$\ddot{F}_2^d - \delta_2 = \ddot{F}_2^d - \ddot{F}_2 = \ddot{e}_2 \tag{54}$$

Therefore, the error dynamics is defined as:

$$0 = \ddot{e}_1 + \rho_1 \dot{e}_1 + \rho_0 e_1 \tag{55}$$

$$0 = \ddot{e}_2 + \tilde{\rho}_1 \dot{e}_2 + \tilde{\rho}_0 e_2 \tag{56}$$

The gains of the flat controller are designed by trial and error. If the control laws gains are correctly chosen, that can ensure exponential stability of the system [74]. Therefore, the control gains were chosen to be:

$$\rho_{0x} = 2, \qquad \rho_{1x} = 1, \qquad \tilde{\rho}_{1y} = 6, \qquad \tilde{\rho}_{0y} = 9$$

Furthermore, the effectiveness of this controller was tested. The open-loop response, which is the response of WMR without a controller, was compared with the response of the robot with the flatness-based controller. Simulations were performed on MATLAB and the following results were obtained:

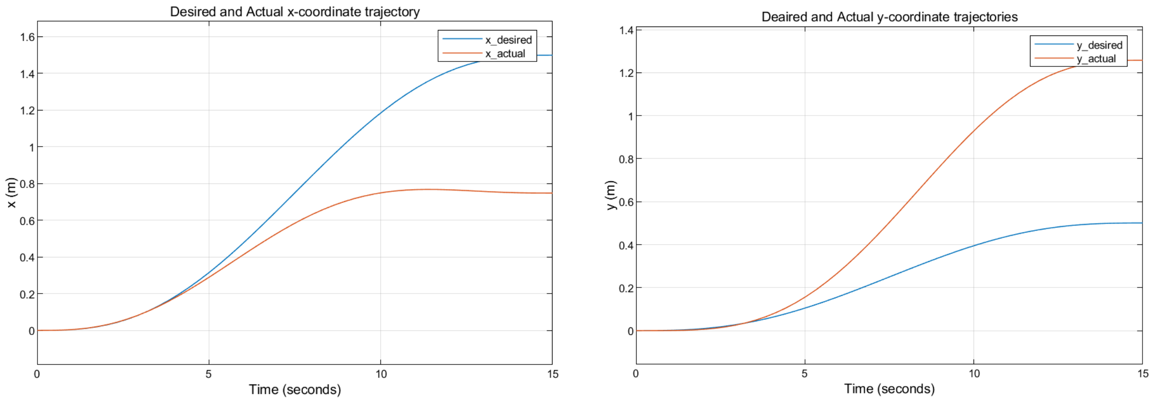


Figure 13: Flat output Trajectory tracking open-loop response

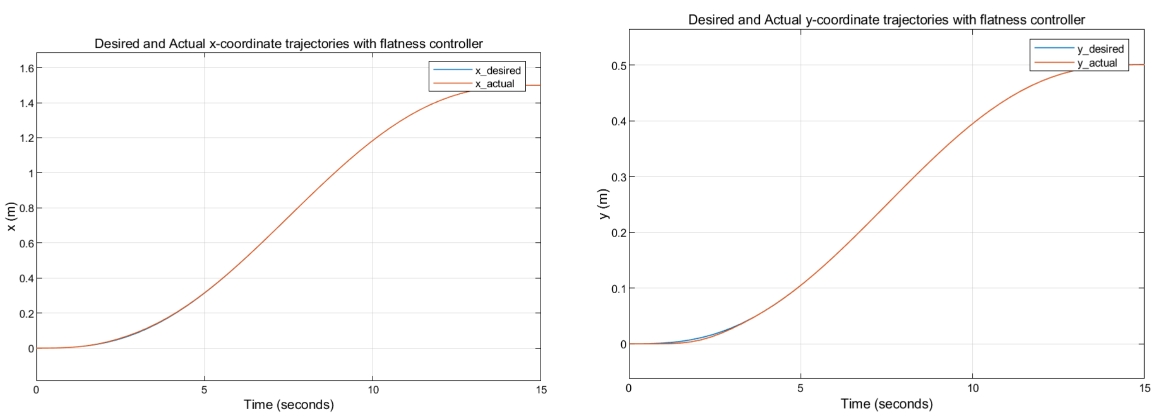


Figure 14: Flat output Trajectory tracking response with flatness-based controller

In Figure 13, although the desired final terminal condition of the x-coordinate is 1.5m, the actual value was found to be 0.78m. Also, instead of the desired 0.5m, the final y-coordinate value reached by the WMR was 1.25m. This is evident that without a controller, the robot fails to track the desired trajectories. However, in Figure 14, the flatness-based control performance is good, and the desired trajectories are reached. Thus, the robot can accurately track the desired trajectories.

4.5 Leader-Follower Formation Control

As mentioned in the previous sections, this study focuses on the leader-follower formation. This means that the purpose of the study is to design a controller that will aid the follower to follow the leader while keeping a specified distance and bearing (l^{ref}, φ^{ref}) from the leader. The formation is made up of three similar nonholonomic wheeled mobile differential drive robots. One of the robots is a leader and the other two are followers. The aim of a controller is to find values of translational and angular velocities such that the separation and bearing angle error is zero. The error of the follower robot is therefore obtained as:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} x_{ref} - x_i \\ y_{ref} - y_i \\ \theta_{ref} - \theta_i \end{bmatrix} \tag{57}$$

Also, the velocity error is given by:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \dot{x}_{ref} - \dot{x}_i \\ \dot{y}_{ref} - \dot{y}_i \\ \dot{\theta}_{ref} - \dot{\theta}_i \end{bmatrix} \tag{58}$$

And the tracking algorithm is defined as:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} cos\,\theta_i & sin\,\theta_i & 0 \\ -sin\,\theta_i & cos\,\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref} - x_i \\ y_{ref} - y_i \\ \theta_{ref} - \theta_i \end{bmatrix} \tag{59}$$

4.5.1 Flatness-based Formation Controller

With consideration of the terminal conditions in Table 5, desired trajectories of the flat outputs trajectories of the followers can be constructed. It is important to note that in the leader-follower approach, only the leader is involved in trajectory generation, the rest of the members just follow the leader. To find the reference trajectories of the followers, equation (28) is used in flat output space. (F_1^{f1}, F_2^{f1}) are the flat outputs for the first follower while (F_1^{f2}, F_2^{f2}) the second follower's flat outputs. For the first follower:

$$\begin{bmatrix} F_1^{f1}{}_{ref} \\ F_2^{f1}{}_{ref} \end{bmatrix} = \begin{bmatrix} F_1 - l^{ref1}cos(\phi^{ref1} + \theta_{i1}) \\ F_2 - l^{ref1}sin(\phi^{ref1} + \theta_{i1}) \end{bmatrix} \tag{60}$$

And for the second follower:

$$\begin{bmatrix} F_1^{f2}{}_{ref} \\ F_2^{f2}{}_{ref} \end{bmatrix} = \begin{bmatrix} F_1 - l^{ref2}cos(\phi^{ref2} + \theta_{i2}) \\ F_2 - l^{ref2}sin(\phi^{ref2} + \theta_{i2}) \end{bmatrix} \tag{61}$$

These desired Flat output trajectories are shown in figure 15 and will be used as reference for the controller.

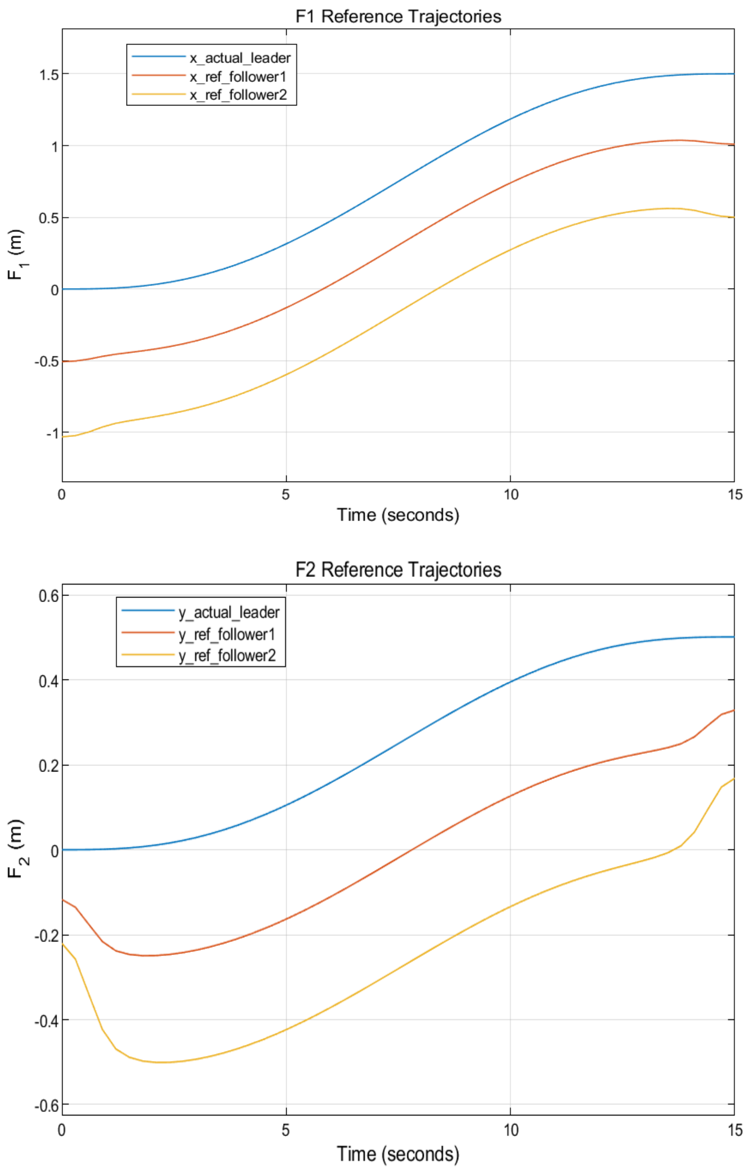


Figure 15: F_1 and F_2 XY Formation Reference Trajectories

Having constructed the desired trajectories, the next step is to define the error.

$$e_1{}^{f1} = F_1^{f1}{}_{ref} - F_1^{f1}$$

$$e_2{}^{f1} = F_2^{f1}{}_{ref} - F_2^{f1}$$

$$e_1{}^{f2} = F_1^{f2}{}_{ref} - F_1^{f2}$$

$$e_1{}^{f2} = F_2^{f2}{}_{ref} - F_2^{f2} \quad (62)$$

Table 5: Desired Terminal Conditions for followers

Robot	Initial condition (F1) m	Final condition (F1) m	Initial condition (F2) m	Final condition (F2) m	Distance from leader (d^{ref})m	Bearing from leader (ϕ^{ref}) rad
Leader	0	1.5	0	0.5	-	-
Follower 1	-0.49	1	-0.11	0.33	0.52	0.2276
Follower2	-1.03	0.5	-0.22	0.17	1.054	0.2109

Let $\begin{bmatrix}\ddot{F}_1^{fs}\\ \ddot{F}_2^{fs}\end{bmatrix} = \begin{vmatrix}\delta_1{}^{fs}\\ \delta_2{}^{fs}\end{vmatrix}$ then the feedback control laws to the new inputs can then be defined as

$$\delta_1{}^{fs} = \ddot{F}_1^{fs}{}_{ref} + \rho_1{}^{fs}\dot{e}_1{}^{fs} + \rho_0{}^{fs}e_1{}^{fs} \quad (63)$$

$$\delta_2{}^{fs} = \ddot{F}_2^{fs}{}_{ref} + \tilde{\rho}_1{}^{fs}\dot{e}_2{}^{fs} + \tilde{\rho}_0{}^{fs}e_2{}^{fs} \quad (64)$$

Where $\tilde{\rho}_1{}^{fs}, \tilde{\rho}_0{}^{fs}, \rho_1{}^{fs}, \rho_0{}^{fs}$ are control the gains and subscript s = [1,2] specifies the robot.

Next the error dynamics of the system in flat output space is determined.

$$\ddot{F}_1^{fs}{}_{ref} - \delta_1{}^{fs} = \ddot{F}_1^{fs}{}_{ref} - \ddot{F}_1 = \ddot{e}_1{}^{fs}$$

$$\ddot{F}_2^{fs}{}_{ref} - \delta_2{}^{fs} = \ddot{F}_2^{fs}{}_{ref} - \ddot{F}_2 = \ddot{e}_2{}^{fs} \quad (65)$$

Therefore, the error dynamics is defined as:

$$0 = \ddot{e}_1{}^{fs} + \rho_1{}^{fs}\dot{e}_1{}^{fs} + \rho_0{}^{fs}e_1{}^{fs}$$

$$0 = \ddot{e}_2{}^{fs} + \tilde{\rho}_1{}^{fs}\dot{e}_2{}^{fs} + \tilde{\rho}_0{}^{fs}e_2{}^{fs} \quad (66)$$

4.5.2 PID Formation Controller

Tests were conducted to compare the flatness-based controller design with the conventional PID controller. The PID controllers produces control signals that has a part that is proportional to the tracking error, a part that is the integration of the error and a part that is the error's derivative. Thus, the equation of PID controller is as follow:

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{d}{dt} e(t) \tag{67}$$

Where $u(t)$ and $e(t)$ are the control signal and error signal respectively, while k_p, k_i and k_d represent the of the proportional, integral, and derivative coefficients, respectively.

The coefficients were chosen by means of manual tuning. The PID controller is used in order to minimise the position error. Equation of the PID controlled formation is presented in [68] as:

$$v_i = v_j[k_{px} e_x(t) + k_{ix} \int_0^t e_x(t)dt + k_{dx} \frac{d}{dt} e_x(t)]$$

$$\omega_i = k_{py} e_y(t) + k_{iy} \int_0^t e_y(t)dt + k_{dy} \frac{d}{dt} e_y(t) \tag{68}$$

4.6 Conclusion

In this chapter, the concept of differential flatness theory was introduced. A differential flatness analysis of a differentially driven nonholonomic wheeled mobile robot was conducted, and the flat outputs were chosen. Next, differential flatness theory was used to generate feasible trajectories with consideration of the desired terminal conditions. To track these trajectories a differential flatness-based controller was designed. To test the effectiveness of the controller a simulation software was used, and results were recorded.

From the results it was seen that the robot successfully tracked its desired trajectories. Focus was thereafter given to the cooperative system modeling and control. A leader-follower approach was used. To maintain the formation, two controllers were designed, which is the PID and the flatness-based controller. The next chapter will present the simulation result of test done to compare the effectiveness of this controllers.

CHAPTER 5: SIMULATION TESTS, RESULTS AND DISCUSSION

5.1 Introduction

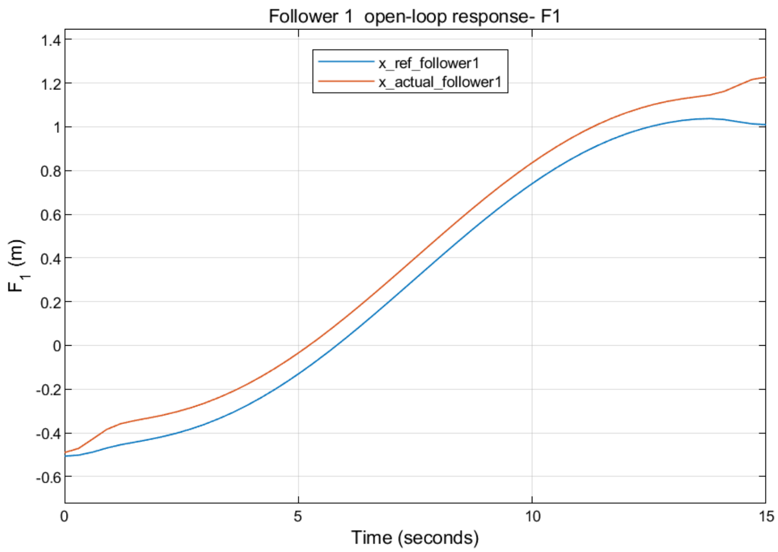
The objective of this chapter is to assess the efficiency of the developed controller model presented in the previous chapter. That is, the simulation results that illustrate the follower robot's ability to follow its leader accurately are presented. The MATLAB/Simulink simulation software is used to demonstrate the trajectory tracking abilities as well as formation maintenance of the robots. Two main cases are studied and that is, the ability of the robots to maintain their formation in open-loop or a closed-loop control system. An open-loop control system is defined as control system in which the output of the system does not have any effect on its input, that is the system acts completely based on the input. The open loop control system has no feedback to determine if the desired output has been achieved. Appendix 2 shows the Simulink diagrams of the open-loop control cooperative system model. Alternatively, in a closed-loop control system the output of the system is fed back to the input of the system to adjust the input such that the desired output is attained and maintained. PID and the flatness-based controller are the two closed-loop controllers used in this study. The performance of the robots is compared when a PID controller is used versus flatness controller. Appendices 3 and 4 show the Simulink diagrams of the cooperative system with flatness and PID controllers respectively.

5.2 Simulation results

As mentioned in the previous section, the MATLAB/Simulink simulation software is used to demonstrate the effectiveness of the formation controllers. The flat output trajectories have been calculated in the previous chapter. It was also seen that the leader robot was able to successfully track its reference trajectories when the flatness-based controller was incorporated (Figure 14). Therefore, the simulations below only demonstrate the ability of the followers to track their reference flat output trajectories which are essentially the x-y coordinates. The other focus of this section will be to study the ability of the followers to maintain a predefined distance and bearing to the leader robot.

5.2.1 Effect of open-loop control on the formation

First, the response of the follower without any feedback control system is examined. Both follower 1 and follower 2 will be observed on their ability to follow their reference flat output trajectories, F_1^{f1} and F_2^{f1}, for follower 1 and, F_1^{f2} and F_2^{f2} for follower 2. These reference trajectories ensure the maintenance of the desired distance and bearing (l^{ref}, φ^{ref}), to the leader. The desired flat output trajectories have been shown in Figure (15) and the initial and desired final conditions are shown in Table 5 in the previous chapter. Note that there are no disturbances added to the system. Firstly, the response of the first follower to correctly track the desired trajectories of the first flat output (F_1^{f1}) and the second flat output (F_2^{f1}) is studied. Figure 16 shows the results obtained.



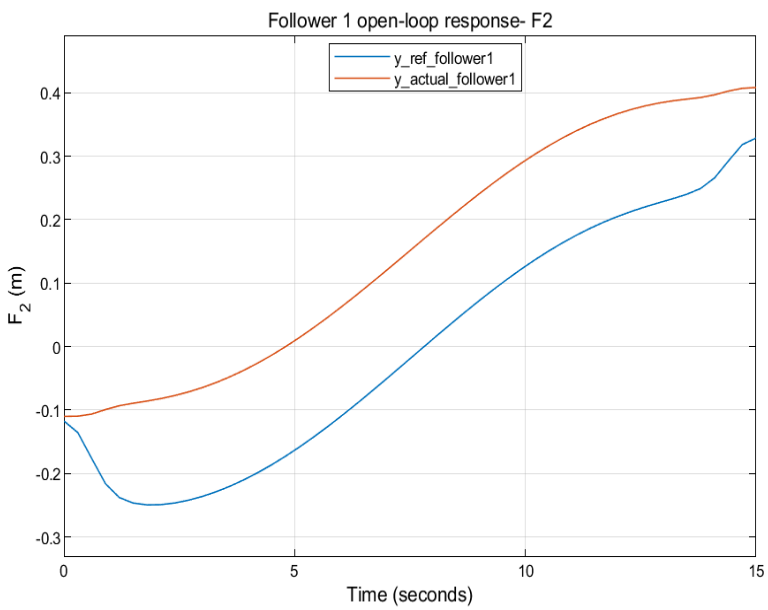


Figure 16: First follower's open loop response: F_1 and F_2 trajectory tracking

Recall that the reference trajectories were calculated such that at every instant of time, distance and bearing (l^{ref}, φ^{ref}) between follower and leader are constant. Thus, failing to track trajectories means failing to maintain formation. As seen in Figure 16, follower 1 failed to track the desired trajectories, thus formation was not kept, as evident in Figures 17 and 18.

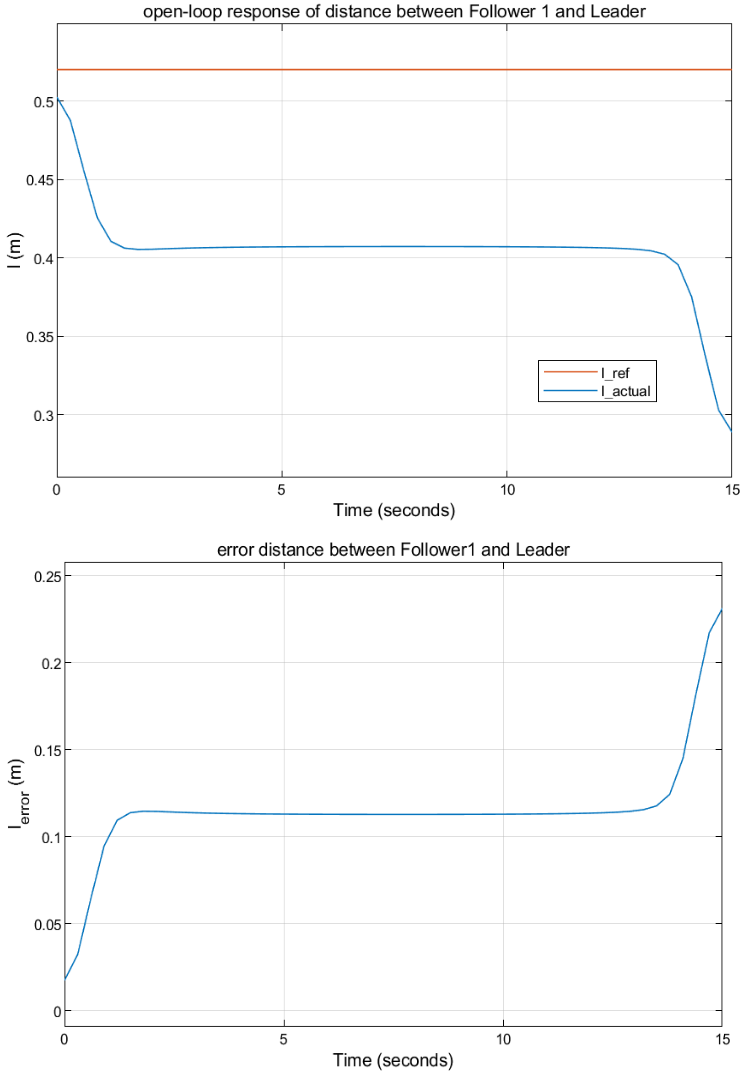


Figure 17: Separation Distance l^{ref} Open-Loop Response of Follower1

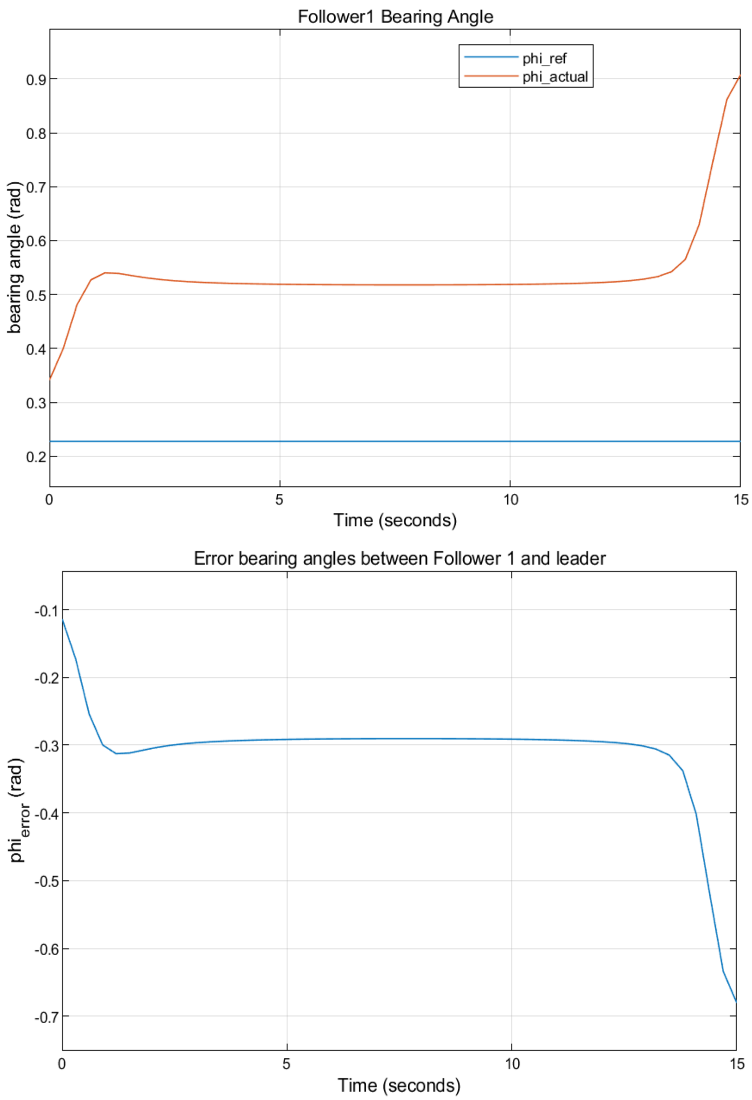


Figure 18: Separation Bearing φ^{ref} Open-Loop Response of Follower1

Likewise, the open-loop response of the second follower robot is shown in Figure 19. Just like the first follower, the second follower failed to track the reference trajectories.

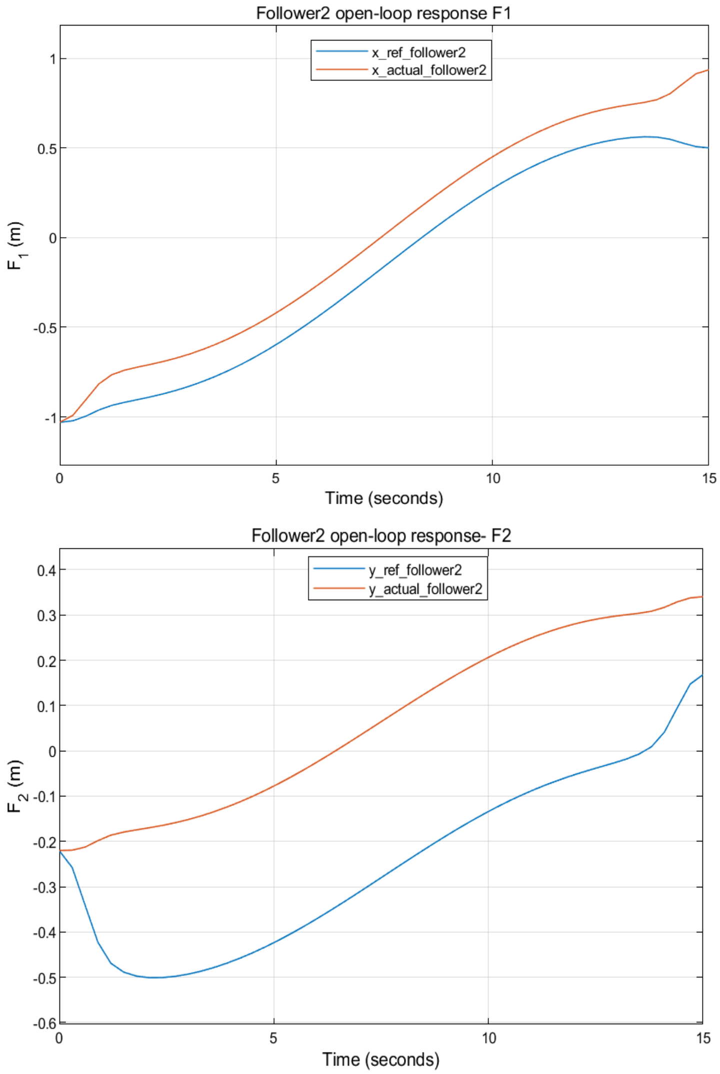


Figure 19: Second follower's open loop response: F_1 and F_2 trajectory tracking

Figures 20 and 21 therefore show errors in the separation distance and bearing from the leader. The error is significant as a result the cooperative system failed to stay in formation. From these simulations it is evident that there is a need for a controller to compensate for the open-loop insufficiency.

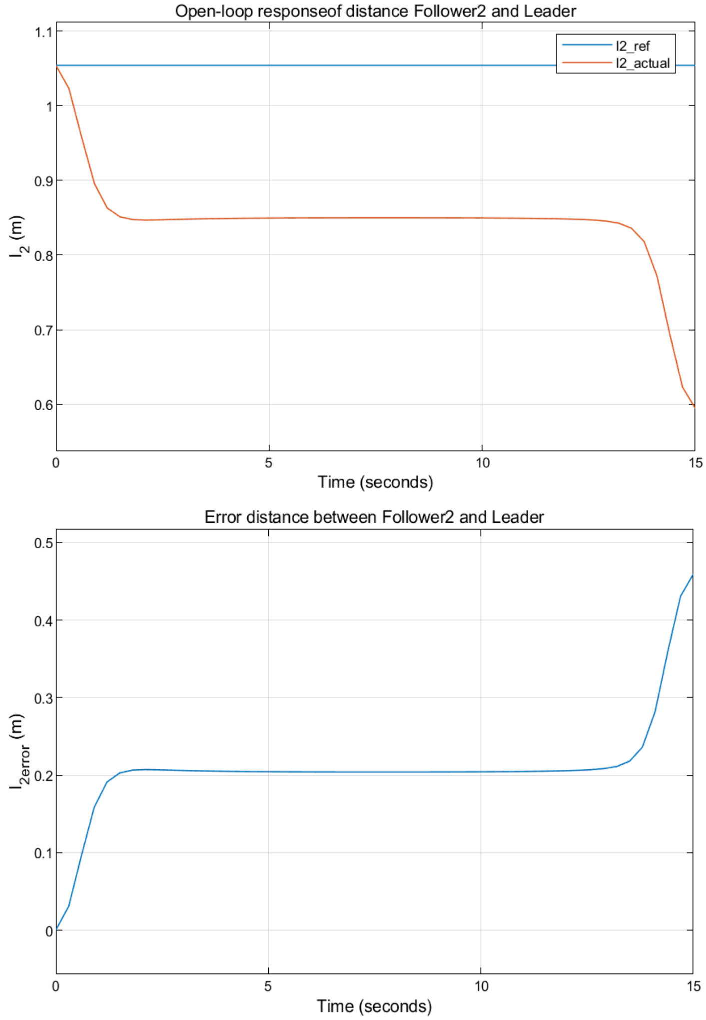


Figure 20: Separation Distance l^{ref} Open-Loop Response of Follower2

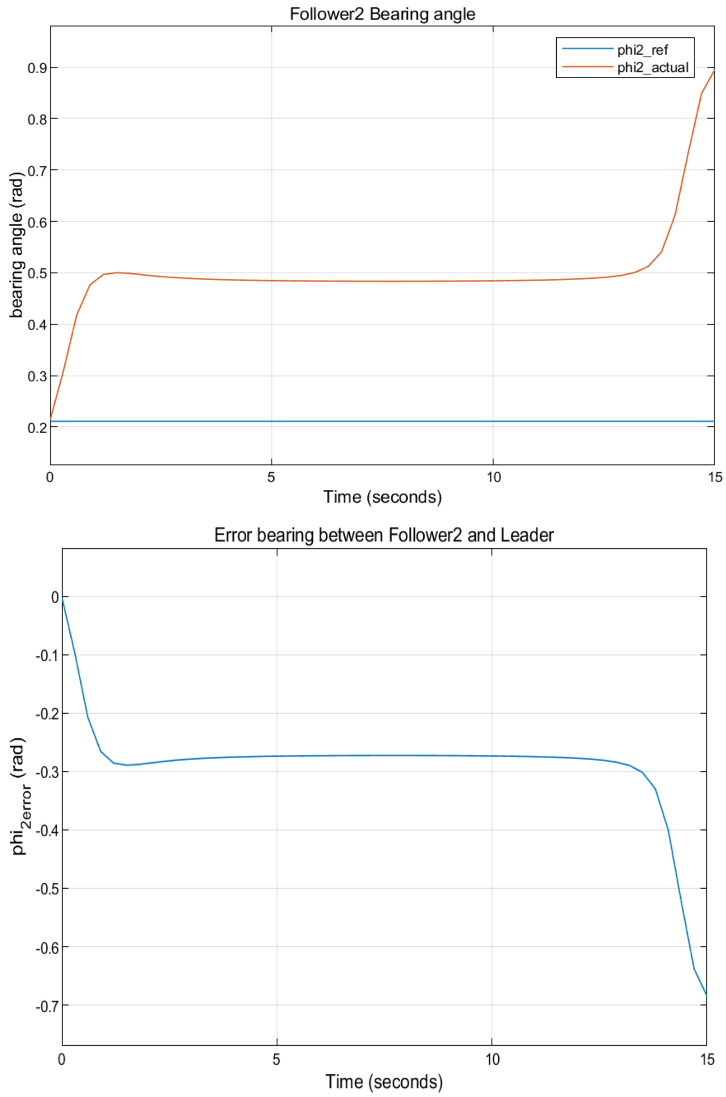


Figure 21: Separation Bearing φ^{ref} Open-Loop Response of Follower2

5.2.2 Effect of closed-loop control on the formation

5.2.2.1 PID Formation Controller Results

A PID controller is used to better the formation accuracy of the open loop control. The gains that were obtained by using a PID tuner application on Simulink. The tuner automatically calculates PID parameters that stabilise the system robustly. Thus, for the first follower the gains are:

$$k_{px1} = 5.028,\ k_{ix1} = 0.197,\ k_{dx1} = -0.217,\ k_{py1} = 10,\ k_{iy1} = 0.2,\ k_{dy1} = 1$$

The Figure 22 shows xy-trajectory response when a PID controller is used on follower 1.

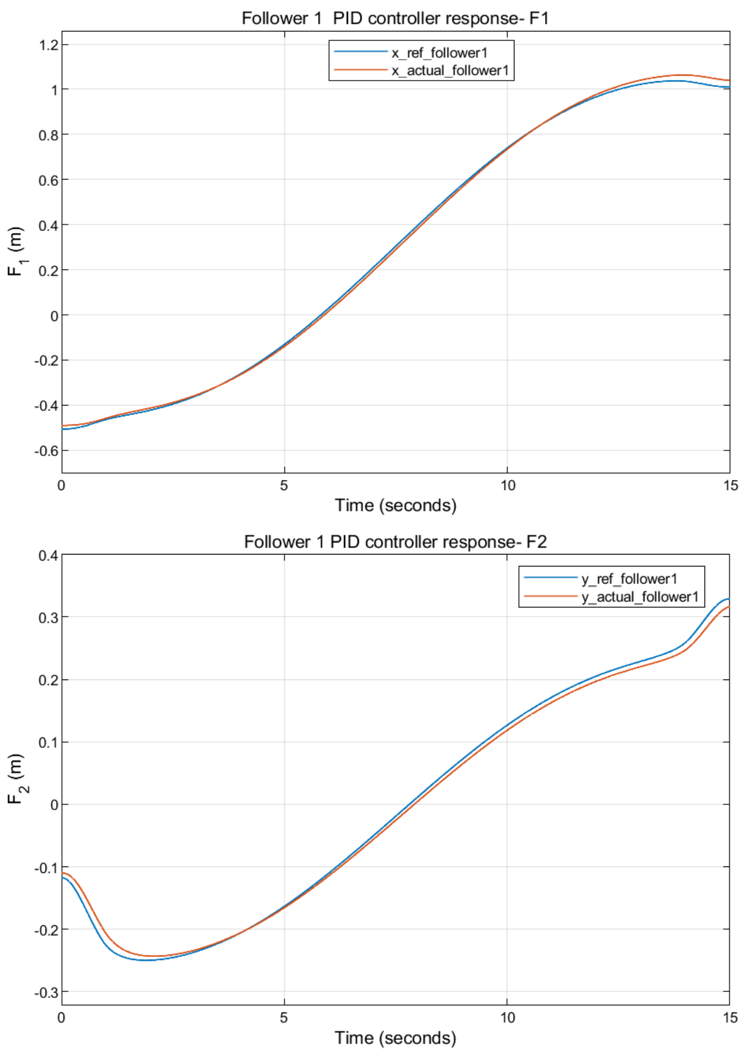


Figure 22: First follower's PID controller response: F_1 and F_2 trajectory tracking

Figure 22 validates that the PID controller has significantly improved on the tracking ability of the robot. On the first trajectory it successfully tracks the trajectory F_1 until towards the end at about 14 seconds where it starts shooting and shaking and thus fails to reach the desired terminal condition of F_1.

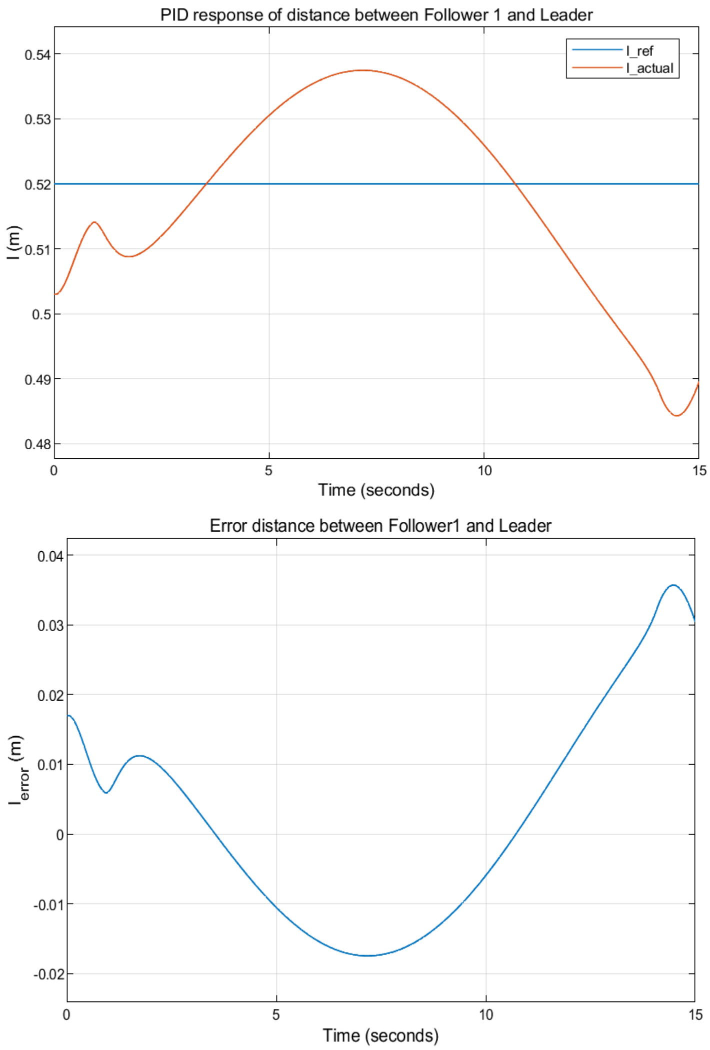


Figure 23: Separation Distance l^{ref} PID Response of Follower1

On the second trajectory also, the robot starts off with an error but later at around 2 seconds its able to correctly track the trajectory. However, at around 7 seconds it starts drifting away from the reference trajectory thus also fails to reach the desired terminal condition of F_2.

Figures 23 and 24 demonstrate the ability of the robot to stay in formation. It can be seen that the robot was not able to maintain a perfectly constant separation distance. However, the error graph shows that the error is significantly small ranging from -0.02m to 0.035m.

Similarly, the separation bearing was also not perfectly constant, with error ranging from -0.15 rad to 0.05rad.

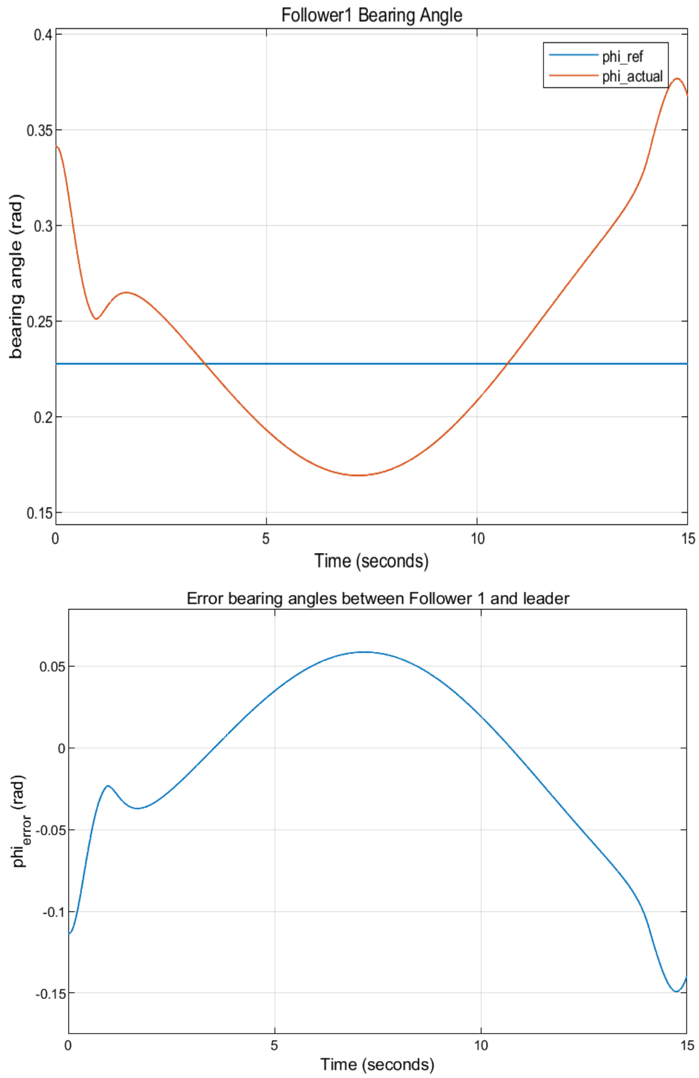


Figure 24: Separation bearing φ^{ref} PID Response of Follower1

Likewise, a PID controller was used on the second follower robot. The gains on PID were:

$K_{px2} = 6.064,\ k_{ix2} = 0.169,\ k_{dx2} = -0.256,\ K_{py2} = 9.84,\ k_{iy2} = 0.24,\ k_{dy2} = 1.04$

These were also obtained by using a PID tuner application on Simulink. The xy-trajectory response is presented in Figure 25. Here, also a major improvement to the tracking ability of the robot is be seen.

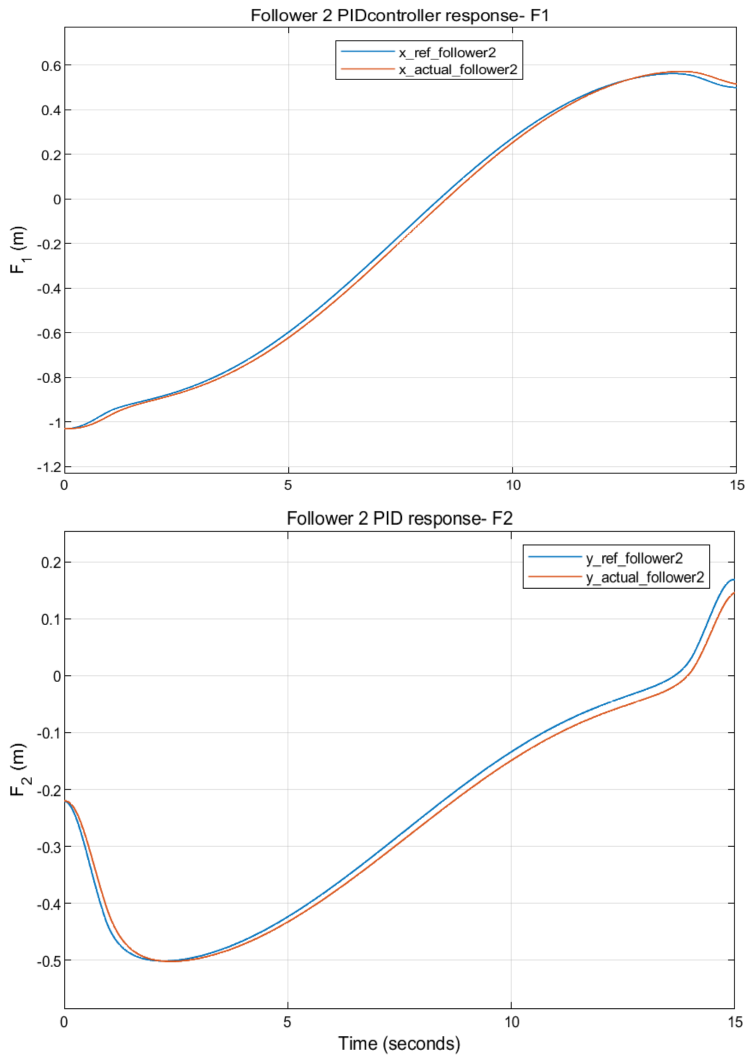
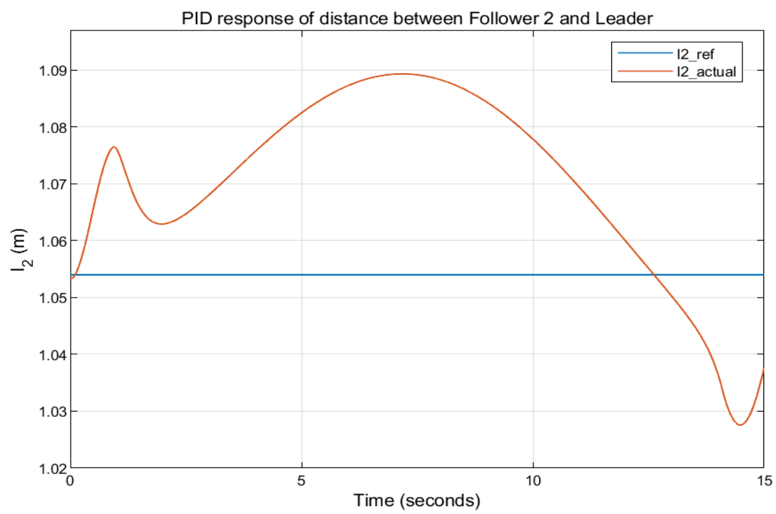
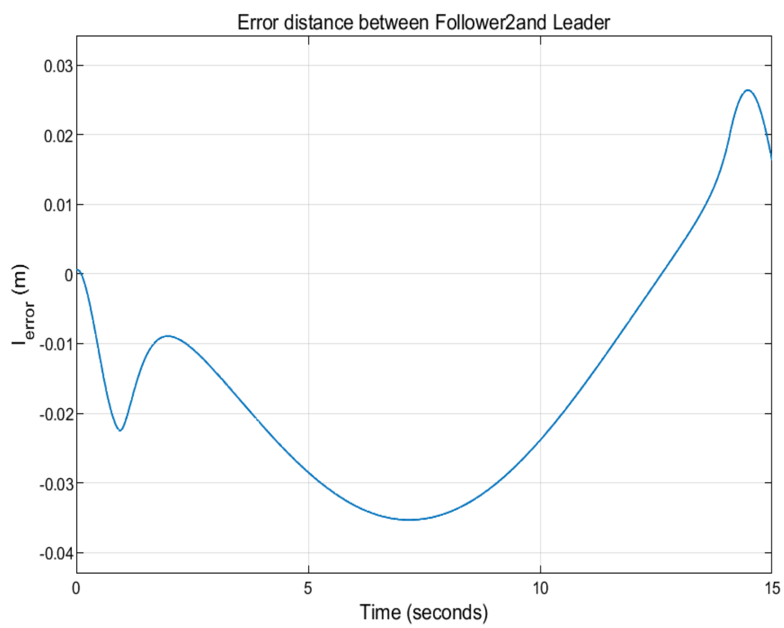
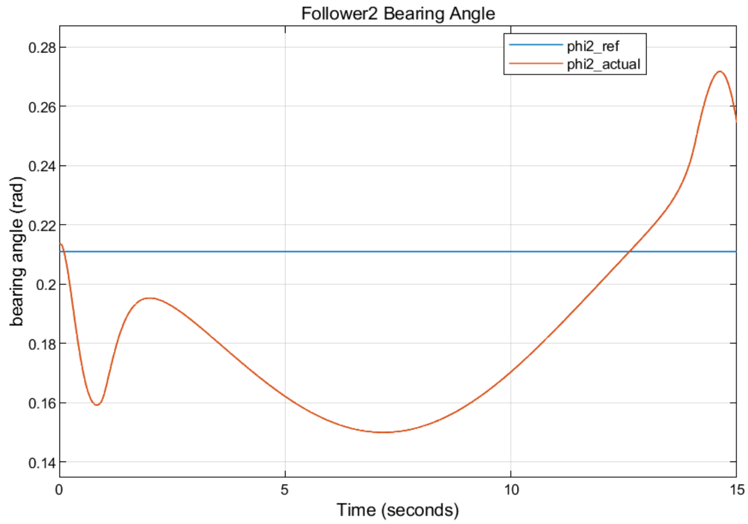


Figure 25: Second follower's PID controller response: F_1 and F_2 trajectory tracking

As seen in Figure 25 above, the robot tracks very close to the reference trajectories but is unable to track on the reference trajectory, thus fails to reach the desired terminal condition. Figure 26 shows the ability of the robot to maintain formation. Like the first follower, this robot was not able to maintain a perfectly constant separation distance. However, the error graph shows that the errors are significantly small ranging from -0.035m to 0.028m. Also, the separation bearing was also not perfectly constant, with error ranging from -0.06rad to 0.05rad







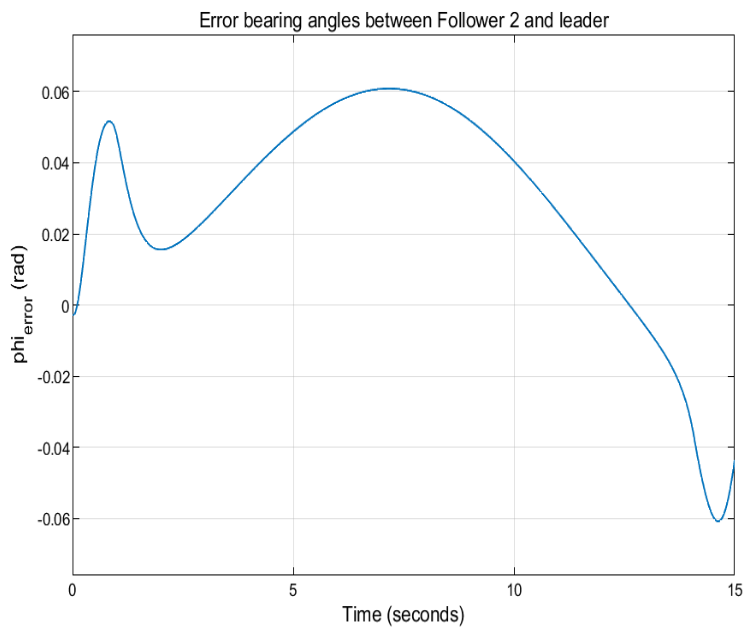


Figure 26 :**Separation Distance l^{ref} and bearing φ^{ref} PID Response of Follower2**

Finally, according to these simulation results it is evident that the PID controller gives reasonably good results.. Therefore, this shows that formation control is still in need of improvement. Also, another factor that contributes to errors might be due to inaccurate tuning caused by computational approximations.

5.2.2.2 Flatness-based Formation Controller

A flatness controller is used to reduce or eliminate the PID controller errors. For the first follower robot, control gains are chosen to be:

$\rho_0 = 3, \quad \rho_1 = 2, \quad \tilde{\rho}_1 = 5, \quad \tilde{\rho}_0 = 8$

Also, for the second follower robot, control gains are chosen to be:

$\rho'_0 = 2, \quad \rho'_1 = 1, \quad \widetilde{\rho'}_1 = 6, \quad \widetilde{\rho'}_0 = 9$

This are chosen by trial and error. Figure 27 below shows the tracking response of the first follower robot with a flatness-based controller.

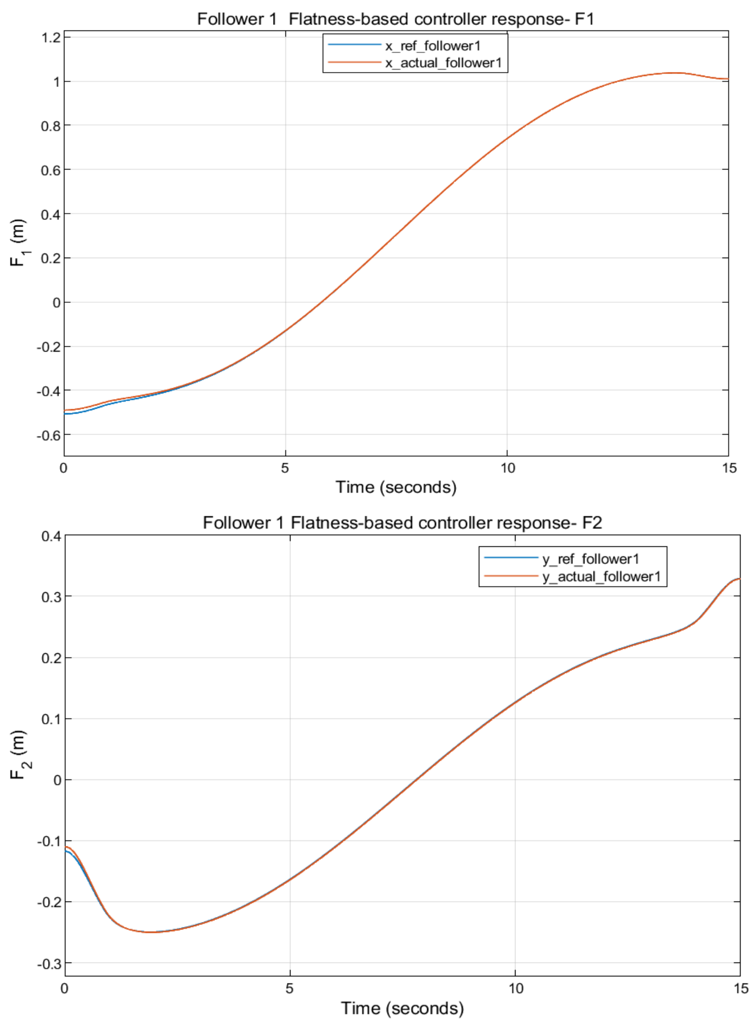


Figure 27: Flatness- Based Trajectory Tracking Response of the First Follower Robot

It can be seen that the the tracking ability of the robot has improved and the reference final coordinates are successfully reached. Figures 28 and 29 show the distance and bearing errors respectively. It can be seen that the error converges to zero and thus a constant distance and bearing is maintained.

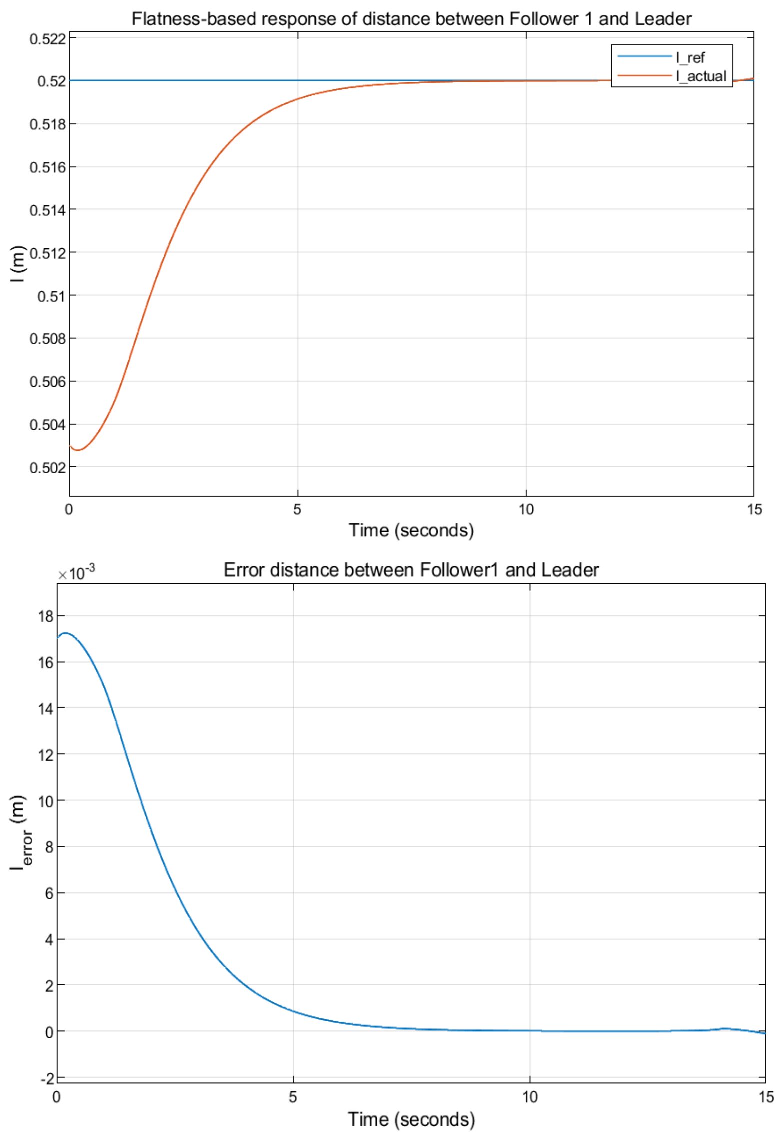


Figure 28: Flatness- Based Distance Maintenance Response of the First Follower Robot

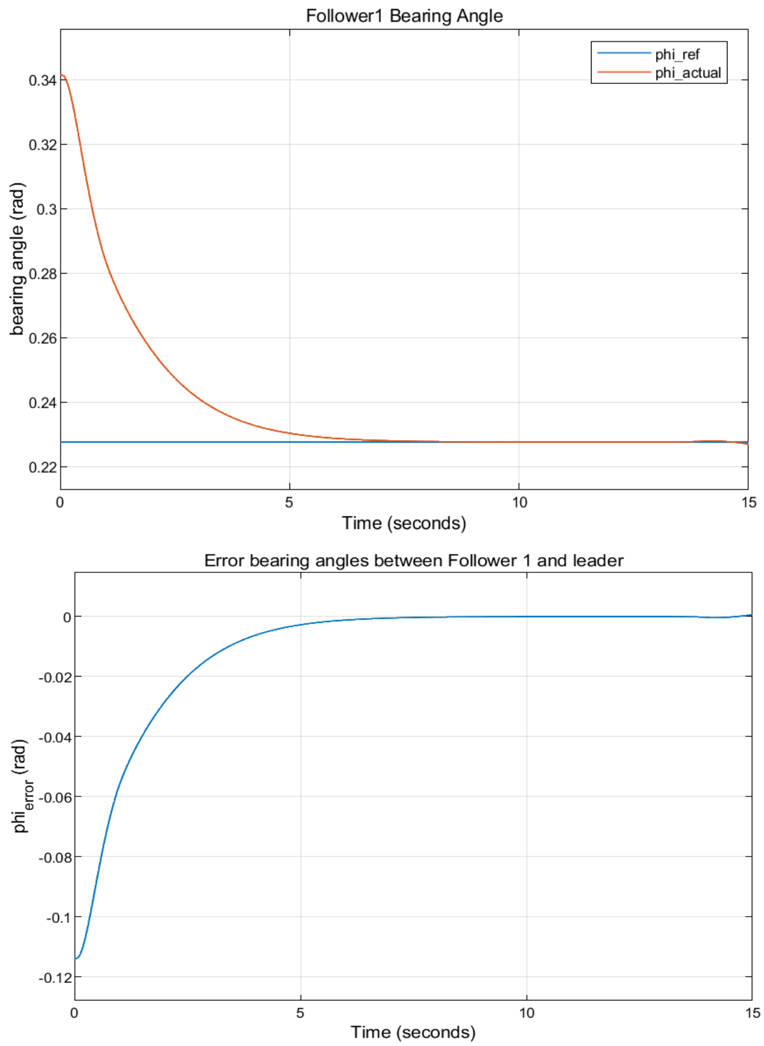


Figure 29: Flatness- Based Bearing Maintenance Response of the First Follower Robot

Like the first follower, Figure 30 shows the second robot's tracking response with a flatness-based controller. The second follower successfully follows its reference trajectory and thus maintaining a constant bearing and distance between the leader and itself. In the case of the second follower, the distance and bearing errors have successfully been reduced to zero (Figure 31), thus formation has successfully been maintained.

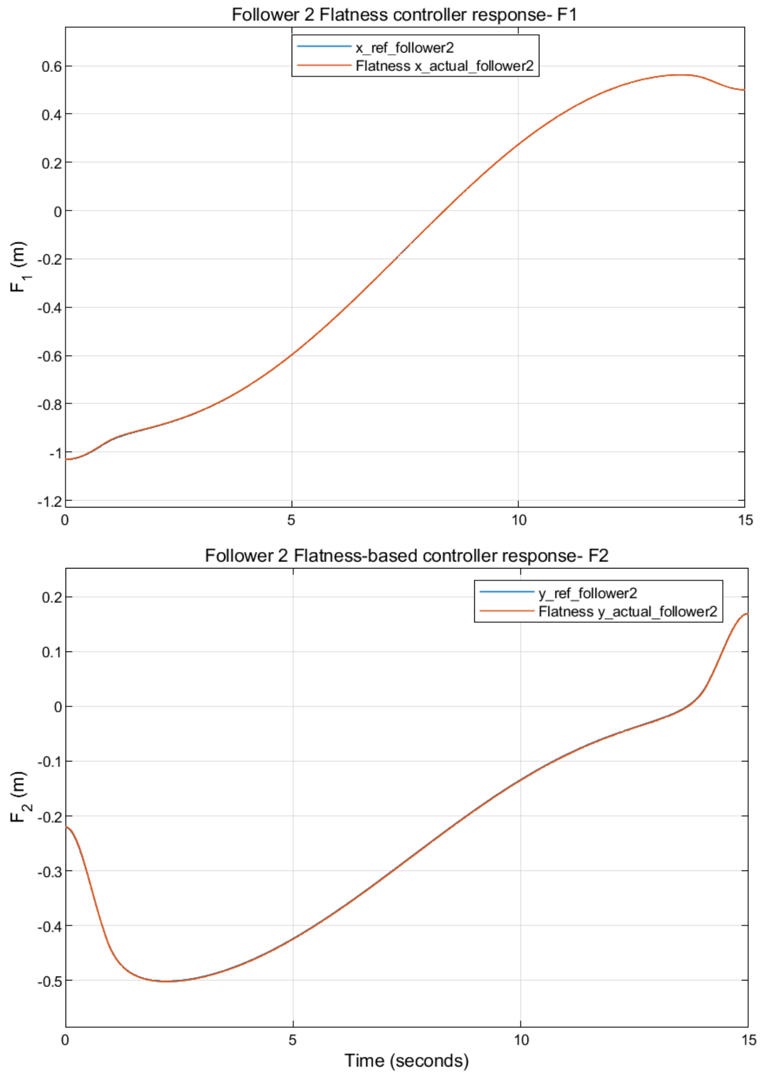
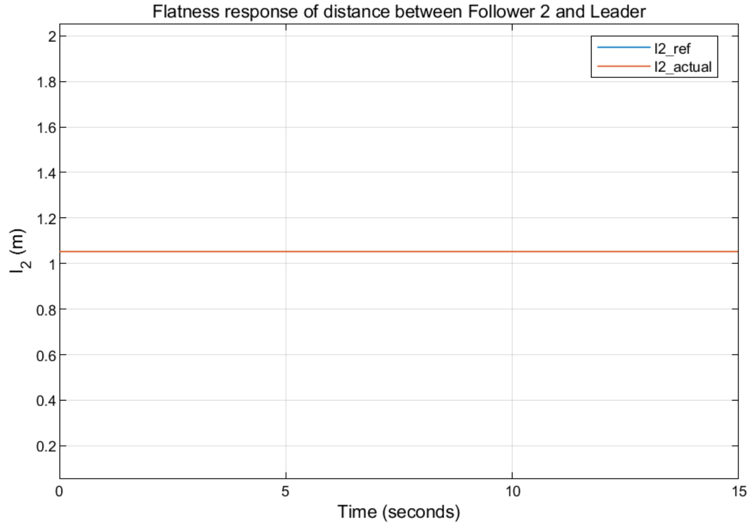
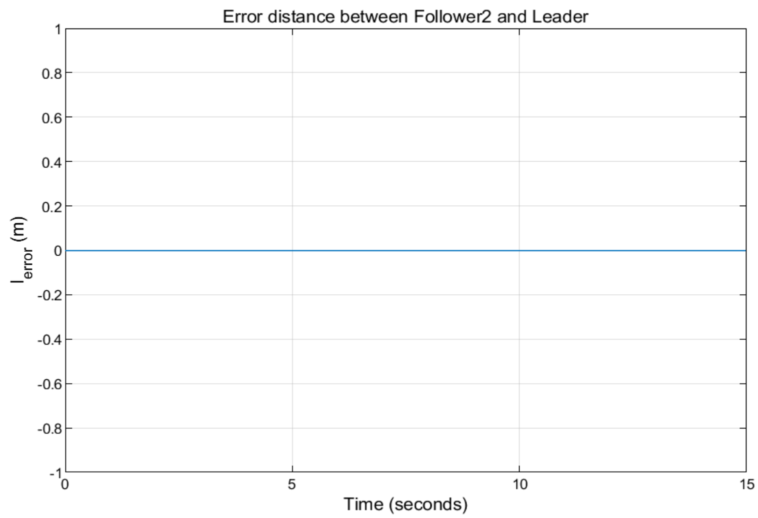


Figure 30: Flatness- Based Trajectory Tracking Response of the Second Follower Robot


Flatness response of distance between Follower 2 and Leader
l2_ref
l2_actual
l_2 (m)
Time (seconds)


Error distance between Follower2 and Leader
l_{error} (m)
Time (seconds)

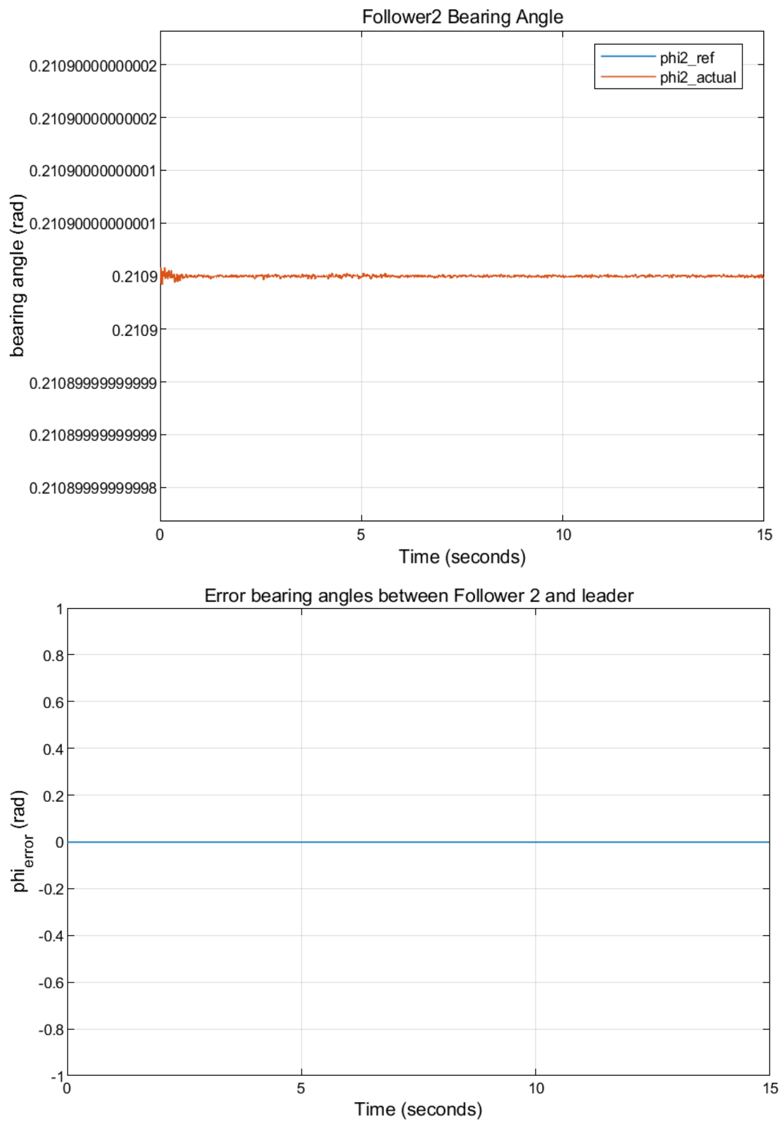
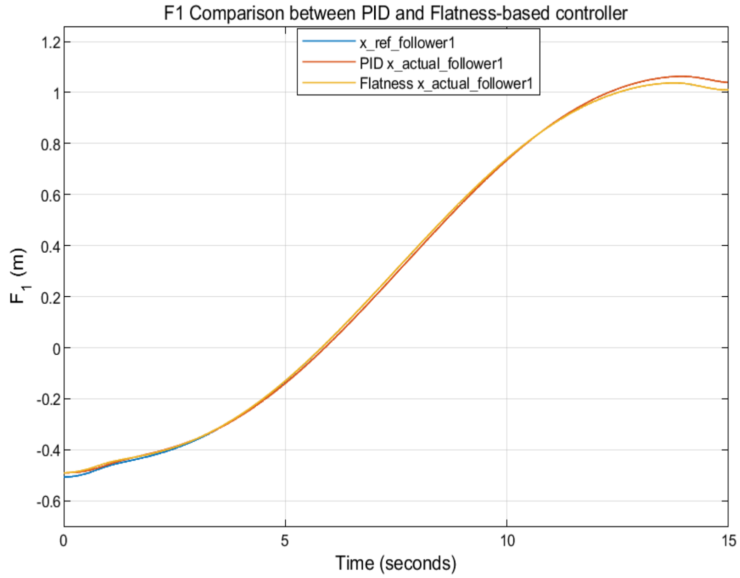
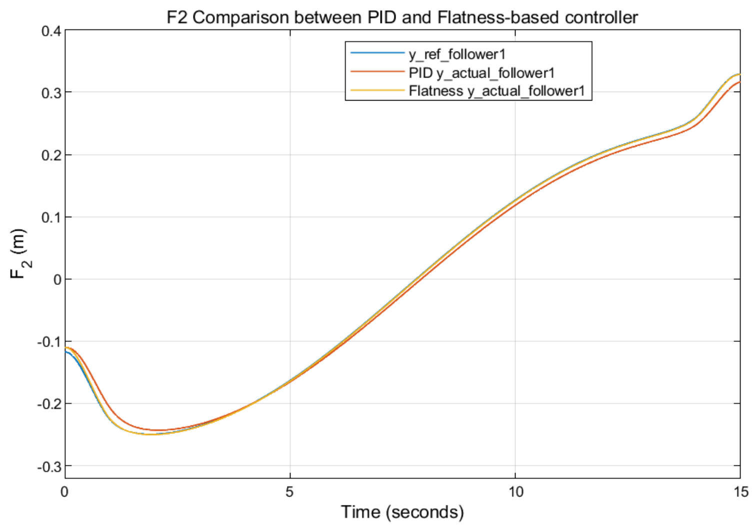


Figure 31: Flatness- Based Distance and Bearing Maintenance Response of the First Follower Robot

5.2.3 An analysis of flatness- and PID-based formation controllers

A comparison was made on trajectory tracking abilities between the widely used PID controller and the flatness-based controller. The below results were obtained:


F1 Comparison between PID and Flatness-based controller
x_ref_follower1
PID x_actual_follower1
Flatness x_actual_follower1
F_1 (m)
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
5
10
15


F2 Comparison between PID and Flatness-based controller
y_ref_follower1
PID y_actual_follower1
Flatness y_actual_follower1
F_2 (m)
Time (seconds)
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
5
10
15

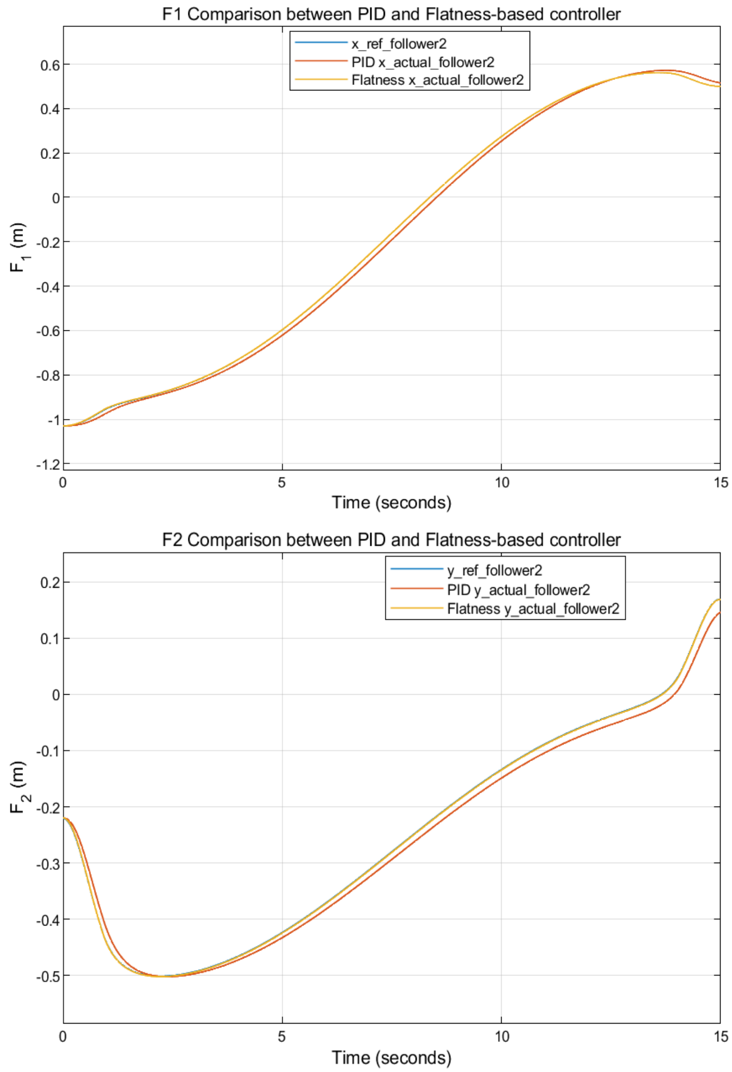
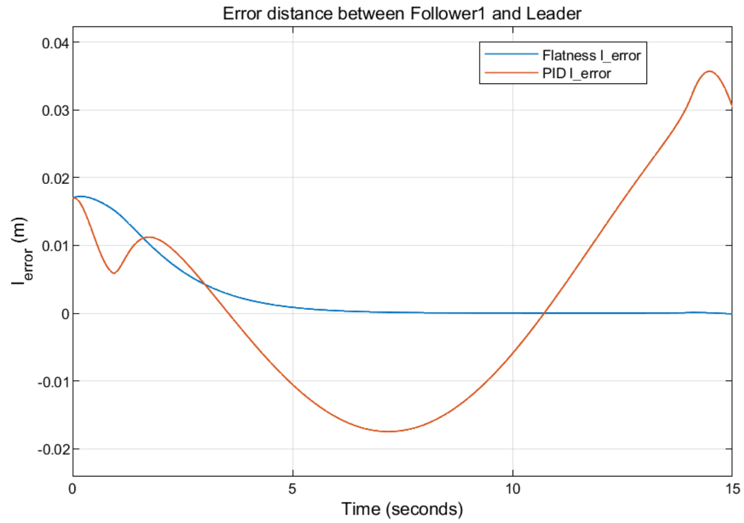
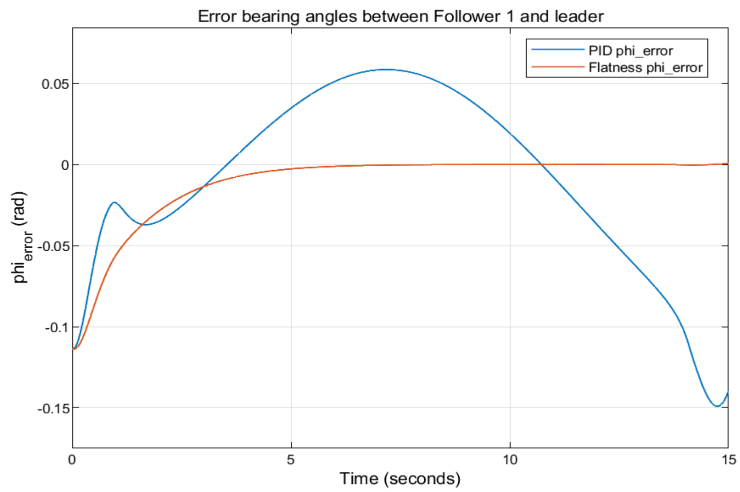


Figure 32: Trajectory Tracking Response of the Follower Robots- PID versus Flatness-based Controller

The flatness has successfully improved the trajectory tracking of the formation system (Figure 32). The follower robots are able to track the leader while maintaining a specified distance and bearing. Additionally, it can be seen that the use of a flatness controller has decreased the distance and bearing error to zero (Figure 33). It can be seen however that the robot oscillate a little before reaching a steady state of zero. This might be caused by computational errors and approximation of parameters. However, the robotic formation is maintaned successfully.


Error distance between Follower1 and Leader
Flatness l_error
PID l_error
l_error (m)
Time (seconds)


Error bearing angles between Follower 1 and leader
PID phi_error
Flatness phi_error
phi_error (rad)
Time (seconds)

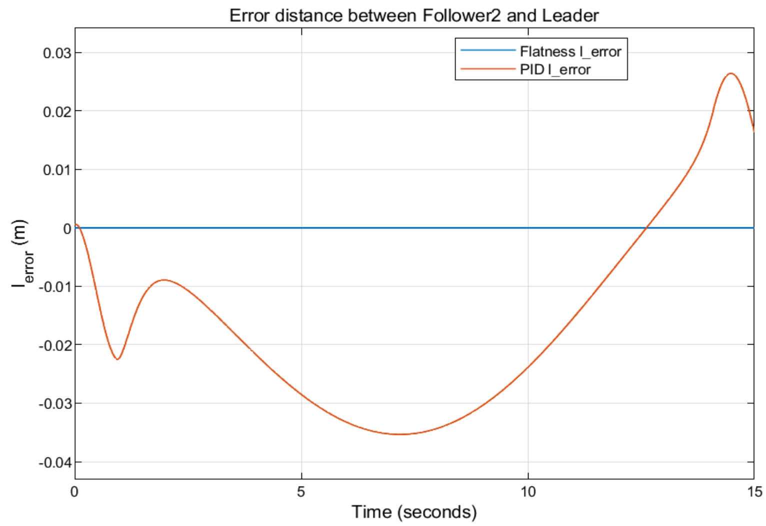


Figure 33: Distance and Bearing Maintenance Response of the First Follower Robots- PID versus Flatness-based Controller

Table 6 analyses the performance differences between the two controllers for follower 1.

Table 6: Error differences between PID and Flatness-based controller

Controller	**Distance from leader Error range(m)**	**Bearing from leader Error range(rad)**
PID	-0.02 to 0.035	-0.15 to 0.05
Flatness-based	0.018 to 0	0.1 to 0

It is readily apparent that the error of the PID has a wider range than of the flatness controller. From Figure 33 it is seen that the error of the PID controller never settles at zero, while for the flatness controller the error converges to zero at 0 seconds for both the distance and bearing respectively in follower 2. This means that the classical PID controller is more prone to oscillations and vibrations than the flatness-based controller. Also, when trying to tune the PID further so as to get more accurate control, it became saturated, but the gains of the flat controller were increased easily without any saturation. However, it can be seen that the PID error is small, meaning it can still be used for formation control and produce reasonably correct results.

5.3 Results Discussion

Simulation results of the effectiveness of the flatness-based formation controller were illustrated in the previous section. Firstly, results were compared between an open-loop response and a closed loop response of the trajectory tracking ability of a multirobot system. Figure 16 illustrates the response graph for the open-loop system. In open loop there is no output feedback signal to the input to ensure that the output maintains a desired value. As seen in figures16 to 19, both followers failed to track their reference trajectories. This is because without any feedback the system cannot correct any errors, thus the followers remain out of track. Thus, the open-loop control system becomes insufficient. It is for this reason that a closed-loop controller is therefore necessary to compensate for the open-loop inadequacy. Figures 22 to 26 show how the PID controller has improved the ability of the robot to follow their reference trajectories. Figure 22 and 25 shows that both followers closely track their reference trajectories on for a few first seconds until the graph shows the followers slightly drifting away from the reference trajectory. This output instability is due to inaccurate tuning caused by computational approximations. However, the overall performance of the PID is great improvement to that of the open-loop control. Another feedback controller that was used in the study for trajectory tracking was the flatness-based controller. Figures 27 to 31 evidently shows that flatness-based controller has significantly improved the tracking ability of the two followers. The followers are now able to track the reference trajectory and thus maintaining a constant bearing and distance with the leader. The distance and bearing errors have successfully been reduced to zero (Figure29 and 31), thus formation has successfully been maintained. It is seen however in figure 29 that the error takes some time before converging to zero, this might be due to approximation of parameters and calculation errors. When comparing the two feedback controllers, table 6 shows that the flatness controller has the least error, thus formation is better maintained. Finally there this is enough evidence to prove that the flatness-based controller improves coordination of the cooperative robotic system.

5.4 Conclusion

In this chapter, the simulation result of tests done to confirm the effectiveness of the Flatness and PID controller were presented. Firstly, the robotic system was tested to see if it could maintain formation when no controller is used. It was evident that for the robots to maintain a formation, a controller is essential. Therefore, the PID and the Flatness controller were tested, and the simulation results recorded. Both controllers proved to be effective in maintaining the robotic formation. A comparison was made between these two controllers. It can be seen that the Flatness controller greatly reduces the tracking error of the cooperative system, whereas the PID has slightly higher errors for both the separation distance and orientation to the leader. This is due to the fact that when the gains of the PID are changed passed some values the PID gets saturated. Conversely, the Flatness controller can be tuned without any concern of saturation.

CHAPTER 6: CONCLUSION AND FUTURE STUDIES

6.1 Introduction

In the previous chapters, the concept of differential flatness was examined and how it could be used to trivialise the challenge posed by controlling cooperative robotic formation. A mathematical model of the robot was developed, and a differential flatness analysis was made to confirm the flatness properties of the robots involved in the formation. A differential flatness-based controller was then designed to control the multi-robot systems, and tests were conducted on a simulation software to test its effectiveness. This chapter provides the final conclusion of this study, along with recommendations for future research

6.2 Initial objective and Research Findings

The purpose of this project was to improve coordination control of a model based cooperative multiple mobile robotic system using differential flatness theory. A kinematic and dynamic model of cooperative multiple mobile robotic system was determined, and a differential flatness characterisation of robots involved in the cooperative system was conducted. It was found that a robot can be represented by a chosen set of variables known as a flat output, which are equal to the number of the system's control inputs. These flat outputs can represent the whole robot because all the robot states and inputs can be determined from these outputs without integration. This reduces the computational cost when designing a controller especially in a cooperative system with many robot- members. Furthermore, the robots in this study are nonholonomic and thus had motion constraints. It was found that when using flat outputs, motion constraints have no impact on the robot formation. This is because the constraint is embedded in a semi-algebraic set that is adaptive to the differential flatness parametrisation.

6.3 Conclusions

In this study, a differential flatness characterisation of a nonholonomic differentially driven wheeled mobile robot was conducted. This enabled the linearisation of the system to a stable linear equivalent system. Also, the whole system is represented by a reduced number of variables and thus the robot calculation task is significantly reduced especially when dealing with multiple robots that can otherwise entail solving large robotic model differential equations. Furthermore, in flat output space a simple polynomial-based trajectory planning can be used, that is simplifying the trajectory generation problem.

Using the kinematic and dynamic models of the robot, the study conducted a differential flatness characterisation of the robots in the cooperative system. This enabled the exploitation of the flatness properties. An effective controller was designed based on the flatness properties. Trajectory generation was also simplified in flat output space and the flatness controller was used to effectively track them. To maintain the cooperative formation, the controller was used to maintain a constant separation distance and bearing angle between the leader and followers. Thus, the study successfully used differential flatness theory to improve coordination control of a model based cooperative multiple mobile robotic system. The key findings of the research were that a differential flatness characterisation enabled the linearisation of the system. Also, representation of the whole system by fewer variables significantly reduced the multi-robotic model for easier calculations. Also, in flat output space the trajectory generation problem is simplified. This is because in flat output space simple polynomial-based trajectory planning can be used, thus, trajectories are solved without integrating robot model differential equations.

Finally, through the research findings this study concluded that the differential flatness theory can be utilised in order to effectively control the formation of cooperative behaviour among multiple robots. Thus, it is concluded that by using differentially flat outputs, trajectory generation problem is reduced to an algebra problem thus trivialising it. Also, the trajectory generation problem is trivialised by the fact that parameters can be easily estimated when using differential flatness without affecting the stability of the formation. Also, when a set of feasible reference trajectories is found using flatness, the constraints of the system are satisfied automatically. Additionally, the computational time for designing a controller that controls multiple robots is significantly reduced because all the robot variables are represented by smaller set of variables (flat outputs). Finally, it is concluded from this study and its findings that differential flatness theory significantly improves coordination control of cooperative multiple mobile robotic systems.

6.4 Recommendations for future work

Having completed this project successfully, there are several promising areas for further study. In this study, it was assumed that no external disturbances are added to the system. For further study the effects of flatness trajectory planning can be tested in mobile systems under friction, or with modelling errors, unreliable estimates, having to avoid obstacles and where there is additional perturbations. Also, the effect of the trajectory execution time and tuning parameters selection on trajectory generation can be further explored.

REFERENCES

[1] Markus, E.D., Agee, J., Jimoh, A.G. and Hamam, Y., 2015. *Coordinated control of multiple robotic manipulators based on differential flatness* (Doctoral dissertation, Tshwane University of Technology).

[2] Dasgupta, P., Baca, J., Guruprasad, K.R., Muñoz-Meléndez, A. and Jumadinova, J., 2015. The comrade system for multirobot autonomous landmine detection in post conflict regions. *Journal of Robotics*.

[3] de Alcantara Andrade, F.A., Reinier Hovenburg, A., Netto de Lima, L., Dahlin Rodin, C., Johansen, T.A., Storvold, R., Moraes Correia, C.A. and Barreto Haddad, D., 2019. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. *Sensors*, *19*(19), p.4067.

[4] Gao, S., Song, R. and Li, Y., 2018. Cooperative control of multiple nonholonomic robots for escorting and patrolling mission based on vector field. *IEEE Access*, 6, pp.41883-41891.

[5] Atallah, S., Parra-Davila, E. and Melani, A.G.F., 2019. Assessment of the Versius surgical robotic system for dual-field synchronous transanal total mesorectal excision (taTME) in a preclinical model: will tomorrow's surgical robots promise newfound options. *Techniques in Coloproctology*, *23*(5), pp.471-477.

[6] Verma, J.K. and Ranga, V., 2021. Multi-robot coordination analysis, taxonomy, challenges, and future scope. Journal of intelligent & robotic systems, 102(1), pp.1-36.

[7] Darmanin, R.N. and Bugeja, M.K., 2017, July. A review on multi-robot systems categorised by application domain. In 2017 25th Mediterranean conference on control and automation (MED) (pp. 701-706). IEEE.

[8] Zhou, L. and Tokekar, P., 2021. Multi-robot coordination and planning in uncertain and adversarial environments. Current Robotics Reports, 2(2), pp.147-157.

[9] Wiedemann, T., Shutin, D. and Lilienthal, A.J., 2019. Model-based gas source localization strategy for a cooperative multi-robot system—A probabilistic

approach and experimental validation incorporating physical knowledge and model uncertainties. *Robotics and Autonomous Systems*, *118*, pp.66-79.

[10] Sapiee, M.R.M. and Annuar, K.A.M., 2018. Synchronous mobile robots formation control. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *16*(3), pp.1183-1192.

[11] Wang, J.W., Guo, Y., Fahad, M. and Bingham, B., 2019. Dynamic plume tracking by cooperative robots. *IEEE/ASME Transactions on Mechatronics*, *24*(2), pp.609-620.

[12] Nuño, E., Loría, A. and Panteley, E., 2021. Leaderless consensus formation control of cooperative multi-agent vehicles without velocity measurements. *IEEE Control Systems Letters*, *6*, pp.902-907.

[13] Paniagua-Contro, P., Hernandez-Martinez, E.G., González-Medina, O., González-Sierra, J., Flores-Godoy, J.J., Ferreira-Vazquez, E.D. and Fernandez-Anaya, G., 2019. Extension of leader-follower behaviours for wheeled mobile robots in multirobot coordination. *Mathematical Problems in Engineering*, *2019*.

[14] Cen, H. and Singh, B.K., 2021. Nonholonomic Wheeled Mobile Robot Trajectory Tracking Control Based on Improved Sliding Mode Variable Structure. *Wireless Communications and Mobile Computing*, 2021.

[15] Cui, M., Liu, W., Liu, H., Jiang, H. and Wang, Z., 2016. Extended state observer-based adaptive sliding mode control of differential-driving mobile robot with uncertainties. *Nonlinear Dynamics*, *83*(1), pp.667-68

[16] Falsafi, M., Alipour, K. and Tarvirdizadeh, B., 2019. Fuzzy motion control for wheeled mobile robots in real-time. *Journal of Computational & Applied Research in Mechanical Engineering (JCARME)*, *8*(2), pp.133-144.

[17] Ammar, H.H., and Azar, A.T., 2019, March. Robust path tracking of mobile robot using fractional order PID controller. *In International Conference on Advanced Machine Learning Technologies and Applications* (pp. 370-381). Springer, Cham.

[18] Rudra, S., Barai, R.K. and Maitra, M., 2016. Design and implementation of a block-backstepping based tracking control for nonholonomic wheeled mobile

robot. *International Journal of Robust and Nonlinear Control*, *26*(14), pp.3018-3035.

[19] Mars 2020 Perseverance Rover - NASA Mars https://mars.nasa.gov/mars2020/ Accessed: 2020-12-08

[20] DoD Office Launches Swarm Of 103 Micro UAVs https://www.defensedaily.com/dod-office-launches-swarm-of-103-micro-uavs/unmanned-systems/ Accessed: 2021-08-18

[21] RoboBees: Autonomous Flying Microrobots https://wyss.harvard.edu/technology/robobees-autonomous-flying-microrobots/ Accessed: 2021-08-18

[22] Assembly, G., 2015. Resolution adopted by the General Assembly on 11 September 2015. A/RES/69/315 15 September 2015. New York: United Nations.

[23] Sira-Ramirez, H., and Agrawal, S.K., 2018. *Differentially flat systems*. Crc Press.

[24] Singh, N.H. and Thongam, K., 2018. Mobile robot navigation using fuzzy logic in static environments. *Procedia Computer Science*, 125, pp.11-17.

[25] Patle, B.K., Parhi, D.R.K., Jagadeesh, A. and Kashyap, S.K., 2019. Application of probability to enhance the performance of fuzzy based mobile robot navigation. *Applied Soft Computing*, 75, pp.265-283.

[26] Štefek, A., Pham, V.T., Krivanek, V. and Pham, K.L., 2021. Optimization of Fuzzy Logic Controller Used for a Differential Drive Wheeled Mobile Robot. *Applied Sciences*, *11*(13), p.6023.

[27] Singh, N.H. and Thongam, K., 2019. Neural network-based approaches for mobile robot navigation in static and moving obstacles environments. *Intelligent Service Robotics*, *12*(1), pp.55-67.

[28] Asai, M., Chen, G. and Takami, I., 2019, March. Neural network trajectory tracking of tracked mobile robot. *In 2019 16th International Multi-Conference on Systems, Signals & Devices* (SSD) (pp. 225-230). IEEE.

[29] Zhao, L., Jin, J. and Gong, J., 2021. Robust zeroing neural network for fixed-time kinematic control of wheeled mobile robot in noise-polluted environment. *Mathematics and Computers in Simulation*, 185, pp.289-307.

[30] Azar, A.T., Ammar, H.H., Brito Silva, G.D. and Razali, M.S.A.B., 2019, March. Optimal proportional integral derivative (PID) controller design for smart irrigation mobile robot with soil moisture sensor. *In International Conference on Advanced Machine Learning Technologies and Applications* (pp. 349-359). Springer, Cham.

[31] Wang, T.Y. and Chang, C.D., 2018, April. Hybrid fuzzy PID controller design for a mobile robot. *In 2018 IEEE International Conference on Applied System Invention (ICASI)* (pp. 650-653). IEEE.

[32] Yu, X., Fan, Y., Xu, S. and Ou, L., 2021. A self-adaptive SAC-PID control approach based on reinforcement learning for mobile robots. *International Journal of Robust and Nonlinear Control*.

[33] Goswami, N.K. and Padhy, P.K., 2018. Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance. *Computers & Electrical Engineering*, 72, pp.307-323.

[34] Moudoud, B., Aissaoui, H. and Diany, M., 2022. Fuzzy adaptive sliding mode controller for electrically driven wheeled mobile robot for trajectory tracking task. *Journal of Control and Decision*, *9*(1), pp.71-79.

[35] Phuc, P.T., Tho, T.P., Hai, N.D.X. and Thinh, N.T., 2021. Design of adaptive fuzzy sliding mode controller for mobile robot. *International Journal of Mechanical Engineering and Robotics Research, 10*(2), pp.54-59.

[36] Markus, E.D., 2018, October. Differential flatness based synchronization control of multiple heterogeneous robots. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society* (pp. 3659-3664). IEEE.

[37] Fliess, M., Lévine, J., Martin, P. and Rouchon, P., 1995. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, *61*(6), pp.1327-1361.

[38] Lévine, J., 2011. On necessary and sufficient conditions for differential flatness. Applicable Algebra in Engineering, Communication and Computing, 22(1), pp.47-90.

[39] Boubezoula, M., Hassam, A. and Boutalbi, O., 2018. Robust-flatness controller design for a differentially driven wheeled mobile robot. *International Journal of Control, Automation and Systems*, *16*(4), pp.1895-1904.

[40] Khesrani, S., Hassam, A., Boutalbi, O. and Boubezoula, M., 2021. Motion planning and control of nonholonomic mobile robot using flatness and fuzzy logic concepts. *International Journal of Dynamics and Control*, *9*(4), pp.1660-1671.

[41] Abadi, A., Mekki, H., Brahim, A.B.H., El Amraoui, A. and Ramdani, N., 2017, December. Optimal trajectory generation and flatness tracking control for a mobile robot. In *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)* (pp. 223-228). IEEE.

[42] Abadi, A., Mekki, H., Brahim, A.B.H., El Amraoui, A. and Ramdani, N., 2017, December. Optimal trajectory generation and flatness tracking control for a mobile robot. In *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)* (pp. 223-228). IEEE.

[43] Abadi, A., El Amraoui, A., Mekki, H. and Ramdani, N., 2020. Flatness-Based Active Disturbance Rejection Control for a Wheeled Mobile Robot Subject to Slips and External Environmental Disturbances. *IFAC-PapersOnLine*, *53*(2), pp.9571-9576.

[44] Petrovsky, A., Kalinov, I., Karpyshev, P., Tsetserukou, D., Ivanov, A. and Golkar, A., 2022. The two-wheeled robotic swarm concept for Mars exploration. *Acta Astronautica*, *194*, pp.1-8.

[45] Li, Y., Li, M., Zhu, H., Hu, E., Tang, C., Li, P. and You, S., 2020. Development and applications of rescue robots for explosion accidents in coal mines. *Journal of Field Robotics*, *37*(3), pp.466-489.

[46] Lei, G. and Zheng, Y., 2021. Research on cooperative trajectory planning algorithm based on Tractor-Trailer Wheeled Robot. *IEEE Access*.

[47] Hausman, K., Müller, J., Hariharan, A., Ayanian, N. and Sukhatme, G.S., 2015. Cooperative multi-robot control for target tracking with onboard sensing. *The International Journal of Robotics Research*, *34*(13), pp.1660-1677.

[48] Panagou, D. and Kumar, V., 2014. Cooperative visibility maintenance for leader–follower formations in obstacle environments. *IEEE Transactions on Robotics*, *30*(4), pp.831-844.

[49] Jain, R.P., Aguiar, A.P. and de Sousa, J.B., 2018. Cooperative path following of robotic vehicles using an event-based control and communication strategy. *IEEE Robotics and Automation Letters*, *3*(3), pp.1941-1948.

[50] Xu, P., Zheng, J., Zhang, J., Zhang, K., Cui, Y. and Tang, Q., 2021, July. Distributed Position-Force Control for Cooperative Transportation with Multiple Mobile Manipulators. In *International Conference on Swarm Intelligence* (pp. 111-118). Springer, Cham.

[51] Ren, W. and Cao, Y., 2008. Simulation and experimental study of consensus algorithms for multiple mobile robots with information feedback. *Intelligent Automation & Soft Computing*, *14*(1), pp.73-87.

[52] Wei, H., Lv, Q., Duo, N., Wang, G. and Liang, B., 2019. Consensus algorithms based multi-robot formation control under noise and time delay conditions. *Applied Sciences*, *9*(5), p.1004.

[53] Wang, G., Wang, C., Du, Q., Li, L. and Dong, W., 2016. Distributed cooperative control of multiple nonholonomic mobile robots. *Journal of Intelligent & Robotic Systems*, *83*(3), pp.525-541.

[54] Liu, L., Yu, J., Ji, J., Miao, Z. and Zhou, J., 2019. Cooperative adaptive consensus tracking for multiple nonholonomic mobile robots. *International Journal of Systems Science*, *50*(8), pp.1556-1567.

[55] Du, H., Wen, G., Cheng, Y., He, Y. and Jia, R., 2016. Distributed finite-time cooperative control of multiple high-order nonholonomic mobile robots. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), pp.2998-3006.

[56] Akbari, A., Muhayyuddin and Rosell, J., 2019. Knowledge-oriented task and motion planning for multiple mobile robots. *Journal of Experimental & Theoretical Artificial Intelligence*, *31*(1), pp.137-162.

[57] Gutiérrez, H., Morales, A. and Nijmeijer H., 2017. Synchronization control for a swarm of unicycle robots: analysis of different controller topologies. Asian Journal of Control, *19*(5), pp. 1822-1833.

[58] Xiang, X., Liu, C., Lapierre, L. and Jouvencel, B., 2012. Synchronized path following control of multiple homogenous underactuated AUVs. *Journal of Systems Science and Complexity*, *25*(1), pp.71-89.

[59] Sun, D., Wang, C., Shang, W. and Feng, G., 2009. A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. *IEEE Transactions on Robotics*, *25*(5), pp.1074-1086.

[60] Ge, H., Chen, G. and Xu, G., 2018. Multi-AUV cooperative target hunting based on improved potential field in a surface-water environment. *Applied Sciences*, *8*(6), p.973.

[61] Tsiu, L. and Markus, E.D., 2020. A Survey of Formation Control for Multiple Mobile Robotic Systems. International Journal of Mechanical Engineering and Robotics Research, *9*(11).

[62] Khaledyan, M. and de Queiroz, M., 2019. A formation maneuvering controller for multiple non-holonomic robotic vehicles. *Robotica*, *37*(1), pp.189-211.

[63] Zhu, T., Ling, H. and He, W., 2018, June. A cooperative control approach of UAV autonomous formation and reconfiguration. In *2018 Chinese Control and Decision Conference (CCDC)* (pp. 2415-2420). *IEEE*.

[64] Ge, H., Chen, G. and Xu, G., 2018. Multi-AUV cooperative target hunting based on improved potential field in a surface-water environment. *Applied Sciences*, *8*(6), p.973.

[65] Zhao, R., Xiang, X., Yu, C. and Jiang, Z., 2016, September. Coordinated formation control of autonomous underwater vehicles based on leader-follower strategy. In *OCEANS 2016 MTS/IEEE Monterey* (pp. 1-5). IEEE.

[66] Wheel Control Theory http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html. Accessed: 2022-01-15.

[67] Tzafestas, S.G., 2013. *Introduction to Mobile Robot Control*. Elsevier.

[68] Tang, C.P., 2009, December. Differential flatness-based kinematic and dynamic control of a differentially driven wheeled mobile robot. In 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 2267-2272). IEEE.

[69] Choi, I.S. and Choi, J.S., 2012, October. Leader-follower formation control using PID controller. In International Conference on Intelligent Robotics and Applications (pp. 625-634). Springer, Berlin, Heidelberg.

[70] Dai, Y., Tran, V.H., Xu, Z. and Lee, S.G., 2010, June. Leader-follower formation control of multi-robots by using a stable tracking control method. In International Conference in Swarm Intelligence (pp. 291-298). Springer, Berlin, Heidelberg.

[71] Chetty, R.K., Singaperumal, M. and Nagarajan, T., 2011. Cooperative Formation Planning and Control of Multiple Mobile Robots. In Mobile Robots-Control Architectures, Bio-Interfacing, Navigation, Multi Robot Motion Planning and Operator Training. IntechOpen.

[72] Ryu, J.C. and Agrawal, S.K., 2011. Differential flatness-based robust control of mobile robots in the presence of slip. The International Journal of Robotics Research, 30(4), pp.463-475.

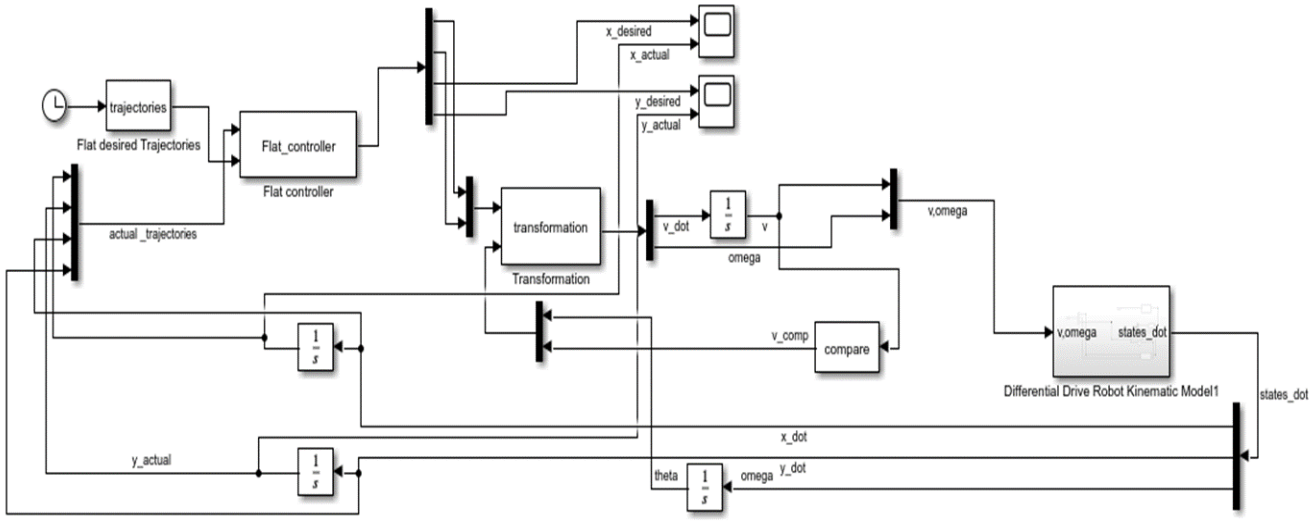
[73] Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G., 2009. Trajectory Planning. Robotics: Modelling, Planning and Control, pp. 163-168.

[74] Silva-Ortigoza, R., Marciano-Melchor, M., García-Chávez, R.E., Roldán-Caballero, A., Hernández-Guzmán, V.M., Hernández-Márquez, E., García-Sánchez, J.R., García-Cortés, R. and Silva-Ortigoza, G., 2022. Robust Flatness-Based Tracking Control for a "Full-Bridge Buck Inverter–DC Motor" System. Mathematics, 10(21), p.4110.

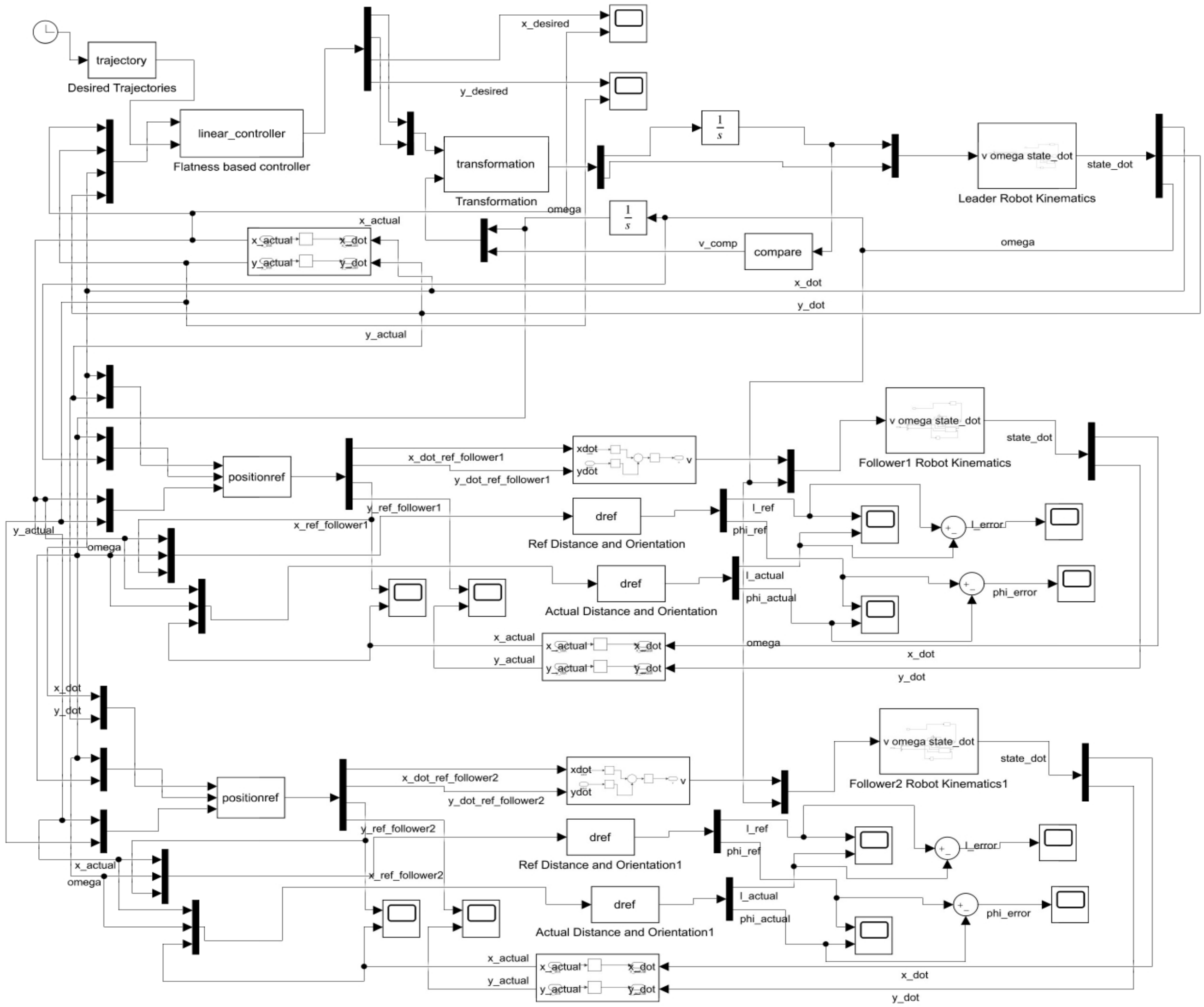
APPENDICES

7.1 MATLAB/SIMULINK Simulation Diagrams

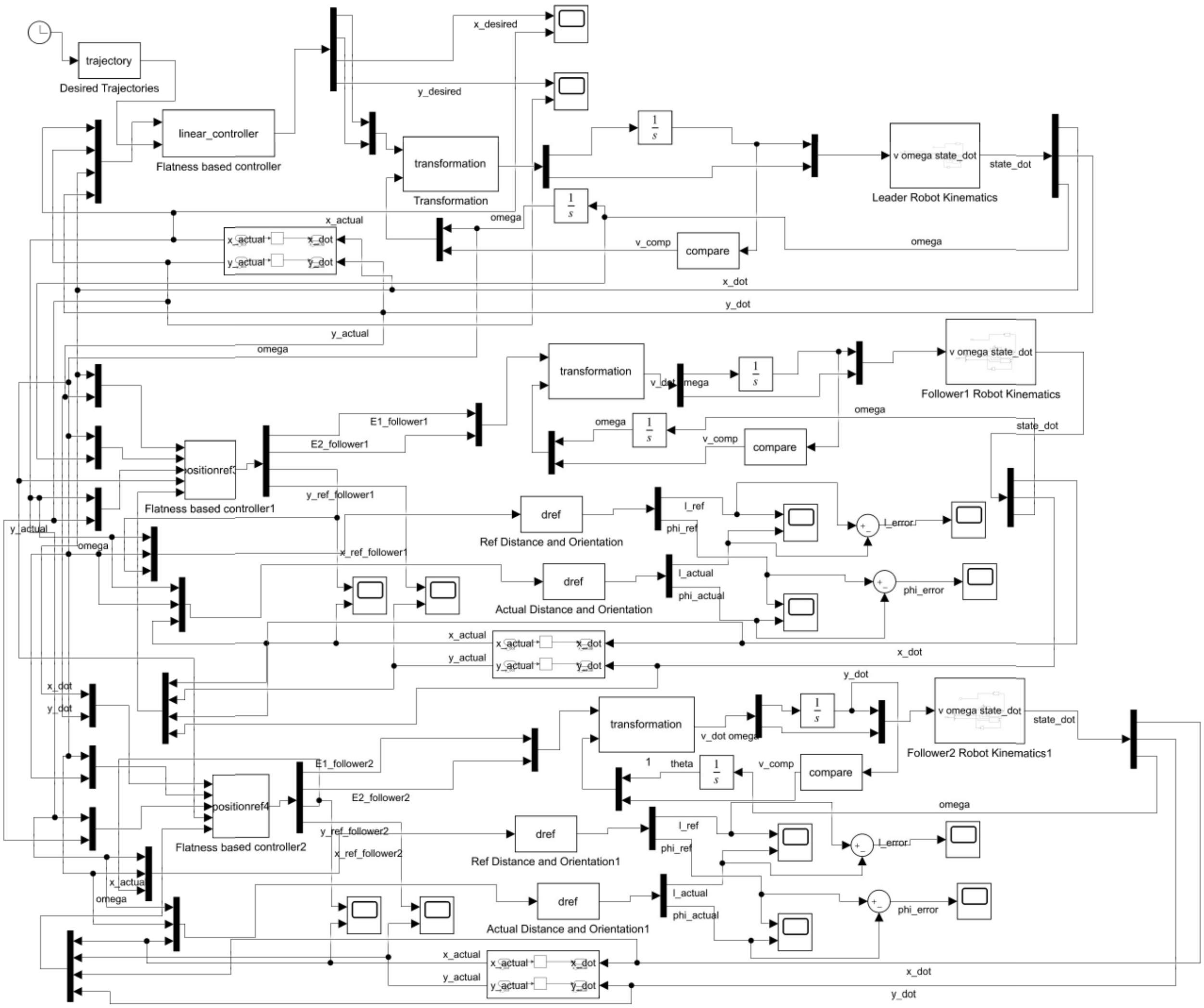
7.1.1 Appendix 1: Single robot flatness control in SIMULINK



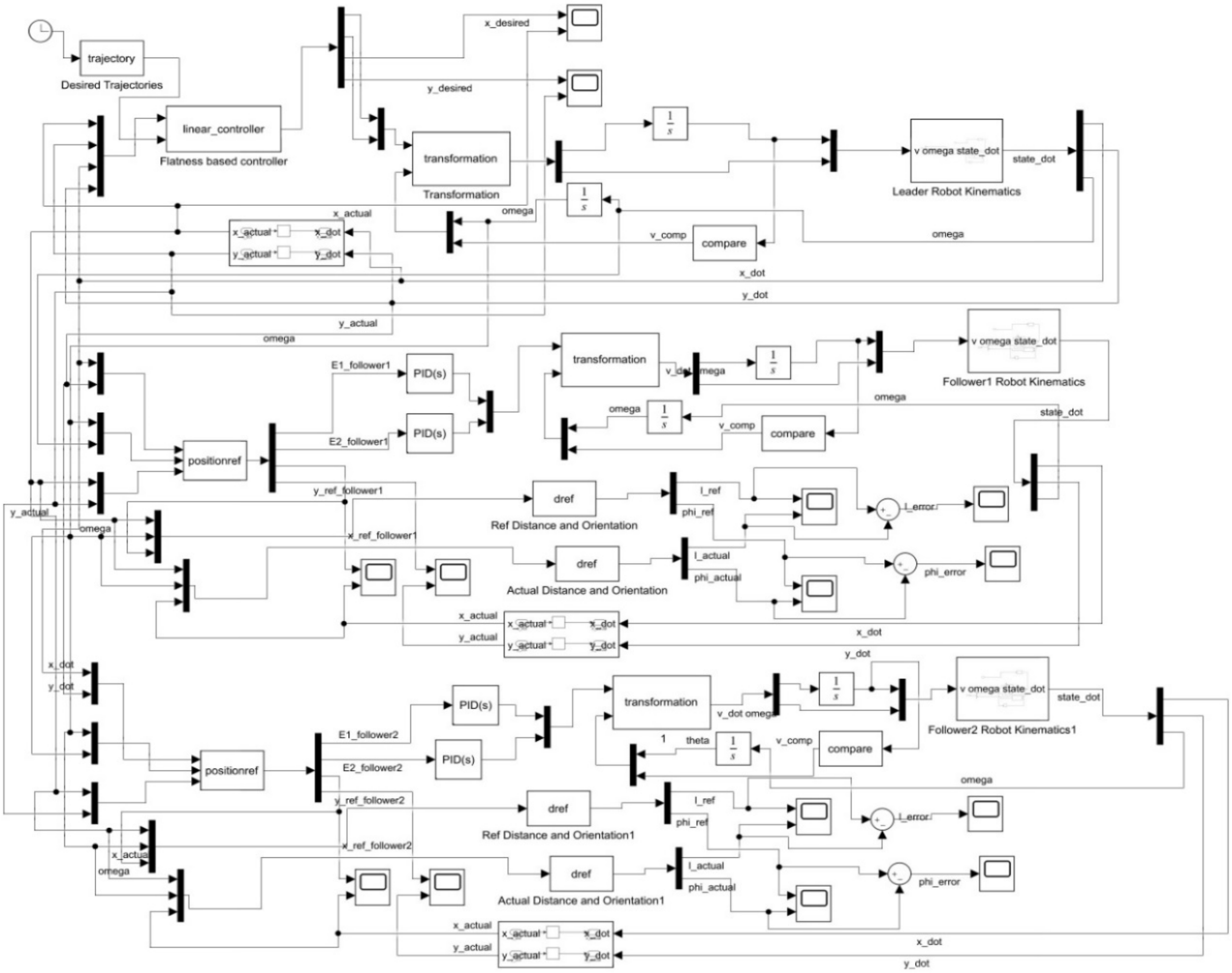
7.1.2 Appendix 2: Cooperative robotic System open-loop control in SIMULINK



7.1.3 Appendix 3: Cooperative robotic System with Flatness controller in SIMULINK



7.1.4 Appendix 4: Cooperative robotic System with PID



NERESHNEE GOVENDER COMMUNICATIONS (PTY) LTD

REGISTRATION NUMBER: 2016/369223/07
DR NERESHNEE GOVENDER (PhD)
neresh@ngcommunications.co.za
0847022553
WRITING PRACTITIONER • EDITOR • COPYWRITER • TRAINER

PhD-Management Sciences: Marketing (gender and media); *PG DIP - Higher Education - Academic Developers (Cum laude;)* M-Tech Public Relations; B-Tech Public Relations *(Cum laude)*; B-Tech Journalism *(Cum laude)*; *N-Dip Journalism*

07/10/2022

LINTLE TSIU

Central University of Technology (CUT)

Free State

RE: EDITING CERTIFICATE

FOCUS AREA: FLATNESS BASED COOPERATIVE CONTROL OF MULTIPLE MOBILE ROBOTIC SYSTEMS

Submitted in fulfilment of the requirements for the degree: Master of Engineering in Electrical Engineering in the Department of Electrical, Electronic and Computer Engineering

Faculty of Engineering, Built Environment and Information Technology

Central University of Technology, Free State.

This serves to confirm that this research report has been edited for clarity, language and layout.

Kind regards,

Nereshnee Govender (PhD)