6-1-2013

Design and Development of an Integrated Mobile Robot System for Use in Simple Formations

Christopher D. Brune
University of Denver

DESIGN AND DEVELOPMENT OF AN INTEGRATED MOBILE ROBOT SYSTEM FOR USE IN SIMPLE FORMATIONS

A Thesis

Presented to

the Faculty of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Christopher D. Brune

June 2013

Advisor: Matthew J. Rutherford, Ph.D.

Author: Christopher D. Brune
DESIGN AND DEVELOPMENT OF AN INTEGRATED MOBILE ROBOT SYSTEM FOR USE IN SIMPLE FORMATIONS
Advisor: Matthew J. Rutherford, Ph.D.
Degree Date: June 2013

Abstract

In recent years, formation control of autonomous unmanned vehicles has become an active area of research with its many broad applications in areas such as transportation and surveillance. The work presented in this thesis involves the design and implementation of small unmanned ground vehicles to be used in leader-follower formations. This mechatronics project involves breadth in areas of mechanical, electrical, and computer engineering design. A vehicle with a unicycle-type drive mechanism is designed in 3D CAD software and manufactured using 3D printing capabilities. The vehicle is then modeled using the unicycle kinematic equations of motion and simulated in MATLAB/Simulink. Simple motion tasks are then performed onboard the vehicle utilizing the vehicle model via software, and leader-follower formations are implemented with multiple vehicles.

Acknowledgements

I would like to offer my sincere appreciation to all of those at the University of Denver who provided me with the opportunity to complete this work. First of all, I would like to thank my advisors Dr. Matthew Rutherford and Dr. Kimon Valavanis for their careful guidance throughout my degree and for continuing to challenge me throughout my work. Additionally, I would like to thank Dr. Michail Kontitsis and my oral defense committee chair Dr. Robert Stencel for their support in this project. I am also very grateful to all of the people at DU^{2}SRI who have worked with me during the past two years: Gonçalo Martins, Allistair Moses, Konstantinos Kanistras, Jonathan Girwar-Nath, Daniele Sartori, Jessica Alvarenga-Veyna, Tanarat Dityam, Florence Mbithi, Nikos Vitzilaios, and Alessandro Benini. I have learned a lot from each of them on how to perform quality research. Finally, I would like to thank my friends and family, especially my parents, Robert and Catherine, for their endless love, support, and encouragement throughout this process.

Table of Contents

List of Tables

List of Figures

Chapter 1

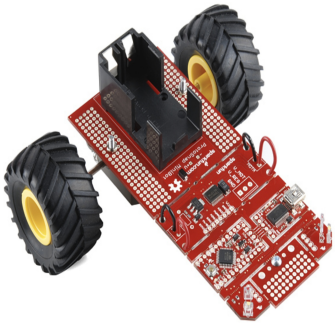
Introduction

The topic of this thesis is the study and design of mobile robots and their implementation in simple motion control and formations. There are numerous applications that can utilize formation and cooperative control in robotics. However, there still exist many technological and scientific challenges before a wide application of multi-vehicle systems becomes feasible. This chapter discusses the motivation and background behind current research in mobile robotics.

1.1 Motivation

In recent years, the study of formation and cooperative control in robotics and unmanned systems has increased in popularity due to its many broad applications [2]. Formation control of autonomous unmanned vehicles has many potential applications. In particular, formation control methods can be used for civilian applications including monitoring, searching, and rescue missions in hazardous environments [3]. It can also be used for military applications, including reconnaissance and area coverage [4]. Additionally, formations of autonomous vehicles can be used in automated highway systems [5] or other platooning

(a) Parallax Shield-Bot



(b) ProtoSnap MiniBot

(c) iRobot Roomba

Figure 1.1: Commercially Available Hilare-type Mobile Robots

applications such as transportation or mining [6]. Finally, the study of formation control in research has also helped in the understanding of biological systems, such as the swarming of insects or the flocking of birds where self-organizing and adaptive behaviors can be observed [7] [8].

1.2 Background

1.2.1 Mobile Robots

There exist two common types of mobile robots with well-designed kinematics. These are the unicycle-type and car-like mobile robots. The unicycle-type vehicle, also known as the Hilare-type mobile robot, has two independently driven wheels. These wheels act as the main drive mechanism and are commonly balanced by a passive castor wheel in front or back. In general, this type of vehicle has good maneuvering capability due to its zero minimum turn radius. It is also relatively easy to control. The drawback to this type of vehicle is that it has difficulty driving along a straight trajectory. The unicycle-type vehicle is also easier to design and manufacture due to its simple drive mechanism. Figure 1.1 shows some common unicycle-type mobile robots.

The unicycle-type mobile robot can have different shapes and sizes as can be seen in Figure 1.1. However, the mathematical modeling of such vehicles is typically similar in structure due to the simple drive mechanism that they share. In general, the mathematical models can be adjusted for any unicycle-type mobile robot by using the physical parameters that correspond to that particular vehicle, as long as the modeling assumptions are still valid. The unicycle mobile robot is commonly used in motion control and formation control research.

The other common type of mobile robot is the car-like robot. Like the name suggests, the car-like robot has a drive mechanism that is similar to that of a car. Car-like mobile robots are driven by a single motor that powers a differential. The differential then distributes torque to the rear wheels to move the vehicle. The front wheels contain a steering mechanism that is driven by a motor that generates steering angles to steer the vehicle. A group of small car-like mobile robots can be seen in Figure 1.2. The car-like mobile robot is also commonly found in motion control and formation control research.

Figure 1.2: Car-Like Mobile Robots

1.2.2 Applications

Mobile robot systems have many applications in a number of industries and military environments, and they continue to be a major area of current research in robotics and control. The focus of this thesis is on research involving small unmanned ground vehicles, or UGV's. Among the numerous applications in mobile robots, there is environment monitoring and inspection, security and surveillance, and even automated home assisting. Examples include inspecting buildings and trouble sites remotely to reduce emergency visits, map the environment, and avoid obstacles.

In order for unmanned ground vehicles to perform the applications mentioned above, they must be able to perform certain tasks. The vehicle should be able to sense the environment and obstacles, perform motion planning, and control the drive mechanism so that the physical vehicle is actually able to carry out the planned motion in reality. This thesis focuses on the physical design of small unmanned ground vehicles, motion control of unicycle-type mobile robots, and the application of vehicles in formation.

1.3 Problem Statement

In this project, we are particularly interested in the problem of existing mobile robot systems being high in cost and relatively complex. Many small mobile robot systems that currently exist are not ideal for swarm vehicle research due to their overall cost and lack of accommodation with additional sensors for different applications. As a result, there was a desire to create a small unmanned ground vehicle (UGV) to be used for multi-vehicle systems research that was more cost effective than other existing systems and relatively simple in design. There are a number of mobile robot systems that currently exist, however, most tend to be very expensive and limited by their complexity.

1.4 Contributions

This research focuses on the design and development of a small unmanned ground vehicle (UGV). A small UGV of unicycle-type is designed, and several vehicles are manufactured with focus on small size and low cost. Motion control of the mobile robots is then studied using the vehicle's kinematic model. This model is then applied in software onboard the vehicle to operate it using system model inputs. This work is then extended to the implementation of the vehicle in leader-follower formations using sensor-based feedback. The follower vehicles in formation contain a decentralized controller which maintains a particular heading and distance from the vehicle immediately in front of it, also known as the leader. The resulting formation is scalable and useful for further work to be done on formation and cooperative control at the University of Denver's Unmanned Research Institute.

1.5 Organization of Thesis

This thesis is organized as follows. Chapter 2 provides a literature review where background information on formation and cooperative control is investigated and compared with other existing methods in research. The literature review provides background information on research methods related to formation and cooperative control including behavior-based methods, virtual structure methods, and leader-follower methods. Chapter 3 describes two generations of unmanned ground vehicle that were designed and developed throughout the course of this work. Chapter 4 discusses the theory behind motion control of a unicycle-type mobile robot including kinematic modeling, trajectory tracking, and leader-follower formation control. Chapter 5 describes the simulation, implementation, and results of the designed miniature unmanned ground vehicle in various experiments performed throughout

the course of this project. Finally, Chapter 6 provides conclusions and recommendations for future work.

Chapter 2

Literature Review

Substantial research and development work has been performed in the areas of cooperative and formation control for mobile robots and other unmanned systems. Many technological challenges involving the development of formation and cooperative control strategies currently exist. These challenges can deal with certain constraints in communication, localization of robot positions, environment mapping such as simultaneous localization and mapping (SLAM), sensing, learning, and mimicking biological system behaviors such as flocking and schooling.

2.1 The Behavior-Based Method

In literature, there are three different approaches towards formation and cooperative control behavior. The first of these is the behavior-based approach, where several desired behaviors, such as formation retaining and collision avoidance are prescribed to each vehicle. The vehicle's final action is then derived by weighing the relative importance of each behavior. A major advantage to this method is that all vehicles in the formation can be controlled in a decentralized fashion. However, the mathematical analysis of this approach tends to be

difficult, and consequently it is not easy to guarantee the convergence of a formation to the desired configuration.

Arkin [4] at the Georgia Institute of Technology presents a behavior-based approach that utilizes reactive behaviors to implement formations in multi-robot teams. The formation behaviors used are integrated with navigational behaviors to allow the robot team to reach navigational goals, perform collision avoidance, and simultaneously remain in formation. The behaviors are implemented in simulation as motor schemas within the Autonomous Robot Architecture (AuRA) and as steering and speed behaviors within the Unmanned Ground Vehicle (UGV) Demo II architecture.

Xiaomin et al. introduce a behavior-based high dynamic autonomous formation and control method for use with multi-missile systems [9], which integrates the behavior-based approach with a leader-follower scheme to ensure stability in the formation. The individual behaviors prescribed to each missile include flying toward a destination, holding a formation, obstacle avoidance, and collision avoidance. The appropriate behavior is chosen by a behavior inhibition selection mechanism. The behavior of flying to a destination or leader is achieved by using an optimal route-planning algorithm, which is based on tree topologic data and dynamic programming theory. The behaviors of holding formation and obstacle/collision avoidance of followers are achieved by using a global formation-hold control algorithm and obstacle/collision avoidance control algorithm, respectively. This method is implemented in a digital simulation.

In [10], an entrapment/escorting mission is implemented using a kinematic control method based on a new type of behavior-based control known as null space-based behavioral (NSB) control. This method differs from other existing behavior-based formation control methods in the way that the outputs of the single elementary behaviors are merged to compose the final behavior. The control strategy is validated both in simulation as well as in several experimental cases, where a team of six mobile robots entraps a moving tar-

get represented by a tennis ball that is randomly pushed by hand. In addition, the control approach is made robust so that in the case of one or more vehicle failures, the system can autonomously reconfigure itself to still complete the mission.

Mead et al. in [11] present a behavior-based approach that treats the formation of robots as a type of cellular automation, where each robotic unit is treated as a cell. The robot's behavior is governed by a set of rules for changing its state with respect to its neighbors. By selecting one of the robots as an initiator, human intervention would change its state, which would propagate to its neighbors, instigating a chain reaction in behaviors from all other robots.

2.2 The Virtual Structure Method

The second approach to formation and cooperative control behavior is the virtual structure approach. In this method, the formation is considered as a virtual rigid structure such as a circle or a square. An advantage to this method is that all vehicles are mutually coupled to each other, however it is often more difficult to determine stability of the formation than other methods. Also, it is necessary for the position and control variables of each individual vehicle or the full state of the virtual structure to be communicated to each individual vehicle in the formation.

A general control strategy for the virtual structure approach is first developed in [12] where control methods force a group of robots to behave as particles that are embedded in a rigid structure. The virtual structure is first aligned with the current position of the individual robots. The virtual structure is then moved with a certain angle and distance while the individual robots determine desired trajectories that are based on the virtual structure end points. The method is tested using both simulation and experimentation with a group of three robots.

In [13], a virtual structure control method is used for a group of unicycle-type mobile robots using the dynamics of the vehicle model. A combination of path-tracking and the virtual structure approach is used with a small amount of communication between vehicles. An output feedback controller is designed for each individual robot so that a path derivative is left as a free input in order to synchronize the vehicles' motion. The proposed controller is demonstrated in simulation.

Similarly, a nonlinear formation control architecture is presented in [14], which uses a combination of both virtual structure and path following approaches. A formation controller is designed for the kinematic model of two-degree-of-freedom unicycle-type mobile robots and takes into account the physical dimensions and dynamics of the robots. The desired motion for each mobile robot is defined from the motion of the virtual structure's center while the path following problem for each mobile robot is solved individually by introducing a virtual target that propagates along the path. Coordination of the formation is achieved by synchronizing the parametrization states that capture the positions of the virtual targets with the parametrization states of the virtual structure center.

Another approach to the virtual structure method can be seen in [15] and [16] where a unified, distributed formation control architecture is proposed, which accommodates an arbitrary number of group leaders and allows for arbitrary information flow among vehicles. This method reduces overall complexity to the control law design and analysis due to an extended consensus algorithm that is applied on the group level to estimate the time-varying formation trajectory information. This is done in a distributed manner. Using the estimated formation trajectory information, a consensus-based distributed formation control strategy is then applied for vehicle level control. This proposed formation control architecture is implemented on a multi-robot platform.

In [17], a virtual structure control strategy with mutually coupling between robots is proposed. This controller is designed for nonholonomic unicycle mobile robots and is

based on the vehicle's kinematic model. Mutual coupling terms are introduced between robots to ensure robustness of the formation in the presence of disturbances. The controller design is validated using experiments with a group of two mobile robots controlled over a wireless communication network.

Additionally in [18], the virtual structure approach is again studied. In this work, the formation control of a group of nonholonomic mobile robots is achieved using implicit and parametric descriptions of the desired formation shape. The presented strategy utilizes implicit polynomial representations to generate potential fields for achieving a desired formation shape. The strategy also utilizes elliptical Fourier descriptors to maintain the formation once it has been achieved. Coordination of the vehicles is modeled by linear springs between each vehicle and its two nearest neighbors. This method is scalable to different formation sizes and is validated using simulations with robot groups of different sizes.

2.3 The Leader-Follower Method

The third approach to cooperative and formation control is the leader-follower approach. In this method, some robots are designated as leaders that follow predefined trajectories, while the remaining robots are designated as followers that follow according to a relative position or posture of the leader robot. The main advantage to this approach is that it is relatively easy to understand and implement. Also, it is often less difficult to determine stability with the leader-follower approach than with other methods such as the virtual structure approach [17]. However, this method holds a disadvantage in that there is typically no explicit feedback from the followers to the leader. For example, if a follower is disturbed, the formation cannot be maintained due to the lack of explicit follower feedback. Another disadvantage to this method is that the position of the leader must be communicated to the followers through communication or sensor feedback so that follower robots can use this

information as a control input. This makes the leader-follower more ideal for a centralized control strategy, which is less ideal for a large number of robots.

The focus of this thesis relates to the leader-follower approach with respect to small unmanned ground vehicles. In literature, most designs of formation controllers are based on the kinematic model of nonholonomic unicycle mobile robots. In [5], Desai et al. present a leader-follower formation control method based on local sensor-based information. One robot follows another by controlling relative distance and orientation between itself and the leader. This method is applicable when each follower has only one leader, which allows for the robots to follow in single file. Feedback linearization is used to provide stability to the system, and the method is demonstrated in simulation using six robots.

The idea of using feedback linearization has become common in research with mobile robots with nonholonomic constraints due to their nonlinearity. Many feedback linearization methods performed have been based on ideas provided in [19], [20], and [21]. This approach is very common in the research of formation control, particularly in leader-follower applications.

Similarly, in [22] and [6], Klančar et al. present a leader-follower method which deals with platoons of wheeled mobile robots with nonholonomic constraints. The control strategy presented addresses platooning of a group of wheeled mobile robots that rely on relative sensor information to precisely follow the path of a vehicle in front of it. The vehicles only have local distance and heading information, and there is no explicit communication between vehicles or global information such as GPS. The reference position and orientation of the follower is determined by the estimated path of the leading vehicle. The platooning control strategy presented in this work is validated experimentally using a group of small-sized mobile robots.

In [23], a second-order kinematic model for a leader-follower type of mobile robot formation is presented. The proposed model is derived in terms of relative motion states

between robots and the follower robot's local motion. Feedback linearization is then used to achieve the desired formation as well as maintain it. In the proposed control law, acceleration commands are generated and then converted into velocity inputs for the follower vehicle by multiplying by the control period. Reference velocities for the follower vehicle can then be found by the current velocity to any velocity variations. In addition, an adaptive controller is developed to add robustness to the system. The adaptive component of the design deals with parametric uncertainty while the robust control component compensates for any uncertainty with the acceleration of the leader robot. The proposed controller is demonstrated in simulation and experimentation.

Another approach to the leader-follower method is introduced in [24] where it is assumed that the desired angle between the follower and leader is measured in the follower frame instead of the leader. This method allows for smoother trajectories for the follower as well as lower control effort. Suitable leader conditions for velocity and trajectory are found so that the follower may maintain the formation and satisfy its own velocity constraints. The follower position is not rigidly fixed with respect to the leader.

In [25], a decentralized leader-follower formation controller is proposed for a group of two nonholonomic mobile robots. Using a fixed delay time, the follower tracks the leader's trajectory with a convenient separation. Only the leader's position measurement is available for the follower. Simulations are presented to demonstrate the proposed approach.

In [26], the leader-follower problem is formalized in a geometric framework where the properties of the formation are shown to affect the set of admissible curvatures of the leader's trajectory and the aperture of the cone containing the admissible positions of the follower. The proposed formation is controlled by maintaining a distance and an angle, however, an additional angle is introduced which is referred to the follower and not the leader. Simulation is used to validate the proposed control. Similarly, in [27], a leader-follower scheme is proposed that shows how the geometric properties of the formation

affect the admissible curvatures of the leader's trajectory as well as the velocity bounds of the followers. The desired angle between the leader and follower is measured in the follower frame and not in the leader frame. The proposed control strategy guarantees smoother trajectories as well as lower control effort and is demonstrated in simulation.

The leader-follower approach is related to the concept of string stability, which can be implemented as a vision-based leader-follower approach such as that presented in [28], [29], and [30] where the follower tracks the leader by means of a camera. The leader-follower approach can also be looked at as in a chained form such as that presented in [31] and [32] which is commonly used in automated highway systems.

It is also popular in literature to perform leader-follower formation control using a virtual vehicle approach. A combination of virtual vehicle and trajectory tracking is used to derive formation architecture in [33]. A virtual vehicle is driven in such a way so that it converges and stabilizes to a reference position defined by the leader. The velocity of the virtual vehicle is used to provide a control law for the follower. Position tracking control is then designed for the follower using Lyapunov and Backstepping techniques to track the virtual vehicle. The proposed approach is limited in that the leader must maintain visual line-of-sight contact with all followers. The proposed method is validated in simulation.

Another popular approach to the leader-follower method is that with limited information due to sensor limitations or noise. In [34], a distributed tracking control of leader-follower formations is proposed under partial and noisy measurements. Each follower can only measure the relative positions of its neighbors in a noisy environment. Using distributed estimators and a neighbor-based tracking protocol, the stability of the proposed closed loop tracking control strategy is shown in simulation. Similarly in [35], a leader-follower adaptive control method is proposed for nonholonomic mobile robots with limited information. An adaptive observer is designed under the assumption that the velocity measurement is not available and neural network is employed to compensate for actuator satura-

tion. Additionally in [36], a distributed nonlinear controller is proposed for leader-follower formations of nonholonomic mobile robots without global position measurements.

The leader-follower approach with obstacle avoidance has also been a popular area of research within recent years. A decentralized control scheme that achieves leader-follower formation control is proposed in [37] where a Lyapunov feedback technique guarantees trajectory tracking and obstacle avoidance for multiple nonholonomic mobile robots. Similarly, an event systems based formation control framework is presented for leader-follower formation control in [38] where all follower vehicles maintain a predetermined formation with the leading vehicle while being adaptable to obstacles in the environment. Additionally, a leader-follower formation control scheme is implemented in [39] where obstacle avoidance is introduced using the concept of fuzzy logic. Simulations are provided to demonstrate the effectiveness of the proposed approach.

Another common application in leader-follower formation control is the method of mimicking biological systems, also known as flocking. Flocking can be implemented in many ways with the same goal of achieving coordinated and cooperative motion of a group. In [40], a fault tolerant flocking algorithm is proposed to select the active mobile robots from a group. A disadvantage to this method is that the proposed algorithm only considers initially faulty robots. In [41], a flocking algorithm is proposed that uses a set of artificial potential field functions to attract desired targets and avoid obstacles. Both proposed methods are validated through computer simulation.

In this project, a decentralized leader-follower controller is implemented through the use of sensor feedback. Other methods, such as those discussed above, are based on synchronization, graph theory, artificial potentials, and generalized coordinates. In this particular work, we are specifically interested in the problem of leader-follower formation control based on distance and orientation sensor information. This differs from other techniques which perform this approach using information communicated from the leader

vehicle. Additionally, the techniques discussed above utilize a closed loop system, which takes into account system measurements from interoceptive sensors, whereas the method implemented in this project is based off of exteroceptive sensor information.

Chapter 3

System Design

The mobile robot platform designed and used in this research can be divided into two generations. The first generation design was used to demonstrate the feasibility of designing such a vehicle and implementing basic motion control applications such as radio and wifi controlled motion. The second generation system builds upon the first generation through a revised bumper system design and the addition of a second level top plate that allows more space for additional sensors and/or actuators.

3.1 Existing Systems

With regards to mobile robot systems, a number of alternative designs currently exist. Many commercially available systems come in different shapes and sizes, which may or may not be convenient for the addition of sensors and actuators for other applications. Table 3.1 describes several of the more popular choices that are commercially available.

As can be seen from the table, many existing systems are available in different shapes and sizes. These systems also have a relatively high cost for a platform that does not necessarily include a microcontroller to perform software development with. An example

Vehicle	Microcontroller	Dimensions (mm)	Price
Parallax Robotics Shield Kit	No	102 x 77.5	129.99
ProtoSnap Minibot Kit	No	50.8 x 102	74.95
Picaxe Robot	Yes	200 x 290	74.95
Rover 5 Robot Platform	No	225 x 245	59.95
Pololu 3pi Robot	Yes	95 (diameter)	99.95

Table 3.1: Existing Mobile Robot Systems

of this would be the Parallax Robotics Shield Kit, which has the highest cost of the systems shown in Table 3.1 and does not contain an Arduino microcontroller.

3.2 UGV Design Considerations

The initial goal of this project was to design and manufacture a small unmanned ground vehicle for use in an indoor environment or in areas where positioning information, such as GPS, is initially unavailable. A unicycle-type mobile robot design was chosen due to its simple drive mechanism and minimal cost of components. The unicycle design is also each to replicate. The unicycle-type vehicle utilizes two independent motors, each of which power one of the vehicle's wheels. In addition, there is a passive castor wheel that balances the vehicle.

Components
2 Continuous Rotation Servo Motors
2 Wheels with grips
Chassis
Ball Castor
RCRX
XMOS XK-1A Microcontroller Board

Table 3.2: Initial Vehicle Components

Initial design considerations of the UGV design specified that the vehicle should consist of several components. These primary vehicle components can be seen in Table 3.2. Two

continuous rotation servo motors with wheels, along with a chassis and ball castor, were used to create the unicycle-type drive mechanism on the vehicle. Additionally, other design considerations for the vehicle included those that can be seen in Table 3.3, which could be used for other applications such as motion control or localization.

Components
Breakout Board
Sonar (4 directions)
Laser Range Finder
Xbee/WiFly Wireless Receiver
Data Logger
Indoor ”GPS” Module

Table 3.3: Additional Nonessential Vehicle Components

Three motion control applications were considered when developing the vehicle. Initially, it was desired to obtain manual motion control of the vehicle through the use of a radio control device. The radio controller device used for this application was the Spectrum DX6i. After achieving this relatively simple goal, it was desired to obtain manual motion control of the vehicle through the XMOS XK-1A microcontroller board onboard the vehicle. A Futaba 319DPS radio receiver was interfaced with the XMOS XK-1A development board, and pulse-width modulation (PWM) commands generated based on the signals received from the Spectrum DX6i were transmitted to the servo motors to drive the vehicle. This application involved a simple software task that created the unicycle-type drive mechanism. Achievement of both of these applications validated the basic operation of the first generation vehicle design. It also served as a milestone in eventually achieving the third motion control application, which involved fully autonomous pre-planned vehicle motion control and is the basis for this thesis.

3.2.1 XMOS Microcontroller

The XMOS XK-1A microcontroller development board was used as the main embedded controller for this project. The XMOS XK-1A is a low cost development board which is based on a single XS-L1 device. It consists of a single XCore, which comprises an event-driven multi-threaded processor with general purpose I/O pins and 64 KBytes of on-chip RAM. In addition, the device consists of two XSYS 20-way IDC headers (one female and one male), an SPI interface to FLASH memory, a 20 MHz crystal oscillator, and two expansion areas with 12 I/O pins each [42]. Programming the device can be done through the use of a JTAG device known as an XTAG that has a USB interface. The device can be programmed with XC programming language in an Eclipse-based environment known as XMOS Development Environment (XDE), which is supplied by XMOS. Figure 3.1 shows the XK-1A microcontroller development board and XTAG interface.

XMOS technology was chosen for the development of the miniature UGV due to its numerous advantages over current processors used in intelligent, unmanned autonomous systems. The XMOS microcontroller utilizes an event-driven parallel processor, which unifies high-level processing with low-level control to overcome the complexity of handling multiple I/O streams while simultaneously performing complex computational tasks required for intelligent behavior [43]. However, the XMOS device also contains a number of disadvantages as well. For example, the small amount of memory that the device carries makes it not very applicable for vision-based control applications.

3.2.2 Servo Motors

Another critical component to the vehicle design is the servo motor. The miniature UGV consists of two standard analog servo motors. The servo motors were modified to be continuous rotation so that they could be used for the unicycle-type drive mechanism. The

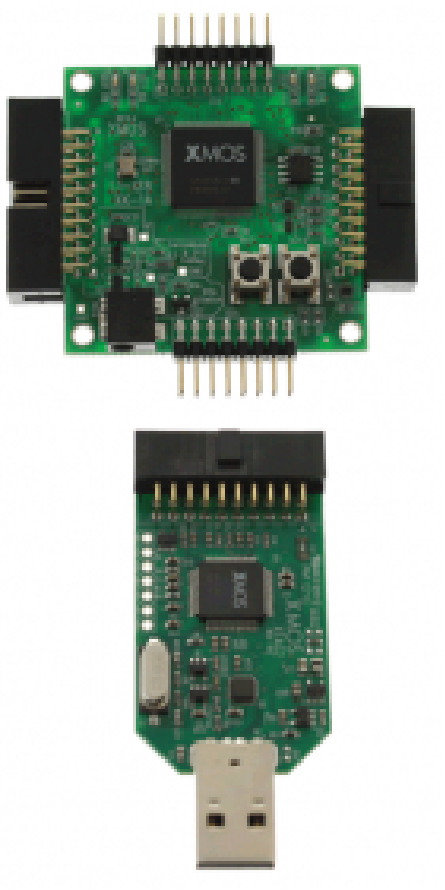


Figure 3.1: XMOS XK-1A Development Board and XTAG

reason for this was primarily due to the reduced cost of standard servo motors over traditional continuous rotation motors. The servo motor used in the miniature UGV is the HK15138 Standard Analog Servo motor from Hobby King (see Figure 3.2). This servo motor can be easily operated with a voltage of 4.8 V to 6 V. The motor provides a torque of 3.8 kg to 4.3 kg and a maximum speed of 0.21 seconds per 60 degrees to 0.17 seconds per 60 degrees.

Figure 3.2: Hobby King HK15138 Standard Analog Servo Motor

3.3 First Generation Vehicle Design

The first generation miniature UGV design was developed using SolidWorks 3D CAD design software. A basic drawing of the first design revision can be seen in Figure 3.3. A simple prototype, as seen in Figure 3.4, was designed in the University of Denver's Unmanned Systems Research Institute and manufactured in the University of Denver's machine shop. The chassis was designed in SolidWorks and cut with aluminum to assemble the unicycle-type drive mechanism with two continuous rotation servo motors and a ball castor for initial proof of concept. This first vehicle design measured at 75mm by 142mm and contained only holes for the two servo motors and standoffs for the XMOS microcontroller.

The prototype above in Figure 3.4 was used to perform manual motion control of the vehicle using radio control both directly through a radio receiver and subsequently through an interface with the XMOS XK-1A microcontroller development board. Following the verification of manual motion control, the vehicle design was then modified in SolidWorks to be manufactured using 3D printing. In particular, the design was made to be more durable for the plastic material it was to be printed with. As a result, the overall thickness of the 3D printed chassis was increased to 3 mm due to the constraints of the 3D printer that

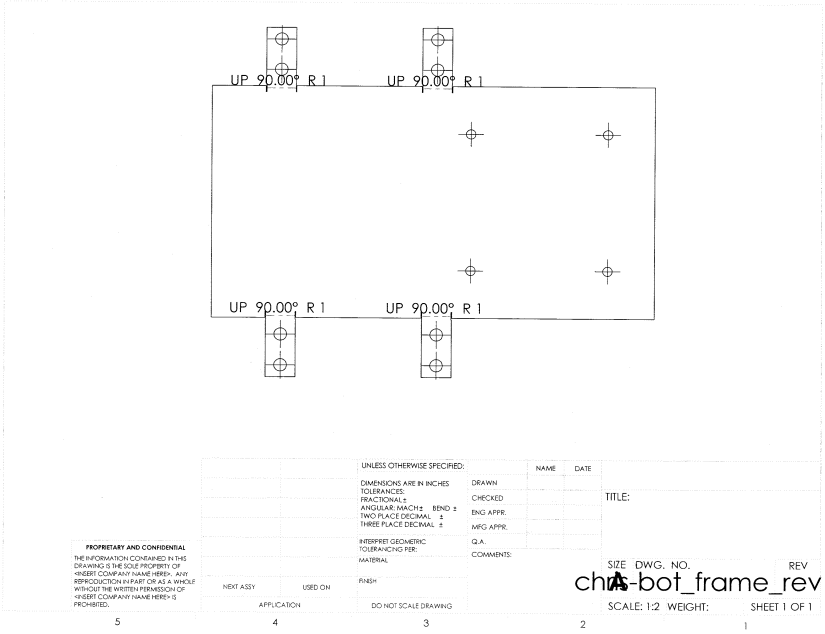


Figure 3.3: Vehicle Drawing of Miniature UGV, rev. 1

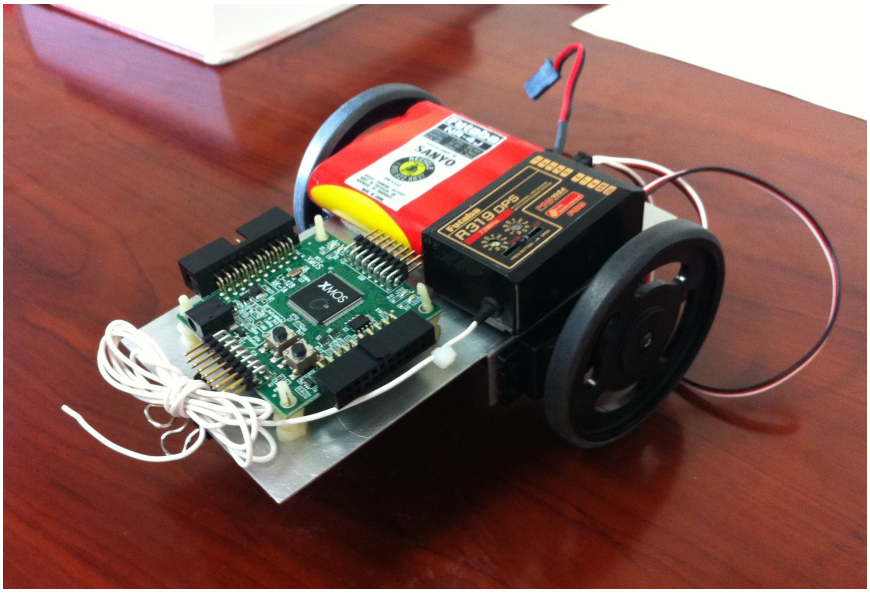


Figure 3.4: Initial Prototype of First Generation Miniature UGV

was used to develop it. Additionally, the vehicle chassis was designed to accommodate the size of a standard analog servo motor. This allows for any brand of standard servo motor to be installed on the vehicle quite easily. The servo motors onboard the miniature UGV can

be driven with pulse-width modulation (PWM) signals. These signals vary in duty cycle between 1.0 ms and 2.0 ms and are used as the inputs to the system to control the velocity of the wheels through software implemented on the XMOS microcontroller.

The first generation SolidWorks design can be seen in Figure 3.5. The resulting printed chassis can be seen in Figures 3.6 and 3.7 and was used to assemble the first generation of the miniature UGV that contained all other necessary components.

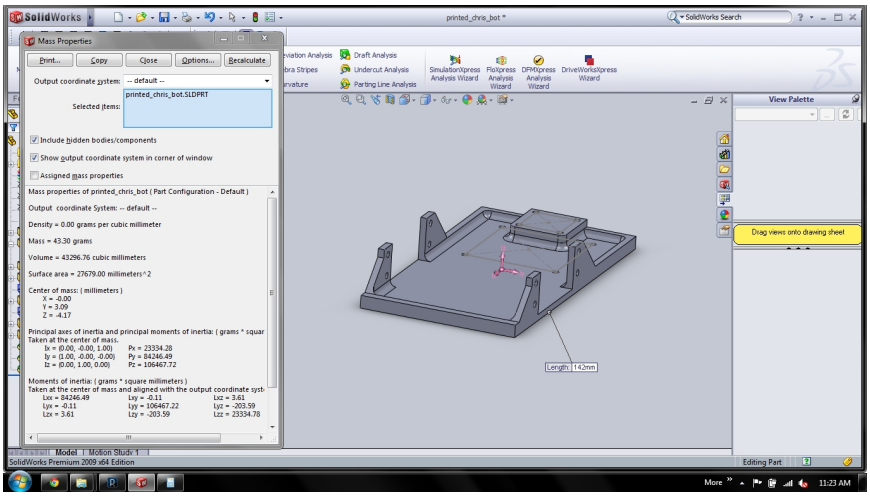


Figure 3.5: Miniature UGV SolidWorks Design, Version 1

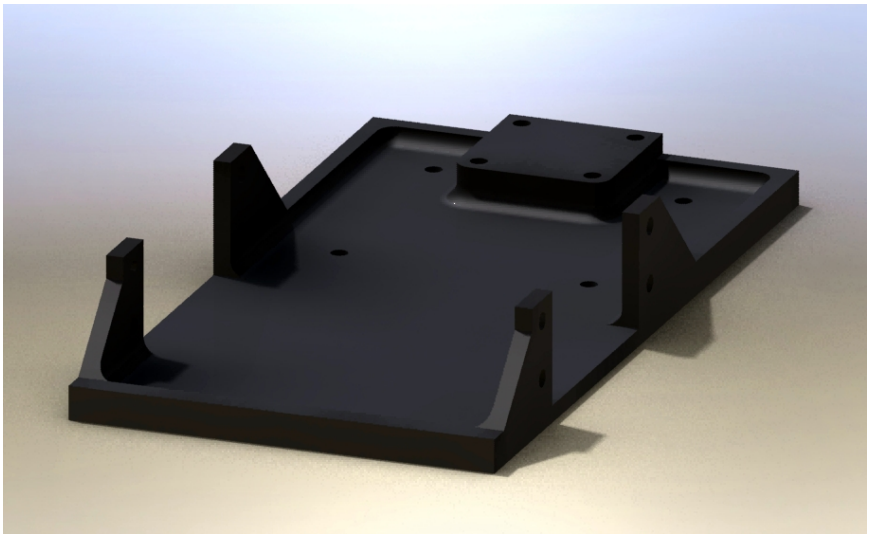


Figure 3.6: 3D Printed Chassis

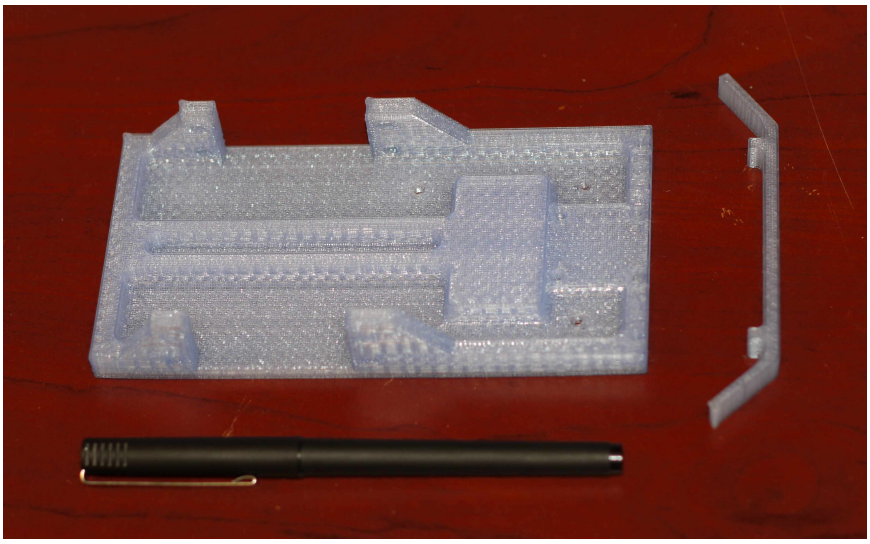


Figure 3.7: 3D Printed Chassis and Bumper

The fully assembled miniature UGV can be seen in Figure 3.8. The vehicle consists of a XMOS XK-1A microcontroller board, breakout printed circuit board, 5 V battery pack, continuous rotation servo motors with wheels, tactile bumper sensors, and compass module. A total of 15 vehicles were initially assembled at the University of Denver's Unmanned Systems Research Institute and can be seen in Figure 3.9. This initial group of vehicles was manufactured and used for the University of Denver's Embedded Systems Programming Course, ENCE 4800. Applications performed in the course involved the vehicle using Breadth First Search and A* Star Search algorithms to navigate through a maze. Additionally, the vehicle was interfaced with tactile bumper sensors and a compass to perform simple navigational and obstacle avoidance tasks.

Several faults in the first generation vehicle design were discovered over the course of the vehicle's extensive use in the University of Denver's Embedded Systems Programming class during Spring 2012. The most immediate fault was easily found in the vehicle's bumper mechanism. The initial bumper system was attached to the vehicle with epoxy that was applied to two tactile bumper switch sensors made from aluminum. This simple design

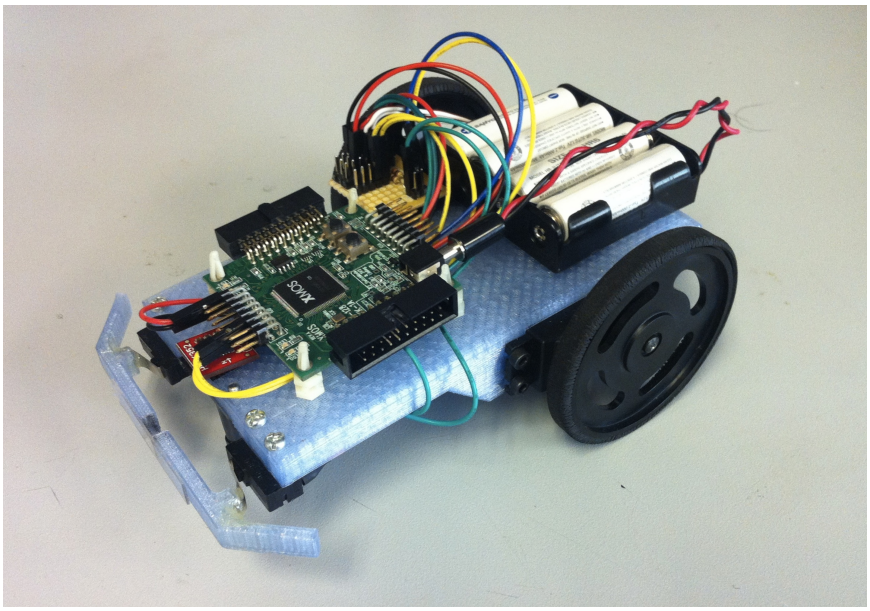


Figure 3.8: Fully Assembled First Generation Vehicle

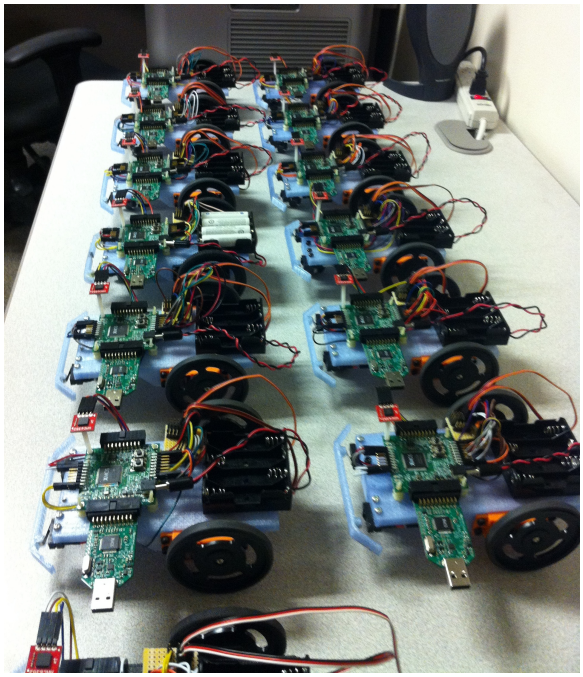


Figure 3.9: Initial Fleet of 3D Printed Miniature UGV's

was found to lack durability when the vehicle experienced a collision. Other faults with the design related to the lack of space on the vehicle for additional sensors.

3.4 Second Generation Vehicle Design

The second generation miniature UGV design, seen in Figure 3.10, makes several improvements over the first generation. The two major revisions involved the vehicle chassis and bumper system. These revisions were designed to improve the faults discovered with the first generation vehicle design as well as reduce overall cost of the vehicle.

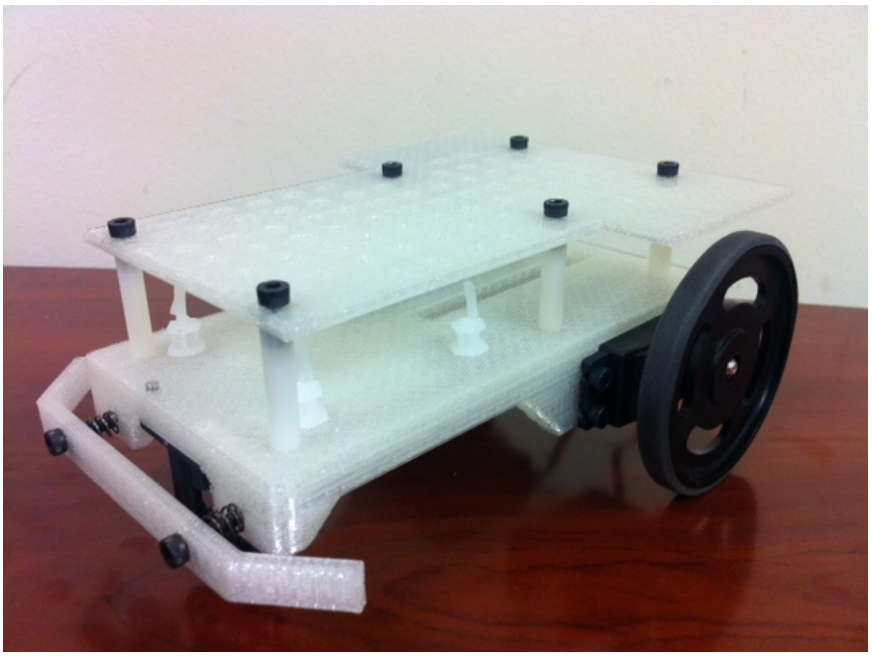


Figure 3.10: Assembled Second Generation Vehicle

3.4.1 Chassis Design

The main improvement on the second generation vehicle was a redesign of the vehicle's chassis to incorporate a top plate to allow for more space on the vehicle as seen in Figures 3.13 and 3.14. This involved adding additional standoffs to the main chassis design in Solidworks and designing a separate top plate to be placed on them. Also, the mounting location of the servo motors was moved back to create more room underneath the vehicle for a battery pack as can be seen in Figure 3.15.

Figure 3.11: Second Generation Vehicle with Laser Range Finder

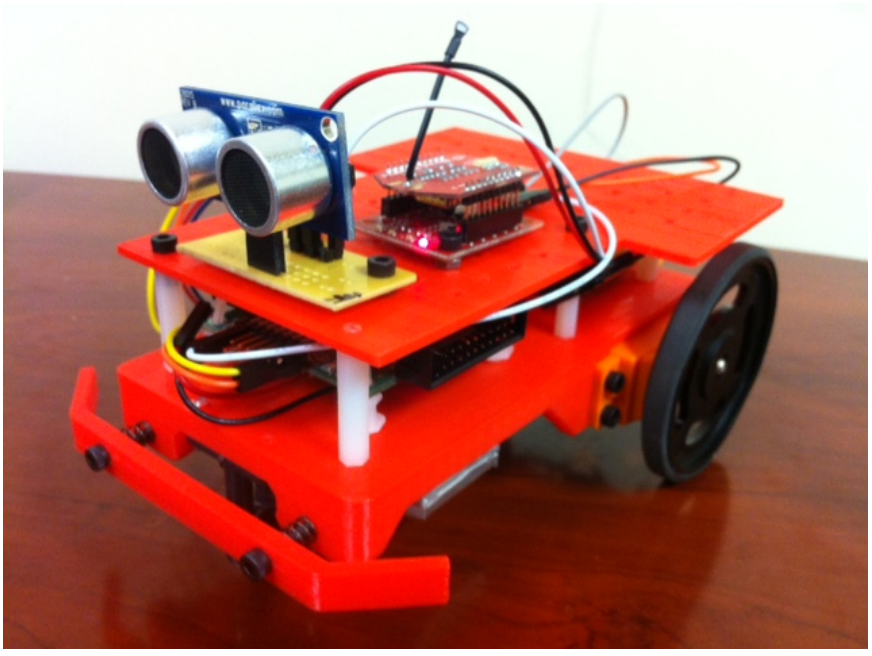


Figure 3.12: Second Generation Vehicle with Sonar, Compass, and WiFly Antenna

The second generation chassis dimensions are 75 mm by 142 mm, which is the same as the previous generation vehicle. There was no significant increase to the second generation chassis due to the size constraints on the 3D printer that was used to develop it. The standoff holes are spaced in a 54 mm by 122 mm rectangular configuration along the length of the vehicle, and the standoffs lift the vehicle top plate 25 mm above the chassis. This additional height gives the top plate clearance over the vehicle's wheels so that there is additional space for many different applications.

Overall, the second generation vehicle was designed to accommodate for a number of different sensors and actuators for different research applications. Figures 3.11 and 3.12 above show two different configurations of the second generation vehicle design for different types of sensors. These particular vehicles are the ones primarily used in this work.

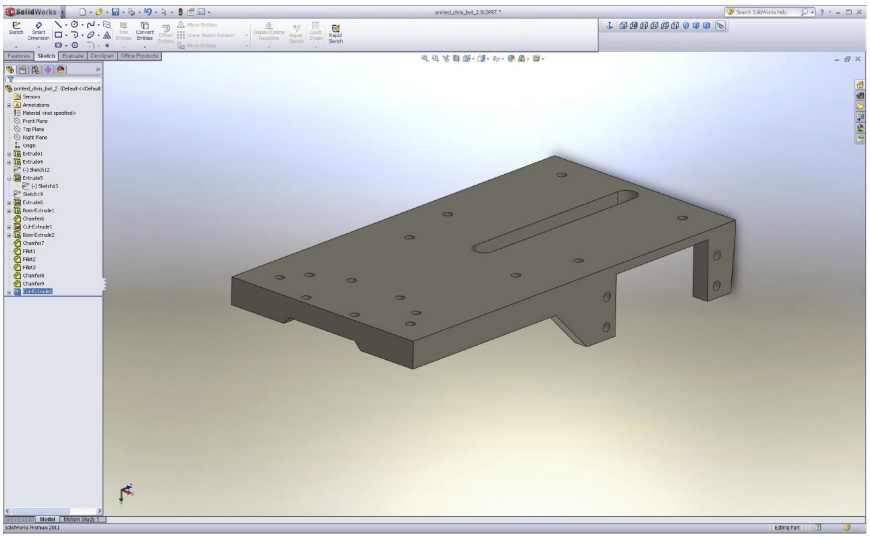


Figure 3.13: Chassis Re-Design with Standoff Holes

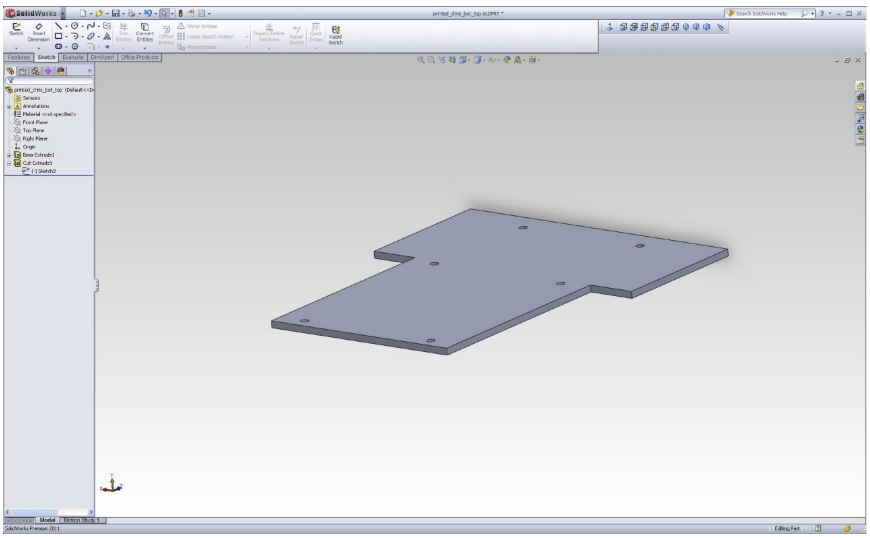


Figure 3.14: Chassis Top Plate

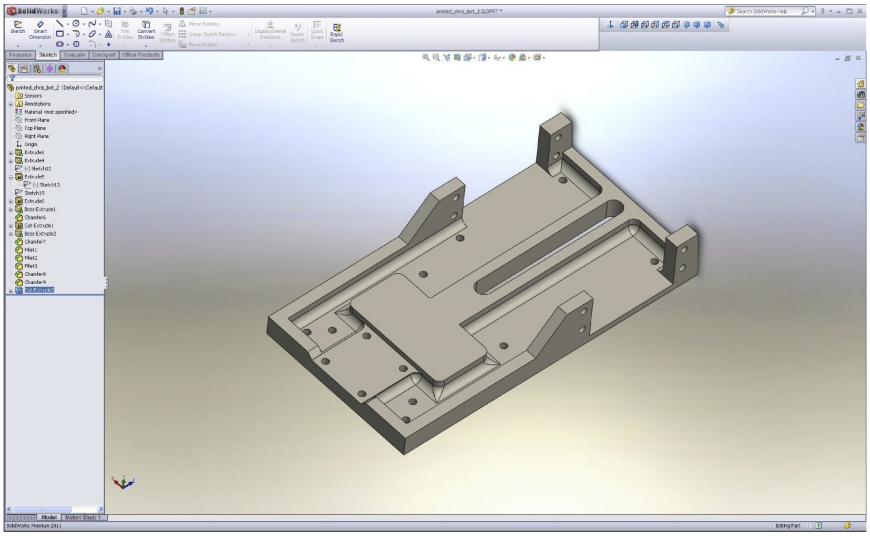


Figure 3.15: Chassis Re-Design with Adjusted Servo Mount

3.4.2 Bumper Design

Additionally, a new bumper system was designed for the vehicle to add durability for collision detection during forward motion. The main problem with the original bumper design was that it was not durable enough to withstand a large number of collisions during regular use. This was primarily due to the original bumper being attached to aluminum tactile

bumper switch sensors with epoxy. Figure 3.16 shows the original bumper design, which could easily be detached from the vehicle with minimal force when small collisions occurred. As a result of the poor performance of the original bumpers, there was a desire on the second generation vehicle design to incorporate a more durable bumper system that could withstand all collisions. The redesigned bumper system can be seen in Figures 3.17 and 3.18.

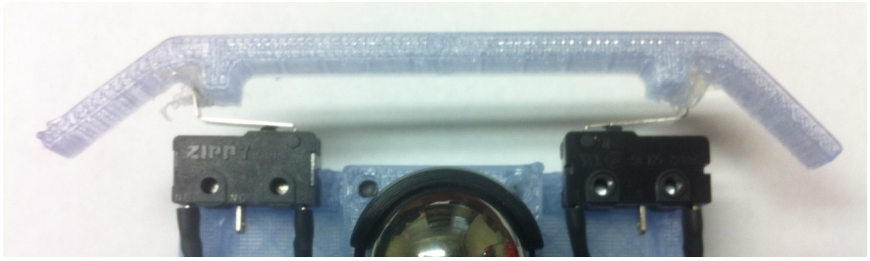


Figure 3.16: Original Bumper Design

Figure 3.17: Second Generation Bumper Design

The second generation bumper system design is comprised of a 3 mm thick 3D printed plastic bumper that was designed in Solidworks. This bumper design differed from the original with two M3 screw holes that were used to attach the bumper to the vehicle with two 20 mm long M3 screws. Using 9.4 mm compression springs, the M3 screws are mounted on the front of the vehicle so that compression may occur when the vehicle experiences front side collisions. When the bumper compresses, a tactile bumper switch sensor is compressed sending the signal of a collision to the microcontroller.

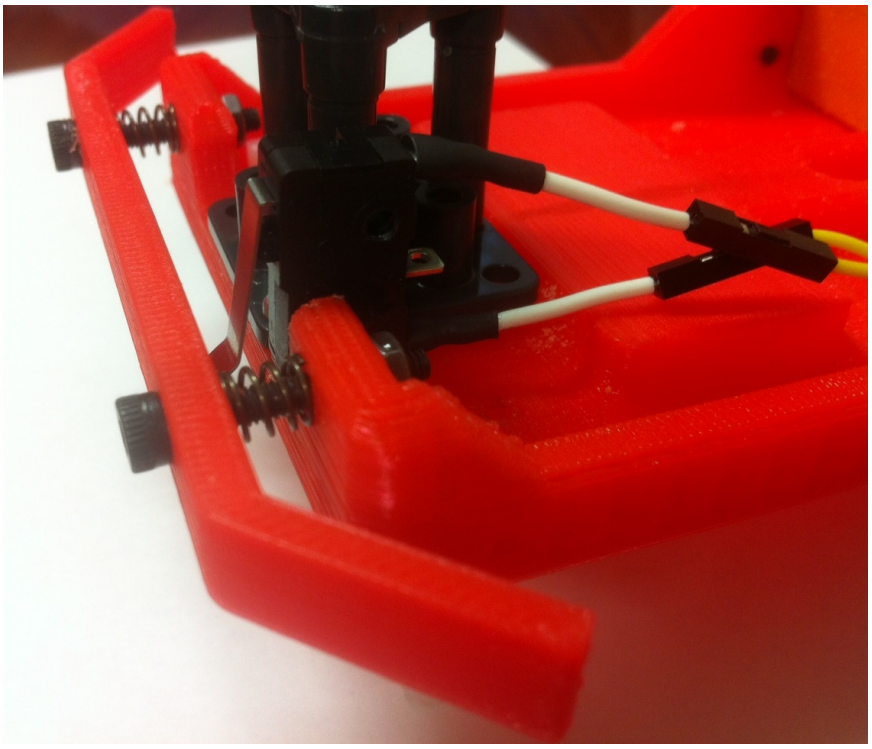


Figure 3.18: Second Generation Bumper Design (angled view)

3.4.3 Power Budget

For the particular application in this project, it is necessary for a power source to be chosen that can allow the vehicle to operate normally for an acceptable amount of time. It is desired that a battery be chosen so that the vehicle may operate for a long enough period in order to perform experiments in a number of different applications. Batteries typically used in robotics include alkaline, fuel cell, lead acid, lithium, NiCad, and NiMH. The battery that was available and chosen for this project was a 11.1 V Lithium-Polymer Battery with 730 mAh. A power budget was made to determine the amount of battery life available with the vehicle for this project.

The power budget presented in Table 3.4 is useful in determining the total power required to operate the mobile robot system. This table shows the maximum current draw of all sensors and actuators onboard the vehicle designed above. The total current of the

Component	Voltage	Maximum Current Draw
XMOS XK-1A	5 V	200 mA
Standard Servo	5 V	200 mA
Standard Servo	5 V	200 mA
Hokuyo URG-04LX	5 V	800 mA
Honeywell Compass	3.3 V	10 mA
Data Logger	3.3 V	6 mA
Roving Networks WiFly	3.3 V	180 mA

Table 3.4: Vehicle Power Budget

system with all components is 1596 mA, or 1.596 Amps. This information allows us to find the approximately battery life of the vehicle, which is approximately 27 minutes. This means that with all components drawing maximum current, the vehicle's battery would only last this much time.

3.5 Cost Analysis

A small unmanned ground vehicle was designed with focus on low cost and simplicity in design. As discussed above, many small mobile robot systems exist, which are relatively high in cost and lack components or space required for swarm behavior research. Table 3.5 shows the overall cost of parts that are required to build the small UGV designed in this project.

As can be seen from the vehicle parts list in Table 3.5, the total cost to build one of the second generation small UGV's discussed in this chapter is $32.01. It is important to note that this vehicle cost does not include the cost of the XMOS microcontroller, which is sold by XMOS for $59.00. This total cost, excluding the XMOS microcontroller, is significantly less than the other existing mobile robot systems described in Table 3.1. It is also important to note that some of these systems do not include a microcontroller. Also, the systems that do contain a microcontroller have one that is significantly cheaper than the

Part	Count/Vehicle	Price/Unit ($)	Total
Standard Servo	2	3.12	6.24
Wheel w/ Band	2	3.5	7.00
Ball Caster Kit	0.5	6.00	3.00
Vehicle Material	1	2.66	2.66
Jumper Wires	1	2.50	2.50
AA Battery Pack	1	1.19	1.19
AA Battery	2	0.68	1.36
25 mm M3 Standoff	4	0.45	1.80
8 mm M3 Screws	12	0.04	0.48
20 mm M3 Screws	2	0.06	0.12
M3 Nuts	2	0.01	0.02
9.4 mm Spring	2	2.07	4.14
PCB Standoff	1	1.50	1.50
			$32.01

Table 3.5: Vehicle Parts List

overall cost of the system. The vehicle created in this project is relatively simple in design and is significantly cheaper than other existing small mobile robot systems.

It is also possible to reduce the cost of the vehicle further by purchasing components in bulk and utilizing cheaper materials. For example, the ball caster kit used onboard the vehicle cost $6.00. However, if this component were to be ordered in a large quantity for the manufacturing of many vehicles, the cost could be reduced to $5.63. This applies to many of the components that are listed in Table 3.5 above.

Additionally, as mentioned above, some of the existing systems listed in Table 3.1 include microcontrollers with the mobile robot. However, the cost of these microcontrollers is significantly cheaper than a microcontroller such as the XMOS XK-1A used in this project. Additionally, these other microcontrollers do not necessarily have the ability to operate and perform multiple simultaneous tasks that are required for autonomous unmanned ground systems. For example, the 3pi mobile robot system from Pololu Robotics includes an ATmega328P microcontroller in its $99.95 price. The ATmega328P microcontroller can be purchased for as little as $2.24 from sources such as Mouser Electronics whereas the

XMOS XK-1A costs $59.00 from XMOS. This shows that the cost of these existing systems excluding any microcontroller or processor is significantly more expensive than the cost to produce the vehicle used in this project.

Chapter 4

Vehicle Kinematics and Control

This chapter discusses the modeling and control for the miniature UGV design of unicycle-type presented in Chapter 3. The notion of nonholonomic systems is presented, and the kinematic model of the vehicle is derived. Motion control is applied using feedback linearization techniques, and control is further developed for application in formations utilizing the Leader-Follower formation method. The following text and figures presented in this chapter are adopted from [1] and [44].

4.1 Kinematic Model

The unicycle mobile robot is a vehicle with a simple drive mechanism that is commonly used for indoor applications in current robotics and controls research. Figure 4.1 shows the schematic of a unicycle type vehicle. This type of mobile robot is the primary vehicle used in this work and was the basis for the miniature UGV design presented in Chapter 3. The drive mechanism uses two independent motors, each of which power one of the vehicle's wheels. The kinematic inputs that drive the vehicle as well as affect its speed and direction of motion are the two individual wheel velocities. However, it is convenient in most ap-

plications to choose the linear and angular velocity of the vehicle as a whole as the inputs to the kinematic model. The reason for this is that most commercially available mobile robots, including those seen in Figure 1.1, utilize a low level controller which controls both the linear and angular velocities of the vehicle.

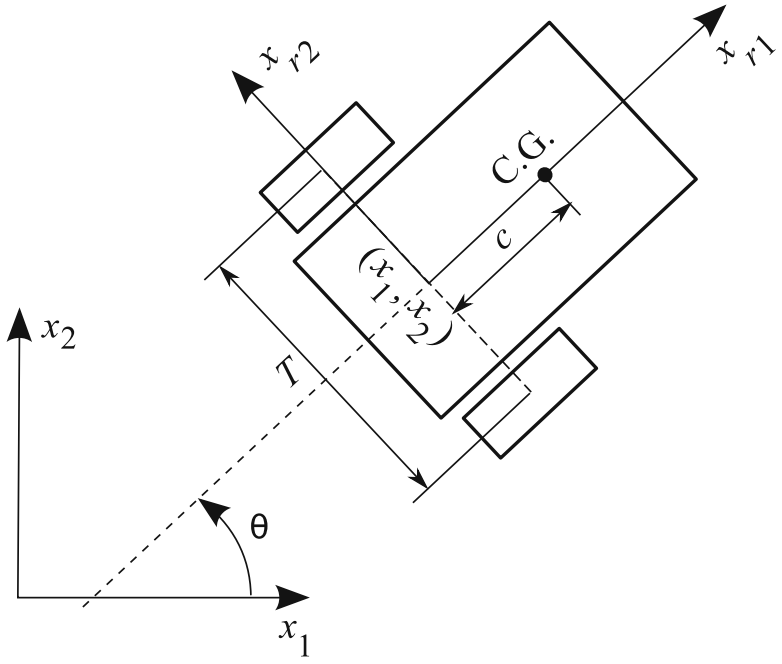


Figure 4.1: Schematic of Unicycle-type Mobile Robot [1]

In Figure 4.1, it is assumed that there is no wheel slipping between the vehicle's wheels and the ground surface while the vehicle is in motion. In other words, it is assumed that the vehicle is moving at a relatively slow speed such that the lateral force and longitudinal traction exerted on the wheels does not exceed the maximum static friction between the wheels and the ground surface in both the lateral and longitudinal directions. As a result of this assumption, the individual velocities of the wheels do not have any lateral components. This means that one can assume that the velocity of the midpoint between the two wheels at (x_1, x_2) does not contain any lateral component and is therefore parallel with the plane of the wheels. This constraint with respect to no lateral velocity with the wheels is known

as the nonholonomic constraint. One can also relate the center point of the wheels (x_1, x_2) with rotational velocity of the wheels.

The configuration variables of the unicycle-type vehicle must be defined before the kinematic equations can be defined. Let the coordinates of the point (x_1, x_2) be the global position of the vehicle with respect to the inertial coordinate system x_1 and x_2. The line in Figure 4.1 that is perpendicular to the wheel axis and goes through the midpoint between the wheels (x_1, x_2) can be considered an orientation reference for the vehicle. The angle that this line makes with the positive x_1 axis is called θ and represents the orientation of the vehicle. Equation 4.1 shows the kinematic equations of motion for the unicycle-type nonholonomic vehicle in matrix form.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4.1}$$

As can be seen from the kinematic model, the three variables that define the geometric configuration of the robot are x_1, x_2, and θ. In addition to the no wheel slip assumption, it can also be assumed that the vehicle at center wheel axis point (x_1, x_2) moves with a linear velocity of v and an angular velocity of ω. These velocities are assumed to be the system inputs and are presented in an input vector in the kinematic model. Equation 4.1 can be numerically integrated to predict the motion of the robot if the input vector is known as a function of time. Different inputs may also be chosen for the system such as rotational velocities of the wheels or the difference between linear velocities of the wheels, however, the linear velocity v and angular velocity ω of the vehicle as a whole are chosen in this work due to the simplicity and well known documentation of this method.

4.1.1 Discrete Time Kinematic Model and Control

The motion tasks performed in this project are implemented on the XMOS processor in discrete time. Therefore, it is necessary to discretize the kinematic model shown in 4.1 and 4.3. If we consider a time interval of Δt, a sampling instant of k, and apply Euler's approximation to the kinematic model of equations in 4.3, we can obtain the following discrete-time model for the unicycle type vehicle shown in 4.2.

$$\begin{aligned} x_1(k+1) &= x_1(k) + v(k)\cos(\theta(k))\Delta t \\ x_2(k+1) &= x_2(k) + v(k)\sin(\theta(k))\Delta t \\ \theta(k+1) &= \theta(k) + \omega(k)\Delta t \end{aligned} \tag{4.2}$$

Equation 4.2 shows the discretization of the unicycle kinematic equations of motion which is implemented on the vehicle in this project. The control inputs of this system of equations are $v(k)$ and $\omega(k)$, which are velocity and angular heading rate of change, respectively. These control inputs are constant over the time interval Δt. In this particular work, predefined trajectories on a finite time horizon are implemented on the vehicle by choosing pairs of v and ω. Each pair of linear and angular velocities is executed on the vehicle and held for a predetermined amount of time. This predetermined amount of time, Δt, can vary with each command and can be chosen such that the vehicle traverses along the desired trajectory.

With regards to the leader-follower formation control implemented in this project, the trajectory, or array of linear and angular velocity commands of the follower vehicles, is produced by a PID control task in software based on sensor feedback. This task is performed over a given time horizon, in this particular case the length of the leader vehicle's trajectory. Each time interval Δt has a constant linear velocity v and angular velocity ω as-

sociated with it. The vehicle path becomes a smooth collection of arcs which demonstrates how the unicycle-type vehicle is unable to travel precisely along a straight trajectory. In this case, command durations Δt are 100 ms, which was determined by the frequency of the laser range finder's sweep. This specific result is discussed in more detail in Chapter 5.

4.2 Trajectory Tracking Control

A common topic in robotics and controls research is trajectory tracking for the nonholonomic unicycle type vehicle. A trajectory tracking controller is useful for robots to be able to follow a planned trajectory. In this section, kinematic trajectory tracking control is presented based on the unicycle-type vehicle. This topic is discussed further in [45] and [1]. The control inputs for the unicycle-type vehicle are assumed to be v and ω as defined above in the previous section. Once again the kinematic equations of motion are written in Equation 4.3.

$$\begin{aligned} \dot{x_1} &= v\cos\theta \\ \dot{x_2} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{4.3}$$

If we assume that the desired trajectory of the vehicle is in the inertial coordinate system and can be defined as functions of time, then two position components $x_1^d(t)$ and $x_2^d(t)$ can be selected as the vehicle's desired positions. The desired velocity components of the vehicle can be derived by differentiating these position components. As a result, we get $\dot{x_1^d}(t)$ and $\dot{x_2^d}(t)$ through differentiation. These velocities are functions of time, which means that the vehicle must follow a desired speed that has been pre-defined. The desired

positions of the vehicle, $x_1^d(t)$ and $x_2^d(t)$, contain information about the desired velocity as well as the geometry of the desired path.

Once the desired velocity components are obtained in $\dot{x_1^d}(t)$ and $\dot{x_2^d}(t)$, the desired heading or orientation of the vehicle must be found. Due to the nonholonomic constraint of the unicycle-type vehicle, there are limitations in the desired orientation that can be selected. The desired orientation must adhere to the vehicle's lateral no-slip condition. In other words, the vehicle has a zero lateral velocity at all times. The nonholonomic constraint of this particular type of vehicle can be conveyed in Equation 4.4. This constraint is well known in literature on control applications of mobile robots or unmanned ground vehicles.

$$\dot{x_2^r} = -\dot{x_1}\sin\theta + \dot{x_2}\cos\theta = 0 \tag{4.4}$$

Once the desired velocity components are obtained, the desired heading or orientation can be derived. This is shown below in Equation 4.5.

$$\theta^d(t) = \arctan\frac{\dot{x_2^d}(t)}{\dot{x_1^d}(t)} \tag{4.5}$$

A common technique utilized in current research literature is to reduced the system to chained form. A new set of configuration variables can be found that can simplify the kinematic model of the unicycle-type mobile robot. If we observe the kinematic model shown in Equation 4.1, the chained form configuration variables can be written as those seen in Equation 4.6.

$$\begin{aligned} z_1 &= x_1 \\ z_2 &= \tan\theta \\ z_3 &= x_2 \end{aligned} \tag{4.6}$$

The chained form is well known in literature pertaining to mobile robots and unmanned ground systems. The chained form control inputs to the kinematic model are shown in Equation 4.7.

$$\begin{aligned} u_1 &= v\cos\theta \\ u_2 &= \omega(1+\tan^2\theta) \end{aligned} \tag{4.7}$$

If we apply these new variables to the original kinematic equations shown in Equation 4.3, then we get a new chained form for the kinematic equations of motion for the unicycle-type vehicle. This chained form can be seen in Equation 4.8. This chain form method can be extended for different vehicle types with different drive mechanisms and configuration variables as well.

$$\begin{aligned} z_1 &= u_1 \\ z_2 &= u_2 \\ z_3 &= z_2 u_1 \end{aligned} \tag{4.8}$$

Once we have the chained form kinematic model, we can find the desired trajectory for the variables shown in Equation 4.6. This is done by substituting the desired position components into Equation 4.6. The control inputs can then be found by substituting Equation 4.6 into Equation 4.8. The new desired trajectory can be seen in Equation 4.9.

$$
\begin{aligned}
z_1^d(t) &= x_1^d(t) \\
z_2^d(t) &= \frac{\dot{x_2^d}(t)}{\dot{x_1^d}(t)} \\
z_3^d(t) &= x_2^d(t)
\end{aligned}
\tag{4.9}
$$

It is desired to control the chained form kinematic equations shown in Equation 4.8 to follow the new trajectory shown in Equation 4.9. The error from this application can be derived simply by subtracting the desired configuration variables from the actual configuration variables. The error equations are nonlinear and can be seen in Equation 4.10 where $\tilde{z_i} = z_i - z_i^d$ for $i = 1, 2, 3$ and $\tilde{u_i} = u_i - u_i^d$ for $i = 1, 2$.

$$
\begin{aligned}
\dot{\tilde{z_1}} &= \tilde{u_1} \\
\dot{\tilde{z_2}} &= \tilde{u_2} \\
\dot{\tilde{z_3}} &= z_2^d \tilde{u_1} + \tilde{z_2} u_1^d + \tilde{z_2} \tilde{u_1}
\end{aligned}
\tag{4.10}
$$

The first step that must be taken is to linearize the nonlinear system of equations. A common technique in current research practices with almost all nonlinear systems is to use the method of feedback linearization. The general idea of this method is to algebraically transform nonlinear systems into linear ones so that linear control techniques can be applied. This method is chosen over the other common method of linearization, known as Jacobian linearization, which for the unicycle kinematic model results in an unstable system [45]. To do this, $\tilde{z_2}\tilde{u_1}$ is neglected and the time-variant feedback control law shown in Equation 4.11 is assumed where k_1, k_2, and k_3 are constant controller gains.

$$\begin{bmatrix} \tilde{u_1} \\ \tilde{u_2} \end{bmatrix} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & k_3/u_1^d \end{bmatrix} \begin{bmatrix} \tilde{z_1} \\ \tilde{z_2} \\ \tilde{z_3} \end{bmatrix} \tag{4.11}$$

The control law in Equation 4.11 is then applied to the linear state error equations shown in Equation 4.10.

$$\begin{bmatrix} \dot{\tilde{z}_1} \\ \dot{\tilde{z}_2} \\ \dot{\tilde{z}_3} \end{bmatrix} = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & k_3/u_1^d \\ k_1 z_1^d & u_1^d & 0 \end{bmatrix} \tag{4.12}$$

This gives us the linear time variant closed loop system, which can be seen below in Equation 4.12. The linear time variant closed loop system in Equation 4.12 is time dependent, however, its characteristic equation is not time dependent. Therefore, one can select controller gains such for the linearized system. The controller gains suggested in [1] are $k_1 = -\lambda_1$, $k_2 = -2\lambda_2$, and $k_3 = -(\lambda_2^2 + \lambda_3^2)$ where λ_1 and λ_2 are positive constant values. With these selected gains, the closed loop system becomes asymptotically stable. As a result, the poles of the system become $-\lambda_1$, $-\lambda_2 + i\lambda_3$, and $-\lambda_2 - i\lambda_3$.

The new inputs of the system can then be calculated from the control law in Equation 4.11 by using the errors in Equation 4.10 with respect to the desired trajectory. Using the chain form control inputs in Equation 4.7, the linear and angular velocities of the unicycle-type vehicle can be found. These are shown in Equation 4.13.

$$\begin{aligned} v &= \frac{\tilde{u_1} + u_1^d}{\cos\theta} \\ \omega &= \frac{\tilde{u_2} + u_2^d}{1 + \tan^2\theta} \end{aligned} \tag{4.13}$$

Tracking control can then be performed for a predefined trajectory, for example a piecewise linear path or a sinusoidal path. To do this, the desired position and velocity components must be derived as functions of time according to the path definition. Then, any errors between the desired configuration values and current configuration values can be computed using Equation 4.10. The computed errors can then be used in Equation 4.11 for the linearized feedback control law. Using these computed values, the linear and velocity commands v and ω can be found using Equation 4.13.

4.3 Leader-Follower Formation Control

An ongoing research area in controls and robotics is the study of multi-vehicle systems and swarming, where it is desired to control a group of robots for a particular application. This type of system holds several advantages as discussed in Chapter 1, and several techniques currently exist in order to address control issues with multiple vehicles moving in formation. These techniques are discussed in more detail in Chapter 2.

In this section, the leader-follower approach is analyzed and discussed for a particular application that utilizes decentralized control. It is assumed that the overall motion of the formation is known to a single vehicle who is designated as the leader, and that all other vehicles, or followers, are aware of the relative position of vehicles directly ahead of them through some type of inter-vehicle communication or vision. The leader follows a given pre-defined trajectory while the follower is controlled by local control laws. The decentralized control for each vehicle is derived from the dynamics of its relative position to the leader. The following is based on the concepts presented in [5] and [1] where a controller is designed that controls the relative distance and heading of the follower relative to the leader vehicle. It is important to know that additional leader-follower control designs exist that involve a follower vehicle that maintains its position in the formation by keeping spec-

ified distance with multiple leader vehicles. The work presented in this thesis demonstrates leader-follower formation control for a single leader vehicle.

The leader-follower formation involving two vehicles, a leader and a follower, is shown in Figure 4.2. The distance between the center of the wheel axis of the leader and the center control point p on the follower is l_{12}. In order to maintain the vehicle formation, the follower vehicle must maintain a desired distance l_{12}^d and desired angle α_{12}^d with respect to the leader vehicle. To do this, a relation of inputs and outputs is required between the system inputs, v_2 and ω_2, and the control outputs, l_{12}^d and α_{12}^d. This type of control is typically called "$l - \alpha$" control in literature.

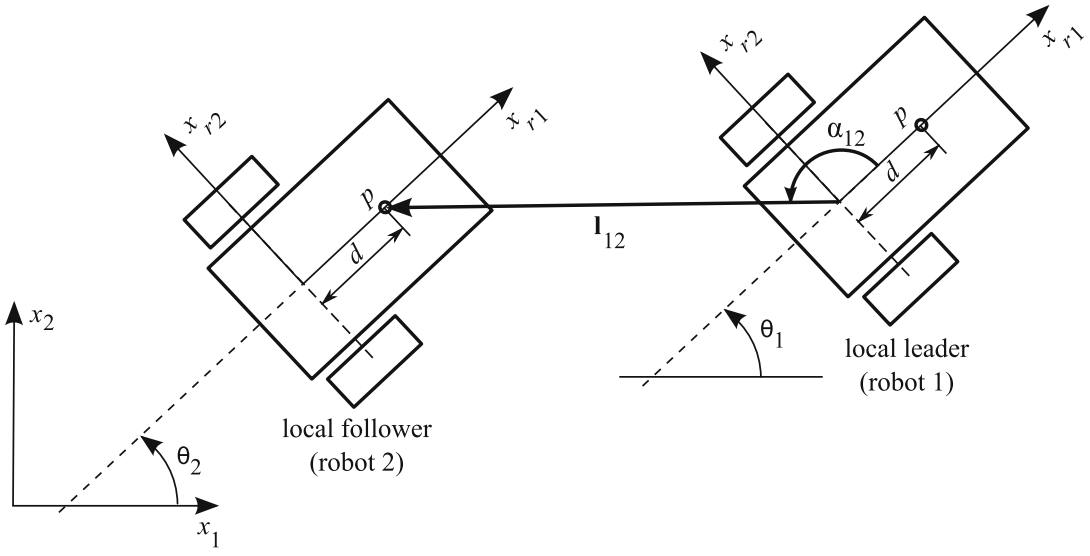


Figure 4.2: Schematic of Leader-Follower Formation [1]

The first step to this method is to perform a kinematic analysis of the relative motion of the vehicles. It is assumed that there is a moving coordinate system with an origin at the center of the wheel axis of the leader vehicle that rotates with the vector $\mathbf{l}_{12}$, which is the distance that the follower vehicle sees with respect to the leader. It is also assumed that there are two points, p_1 and p_2, which are connected to the coordinate frame on the leader vehicle and follower vehicle, respectively. When looking at the follower vehicle from point

p_1 on the leader vehicle, p_2 moves along the distance vector $\mathbf{l}_{12}$. The velocity of p_2, the follower, can be written in terms of its inertial velocity. This can be seen in Equation 4.14, where $\alpha_0 = \theta_1 + \alpha_{12}$, $\dot{\alpha_0} = \omega_1 + \dot{\alpha_{12}}$, and $\mathbf{v}_1$ is the velocity of the leader's center wheel axis.

$$\begin{aligned} \mathbf{v}_{p2} &= (\mathbf{v}_1 + \alpha_0 \hat{\mathbf{k}} \times \mathbf{l}_{12}) + \dot{\mathbf{l}}_{12} \\ \mathbf{v}_{p2} &= \mathbf{v}_2 + \omega_2 \hat{\mathbf{k}} \times \mathbf{d} \end{aligned} \tag{4.14}$$

Equation 4.14 is useful because it relates the system inputs to the follower vehicle, v_2 and ω_2, to the system outputs of the formation, l_{12} and α_{12}. Using the equations for the velocity of the follower vehicle $\mathbf{v}_{p2}$ in Equation 4.14, the outputs of the system $\dot{l_{12}}$ and $\dot{\alpha_{12}}$ can be found. These outputs can be seen in Equation 4.15 where $\gamma_1 = \theta_1 + \alpha_{12} - \theta_2$.

$$\begin{aligned} \dot{l_{12}} &= v_2 \cos\gamma_1 + v_1 \cos\alpha_{12} + d\omega_2 \sin\gamma_1 \\ \dot{\alpha_{12}} &= (v_1 \sin\alpha_{12} - v_2 \sin\gamma_1 + d\omega_2 \cos\gamma_1 - l_{12}\omega_1)/l_{12} \end{aligned} \tag{4.15}$$

Equation 4.15 can also be written in matrix form. This can be seen below in Equations 4.16 and 4.17.

$$\dot{\mathbf{z}} = \mathbf{f} + \mathbf{b}\mathbf{u} \tag{4.16}$$

$$\mathbf{f} = \begin{bmatrix} -v_1 \cos\alpha_{12} \\ (v_1 \sin\alpha_{12} - l_{12}\omega_1)/l_{12} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \cos\gamma_1 & d\sin\gamma_1 \\ -\sin\gamma_1/l_{12} & d\cos\gamma_1/l_{12} \end{bmatrix} \tag{4.17}$$

After obtaining a relation between the system inputs and system outputs, a control law for the nonlinear system can be proposed. The following nonlinear control law is originally

proposed in [1]. The first step is to define stable first-order nonlinear error dynamics. If we define the error as $\tilde{z}_1 = l_{12} - l_{12}^d$ and $\tilde{z}_2 = \alpha_{12} - \alpha_{12}^d$, then the error dynamics can be defined as $\dot{\tilde{z}}_i + k_i \tilde{z}_i = 0$, where $k_i > 0$ for $i = 1, 2$. This error dynamics equation results in a damped linear first order closed-loop system for $k_i > 0$. The desired error behavior can be seen in Equation 4.18.

$$\dot{\tilde{\mathbf{z}}} + \mathbf{K}\tilde{\mathbf{z}} = 0, \mathbf{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \tag{4.18}$$

Using the desired error behavior, we can solve for $\dot{\mathbf{z}}$, which becomes $\dot{\mathbf{z}} = \dot{\mathbf{z}}^d + \mathbf{K}\tilde{\mathbf{z}}$. This can then be substituted into Equation 4.16 to solve for the control input of the leader-follower formation. The control input can be seen in Equation 4.19. The inverse of matrix $\mathbf{b}$ must be inverted in order to calculate the control law. In other words, the determinant of matrix $\mathbf{b}$ cannot be equal to zero, or the follower controller would become uncontrollable.

$$\mathbf{u} = \mathbf{b}^{-1}(\dot{\mathbf{z}}^d - \mathbf{K}\tilde{\mathbf{z}} - \mathbf{f}) \tag{4.19}$$

For more detailed information on leader-follower control schemes as well as specific examples, please refer to [1] and [44]. Additional methods form literature can be found in Chapter 2. These sources discuss different feedback linearization techniques for nonlinear systems and provide analysis of additional control methods for mobile robots.

Chapter 5

Simulation, Experimentation, and Results

This chapter discusses the simulation and experimentation performed during the course of this project using the second generation miniature integrated unmanned ground vehicle design that is presented in Chapter 3. Motion control is simulated in MATLAB and Simulink using the model of kinematic equations of motion presented in Chapter 4 and vehicle modeling is verified. The kinematic equations of motion are then implemented onboard the vehicle in software using different pre-defined trajectories defined by system inputs, and sensor feedback data is analyzed. Finally, a small group of vehicles is put into leader-follower formation and driven along a pre-defined trajectory.

5.1 Simulation

It is important to verify the motion of the unicycle-type vehicle in simulation so that the system can be implemented experimentally onboard the actual vehicle using the system model. To do this, the model of kinematic equations of motion for the unicycle-type mobile

robot as seen in Equation 4.1 is modeled in MATLAB and Simulink to simulate the system with different velocity inputs and outputs. The system model can be seen below in Figure 5.1.

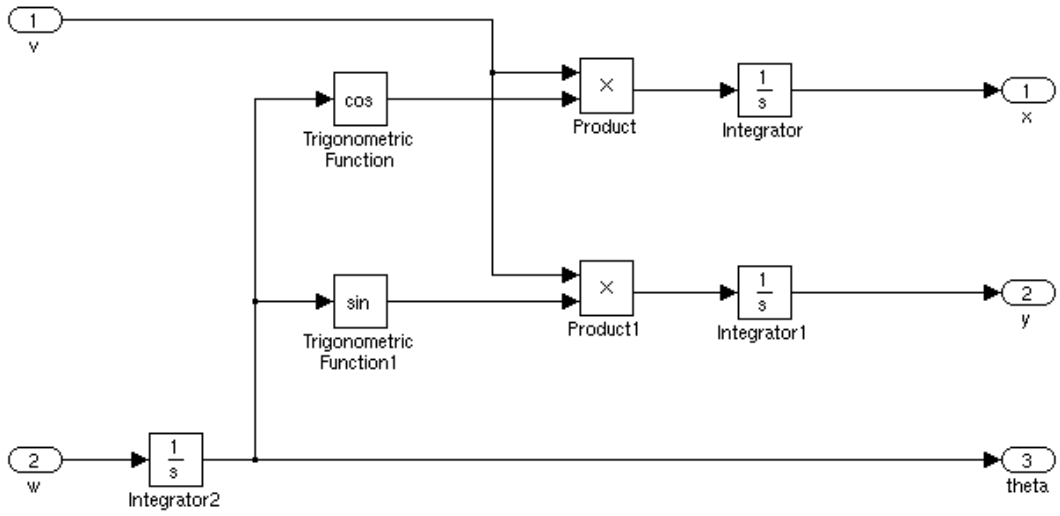


Figure 5.1: Unicycle Vehicle MATLAB Model Equations

To verify the system model in MATLAB and Simulink, the vehicle model shown in Figure 5.1 is put into a subsystem with the inputs and outputs discussed in the previous chapter. The system outputs can be plotted so that they may be analyzed if necessary. This subsystem can be seen in Figure 5.2. Verification is performed by setting the system inputs v and ω to constant values of 1 m/s and 0.52 rad/s, respectively. The resulting output in a 2-dimensional x-y plot is the trajectory of the unicycle-type vehicle traveling in a circle. The vehicle begins at the point $(0, 0)$ and begins driving with constant linear and angular velocities. This simple vehicle trajectory can be seen in Figure 5.3.

A simple controller is then designed in MATLAB to drive the vehicle to a specified (x, y) coordinate reference. A PID controller is interfaced with both the linear and angular velocity inputs of the system to drive the outputs, x, y, and θ to the desired reference values. The feedback for the system is the vehicle's distance from the reference and heading. This

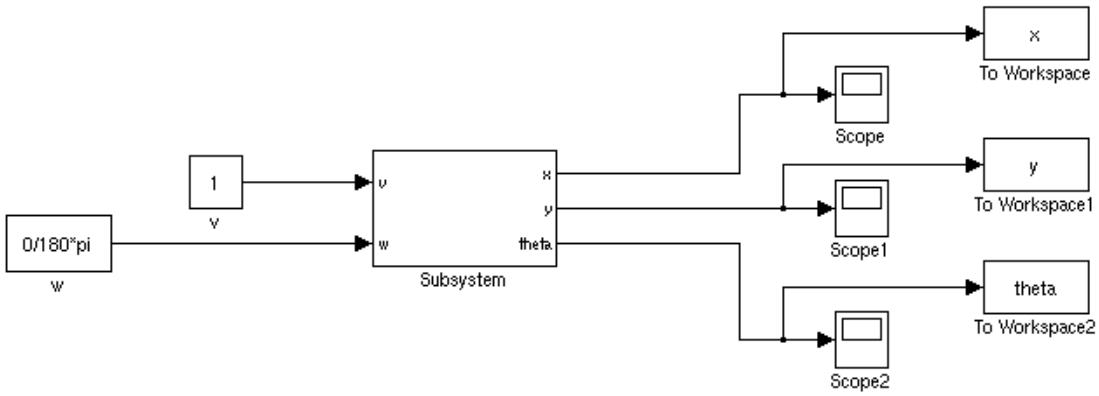


Figure 5.2: MATLAB Vehicle Subsystem

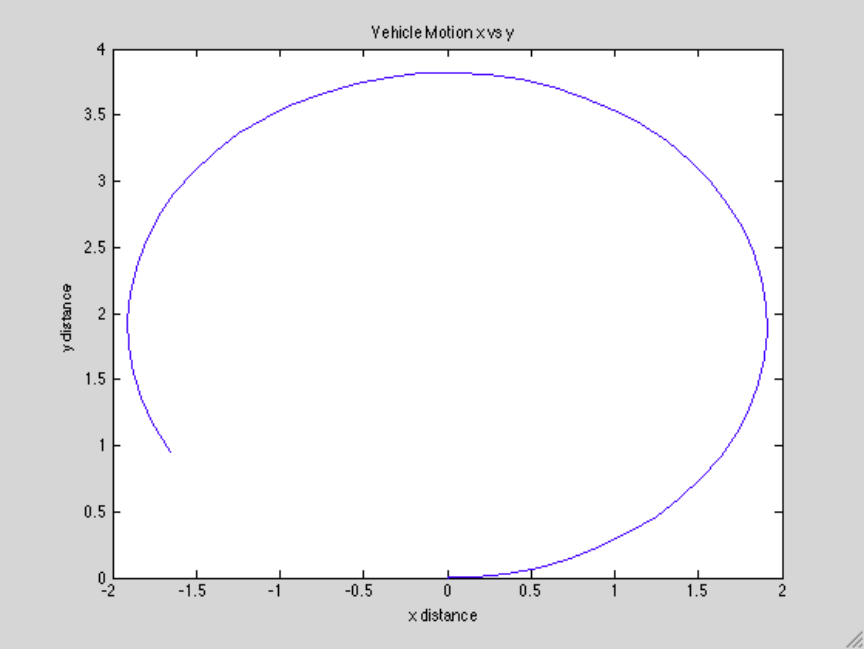


Figure 5.3: Vehicle Trajectory with Constant System Inputs

is compared with the reference values to compute an error and adjust the system inputs accordingly. The controller simulation design can be seen in Figure 5.4.

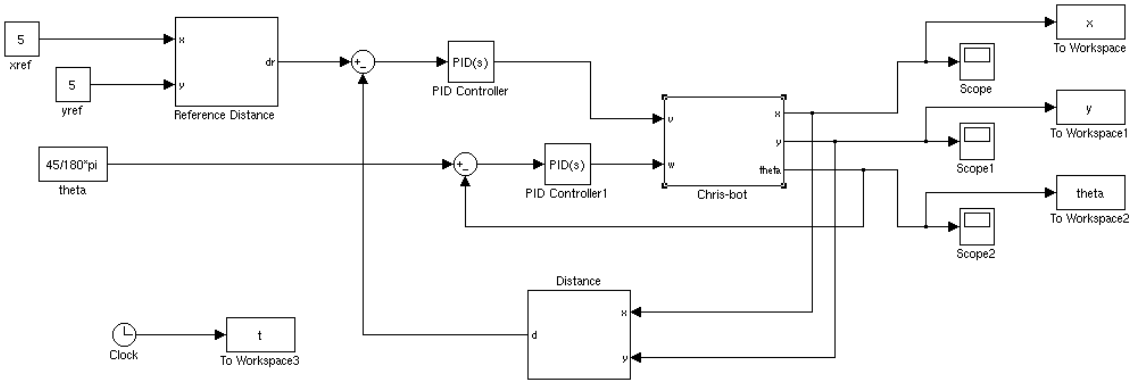


Figure 5.4: MATLAB PID Controller for Motion Control

5.1.1 Simulation Results

Initial conditions of the vehicle were set to $x = 0$ m, $y = 0$ m, and $\theta = 0$ radians. The chosen reference values were $x = 5$ m, $y = 5$ m, and $\theta = 45 \times \pi/180$ radians. The resulting system responses can be seen in Figures 5.5, 5.6, and 5.7.

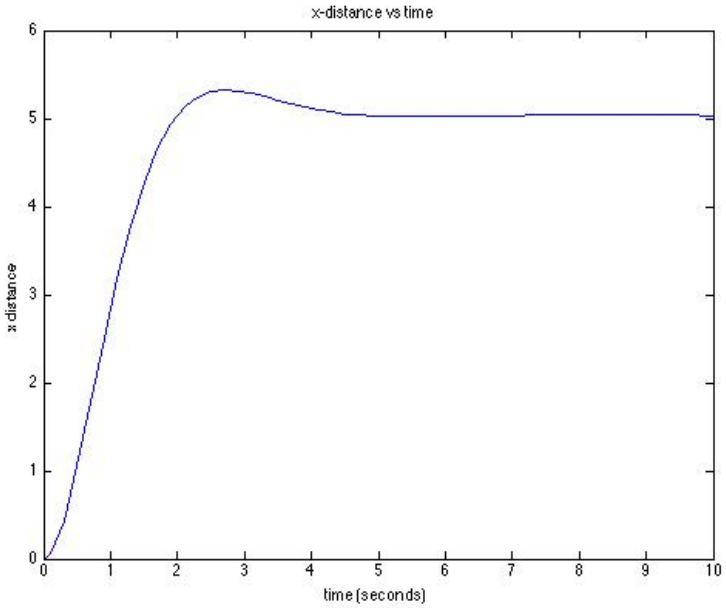


Figure 5.5: System Response in x-direction

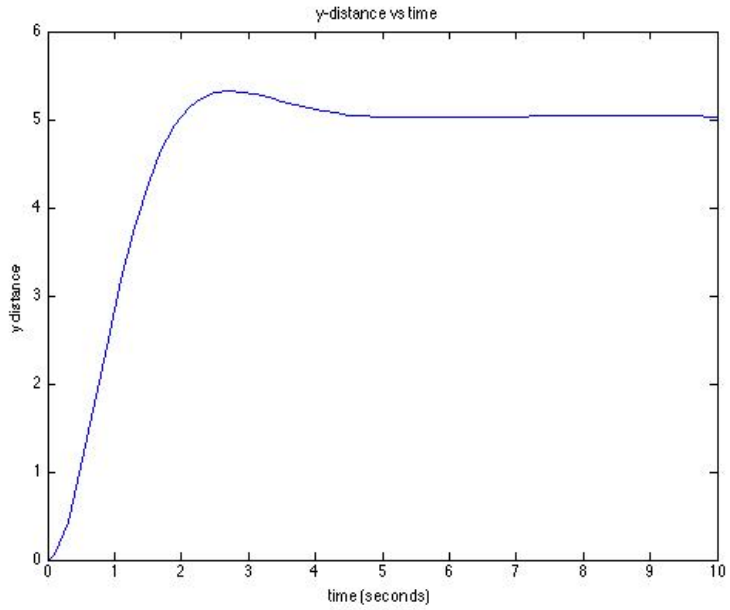


Figure 5.6: System Response in y-direction

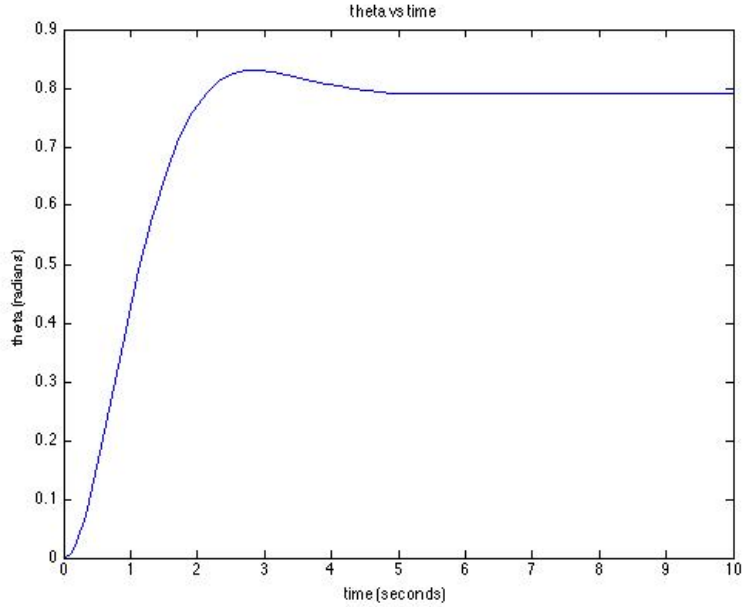


Figure 5.7: System Response of θ

From the simulation results presented in Figures 5.5, 5.6, and 5.7, it can be seen that the distances traveled by the vehicle in the x- and y- directions converge to the desired

reference values. Likewise, the vehicle orientation θ converges to the desired orientation value. This simple simulation verifies the motion control of the vehicle model in a simple waypoint example. The vehicle, starting at position $x = 0$ m and $y = 0$ m with orientation $\theta = 0$ radians, is controlled to the desired position of $x = 5$ m and $y = 5$ m with desired orientation of 0.79 radians. The most difficult component to deal with in this simulation is the feedback of the system. The output of the vehicle model is x, y, and θ while the inputs to the system are v and ω. Calculating the distance traveled by the vehicle was the most convenient way to perform feedback within the system for the given outputs of the system.

5.2 Implementation

After verification of the motion of the unicycle-type vehicle model in MATLAB simulation, the model can then be implemented onboard the vehicle design discussed in Chapter 3. This process was primarily done in software in the XMOS Development Environment and saved to the XMOS XK-1A device's memory. To do this, the inputs to the system had to be identified and translated to PWM signals for the servo motors to be able to read and operate with.

Several characteristics about the miniature UGV design must be known to be able to convert system inputs to PWM signals. For this particular application, characteristics included wheel diameter, wheel axis length, and maximum servo motor velocity. Wheel diameter for the miniature UGV was measured as $d = 0.0698$ m and therefore the circumference of the wheel was found to be $circumference = \pi \times d = 0.219283$ m. The length of the wheel axis on the vehicles was measured to be $a = 98.5$ mm. The maximum velocity was found by performing drive tests on the vehicle along a fixed distance at maximum PWM velocity and measuring the time duration. The maximum velocity for the servo motor in the forward and backward directions was found to be $v = 0.174034$

m/s and $v = -0.174034$ m/s for a 2.0 ms pulse and 1.0 ms pulse, respectively. With this information, the resulting relation between servo motor velocity and PWM pulse width in milliseconds was found to be linear and is shown in Equation 5.1 where p is the pulse-width in milliseconds. This relation for vehicle velocity was found in terms of pulse-width because this is the actual input used in software to drive the individual servo motors.

$$v = 0.348 \times p - 0.522 \tag{5.1}$$

The PWM pulse-width for the left or right servo motor $p_{R,L}$ can then be solved for using Equation 5.1 and is shown below in Equation 5.2.

$$p_{R,L} = \frac{v + 0.522}{0.348} \tag{5.2}$$

In the vehicle software, the PWM signal is determined by the pulse-width, which is specified in milliseconds. Typical pulse-width values for analog servo motors range between 1.0 ms and 2.0 ms. Therefore, the velocities of the left and right servo motors are determined by Equation 5.1. Using Equation 5.1 for linear velocity, a relation between angular velocity ω and pulse-width could be found. This can be seen in Equation 5.3, where a is the axis length between wheels ($a = 0.0985$ m).

$$\omega = \frac{v_R - v_L}{a} \tag{5.3}$$

By substituting Equation 5.1 for both the left and right servo motors into Equation 5.3, a relation between angular velocity ω and PWM pulse-widths for both the left and right servo motors can be obtained. This equation can be reduced to the form shown in Equation 5.4.

$$p_R - p_L = \frac{\omega}{3.53369} \tag{5.4}$$

In the miniature UGV software project, the Equations 5.2 and 5.4 are utilized in a C-file to calculate the actual PWM pulse-width between 1.0 ms and 2.0 ms to send to the individual servo motors. To verify the functionality of the translation between system model inputs v and ω and PWM pulse-width values p_R and p_L, simple test drives were performed using simple pre-defined trajectories consisting of system velocity inputs.

The first vehicle motion test was similar to the motion test performed in MATLAB simulation so that results of the two could be compared. In this experiment, both the linear velocity v and angular velocity ω system inputs were set to constant values. As a result, the vehicle could be observed driving in a circular trajectory. A marker was mounted on the vehicle to observe the error as the trajectory changed primarily due to wheel slipping. The vehicle was then tested with other pre-defined trajectories to see how close the actual trajectory of the vehicle followed a given path. A Python script was written to read larger trajectory information from a text file source into a string of inputs for the system to read. Trajectory information was in the form of linear velocity v, angular velocity ω, and time duration of each command Δt.

Onboard sensors for motion control experiments included front sonar and vehicle heading information. Specifically, the vehicle was equipped with the Parallax Ping sensor and Honeywell HMC6352 Compass among several other sensors used throughout this project. The vehicle was driven along a pre-defined path, in this case a square, with multiple obstacles present in the environment, and sensor data was logged on the OpenLog open source data logger. Positions of all obstacles were known.

5.2.1 Motion Implementation Results

The first motion task implemented on the miniature UGV system involved a trajectory which utilized constant input values for linear velocity v and angular velocity ω. System inputs v and ω were set to small values and repeated over the same time interval, Δt. As a result, the vehicle was in constant motion along a circular trajectory similar to that presented in Figure 5.3. As a result of this test, the miniature UGV was driven along a circular trajectory with a certain amount of error from the desired trajectory. The error over this simple trajectory was observed by a marker that was mounted on the rear the vehicle. The observed trajectory error is presented in Figure 5.8.

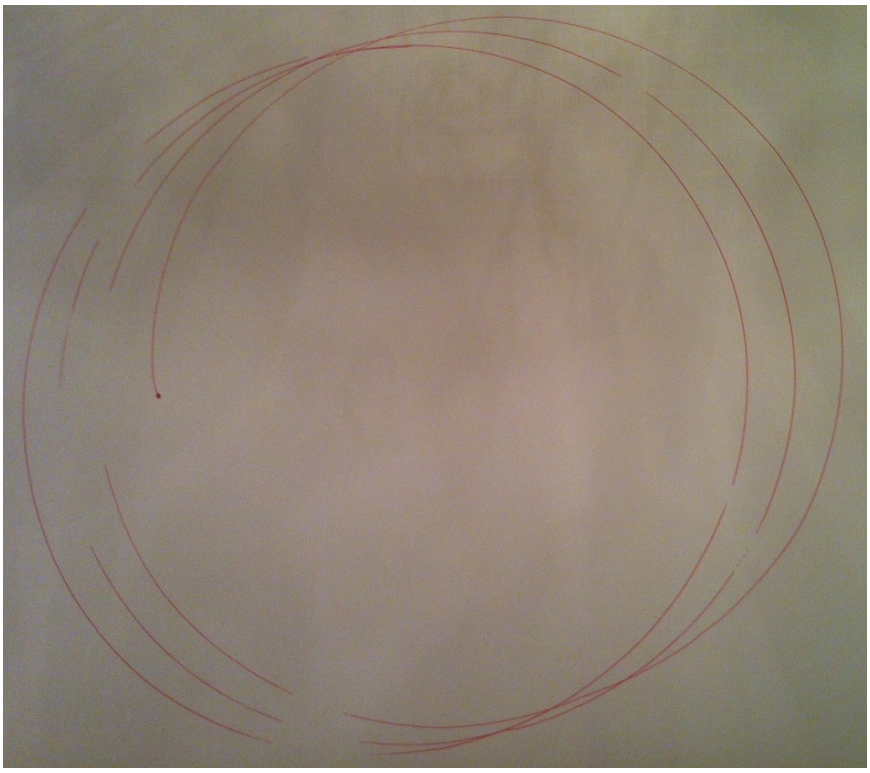


Figure 5.8: Observed Error Over a Circular Trajectory

The trajectory error presented in Figure 5.8 was an expected result of the wheel slipping that occurs with a vehicle of this type. Several control methods developed in current mobile

robotics research assume that no wheel slipping occurs. This can be seen when comparing the simulation result in Figure 5.3 to the actual result in Figure 5.8. Error can also occur due to the environment that the vehicle is operating in. In this particular example, the error can most likely be attributed to the unevenness of the experiment surface. Due to the slope of the surface that the vehicle travels on, the resulting trajectory drifts away from the original desired trajectory slightly.

The next motion test performed involved having the miniature UGV follow the trajectory of a small 400 mm by 400 mm square. Additionally, the environment in which the vehicle operated contained small obstacles. Sensor data from the onboard Honeywell HMC6352 compass and Parallax Ping sonar sensors was logged on the OpenLog data logger. The trajectory used for this experiment contained a linear velocity of 100 mm/s for traversing the edges of the desired square and angular velocity of $\pi/2$ rad/s for performing 90 degree turns at the corners of the square. Data for the experiment can be seen in Figure 5.9.

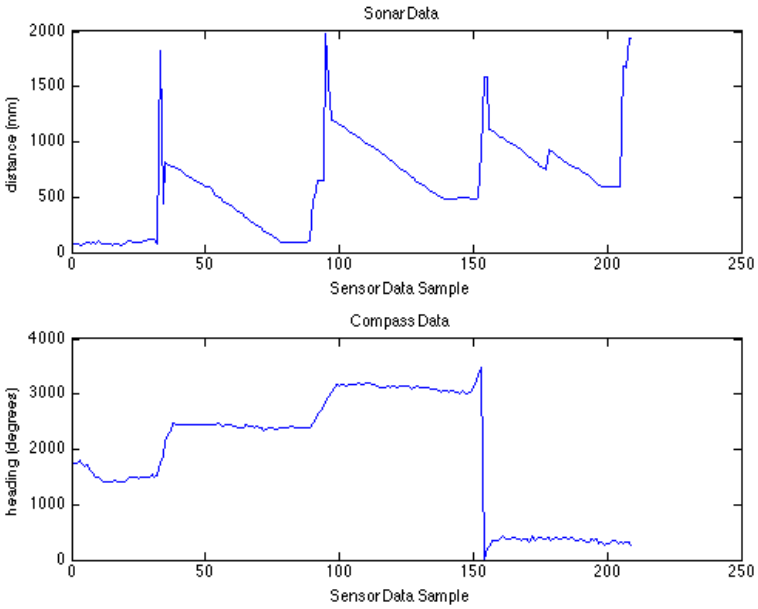


Figure 5.9: Sensor Data over a Square Trajectory with Obstacles

Figure 5.9 shows sonar distance measurements with respect to the front of the vehicle. The peaks within this plot represent obstacles that the vehicle is approaching during its forward motion along the given trajectory. It can be seen that as the vehicle approaches an obstacle, the distance given by the sonar decreases. The distance then increases when the vehicle turns and drives in a different direction. Figure 5.9 also shows the heading of the vehicle throughout the trajectory. Clear transitions of approximately 90 degrees can be seen as the vehicle follows the trajectory. The environment that the vehicle operates in can be visualized from the sensor data to allow for other applications such as localization and obstacle avoidance techniques.

5.3 Leader-Follower Formation

Upon verification of vehicle motion control, vehicle motion was applied to a group of miniature UGV's in formation. The formation method implemented was leader-follower, in which a vehicle designated as the leader follows a pre-defined trajectory while other vehicles designated as the followers maintain a desired distance and orientation with respect to the leader. This method was implemented in a decentralized fashion, meaning that each follower vehicle performed its own follower control based on sensor feedback of where the leader vehicle was positioned with respect to itself.

The leader-follower controller in this project was implemented in software on the XMOS XK-1A microcontroller development board. A sensor was interfaced with the microcontroller onboard the miniature UGV. The sensor used for this application was the Hokuyo URG-04LX Laser Range Finder, which provides both distance to the nearest object and heading associated with that object. A picture of the follower vehicle utilized in this project can be seen in Figure 3.11. The follower vehicle initially reads in data from the laser range finder to find out its distance and orientation to the leader vehicle. This sensor information

is then compared with the desired distance and orientation, which can be initially selected by the user in software. The calculated error between the desired distance and orientation and the desired values is used to compute velocity commands for the vehicle to maintain its desired position relative to the leader.

The controller used in this application is PID, or Proportional-Integrative-Derivative controller, and is implemented in software individually for both the distance and orientation errors. In the miniature UGV software project, a separate C-file was written and used to perform precise PID control computation with the sensor error measurements. Control in this project is considered open-loop due to the lack of direct velocity feedback from the servo motors. The analog servo motors used in this application do not contain any sort of encoder or sensor that offers accurate velocity measurements. Therefore, exteroceptive sensor feedback is used from the laser range finder to perform the control application. Leader-follower formations are implemented utilizing two miniature UGV's and three miniature UGV's in different configurations.

5.3.1 Leader-Follower Formation Results

A decentralized leader-follower controller was designed for vehicles designated as followers. Using sensor feedback from a Hokuyo URG-04LX laser range finder, the follower vehicle maintains a desired distance and orientation from the leader vehicle. The reference given for distance was 250 mm, and the reference for orientation was 0 degrees, or directly behind the leader vehicle. These values were found to be convenient for testing. The leader-follower formation is presented in Figure 5.10.

As a result of the scenario presented in Figure 5.10, the follower vehicle maintained the desired distance and heading over a pre-defined trajectory given to the leader, however, the formation could not be maintained if a disturbance occurred in the system. For example,

Figure 5.10: Leader-Follower Formation with Two Vehicles

if an obstacle presented itself within the sensing range of the follower, the follower would assume that the obstacle was the leader and begin maintaining distance and orientation with respect to it. There was no communication between vehicles, therefore the formation was maintained purely by the local controller on the follower vehicle.

The leader-follower formation was then extended to three vehicles. This formation was driven in the same fashion as the formation with two vehicles. Again, the leader vehicle was primarily responsible for the formation's trajectory. The follower vehicles followed the vehicle directly in front of them by a local controller. The leader-follower formation with three vehicles is presented below in Figures 5.11 and 5.12.

Additionally, the leader-follower formation with three vehicles was implemented in a different configuration, where the two follower vehicles were placed to the side of each other. In this example, both followers were following the same leader vehicle which again was responsible for the formation's overall trajectory. Both follower vehicles each had

Figure 5.11: Leader-Follower Formation with Three Vehicles

Figure 5.12: Leader-Follower Formation with Three Vehicles

local controllers that allowed them both to maintain different orientations with respect to the leader. However, the reference distance for both followers remained the same. This configuration of the leader-follower formation is presented in Figures 5.13 and 5.14.

Figure 5.13: Alternative Leader-Follower Formation Configuration

All local controllers on the follower vehicles utilized the discretized kinematic model of equations along with a discrete PID controller. The PID controller used in this application contained gains of $K_p = 0.2$, $K_i = 0.0001$, and $K_d = 0.02$. These values were chosen through initial testing of the leader-follower formation. The overall goal of choosing the controller gains was to prevent overshoot where the follower vehicle could collide with the leader. All vehicles were operated at a low speed of approximately 50 mm/s so that minimal error from factors such as wheel slipping and environment could occur. Command durations, or Δt was based on the frequency of the laser sweep, which was 100 ms. All vehicles used 730 mAh 11.1 V Lithium-Polymer batteries which provided approximately

Figure 5.14: Alternative Leader-Follower Formation Configuration

30 minutes of testing on a single charge. All follower vehicles maintained formation while the leader followed a pre-defined trajectory. However, the system was found to be highly susceptible to disturbances and lacked robustness when the formation needed to navigate through obstacles.

Chapter 6

Conclusions and Recommendations

In this chapter, conclusions and recommendations are presented based on the results of experimentation that are stated in Chapter 5. The main results of this thesis are concluded and a number of recommendations are made for future research.

6.1 Conclusion

The research presented in this thesis involves the design and implementation of a small unmanned ground vehicle at the University of Denver's Unmanned Research Institute. The primary contributions of this project have been the design of a small integrated mobile robot system, implementation of a kinematic model of equations onboard the vehicle's microcontroller, and implementation of multiple vehicles in a simple sensor-based leader-follower formation. This project encompasses mechatronics design techniques in the areas of mechanical, electrical, computer engineering.

This project addresses the current problem of existing small mobile robot systems being relatively high in cost and not ideal for multi-vehicle or swarm behavior research. Many of the popular commercially available mobile robot systems are not convenient for swarm

behavior research due to their lack of accommodation for different research applications. They are also high in cost and do not contain a microcontroller that is sufficient for research in unmanned systems.

Two generations of a simple unicycle-type vehicle design are presented and discussed. The second generation provides a number of significant improvements over the first, including an additional top plate for sensors, more chassis space for components, and a robust front bumper system. The overall cost of the vehicle design is compared with existing systems and shown to be superior in cost and simplicity. Two different configurations for the second generation vehicle containing different sensors are also shown.

Next, the designed vehicle is then developed to utilize the kinematic model of motion equations for the unicycle type vehicle. The kinematic model is discretized and system inputs v and ω are translated to PWM signals in software for the servo motors to operate with. The system is then tested both in MATLAB simulation and onboard the vehicle with pre-defined trajectories. Error primarily from wheel slipping is observed and shown.

Finally, the designed miniature UGV is implemented in a small leader-follower formation. A leader-follower task is written in software utilizing sensor feedback. The leader follows a pre-defined trajectory while the followers maintain relative distance and orientation through decentralized control. The formation is first implemented with two vehicles followed by the addition of a third. Additionally, the formation is implemented in different configurations.

6.2 Recommendations for Future Work

The successful design and implementation of this small unmanned ground vehicle can serve as a proof of concept and baseline for future research in the field of robotics and control. There are several points that are open for improvement with this research. First, with

regards to the vehicle design, a breakout PCB may be designed and manufactured to incorporate all connections to the onboard microprocessor. This can help reduce the number of wires onboard the vehicle and make the platform more convenient to work with. The vehicle has been designed to be used for numerous applications involving future work. In particular, localization and obstacle avoidance research methods can be applied and studied using this platform.

Several improvements can be made upon the current leader-follower formation technique that is implemented. First, inter-vehicle communication can easily be implemented onboard using devices such as the Roving Networks Wi-Fly wireless antenna. The vehicles are currently equipped with these devices. As an example, it can be useful for the leader to broadcast sensor information to the followers so that they can better maintain the formation. Finally, the system can be made more robust so that the leader-follower formation is less susceptible to disturbances from obstacles or other vehicles.

Bibliography

[1] F. Fahimi, *Autonomous robots: modeling, path planning, and control*. Springer New York, 2009.

[2] Y. Q. Chen and Z. Wang, "Formation control: a review and a new consideration," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 3181–3186, IEEE, 2005.

[3] C. Brune, T. Dityam, J. Girwar-Nath, K. Kanistras, G. Martins, A. Moses, I. Samonas, J. L. S. Amour, M. J. Rutherford, and K. P. Valavanis, "Enabling civilian applications of unmanned teams through collaboration, cooperation, and sensing," in *SPIE Defense, Security, and Sensing*, pp. 83870D–83870D, International Society for Optics and Photonics, 2012.

[4] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 926–939, 1998.

[5] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4, pp. 2864–2869, IEEE, 1998.

[6] G. Klančar, D. Matko, and S. Blažič, "A control strategy for platoons of differential drive wheeled mobile robot," *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 57–64, 2011.

[7] G. Lee and N. Y. Chong, "Flocking controls for swarms of mobile robots inspired by fish schools," *Recent advances in multi robot systems, A. Lazinica, editor, I-Tech Education and Publishing, Vienna, Austria*, 2008.

[8] I. Navarro and F. Matía, "A survey of collective movement of mobile robots," *Int J Adv Robotic Sy*, vol. 10, no. 73, 2013.

[9] M. Xiaomin, D. Yang, L. Xing, and W. Sentang, "Behavior-based formation control of multi-missiles," in *Control and Decision Conference, 2009. CCDC'09. Chinese*, pp. 5019–5023, IEEE, 2009.

[10] G. Antonelli, F. Arrichiello, and S. Chiaverini, "The entrapment/escorting mission," *Robotics & Automation Magazine, IEEE*, vol. 15, no. 1, pp. 22–29, 2008.

[11] R. Mead, J. B. Weinberg, and J. R. Croxell, "A demonstration of a robot formation control algorithm and platform," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence, Vancouver, Canada*, Wiley, 2007.

[12] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[13] K. Do and J. Pan, "Nonlinear formation control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 191–204, 2007.

[14] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, "Formation path following control of unicycle-type mobile robots," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 727–736, 2010.

[15] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324–333, 2008.

[16] N. Sorensen and W. Ren, "A unified formation control scheme with a single or multiple leaders," in *American Control Conference, 2007. ACC'07*, pp. 5412–5418, IEEE, 2007.

[17] T. H. van den Broek, N. van de Wouw, and H. Nijmeijer, "Formation control of unicycle mobile robots: a virtual structure approach," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pp. 8328–8333, IEEE, 2009.

[18] Y. H. Esin and M. Ünel, "Formation control of nonholonomic mobile robots using implicit polynomials and elliptic fourier descriptors," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 18, no. 5, pp. 765–780, 2010.

[19] B. d'Andrea Novel, G. Campion, and G. Bastin, "Control of nonholonomic wheeled mobile robots by state feedback linearization," *The International journal of robotics research*, vol. 14, no. 6, pp. 543–559, 1995.

[20] A. De Luca, G. Oriolo, and M. Vendittelli, "Stabilization of the unicycle via dynamic feedback linearization," in *6th IFAC Symp. on Robot Control*, pp. 397–402, 2000.

[21] S. Ahmed, M. N. Karsiti, and G. M. Hassan, "Feedback linearized strategies for collaborative nonholonomic robots," in *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, pp. 1551–1556, IEEE, 2007.

[22] G. Klančar, D. Matko, and S. Blažič, "Wheeled mobile robots control in a linear platoon," *Journal of Intelligent and Robotic Systems*, vol. 54, no. 5, pp. 709–731, 2009.

[23] S.-C. LIU, D.-L. TAN, and G.-J. LIU, "Robust leader-follower formation control of mobile robots based on a second order kinematics model," *Acta Automatica Sinica*, vol. 33, no. 9, pp. 947–955, 2007.

[24] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.

[25] H. Sira-Ram'rez and R. Castro-Linares, "Trajectory tracking for non-holonomic cars: A linear approach to controlled leader-follower formation," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 546–551, IEEE, 2010.

[26] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "On the control of a leader-follower formation of nonholonomic mobile robots," in *Decision and Control, 2006 45th IEEE Conference on*, pp. 5992–5997, IEEE, 2006.

[27] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "A geometric characterization of leader-follower formation control," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2397–2402, IEEE, 2007.

[28] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis, "Vision-based localization for leader–follower formation control," *Robotics, IEEE Transactions on*, vol. 25, no. 6, pp. 1431–1438, 2009.

[29] P. Renaud, E. Cervera, and P. Martiner, "Towards a reliable vision-based mobile robot formation control," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4, pp. 3176–3181, IEEE, 2004.

[30] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 5, pp. 813–825, 2002.

[31] Z.-P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form," *Automatic Control, IEEE Transactions on*, vol. 44, no. 2, pp. 265–279, 1999.

[32] Z.-Y. Liang and C.-L. Wang, "Robust stabilization of nonholonomic chained form systems with uncertainties," *Acta Automatica Sinica*, vol. 37, no. 2, pp. 129–142, 2011.

[33] J. Ghommam, H. Mehrjerdi, and M. Saad, "Leader-follower based formation control of nonholonomic robots using the virtual vehicle approach," in *Mechatronics (ICM), 2011 IEEE International Conference on*, pp. 516–521, IEEE, 2011.

[34] J. Hu and G. Feng, "Distributed tracking control of leader–follower multi-agent systems under noisy measurement," *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010.

[35] B. S. Park, J. B. Park, and Y. H. Choi, "Adaptive formation control of electrically driven nonholonomic mobile robots with limited information," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, no. 4, pp. 1061–1075, 2011.

[36] T. Liu and Z.-P. Jiang, "Distributed formation control of nonholonomic mobile robots without global position measurements," *Automatica*, 2012.

[37] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[38] G. W. Gamage, G. K. Mann, and R. G. Gosine, "Discrete event systems based formation control framework to coordinate multiple nonholonomic mobile robots," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 4831–4836, IEEE, 2009.

[39] J. Ghommam, H. Mehrjerdi, and M. Saad, "Leader-follower formation control of nonholonomic robots with fuzzy logic based approach for obstacle avoidance," in

Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 2340–2345, IEEE, 2011.

[40] N. Xiong, J. He, J. H. Park, T.-h. Kim, and Y. He, "Decentralized flocking algorithms for a swarm of mobile robots: Problem, current research and future directions," in *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pp. 1–6, IEEE, 2009.

[41] K. S. Raghuwaiya, S. Singh, and J. Vanualailai, "Formation control of mobile robots," *World Academy of Science, Engineering and Technology*, vol. 60, pp. 762–767, 2011.

[42] XMOS, *XK-1A Hardware Manual, REV 1.1.0*, 2012.

[43] G. Martins, A. Moses, M. J. Rutherford, and K. P. Valavanis, "Enabling intelligent unmanned vehicles through xmos technology," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 9, no. 1, pp. 71–82, 2012.

[44] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.

[45] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot motion planning and control*, pp. 171–253, Springer, 1998.