COMPLIANT FORMATION CONTROL OF AN AUTONOMOUS MULTIPLE VEHICLE SYSTEM

By

ERICA ZAWODNY MACARTHUR

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2006

I owe this research to my wonderful husband, Donald, who has been my motivation and inspiration during these years. You suffered with me through the bad times and celebrated with me through the good times. My parents Joseph and Yvonne, and my brother, Nikolas, thank you for all of your support and love. You encouraged me to persevere when things got tough. Thank you.

ACKNOWLEDGMENTS

I would like to thank Dr. Carl Crane for supporting me throughout my graduate school career. Thanks go to my committee members: Dr. John Schueller, Dr. Gloria Wiens, and Dr. Antonio Arroyo. I would also like to extend my thanks and gratitude to the Air Force Research Laboratory at Tyndall Air Force Base, Panama City, Florida, for supporting this work and all of CIMAR throughout the years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

COMPLIANT FORMATION CONTROL OF AN AUTONOMOUS MULTIPLE VEHICLE SYSTEM

By

Erica Zawodny MacArthur

December 2006

Chair: Carl. D. Crane
Major Department: Mechanical and Aerospace Engineering

This research identifies a new strategy called *compliant formation control*, which may be used to coordinate the navigational structure of a team of autonomous vehicles. This technique controls the team's motion based on a given, desired formation shape and a given, desired set of neighboring separation distances, wherein the formation shape is considered general two-dimensional. The strategy establishes how to select, place, and use virtual springs and dampers that conceptually "force" proper interspacing between neighboring team members. The objective is to continuously maintain, in the most optimal way, the desired formation as team motion proceeds.

Research in multiple vehicle systems has addressed follow-the-leader techniques, cooperative mapping, reconnaissance and communication, and learning and adaptation techniques. Each of these areas is interested in multiple vehicle coordination techniques. However, formation control may or may not be a concern, as the following contemplates. For example, follow-the-leader techniques, as the name implies, require formation control while cooperative mapping is mainly concerned with maximizing coverage area and development of a world model. Multiple vehicle reconnaissance heavily depends upon communication between team members and robotic relays have been developed to alleviate the communication

limitations. Learning and adaptation applied to systems of robots draws upon biology to create behaviors that react to environmental stimulus. Behaviors such as foraging and flocking provide a form of loose formation control as dictated by nature. The techniques mentioned above have merit in the realm of multiple vehicle research but lack a clearly defined technique for strict formation control.

This research provides a strategy for formation control that is based on a desired formation shape. In practice, actual robot separation distances will be measured relative to smarter, leader robots that have known position and orientation information at all times (e.g., GPS or INS). The control strategy subsequently commands, in an optimal way, each vehicle by providing a heading and velocity necessary to maintain the desired formation. Such requisite commands result from modeling the compliant displacements of team members as they travel in a network of virtual springs and dampers. One of the primary contributions of this work is the development of a methodology to determine the internal behaviors for the individual mobile robots in order to achieve a desired global formation for the entire system. One of the motivations here is to reduce the cost and increase the navigational effectiveness of a team of mobile robots, since only a select few team members (leaders) are required to be equipped with expensive GPS or INS equipment.

CHAPTER 1
INTRODUCTION

Research at the Center for Intelligent Machines and Robotics at the University of Florida has strived to advance the field of unmanned ground vehicle systems since 1990. Previous research has included innovative sensor systems, sensor fusion, precision navigation, path planning, and obstacle avoidance. The research has since expanded to include multi-vehicle cooperation, unmanned aerial platforms, and collaboration of unmanned aerial and ground vehicle systems.

Faster computing and new sensing technologies have rapidly advanced robotics research. As perception abilities improve, robotic agents can be used more often to perform monotonous and often dangerous tasks in place of human operators. Robots are currently being used to automate military and agricultural tasks. Robots working as teams can take existing technology one step further by allowing several robots to accomplish a series of tasks more efficiently than a single agent. Duplicating several inexpensive robots with different sensing abilities may prove to be more cost effective than a single robot performing the task.

The 1990s sparked an era in robotics that has created a lot of interest in robot teams. The concept of soccer-playing robots was first introduced in 1993. Since then competitions such as RoboCup soccer leagues and the DARPA Grand Challenge have become an international phenomenon in robotics education and research. Currently, there are several robotic soccer leagues ranging from a simulation league to a humanoid league where biped robots play against each other. As a result, a host of research papers have provided insight into artificial intelligence techniques developed for the RoboCup competitions. The DARPA Grand Challenge competition proved to challenge the sensing and mobility of unmanned ground vehicles through rough desert terrain. The next generation of the Grand Challenge involves autonomous

navigation through an urban environment and will challenge the vehicle's ability to make intelligent decisions.

Recently, the Air Force Research Laboratory (AFRL) has become interested in improving EOD range clearance operations and has employed tele-operated unmanned ground vehicles to assist in this task. However, due to the volatility of this task, operators can be several miles away from their targets which lead to problems with communication. Use of multiple ground vehicles as communications repeaters could alleviate or even eliminate communication lags.

Formation control of these unmanned systems is essential for autonomous convoys, search and rescue, and reconnaissance missions. This research investigates collaborative ground vehicles and formation control. The research vision is to apply basic principles to develop a methodology to determine behaviors for members of a robot team that will result in reliable formation coordination of the group.

CHAPTER 2
LITERATURE REVIEW

Research in multi-robot systems has grown significantly since the late 1980s. Prior to this, research had focused on single robot systems or distributed problem-solving systems that were devoid of robotic components. The growth in this field of research is fueled by the increasing realization that teams of robots may provide solutions to problems that were previously unsolvable. Multi-robot teams can provide reliability, robustness, and efficiency by taking advantage of the redundancy built in to the system. This chapter will provide insight into different aspects, methodologies, and applications of this field.

Biological Inspiration

Many researchers have found it beneficial to investigate the social characteristics of animals and insects and apply it to artificial life systems. These techniques have had a predominant influence in cooperative multi-robot research.

> . . . yet we feel quite certain that higher vertebrates - ourselves, certainly – are more than programmed robots solving the same problems generation after generation. [1:211]

Swarms and Flocks

Swarms of insects have the innate ability to effectively collect food, travel, and raise young as a collective. Worker honeybees' very existence is to serve the needs of the colony as a whole. The worker bees work to build the honeycomb, rear the young, clean the colony, feed the queen and the drones, guard the hive, and collect food. Without hive mates a worker bee does not live very long.

Swarm intelligence (SI) is modeled after this very phenomenon of cooperation of a biological system to achieve a collective goal. The concept of "swarm intelligence" was first introduced by Beni and Wang in 1989 [2]. SI systems are typically composed of a population of single agents that can interact with each other and their environment. Typically there is no

centralized control strategy dictating how each individual agent should behave. Surprisingly, local interactions of individual agents lead to the emergence of global behavior which mimics natural systems such as bee colonies, fish schooling, and bird flocking. Figure 2-1 shows a typical swarm of honeybees.

Figure 2-1: Honeybees swarming.

Ant-Like Behaviors

In an effort to build a more perfect robotic system, researchers and designers strive to imitate nature. Ants, for example, are self-organized and have natural mechanisms for the division of labor and that allow the transfer of information among group members (see Figure 2-2).

Ant-like task allocation and recruitment have been applied to cooperative robots [3]. This research addressed the benefit of cooperative robots working together to single robots working in solitary. The task of foraging was assigned to the robot colonies to investigate whether ant-derived algorithms affect the robot cooperation. The foraging robots were programmed according to the components of social organization that govern ant colonies.

Several experiments were performed with 1, 3, 6, 9, and 12 robots. The robots had several different scenarios ranging from uniform food distribution without recruitment to clustered food distribution with recruitment. Recruitment is a behavior observed in many ant species [4] that

allows ants in resource-rich environments to mimic the behavior of others, thus resulting in more resources for the colony. Overall, results indicate that with a group size between 3-9 robots, the efficiency of the colony was greatest. Efficiency was lowest with twelve robots which are due to interference during foraging.

Figure 2-2: Fire ants.

The ant-like behaviors have been applied to optimization problems such as the traveling salesman problem, quadratic assignment, and scheduling [5]. The model developed provides positive feedback which allows for rapid discovery of solutions, and distributed computation avoids premature convergence. The idea is that if an ant is given a choice among different paths, those which were heavily chosen by preceding ants are chosen with higher probability. Therefore, more heavily traveled trails lead to selection of the shortest (optimal) path.

Robot Behaviors and Learning

A behavior, simply put, is a reaction to a stimulus. [6:66]

Schema-Based Behaviors

The ideas of schema-based behaviors or motor schemas form the basic building block for the behavior of a robotic agent. A method of reactive navigation for mobile robots is presented in Arkin [7]. Arkin presents reactive navigation systems as being characterized by a decomposition of robotic goals into primitive behaviors, which are activated via arbitration;

perceptual strategies which provide only information necessary for each activity, and the avoidance of global world models which yields faster real-time response. A series of motor schemas are defined: move-to-goal, avoid-static-obstacle, and noise (random wandering). This research provides an effective method of multi-vehicle coordination using a set of simplistic behaviors while minimizing inter-robot communication.

Conversely, communication among the robot team can be an important feature that can be used to aid in coordination. The value-based communication preservation (VBCP) is a navigation behavior that takes into account shared locations of its teammates, measured communication signal quality and map-based predictions of communications signal quality to calculate movement vectors [8]. A motor schema approach is used to create a vector that guides a robot to its respective place in a formation, relative to its teammates' positions. This vector is calculated by finding the center of mass of the entire team and a line from the center to the goal the team is moving toward. Each robot's position is calculated relative to these features.

This research models the communication signal strength using a standard line-of-sight path loss model, which is calculated from geometric information from an a priori map. The path loss between two unobstructed nodes in an RF network in dB is

$$pathloss = 20\log\frac{4\pi d}{\lambda} \tag{2-1}$$

where d is the distance between the nodes and λ is the wavelength (both having the same units). For this project, it was assumed that each robot must maintain connectivity with at least two teammates. The following function was developed that served to measure communication quality, ensure connectivity between two teammates with similar signal quality, and increase/decrease the "dead zone," or the range of adequate communication quality:

$$v = \frac{1}{\left(1 + e^{-C_1\left(\frac{r_1 + r_2}{100} - C_2\right)}\right)\left(1 + e^{C_3\left(\frac{r_1 - r_2}{100} - C_4\right)}\right)} \tag{2-2}$$

where r_1 and r_2 are the first and second strongest predicted or measured signals with teammates. C_1, C_2, C_3, and C_4 are positive constants. The full VBCP algorithm is summarized in the following steps:

1. The current signal strength of each teammate is measured. The current overall communication quality is calculated using equation (2-2).
2. The predicted signal strength at possible next steps is evenly distributed around a radius representing the distance to be traveled. The overall communication quality is calculated at each next step.
3. A unit vector $\boldsymbol{a_i}$ in the direction of each next step is created and scaled by the overall communication quality.
4. A unit vector b is created in the direction of the sum of the $\boldsymbol{a_i}$ vectors and scaled by the communication quality.
5. Vector $\boldsymbol{b}$ is returned as the best next movement.

Overall, VBCP seems to improve communication quality of service, especially as the number of robots in the team is increased. The percent of missions completed declined slightly when VBCP was used because robot teams became stuck in local minima.

Learning and Adaptation

Nature presents an elegant design to the phenomena of adaptation in insects and animals. These naturally occurring multi-agent systems are successfully able to survive with incomplete information and noisy, non-deterministic environments. The field of multi-robot systems is faced with similar challenges.

Based on the research of Mataric [9], suggestions of control and learning in multi-robot systems must be addressed separately rather than together as another piece to the single-robot puzzle. A bottom-up methodology is proposed to produce desired system behavior. The approach uses behaviors such as group foraging, flocking, socializing rules, and coordinated

object moving. The behaviors couple sensory inputs and effector outputs and are then programmed to construct flexible internal representations such as world models or history data.

A multi-robot learning testbed was entitled "Cooperative Multi-Robot Observation of Multiple Moving Targets" (CMOMMT) [10]. The problem is modeled as a two-dimensional enclosed spatial region, S, a team of m robots with 360° field of view with a set of n targets, $O(t)$. A robot, v_i, is said to be observing a target when the target is within its sensing range. An $m \times n$ matrix is defined such that the (i, j) members are 1 if the robot is observing a target and 0 otherwise. This information is used to maximize a metric, A. The overall goal of the robots is to maximize the average number of targets in S that are being observed by at least one robot throughout the mission of length time, T. The CMOMMT is able to handle a large number of robots and targets which must be taken into account in the design of a cooperative control system.

Robotic Soccer Teams

Robotic soccer has been a test bed for the development of multi-vehicle systems. The coordination of the Azzura Robot Team [11] is based on a communication protocol and is able to coordinate a group of heterogeneous robots from various universities. These robots differ in hardware and software, but work as a team through effective use of UDP-IP protocol, broadcast communication protocol, message exchange protocol, and coordination protocol. A set of status messages (own status, ball status, other players) are used to asses the overall state of the team.

The larger task (formation) was divided into smaller subtasks (roles) that are easily managed by a single robot, and then a method assigning to each robot one of these subtasks was applied. A coordination module is responsible for the team formation and role assignment based on communication messages and internal states. Roles can be dynamically assigned depending on the internal states of the robots. High level coordination among the robots is the selection of

the formation that is most suitable based on the current environmental state. A planner is used to generate concurrent formations based on this information.

The field of robotic soccer will continue to advance the field of multi-robot research in the future. It serves as a catalyst for research in this area of robotics.

Towards Motion Coordination

Follow-the-Leader

In order to have coordinated groups of autonomous robots, they must be organized in some type of hierarchy. In addition, it is beneficial to have a task in which to base the research and experiments. Much research has been pursued in the area of convoying applications. This is particularly applicable to military transport operations (see Figure 2-3). With this type of coordination a leader is required to guide the rest of the fleet. In essence, this is robotic follow-the-leader.

Figure 2-3: Military convoy operation.

Robotic convoy navigation research is a prime example of a formation control application. The research in this area considers a lead vehicle that is either manually driven or autonomous that is followed by a series of automated vehicles single-file. Early research considers basic operations of a convoy of vehicles whose formation control strategy is based on distance measurement [12,13]. Guidance law strategies [14] have been applied to derive control laws for angular and linear vehicle velocities such that a constant following distance is maintained.

Another approach for autonomous convoy navigation involves visual tracking and servoing. In the research of Schneiderman [15] a vision-based autonomous convoy navigation technique is described in which a robotic vehicle is pursuing another by visually tracking a target mounted on the back of the pursued or lead vehicle. Other research applies visual servoing techniques [16, 17, 18, 19, 20] for robot positioning, tracking, and estimating velocities. In position-based visual servoing an estimate of the relative pose of a target is made based on visual data and geometric modeling.

Previous research has proposed a simple discrete-time model of autonomous agents – point or particles – all moving in the plane with the same speeds but different headings [21]. An interaction radius of r was used as the unit to measure distances from other particles as shown in Figure 2-4.

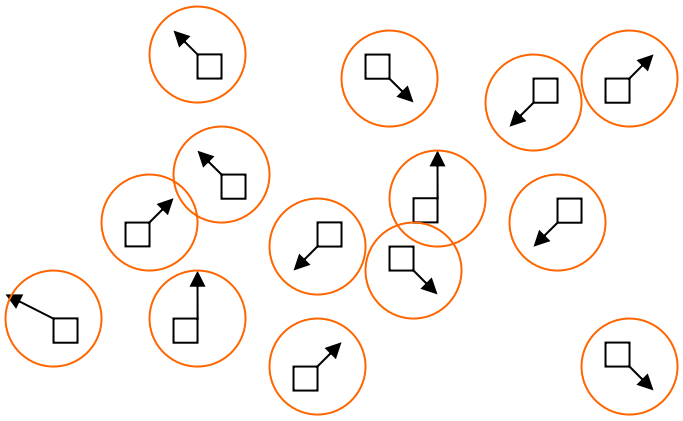


Figure 2-4: An illustration of particles with different headings. The circle represents each particle's interaction radius.

At each time step, t, the velocities $\{\mathbf{v}_i\}$ of the particles were determined and the position was updated using the following:

$$x_i(t+1) = x_i(t) + v_i(t)\Delta t. \tag{2-3}$$

The velocity of a particle $\mathbf{v}_i(t+1)$ was based on a constant velocity magnitude, v along a direction given by the angle $\theta(t+1)$. The angle was obtained from the following:

$$\theta(t+1) = \left\langle \theta(t) \right\rangle_r + \Delta\theta \tag{2-4}$$

The expression $\left\langle \theta(t) \right\rangle_r$ is the average direction of the particles that are within a circle of radius r that surrounds the given particle. The term $\Delta\theta$ represents noise. To summarize, each agent's heading is updated using a local rule based on the average of its own heading plus the headings of neighboring agents. These relatively simplistic equations cause all agents to ultimately move in the same direction.

This same technique was applied to a system of agents; however an additional agent is designated as the group leader [22]. The i-th follower updates its heading in the same manner using the average of its own heading plus the headings of its neighbors. However, what differs is that now each follower's set of neighbors can include the leader and does so whenever the leader is within its bounding circle of radius r. The heading update rule for agent i was developed to have the following form:

$$\theta_i(t+1) = \frac{1}{1+n_i(t)+b_i(t)}\left(\theta_i(t) + \sum_{j\in N_i(t)} \theta_j(t) + b_i(t)\theta_0\right) \tag{2-5}$$

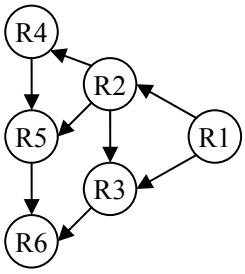
In (2-5) $N_i(t)$ is the set of labels of agent i's neighbors and $n_i(t)$ is the number of labels in $N_i(t)$. The leader's heading is accounted for in the i-th average by defining $b_i(t)$ to be 1 whenever the leader is a neighbor of agent i and 0 otherwise. The goal of this research was to show that for any initial set of followers with different headings, the headings of all n followers converge to the leader's heading.

Other research addresses formation changes combined with obstacle avoidance [23]. A lead robot is assumed to be equipped with all onboard sensors necessary to plan the gross motion for the team of robots. The algorithm developed allows the team of robots to move between any two formations while avoiding obstacles. The formations of robots are modeled as a triple

(g, r, H) where

- g represents the group variables (x, y, θ), of the lead robot in the formation
  - Denotes the gross motion of the team.
- r is a set of shape variables that describe the relative positions of other robots in the team
  - Governs the shape of the formation.
- H is a control graph which describes the control strategy used by each robot
  - The discrete coordination protocol employed by the team.

This research proceeds to model the formations using $n \times n$ adjacency matrices. A system of six robots has the formation shown in figure 2-5 and can be represented by the adjacency matrix G.



$$G = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2-5: Robot formation and adjacency matrix. A 1 in the (i, j) entry represents an edge of the incoming edge of robot R_i to R_j and a 0 corresponds to no edge between R_i to R_j.

This matrix is used as a control graph. When the formation changes, the transition fromone control graph to another results in a transition matrix. This serves to keep track of the difference between the final and initial adjacency matrices. The appearance of a 1 in a column of the control graph will define the controller for that robot.

Cooperative Mapping

Creating maps of an environment is a challenge in mobile robotics. Reliable sensor readings are a necessity and knowing what regions are worthy of exploration require a detailed

strategy. Research in multi-robot exploration has addressed mapping unknown environments using sensor data and odometry.

The research presented in Simmons [24] is a type of centralized control algorithm that approaches the problem with a distributed algorithm for concurrent mapping and localization. This approach makes two assumptions: (1) It assumes the world is static and cannot handle densely populated environments. (2) It assumes that the robots begin in view of one another and know their approximate relative location. The mapping problem is decomposed in a hierarchical fashion: Each robot maintains its own local map while correcting odometry errors as it travels. A central module receives the local maps and combines them into a single global map.

Each robot has a bid that consists of an estimate of the utilities required to travel to a particular location. The bids are sent to a "central executive" who tasks each robot based on all the bids, taking into account the potential overlap in map coverage. Bids are resubmitted once the robot's map has been updated. The bid location with the highest net utility as determined by the executive is assigned to the robot that made the bid. The experiments were performed on a three robot system with the primary performance metric being the amount of coverage and the time needed to completely explore the environment. Overall, the team of robots is able to cover 100% of the area for the cases of a single-corridor environment, and a random environment with and without obstacles.

Another aspect of the cooperative mapping is the idea of maximizing coverage while maintaining line-of-sight contact with the other members of the team. The algorithm outlined in [25] deploys the robots individually with each robot making use of information gathered by the previous robot in order to determine the next deployment location. Several assumptions are made:

1. The algorithm assumes that the robots are homogeneous – have all the same sensors, communication, and mobility.
2. The environment is assumed to be static.
3. The position of every robot is known in a global coordinate system.
4. Prior knowledge of the environment is unknown.

Because the environment is assumed to be unknown a key task of the robot team will be to generate a model of the environment. The algorithm proceeds as follows:

> *Initialization*: The robots are initially set to the state of *waiting*, with the exception of a single robot that is set to *deployed*. This robot is the anchor of the team.
>
> *Selection*: The deployment location is selected for the next node by analyzing a unified map of the environment, obtained from sensor data from the deployed nodes.
>
> *Assignment*: The selected goal is assigned to the first waiting robot, whose state then shifts from *waiting* to *active*.
>
> *Execution*: Active robots are deployed sequentially to their goal locations and the state of each robot is changed from *active* to *deployed* upon arrival at the goal.

The algorithm iterates until all robots have been deployed or the environment is completely covered. The sensor data from all of the deployed robots are combined to form an *occupancy* grid where each cell of the grid is designated as *free*, *occupied*, or *unknown*. A Bayesian technique is used to determine the probability that each cell is occupied; a threshold is applied to this probability to determine the state of each cell.

An interesting aspect of this method is the series of grids that are used to interpret the robot's next deployment location. Based on the occupancy grid a configuration grid and a reachability grid are constructed. The configuration grid represents the robots' configuration space – the clustering of occupied, free, or unknown space. Therefore, a robot can be placed in any free cell in the configuration grid. However, not all such cells are reachable; that is, a robot might be able to "see" free space but the opening to reach it may be too narrow to allow passage. The grids are then used in a set of four goal selection policies:

P1: randomly select a location in free/reachable space.
P2: randomly select a location on the free/unknown boundary.
P3: select the free space location that maximizes the coverage heuristic.
P4: select the free/unknown boundary location that maximizes the coverage heuristic.

A set of experiments were conducted in simulation for a team of 50 robots each equipped with a scanning laser range finder with a 360 degree field of view. For P1 and P2 (stochastic) ten trials were conducted from each of ten initial locations and for policies P3 and P4 (deterministic) a single trial was conducted for each of the ten initial locations. It was found that policies P2-P4 outperformed the control case, P1. Policies P3 and P4 produce a 3-fold improvement over simple random deployment. In addition, it is apparent that most of this improvement can be achieved using the boundary heuristic alone. Policies P3 and P4 are almost indistinguishable suggesting that the coverage heuristic will, in almost all situations, deploy nodes to the free/unknown boundary.

Reconnaissance and Communication

A common thread throughout the field of robotics is the reliance on communications. The ability of an operator or another robot to communicate with one another is of utmost importance. Communication failure can spell certain disaster for a mission.

The research of Nguyen [26] addresses the area of RF and digital communication and its degradation upon entering a building as applied to a group of mobile robots. In order to extend the range of the digital radios used in the experiment and provide service beyond line of sight, the research is exploring the use of mobile autonomous robots as relay nodes.

The communication software uses a link-state protocol where each node in the network has information about all links. The routing algorithm can then choose a suitable path. Each node broadcasts messages to specify the link characteristics and to setup a routing table. The routing table is recomputed when certain network events occur – such as the link quality dropping below

a prescribed level for a desired scenario.

Several relay deployment strategies were presented:

1. The lead robot proceeds alone with relay robots remaining at the base station. The lead robot will stop and call for a relay when its link becomes weak. A relay proceeds to travel to the current position of the lead robot and the lead robot continues on until the signal becomes weak again, and this process repeats.
2. All relay robots follow the lead robot in a convoying fashion. When the link between the lead robot and the base station becomes weak, the last relay robot stops while the rest of the convoy continues on. This process continues until all relays have been deployed.
3. Hybrid strategies that fall between 1 and 2.

The first strategy results in the least energy expenditure of the system. Each relay only has to move the minimum distance required, however this results in delays of the lead robot. More energy is required for the second strategy since the relay robots have to follow the lead robot; however, the lead robot does not suffer any delays. The objective of this research was to have complete transparency of the relaying function and any delay of the lead robot is unacceptable, and as a result strategy two is chosen.

Formation Control

The term "formation control" encompasses a broad field of robotics that addresses the challenge of coordinating a system of mobile robots. This term applies to aerial and ground vehicles as well as to robots of varying sizes. Previous research applies different control techniques and methodologies to achieve the goal of maintaining a formation of robots while in motion.

The incorporation of a "leader" into the formation is a common approach in which the motion of the lead robot is planned. In addition, the robots can have minimal sensing capabilities and can communicate to at least one other robot. In the case of control techniques, l-ψ control and l-l control [27] are examples of feedback controllers for maintaining mobile robot formations. In l-ψ control the aim is to maintain a desired length and a desired relative angle

between two robots in the formation. The task of *l*-*l* control involves stabilizing the distance of a third robot from the other two robots. Similarly, the goal is to maintain the desired lengths of the third robot from the two lead robots. This algorithm can be extended to formations with greater than three robots and also to environments containing obstacles.

Formation control of a system of mobile robots can often lead to other problems such as inter-robot collisions within the formation. This problem can be handled by placing constraints on the system of robots. The concept of a virtual robot and virtual robot tracking control [28] attempts to maintain clearance between the robots while navigating in formation. The robots are assumed to be homogeneous and modeled as differential drive, the workspace is flat and contains no obstacles, and each follower robot is indexed by a unique priority number. The virtual robot tracking control applies to a reference robot, a follower robot, and a virtual robot, all of whose positions and orientations are known. The virtual robot is hypothetical whose orientation is based on the follower robot but is positioned between the reference and follower. The separation created by the virtual robot avoids inter-robot collisions and simulations are presented for a three-robot case.

Maintaining a formation of robots is challenging in an unobstructed environment and becomes even more complicated when obstacles are introduced. A combination of formation control and obstacle avoidance is achieved using input-to-state stability and a general formation keeping algorithm [29]. The robots are modeled as differential drive vehicles and the overall control is based on system stability to environmental perturbations. Navigation of a three robot formation is achieved through a complex series of obstacles. Examples with greater than three robots would further validate this technique.

Formation control of a robot system involves generating robot movements that progress

in a general direction. For a set of disconnected, somewhat autonomous vehicles, this can pose problems. A different approach treats each robot as a particle embedded in a structure [30] called a *virtual structure*. The virtual structure serves to maintain a rigid geometric relationship among the mobile robots. The case of three differential drives robots in a triangular-shaped structure is presented in simulation and on an actual robot system. The virtual structure responds well to pure translations and pure rotations but incurs large errors when transitioning from translation to rotation and vice versa.

This is closely related to the research that will be presented in subsequent chapters however, “virtual springs” and “virtual dampers” are used to maintain robot formation. Additionally, the robots are modeled as front-steered (Ackermann) vehicles and different types of formations and numbers of robots are considered.

CHAPTER 3
PREVIOUS RESEARCH AT THE CENTER FOR INTELLIGENT MACHINES AND ROBOTICS

This chapter will discuss previous research experiences with ground vehicles at the Center for Intelligent Machines and Robotics at the University of Florida. The research performed in the Center requires inter-disciplinary knowledge of several engineering fields. The next few paragraphs will showcase the broad range of skills needed in applied robotics and will also summarize some of the air and ground vehicles currently in operation in the Center for Intelligent Machines and Robotics.

Single Vehicle Applications

Previous research in the laboratory primarily focused on single vehicle applications and most of the research was performed on a single autonomous vehicle that served as the test bed for experiments. This laid the foundation for future ground vehicle research. Over the years more vehicles were developed which allowed for more testing and sensor development.

Several smaller electric vehicles were developed as inexpensive alternative solutions to the larger golf cart sized autonomous vehicle. These smaller UGVs were easier to transport and served as a test bed for sensor development. The two smaller electric UGVs are shown in figure 3-1. In addition to their small size, they are relatively rugged and allowed for a quick turnaround during the automation process. Both platforms were equipped with a single board computer and wireless Ethernet. Different types of sensors can be easily programmed and integrated into the platform. Digital compass, GPS, encoders, vision systems, and laser measurement sensors are some of the sensors that have been integrated into these platforms.

While the electric vehicles were compact and easy to transport and test with, they had physical limitations. They could only run on relatively flat terrain and lacked an adequate suspension. The next generation of ground vehicle that was created was a small

gasoline powered ATV. This UGV is incredibly rugged and had a runtime of approximately six hours.

Figure 3-1: Electric vehicles developed at CIMAR

The platform shown in figure 3-2 is equipped with two computers, a power distribution system with custom enclosure and battery backup, and all necessary actuators for steering, throttle, and brake control.

Figure 3-2: Gasoline powered UGV

Each of these platforms was essential in advancing the lab's ground vehicle research. The platforms were entered in the International Ground Vehicle Competition from 2002-2004. This

competition is designed to test the limits of intelligent ground systems. Entering in these competitions afforded the opportunity to develop sensor systems in addition to testing the UGV's physical capabilities.

This resulted in a host of research in waypoint navigation, obstacle avoidance, image processing, and path planning. A suite of software was developed for testing each individual sensor and then fusing all of the data such that a "best path" could be derived.

Extensive research has been performed in image processing techniques. Developing a robust vision algorithm is a formidable task as subtle variations in lighting affect the overall performance of the algorithm. Most of the image processing research was based on the mean and covariance of the image and then applying a maximum likelihood technique to determine if pixels were within the desired range.

Vector quantization (VQ) was used to segment an image into regions of similar color properties. This technique was used predominantly to group pixels for further processing. A sample of the VQ technique is shown in figure 3-3.

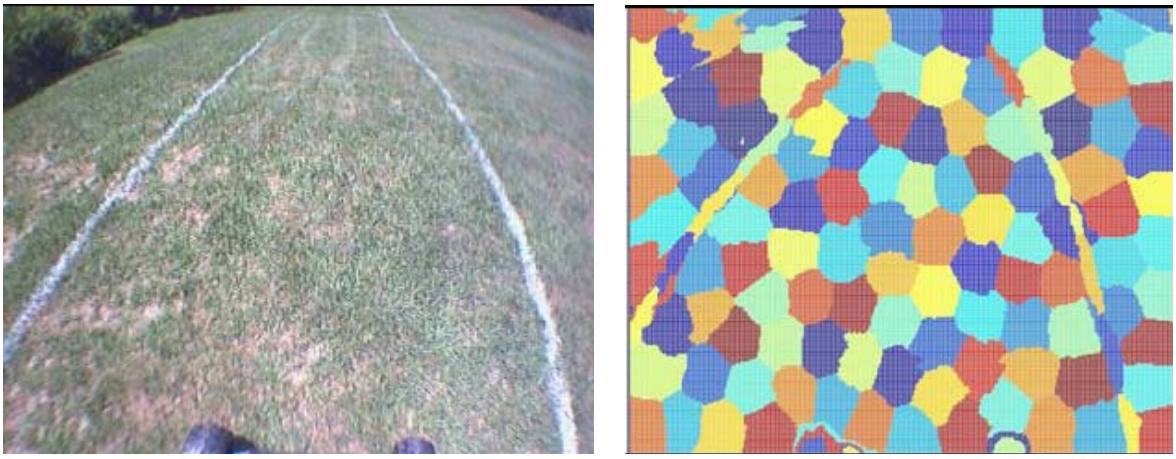


Figure 3-3: Sample image with VQ classified image.

As stated earlier, the majority of the image processing utilized the maximum likelihood technique. The parameters for the maximum likelihood classifier were determined by

segmenting several training images using the provided segmentation program as seen in figure 3-3. The maximum likelihood algorithm is summarized in figure 3-4.

Figure 3-4: The maximum likelihood algorithm.

The mean (μ) and covariance (∑) were both determined for each color channel: R, G, and B using equations (3-1) and (3-2) shown below. The probability of whether a specific pixel was part of the line was determined using equation (3-3). Results of this algorithm can be seen in figure 3-5.

$$\mu = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{3-1}$$

$$\Sigma = \frac{1}{n}\sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^T \tag{3-2}$$

$$p(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right] \tag{3-3}$$

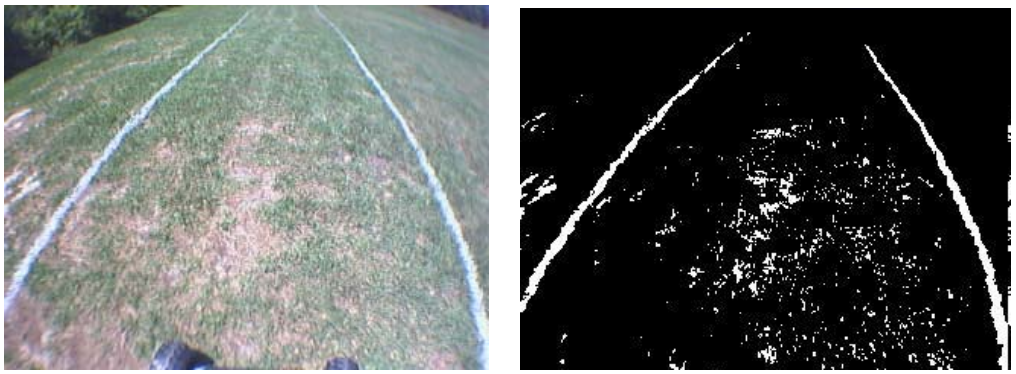


Figure 3-5: Sample image and ML processed image.

For testing purposes it was necessary to develop software that allows the operator to view sensor output in real-time. The vision system was very sensitive to changes in lighting and it became necessary to see the world as the robot sees it. The GUI in figure 3-6 was developed that

displays the camera image, the processed image, and the calculated path based on the sensor information obtained from the vision system.

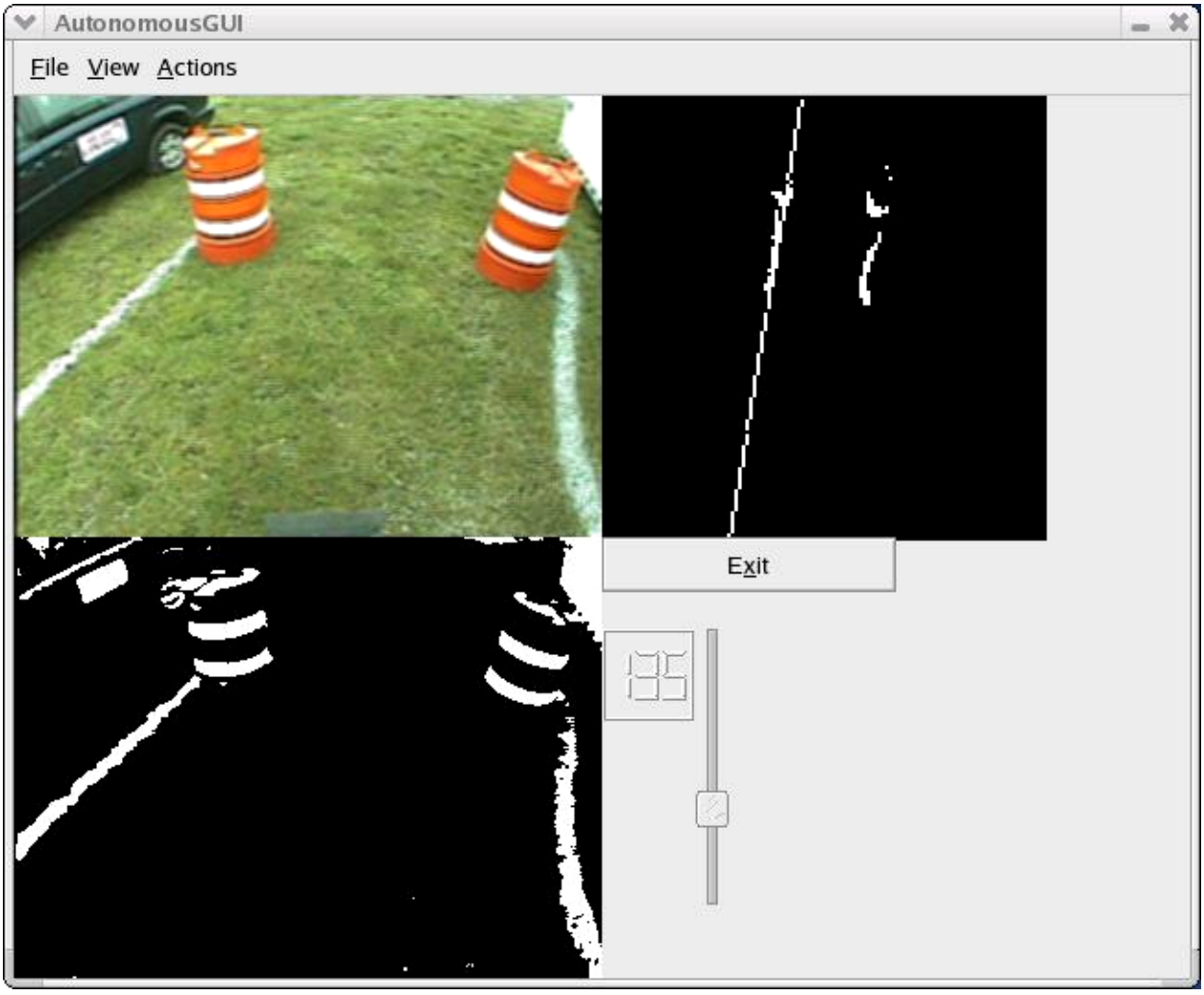


Figure 3-6: Image processing GUI.

In addition to having the ability to extract desired features from an image, it was necessary to be able to detect obstacles and successfully avoid them. The sensor of choice was a laser measurement device which emits a laser beam and then measures the time it takes for the reflection to return. A GUI seen in figure 3-7 was developed for this sensor as well to facilitate in the debugging process.

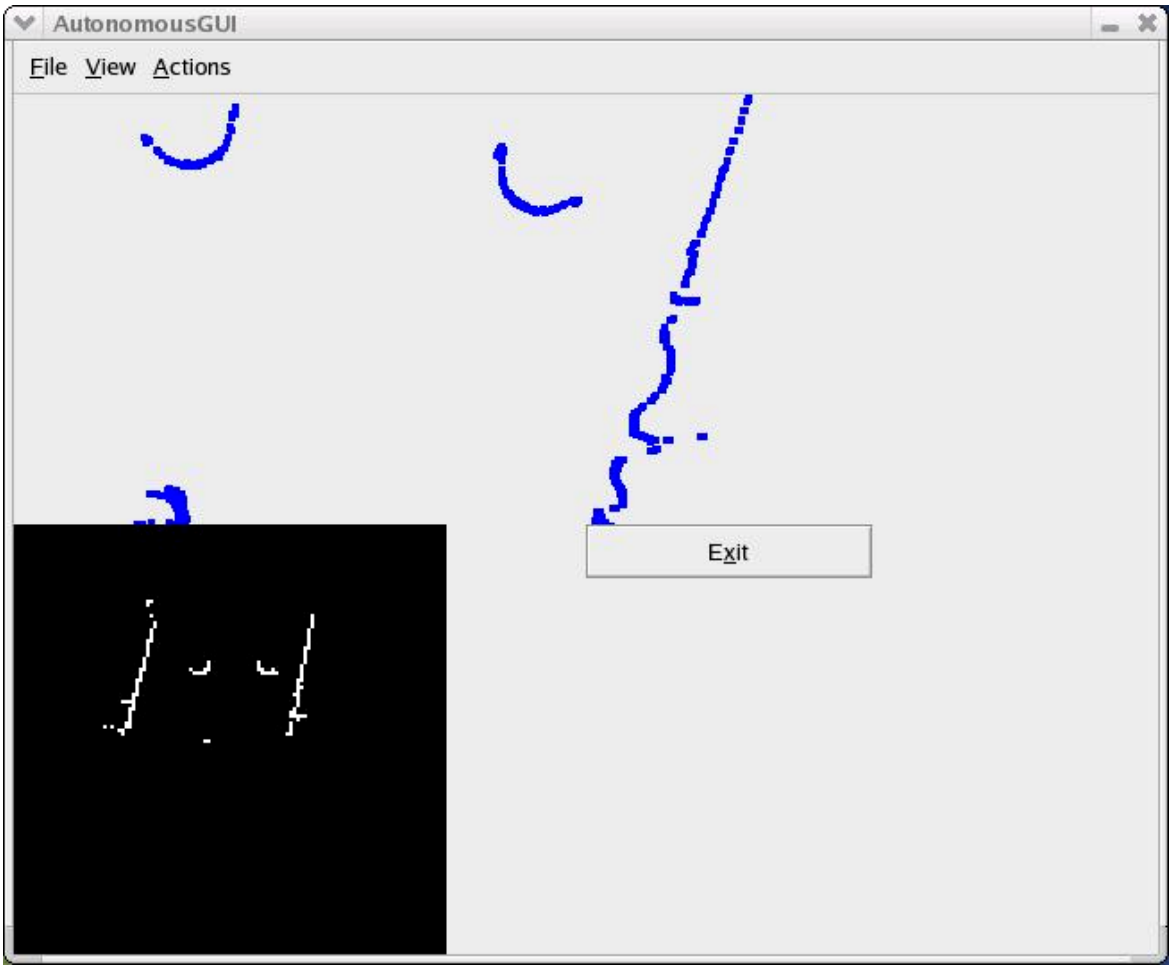


Figure 3-7: Obstacle detection GUI.

Once the data has been captured by the individual sensors an overall view of the area surrounding the vehicle was created. A binary grid in the vehicle's coordinate system was filled as either 1 or 0 based on the sensor readings. Based on the fused sensor data and the resulting grid, a path was planned that consisted of a heading for the vehicle to travel along. Figure 3-8 displays the grid, then an expansion of all points in the grid - this accounts for any errors in sensor readings. Finally the rightmost image shows the path derived from all sensor information.

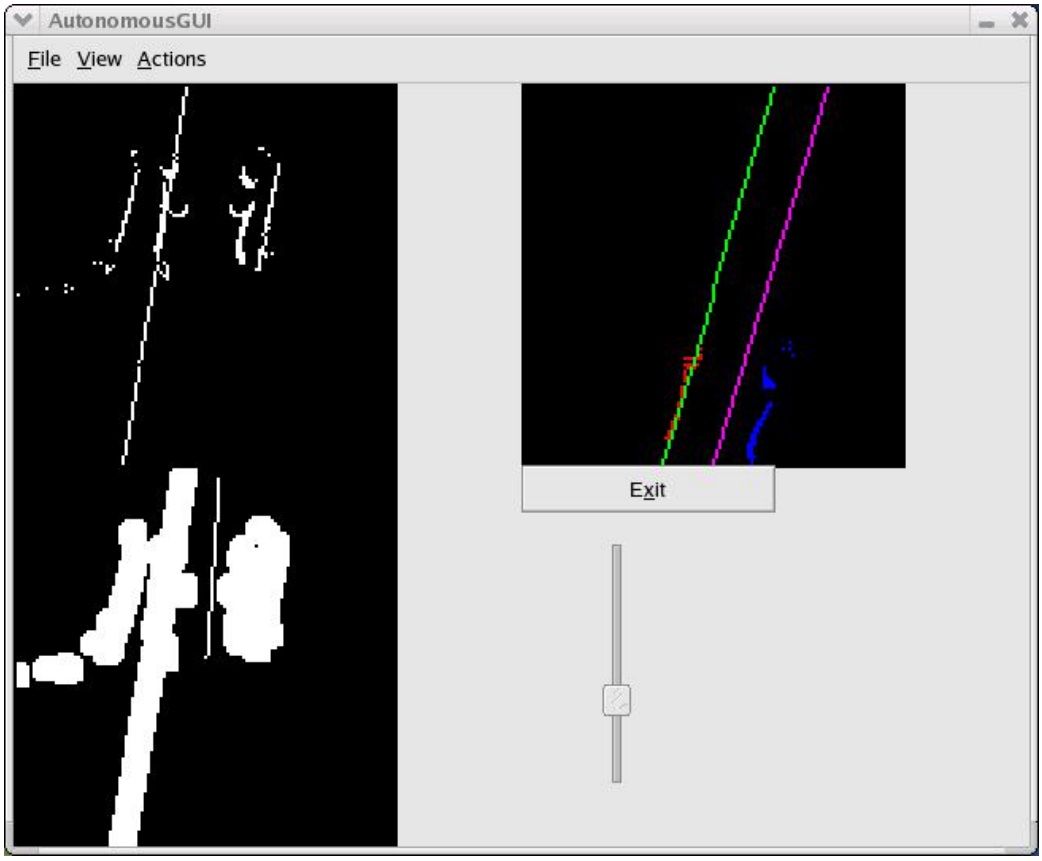


Figure 3-8: Path planner GUI.

Multi-Vehicle Applications

With the development of the robotic fleet and associated technologies it became feasible to begin multi-vehicle investigations. Initial research was performed in a simulation environment. However, the simulation was designed such that implementation on an actual system would be a simple transition.

Preliminary Multi-Vehicle Simulation

A convoying strategy was initially tested for a three vehicle simulation. A leader would navigate to a series of waypoints and the remaining vehicles would follow. The following behavior was based on a distance metric and the calculated following distance was used to regulate the velocity. A block diagram of the control strategy is shown in figure 3-9.

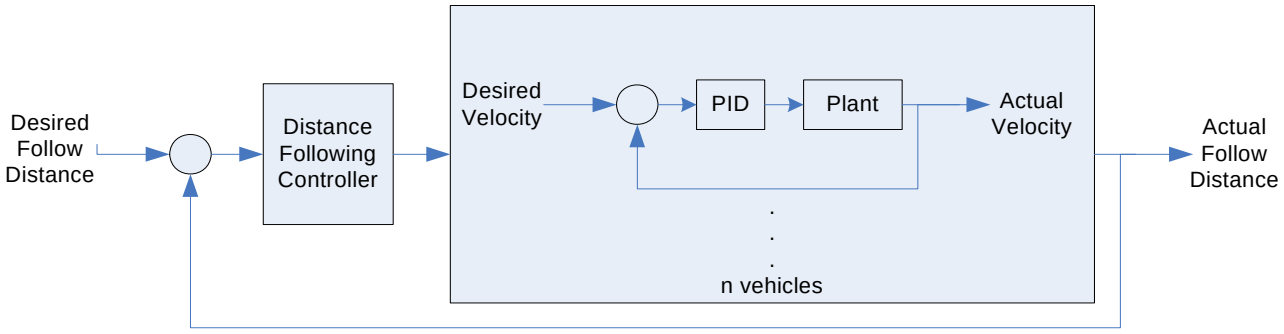


Figure 3-9: Distance following control strategy diagram.

A PD controller was implemented for the distance following. The vehicles were given a desired follow distance of five meters and would regulate their speeds accordingly. For the two follower vehicles, the plots of the following distances are shown in figure 3-10. The vehicles' follow distance is slightly over six meters and this discrepancy can be accounted for due to the method in which the distance from one robot to the next is calculated. This is performed using GPS data which is known to vary over time.

Preliminary Multi-Vehicle Implementation

Part of the allure of engineering is the ability to take an idea, develop a design on paper, then prototype it, and finally test it rigorously. Moving from a robotic simulation to implementation on an actual system has similar challenges. Sometimes the robot reacts in ways that, at first glance, boggle the designer. Similar to the simulation, a convoy strategy experiment was performed. A leader would travel to waypoints while the follower would try to maintain a desired distance of ten meters from the lead vehicle. A plot of the follow distance is shown in figure 3-11. The desired follow distance is ten meters and the two followers have some issues at maintaining this which is again due to GPS drift and ill-tuned vehicle controls.

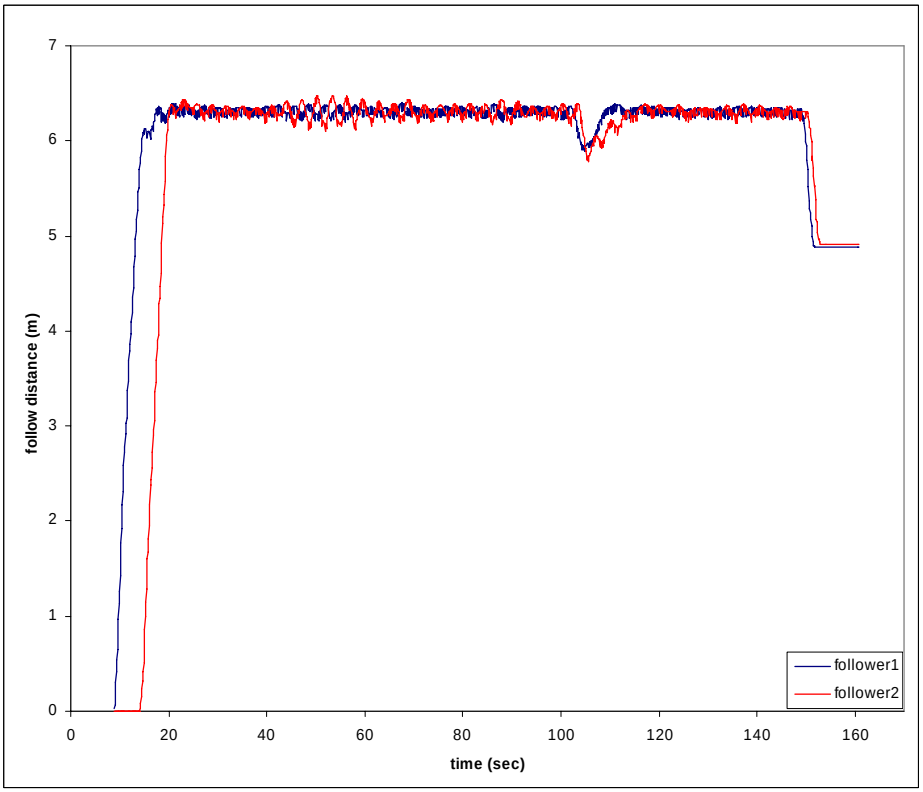


Figure 3-10: Following distances for each vehicle (simulation).

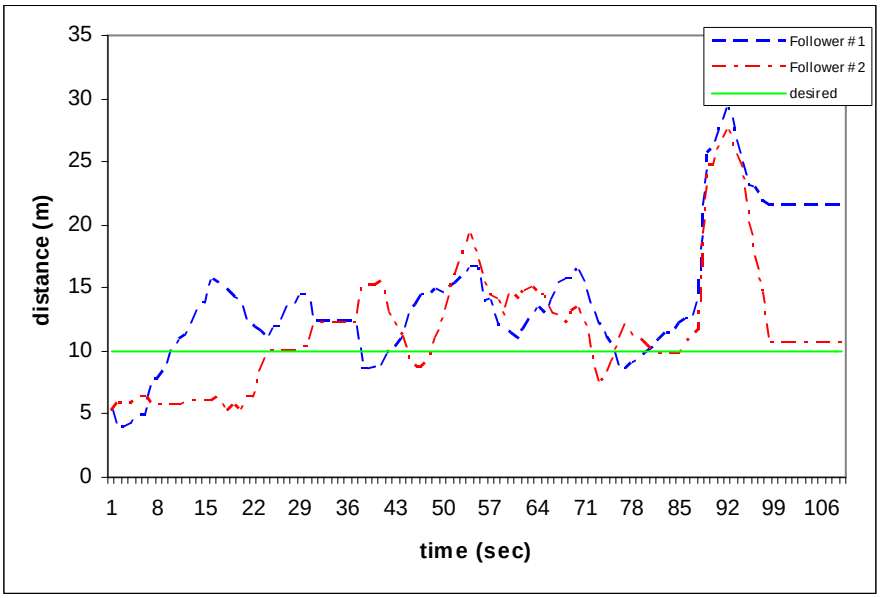


Figure 3-11: Vehicle follow distance on an actual system.

Additional sensor information, such as an encoder, would aid in this experiment so that the follow distance doesn't rely solely on the GPS. These preliminary experiments however have provided insight into all of the challenges that are present in this type of research.

UAV and UGV Applications

Thus far, this chapter has only addressed ground vehicle research. However, there is a vested interest in unmanned aerial vehicle research. There is also interest in having a cooperative UAV/UGV system for possible use in agricultural applications, search and rescue, and military reconnaissance operations. The research presented here addresses explosive ordnance disposal using a UAV/UGV system. This highly dangerous task is currently performed by military personnel and does not allow any room for error. Automation of this task would take the human out of harm's way.

The system consisted of the gas powered UGV and a gas XCell RC style helicopter shown in figure 3-12. The helicopter is equipped with a modular sensor payload that consists of a GPS, digital compass, stereovision camera system, and all necessary computing and power.

Figure 3-12: UGV and UAV in flight.

The experiment was conducted by surveying the target area with the UAV and creating a map of the area. The aerial map was transmitted to the base station and post-processed to extract the locations of the targets and develop waypoints for the ground vehicle to navigate. The

ground vehicle then proceeded to each of the targets, which simulated the validation, and disposal of the ordnance. Results include the aerial map, processed images of the extracted ordnances, and the ground vehicle's ability to navigate to the target points.

For this application, a precise camera model and an image-to-global coordinate transformation [31] were developed. This involved finding the intrinsic and extrinsic camera parameters of the camera system attached to the aerial vehicle. The intrinsic camera parameters were determined using a camera calibration toolbox for MATLAB® [32]. The camera model and image transformation was derived and is described in detail in [33].

In order to evaluate the performance of the UAV/UGV system, the waypoints were surveyed using a Novatel RT-2 differential GPS. This system provides two centimeter accuracy or better when provided with a base station correction signal. Accurate surveying of the visited waypoints provided a baseline for comparison of the results obtained from the helicopter and the corresponding path the ground vehicle traversed.

The UXOs were simulated to resemble BLU-97 ordnance. An aerial photograph of the ordnance along with the camera position and orientation were collected and is shown in figure 3-14. Using the transformation described previously, the global coordinates of the UXOs were calculated. The calculated UXO positions were compared with the precision survey data. A local map of the operating region was generated using the precision survey data. This local map shown in figure 3-13 provided a baseline for all of the position comparisons.

The results compare the positioning ability of the UGV and the ability of the UAV sensor system to accurately calculate the UXO positions. While both the UGV and UAV use WAAS enabled GPS there is some inherent error due to vehicle motion and environmental affects. The UGV's control feedback is based on waypoint to waypoint control versus a path following

control algorithm.

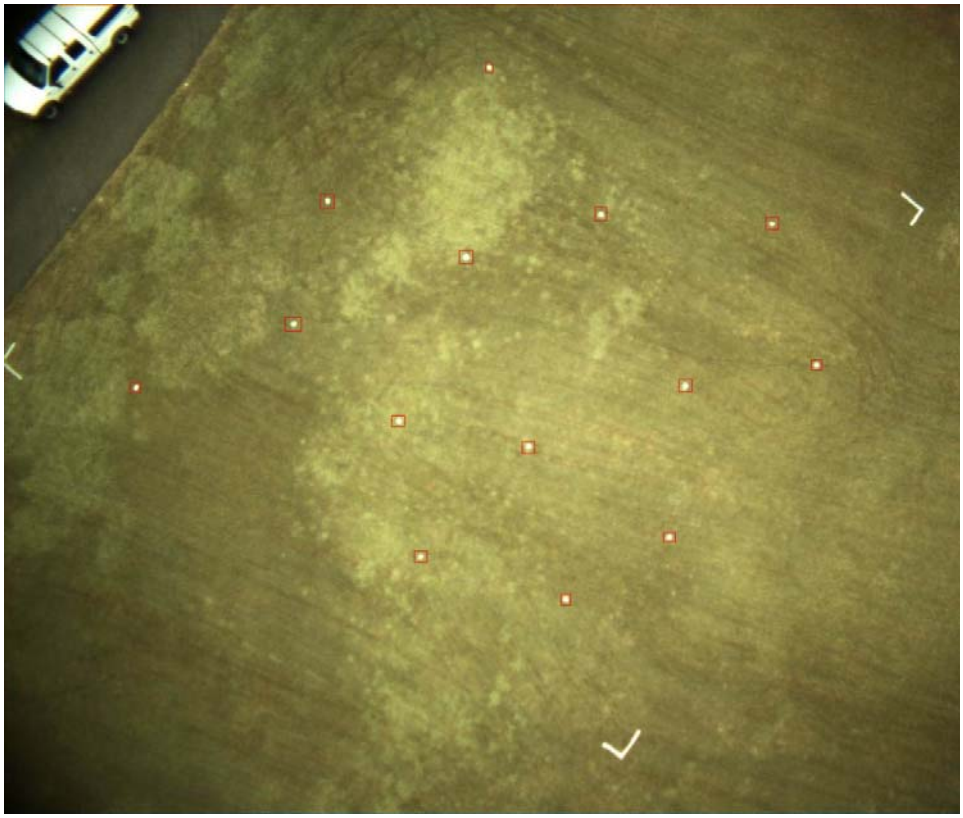


Figure 3-13: Aerial photograph of all simulated ordnance.

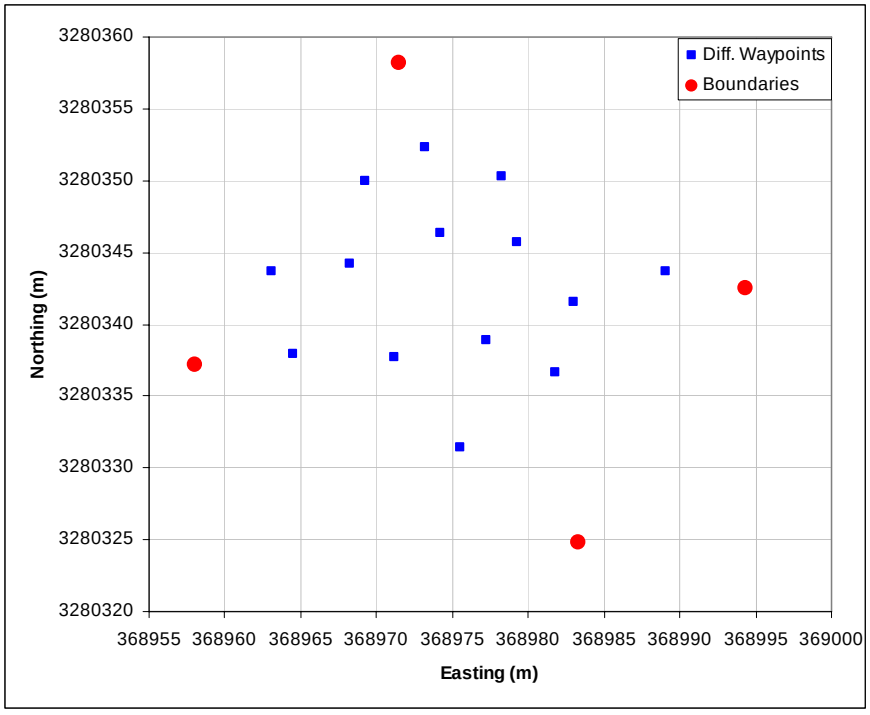


Figure 3-14: Local map generated with Novatel differential GPS.

The UGV was commanded to come within a specified threshold of a waypoint before switching to the next waypoint and is shown in figure 3-15. The UGV consistently traveled within three meters or less of each of the desired waypoints which is within the error envelope of typical WAAS GPS accuracy.

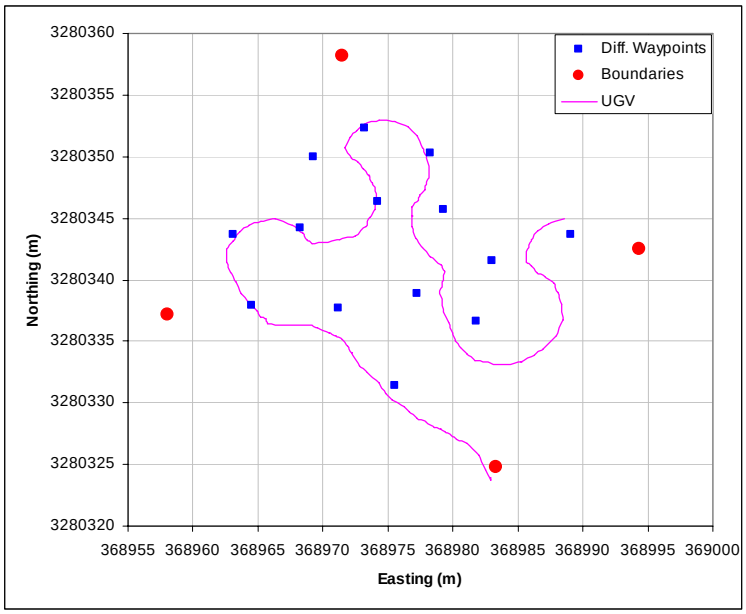


Figure 3-15: A comparison of the UGV's path to the differential waypoints

The UAV calculates the waypoints based on its sensors and these points are compared with the surveyed waypoints. From figure 3-15 there is an offset in the UAV's data due to the GPS and to errors in the transformation from image coordinates to global coordinates. In addition, the waypoints that the UAV determined were given directly to the UGV once they were sorted as seen in figure 3-16. A comparison between the waypoints the UAV calculated and the path the UGV traverses is shown in figure 3-17. The ground vehicle successfully visits each target to within the specified threshold.

The UGV is able to navigate within several meters of the waypoints, however, is limited due to the vehicle kinematics. Further work involves a waypoint sorting algorithm that accounts for the turning radius of the vehicle.

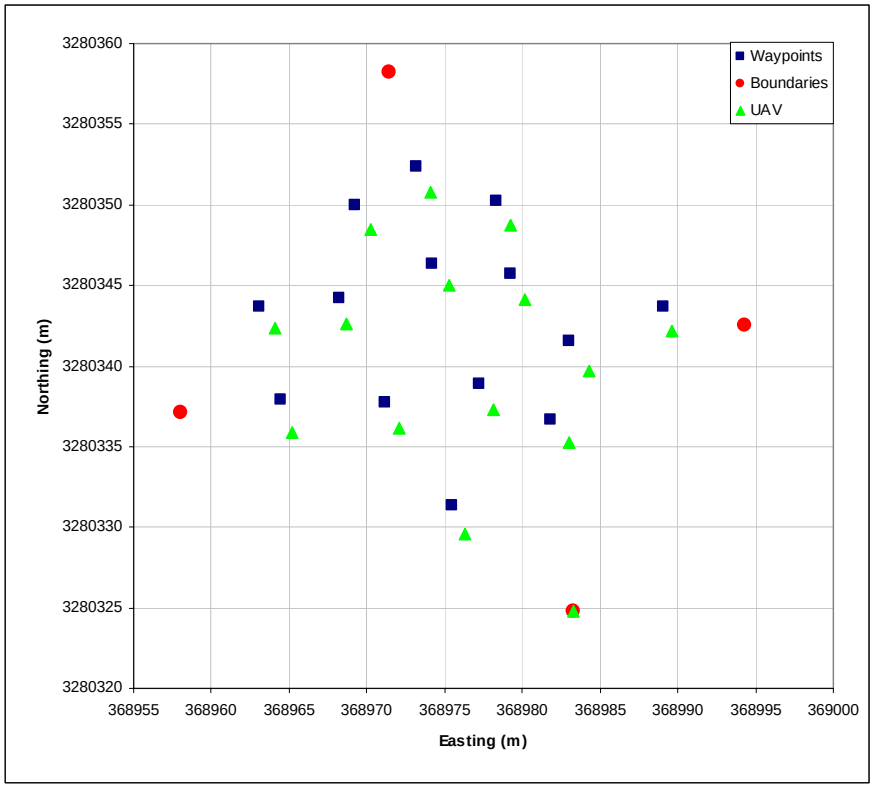


Figure 3-16: UAV vs. differential GPS

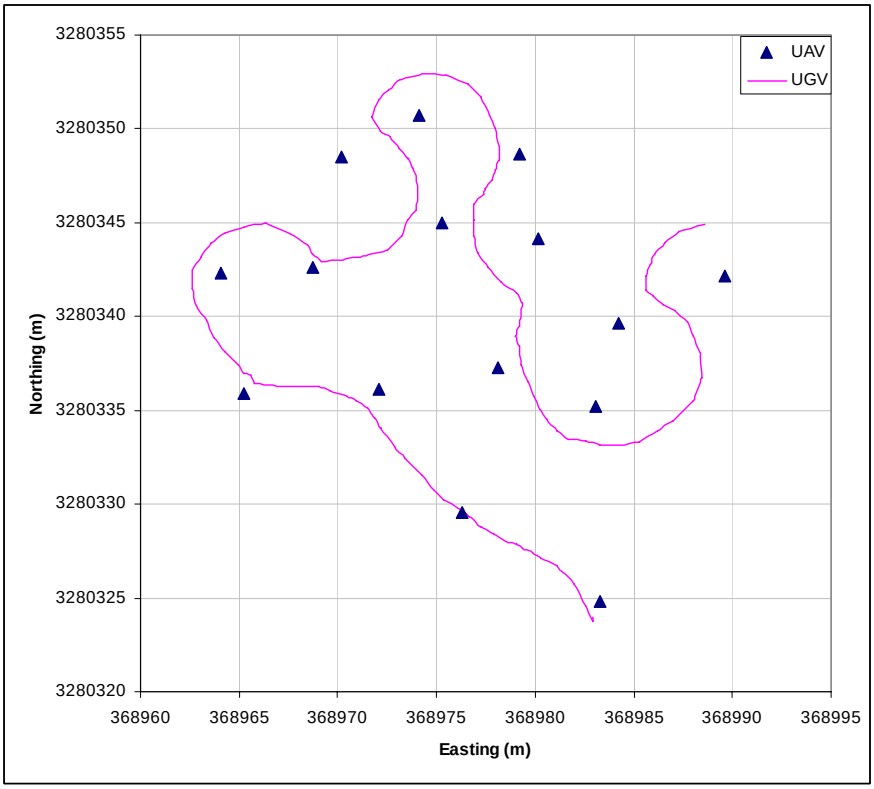


Figure 3-17: UAV waypoints vs. UGV path

CHAPTER 4
DYNAMICS, KINEMATICS, AND FORCE MODELING

Much of the research presented thus far has utilized simulations to validate a formation control algorithm and robot platforms are modeled using differential drive mobility. This chapter derives the equations of motion of a front-steered vehicle and the forces applied to the vehicle. A detailed model of the front-steered vehicle will be developed and will be used in simulation. The formation control laws will be derived based on virtual springs that keep the vehicles in a desired formation. These derivations provide the foundation for compliant formation control and are implemented in the simulation environment.

Modeling Vehicle Dynamics and Kinematics

An Ackermann steering configuration allows only the front wheels of a vehicle to turn. In order to reduce the complexity of the problem the following constraints and assumptions are made: no lateral motion of any wheel, the front wheels turn at the same angle, the wheels roll without slipping.

Figure 4-1 diagrams the vehicle geometry and all associated coordinate systems. The vehicle is modeled after a tricycle in order to reduce the complexity of the analysis. This accounts for the assumption that both front wheels turn at the same angle. The generalized coordinates for the system are as follows:

q_1 = x: The position of point *p* in the $\hat{f}_1$ direction.

q_2 = y: The position of point *p* in the $\hat{f}_2$ direction.

The ‘F’ reference frame is the inertial frame, fixed to ground. The ‘A’ reference frame is fixed in the rear of the vehicle as shown in figure 4-1. Similarly, the ‘B’ reference frame is fixed in the front wheel. For this particular system, the throttle force is constrained to be in the $\hat{a}_1$ direction. A constraint is placed upon the velocity of point *p*. That is, the center point of the rear

axle is constrained to move in the $\hat{a}_1$ direction.

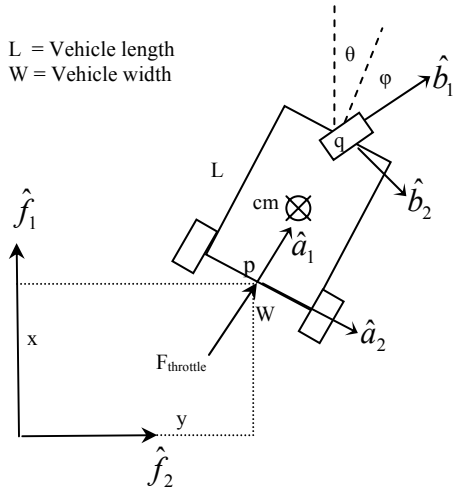


Figure 4-1: Vehicle geometry and coordinate systems.

In addition, a constraint is placed upon the velocity of point *q* – the front wheel. Its velocity is constrained to be in the $\hat{b}_1$ direction. The following constraint equations are developed for point *p* and point *q*:

$$\overline{V}_p = V_p \hat{a}_1 = \sqrt{\dot{x}^2 + \dot{y}^2}\,\hat{a}_1 \tag{4-1}$$

$$\overline{V}_q = V_q \hat{b}_1 \tag{4-2}$$

These two equations constrain the velocity of the rear of the vehicle to be parallel to the rear wheels and the velocity of the front of the vehicle to be in the direction of the angle of the front wheel. Since points *p* and *q* lie in the same rigid body, the following velocity relationship applies to the rigid body:

$$\overline{V}_q = \overline{V}_p + {}^F\overline{\omega}^A \times \overline{r}_{p\to q} \tag{4-3}$$

$$\overline{V}_q = \overline{V}_p + \dot{\theta}\hat{a}_3 \times L\hat{a}_1 \tag{4-4}$$

$$\overline{V}_q = V_p \hat{a}_1 + L\dot{\theta}\hat{a}_2 = V_q \hat{b}_1, where: \hat{b}_1 = \sin\varphi \hat{a}_2 + \cos\varphi \hat{a}_1 \tag{4-5}$$

$$V_q \cos\varphi = V_p = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{4-6}$$

$$V_q \sin\varphi = L\dot{\theta}, where: L\dot{\theta} = \tan\varphi\sqrt{\dot{x}^2 + \dot{y}^2} \tag{4-7}$$

This produces the following relationship:

$$\dot{\theta} = \frac{\tan\varphi\sqrt{\dot{x}^2 + \dot{y}^2}}{L} \tag{4-8}$$

It is assumed that the vehicle center of mass (cm) is located at the center of the vehicle as shown in figure 4-1. Since the center of mass and point *p* lie in the same rigid body, the velocity relationship is applied to the system:

$$\begin{aligned} \overline{V}_{cm} &= \overline{V}_p + {}^F\overline{\omega}^A \times \overline{r}_{p\to cm} \\ &= \sqrt{\dot{x}^2 + \dot{y}^2}\,\hat{a}_1 + \frac{\dot{\theta}L}{2}\hat{a}_2 = \sqrt{\dot{x}^2 + \dot{y}^2}\,\hat{a}_1 + \frac{\tan\varphi\sqrt{\dot{x}^2 + \dot{y}^2}}{2}\hat{a}_2 \end{aligned} \tag{4-9}$$

The result is an expression for the magnitude of the velocity of the center of mass.

$$V_{cm} = \sqrt{\left(\dot{x}^2 + \dot{y}^2\right)\left(1 + \frac{\tan^2\varphi}{4}\right)} \tag{4-10}$$

The rigid-body system is analyzed using Lagrange's method. Therefore, the kinetic and potential energy of the system are derived and the Lagrangian is developed:

$$L = T - V \tag{4-11}$$

$$L = \frac{1}{2}m\overline{V}_{cm} \bullet \overline{V}_{cm} + \frac{1}{2}{}^F\overline{\omega}^A \bullet [I]\,{}^F\overline{\omega}^A - 0 \tag{4-12}$$

$$T = \frac{1}{2}m\overline{V}_{cm} \bullet \overline{V}_{cm} + \frac{1}{2}{}^F\overline{\omega}^A \bullet [I]\,{}^F\overline{\omega}^A = \frac{1}{2}m\left(\dot{x}^2 + \dot{y}^2 + \frac{\tan^2\varphi\left(\dot{x}^2 + \dot{y}^2\right)}{4}\right) + \frac{1}{2}\dot{\theta}^2 I_{zz} \tag{4-13}$$

$$T = \frac{1}{2}\left(\dot{x}^2 + \dot{y}^2\right)\left[m + \frac{m\tan^2\varphi}{4} + I_{zz}\frac{\tan^2\varphi}{L^2}\right] \tag{4-14}$$

$$V = 0 \tag{4-15}$$

The kinetic energy of the system is composed of linear and rotational kinetic energy. This particular system undergoes no change in the z-direction which causes the potential energy term to become zero. The following Lagrange's equations are solved in order to obtain the equations of motion of the system:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1} = Q_1 \tag{4-16}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2} = Q_2 \tag{4-17}$$

where q_1 and q_2, the generalized coordinates, correspond to the *x* and *y* coordinates of point *p*. The generalized forces, Q_1 and Q_2 are derived in the following manner:

$$Q_i = \sum_{i=1}^{2} \overline{F} \bullet \frac{\partial \overline{r}}{\partial q_i}$$
$$where: \overline{r} = x\hat{f}_1 + y\hat{f}_2 \tag{4-18}$$
$$\frac{\partial \overline{r}}{\partial q_1} = \hat{f}_1, \frac{\partial \overline{r}}{\partial q_2} = \hat{f}_2$$

$$Q_1 = f_{throttle}\hat{a}_1 \bullet \hat{f}_1 = f_{throttle}\left(\cos\theta \hat{f}_1 + \sin\theta \hat{f}_2\right) \bullet \hat{f}_1 \tag{4-19}$$

$$Q_1 = f_{throttle}\cos\theta, \quad Q_2 = f_{throttle}\sin\theta \tag{4-20}$$

For the first generalized coordinate, q_1:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1} = Q_1 \tag{4-21}$$

$$\frac{\partial L}{\partial \dot{q}_1} = \dot{x}\left(m + \frac{m\tan^2\varphi}{4} + \frac{I_{zz}\tan^2\varphi}{L^2}\right) \tag{4-22}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) = \ddot{x}\left(m + \frac{m\tan^2\varphi}{4} + I_{zz}\frac{\tan^2\varphi}{L^2}\right) + \dot{x}\dot{\varphi}\left(\frac{m\tan\varphi}{2\cos^2\varphi} + 2I_{zz}\frac{\tan\varphi}{L^2\cos^2\varphi}\right), \frac{\partial L}{\partial q_1} = 0 \tag{4-23}$$

This yields the first equation of motion:

$$\boxed{\ddot{x}\left(m + \frac{m\tan^2\varphi}{4} + \frac{I_{zz}\tan^2\varphi}{L^2}\right) + \dot{x}\dot{\varphi}\left(\frac{m\tan\varphi}{2\cos^2\varphi} + \frac{2I_{zz}\tan\varphi}{L^2\cos^2\varphi}\right) = f_{throttle}\cos\theta} \tag{4-24}$$

For the second generalized coordinate, q_2:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2} = Q_2 \tag{4-25}$$

$$\frac{\partial L}{\partial \dot{q}_2} = \dot{y}\left(m + \frac{m\tan^2\varphi}{4} + \frac{I_{zz}\tan^2\varphi}{L^2}\right) \tag{4-26}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) = \ddot{y}\left(m + \frac{m\tan^2\varphi}{4} + I_{zz}\frac{\tan^2\varphi}{L^2}\right) + \dot{y}\dot{\varphi}\left(\frac{m\tan\varphi}{2\cos^2\varphi} + 2I_{zz}\frac{\tan\varphi}{L^2\cos^2\varphi}\right), \frac{\partial L}{\partial q_2} = 0 \tag{4-27}$$

This yields the second equation of motion:

$$\boxed{\ddot{y}\left(m + \frac{m\tan^2\varphi}{4} + \frac{I_{zz}\tan^2\varphi}{L^2}\right) + \dot{y}\dot{\varphi}\left(\frac{m\tan\varphi}{2\cos^2\varphi} + \frac{2I_{zz}\tan\varphi}{L^2\cos^2\varphi}\right) = f_{throttle}\sin\theta} \tag{4-28}$$

The inputs to the system are the steering angle, φ, and the $f_{throttle}$ command. The system inputs and the initial conditions along with the equations of motion are used to simulate the motion of the vehicle over time. Integration of the acceleration equation will provide the velocity and position equations of the vehicle. These equations are used in the simulation to provide realistic motion of the simulated robots. It is interesting to note that there exists a

singularity in the system when $\varphi = \pm 90°$. This implies that with the front wheel turned to either extreme (right or left), the vehicle cannot move and thus the equations of motion are indeterminate.

Force Modeling and Formation Control Laws

Without Damping

The formation of robots is modeled as a series of masses and virtual springs that stretch and compress based on the dynamics of the robot system. As the system of robots traverses throughout the environment the forces applied on each vehicle changes. The motion of each vehicle is based on the applied forces and responds accordingly while striving to maintain formation.

The robot system consists of one leader vehicle and two or more follower vehicles. The interconnections of the system depend on the desired formation shape. A sample wedge and vee formation for a three-robot system is shown in figure 4-2. The ⌀ in figure 4-2 represents a torsional spring that is placed between the lead vehicle and follower one and two.

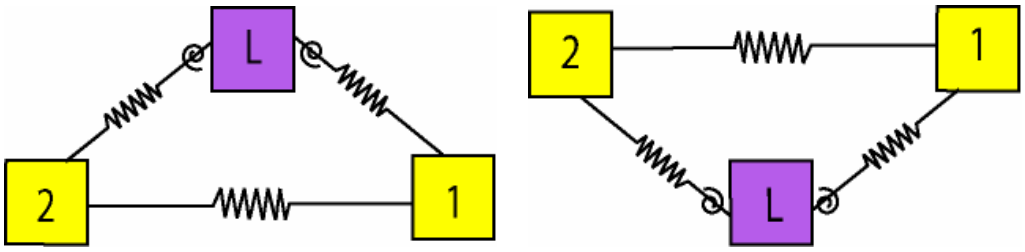


Figure 4-2: Sample wedge formation and vee formation.

The formation control algorithm is dependent on the forces that are applied to each individual robot as the system moves. The torsional spring connected to the leader provides the formation with orientation. The combination of the linear and torsional spring forces acting on each vehicle result in formation control laws for the entire robot system.

The individual forces acting on each vehicle consist of forces due to the linear and torsional springs. A generic notation for the force equations were developed for each spring type.

The generic linear spring force vector notation is presented as:

$$\overline{F_{ij}} = k_{ij}\left[\left(\left|\overline{L_{ij}}\right| - DL_{ij}\right)\left(\frac{\overline{L_{ij}}}{\left|\overline{L_{ij}}\right|}\right)\right] \tag{4-29}$$

where

$\overline{L_{ij}} = \overline{r_j} - \overline{r_i}$ and DL_{ij} is the desired length between vehicle i and vehicle j and

where k_{ij} represents the linear spring constant between vehicle i and vehicle j.

The generic torsional spring force vector notation:

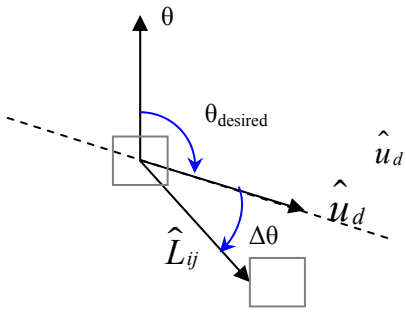
$$\overline{F_{t\_ij}} = \frac{k_{t\_ij}\left(\Delta\theta_{ij}\right)}{\left|\overline{L_{ij}}\right|}\left(\frac{\hat{k}\times\overline{L_{ij}}}{\left|\overline{L_{ij}}\right|}\right) \tag{4-30}$$

where

$$\Delta\theta_{ij}\hat{k} = \tan^{-1}\left(\frac{\left(\hat{u}_d\times\hat{L}_{ij}\right)\hat{k}}{\hat{u}_d\bullet\hat{L}_{ij}}\right) \text{ and } \hat{L}_{ij} = \frac{\overline{L_{ij}}}{\left|\overline{L_{ij}}\right|} \tag{4-31}$$

and where k_{t_ij} represents the torsional spring constant between vehicle iand j.

From the above equation the vector $\overline{u_d}$ requires further explanation. The diagram in figure 4-3 illustrates the desired vector between two vehicles. The resultant forces are calculated for each follower vehicle using equations (4-29) – (4-32). An example of a three-robot system in a wedge formation is shown in figure 4-4. This system will have a desired spacing, or "follow distance" between the robots. Also, a desired follow angle" will dictate the overall shape of the robot formation and is referenced from the lead robot's current heading.



$$\hat{u}_d = \cos\left(\theta \mp \theta_{desired}\right)\hat{i} + \sin\left(\theta \mp \theta_{desired}\right)\hat{j} \tag{4-32}$$

Figure 4-3: Desired and actual vectors and angles for follower robot.

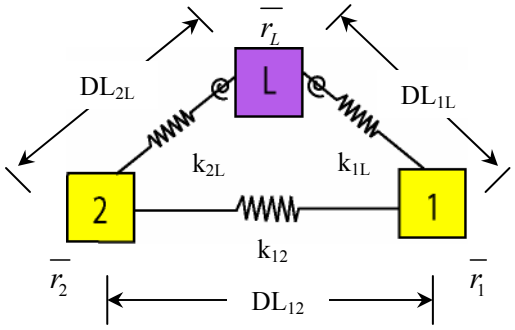


Figure 4-4: Three-robot system in wedge formation.

Free body diagrams of the two follower robots are shown in figure 4-5. Derivation of the forces applied to each robot follows.

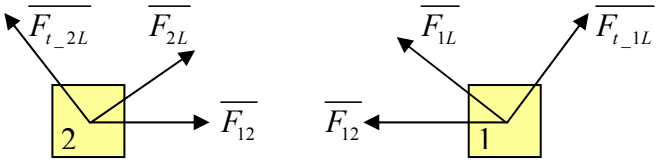


Figure 4-5: Free body diagrams of robot 1 and 2.

The forces applied to robot 1 due to the linear springs:

$$\overline{F_{1L}} = k_{1L}\left(\left[\left|\overline{L_{1L}}\right| - DL_{1L}\right]\frac{\overline{L_{1L}}}{\left|\overline{L_{1L}}\right|}\right) \tag{4-33}$$

$$\overline{F_{12}} = k_{12}\left(\left[\left|\overline{L_{12}}\right| - DL_{12}\right]\frac{\overline{L_{12}}}{\left|\overline{L_{12}}\right|}\right) \tag{4-34}$$

The force applied to robot 1 due to the torsional spring:

$$\overline{F_{t\_1L}} = \frac{k_{t\_1L}\Delta\theta_{1L}}{\left|\overline{L_{1L}}\right|}\left(\frac{\overline{L_{1L}}}{\left|\overline{L_{1L}}\right|}\right) \tag{4-35}$$

The resultant force is the sum of each of the forces applied to the body:

$$\overline{F_{1\_result}} = \overline{F_{1L}} + \overline{F_{12}} + \overline{F_{t\_1L}} \tag{4-36}$$

The forces applied to robot 2 due to the linear springs:

$$\overline{F_{2L}} = k_{2L}\left(\left[\left|\overline{L_{2L}}\right| - DL_{2L}\right]\frac{\overline{L_{2L}}}{\left|\overline{L_{2L}}\right|}\right) \tag{4-37}$$

$$\overline{F_{12}} = k_{12}\left(\left[\left|\overline{L_{12}}\right| - DL_{12}\right]\frac{\overline{L_{12}}}{\left|\overline{L_{12}}\right|}\right) \tag{4-38}$$

The force applied to robot 2 due to the torsional spring:

$$\overline{F_{t\_2L}} = \frac{k_{t\_2L}\Delta\theta_{2L}}{\left|\overline{L_{2L}}\right|}\left(\frac{\hat{k}\times\overline{L_{2L}}}{\left|\overline{L_{2L}}\right|}\right) \tag{4-39}$$

The resultant force is the sum of each of the forces applied to the body:

$$\overline{F_{2\_result}} = \overline{F_{2L}} + \overline{F_{12}} + \overline{F_{t\_2L}} \tag{4-40}$$

With Damping

For the case including damping, the formation of robots is modeled as a series of masses, virtual springs, and virtual dampers that stretch and compress based on the dynamics of the robot system. As the system of robots traverse throughout the environment the forces applied on each vehicle changes. It is hypothesized that addition of the dampers to the robot systems will reduce any oscillatory motion of the system.

The robot system consists of one leader vehicle and two or more follower vehicles. The interconnections of the system depend on the desired formation shape. A sample wedge and vee

formation for a three-robot system with damping is shown in figure 4-6.

Similar to the case without damping there are forces acting on each vehicle due to the linear and torsional springs with the addition of forces due to the dampers.

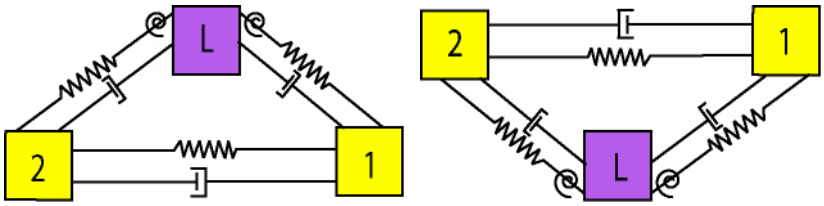


Figure 4-6: Sample wedge and vee formation with damping.

The force due to damping may be written as:

$$\overline{F_{D\_ij}} = c_{D\_ij}\left[\frac{\left(\left|\overline{L_{ij\_previous}}\right| - \left|\overline{L_{ij\_current}}\right|\right)}{\Delta t}\left(\frac{\overline{L_{ij}}}{\left|\overline{L_{ij}}\right|}\right)\right] \tag{4-41}$$

where c_{D_ij} represents the damping constant between vehicle i and j.

Therefore, the resultant forces on robot 1 and robot 2 will have an additional term due to the damping force. The corresponding resultant forces for robot 1 and robot 2 are in equations 4-42 and 4-43.

$$\overline{F_{1\_result}} = \overline{F_{1L}} + \overline{F_{12}} + \overline{F_{t\_1L}} + \overline{F_{D\_12}} + \overline{F_{D\_1L}} \tag{4-42}$$

$$\overline{F_{2\_result}} = \overline{F_{2L}} + \overline{F_{12}} + \overline{F_{t\_2L}} + \overline{F_{D\_12}} + \overline{F_{D\_2L}} \tag{4-43}$$

Velocity and Force Control

The resultant force applied to the body is ultimately used to regulate the velocity and steering angle of the follower robots. There are two layers of control that are applied to the robot system. The outer loop feedback layer controls the vehicle formation by regulating the applied force by commanding a desired velocity. The inner loop feedback layer tracks the desired velocity using throttle actuation. The controllers are conceptualized as follows:

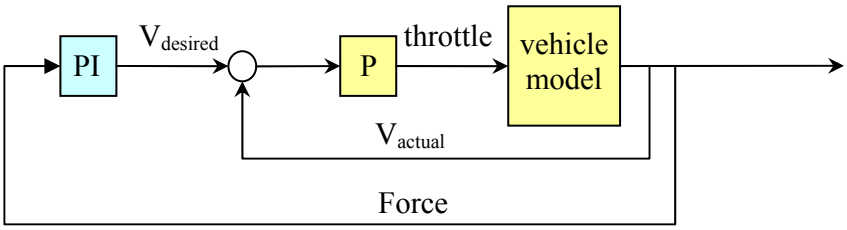


Figure 4-7: Force and velocity control block diagram.

The desired velocity controller utilizes a proportional-integral (PI) form of control. This form of control provides quick system response and drives the steady-state error to zero or nearly zero. The desired velocity command is based on the linearization and integration of the force error:

$$V_{desired} = k_p F_{res} + k_i \int F_{res} dt \tag{4-41}$$

The proportional term in equation 4-41 handles the transient response of the robot while the integral term is used to minimize the steady state error. Initial experiments were performed with proportional control and the system response behaved in a harmonic manner similar to spring-mass system without any damping. The steady state error oscillated and never reached zero. Addition of the integral term eliminated this behavior and minimized the steady-state error. Also, the integral term ensures that the desired velocity does not become zero.

Once the resultant force is calculated, a desired heading angle is calculated that is based on the components of this force. The steering command is derived from the angle between the resultant force vector and the vehicle heading vector. A proportional controller is used to control the steering angle of each of the follower vehicles.

Potential Energy Calculation

As the system of robots traverse the desired trajectory, it is visually apparent if the robots can or cannot maintain formation. However, it is necessary to quantify and validate the ability of

the robot system to maintain a formation. Since the system is modeled as a series of springs, masses and dampers it is advantageous to measure the potential energy of the system. As the system reaches equilibrium the potential energy will tend to zero and this indicates that the robots are in formation. The potential energy of the three-robot system in figure 4-4 is calculated as follows:

$$\Sigma PE = \frac{1}{2}k_{1L}\left(DL_{1L} - \left|\overline{L_{1L}}\right|\right)^2 + \frac{1}{2}k_{2L}\left(DL_{2L} - \left|\overline{L_{2L}}\right|\right)^2 + \frac{1}{2}k_{12}\left(DL_{12} - \left|\overline{L_{12}}\right|\right)^2 \qquad (4\text{-}42)$$

It is expected that for a system of robots in formation, the potential energy will have an initial overshoot and then proceed to track to zero or nearly zero. Slight errors may be introduced due to the dynamics of the system and will slightly affect the potential energy measurements.

Other factors will also affect the potential energy of the system. The potential energy of the system can be affected by the trajectory that the robot formation is attempting to track. Complicated trajectories will result in an increase in the potential energy of the system due to rapid changes in steering direction. The addition of more follower robots to the system will have an affect on the potential energy as well. Doing so will initially increase the potential energy and slightly increase the time in which it takes the system to reach equilibrium or steady-state.

Preliminary Experiments

Development of the compliant formation control algorithm occurred in several phases. The first phase involved using only linear springs in between each robot in the formation. The second phase introduced a torsional spring between the leader and the follower vehicles. The third phase introduced dampers in between each robot in the formation. The results from several preliminary experiments were used to provide an indication of how the dampers and linear and torsional springs affect the overall system. A series of eight experiments were performed on a

three robot system for a wedge formation and the details are presented in table 4-1 and 4-2. The spring constants between followers one and two are denoted as: k_12 and between leader and follower one as: k_1L and so on. A similar notation is used for the damping constants as well.

Test #	k_12	k_1L	k_2L	k_t
1	1.0	1.0	1.0	0.0
2	2.0	1.0	1.0	0.0
3	2.0	1.0	1.0	3.0
4	3.0	1.0	3.0	0.0

Table 4-1: Details of experiments 1-4. These experiments were performed without damping.

Test #	k_12	k_1L	k_2L	k_t	c_12	c_1L	c_2L
5	1.0	1.0	1.0	0.0	0.2	0.2	0.2
6	1.0	1.0	1.0	0.0	0.4	0.4	0.4
7	1.0	1.0	1.0	3.0	0.4	0.4	0.4
8	3.0	1.0	3.0	3.0	0.2	0.2	0.2

Table 4-2: Details of experiments 5-8. These experiments were performed with damping.

A comparison of the potential energies for each set of experiments (with and without damping) is made in an attempt to understand how variations in the spring and damper constants affect the performance of the overall system. The potential energy plots are seen in figures 4-8 and 4-9.

The potential energy plots reveal that the addition of the torsional spring (test 3, test 7, test 8) causes temporary spikes in the potential energy.

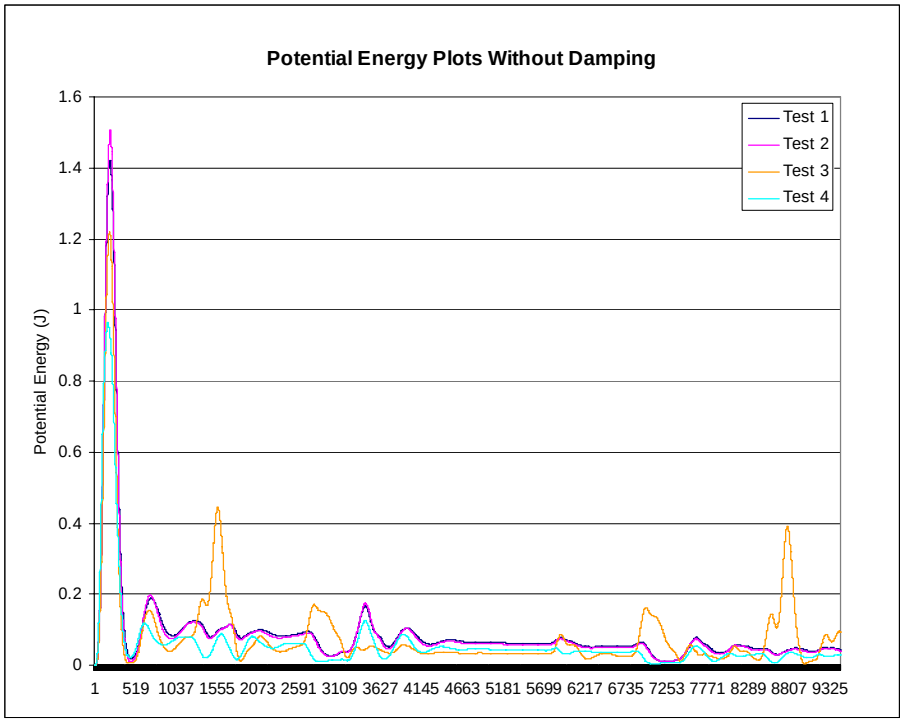


Figure 4-8: Potential energy plot comparisons across each test without damping.

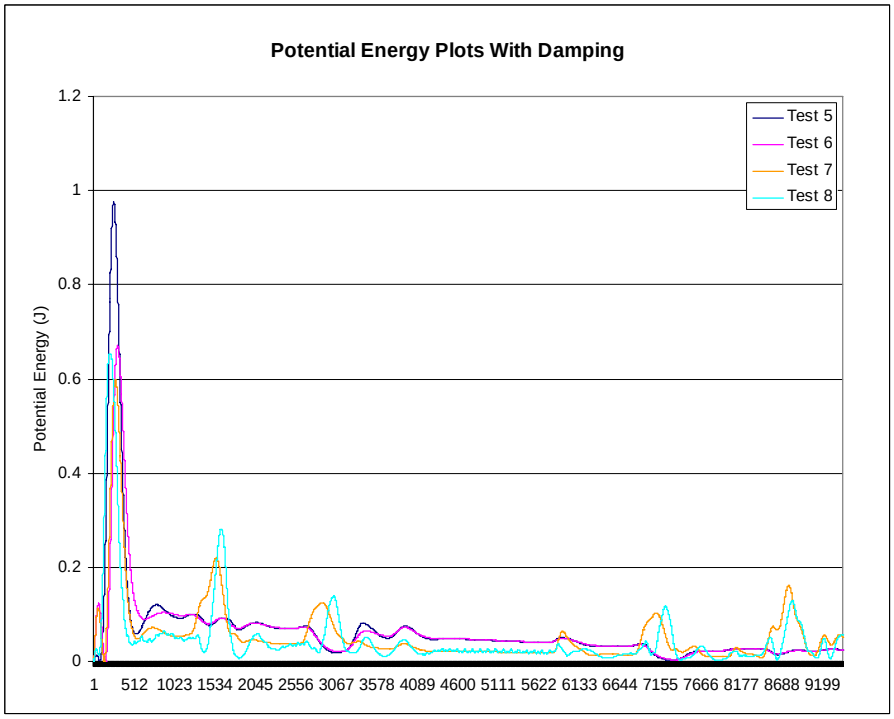


Figure 4-9: Potential energy plot comparisons across each test with damping.

However, the overall potential energy is lower when compared with the other cases. The cases that do not include a torsional spring appear to have very desirable results. This data does not

reveal that the *orientation* of the formation shape is not maintained even though the potential energy is low. That is, without the torsional spring to maintain the follow angle the formation shape can rotate about the leader. This behavior was observed in the simulation.

Another comparison is made between the errors of the free lengths of springs. This comparison provides an overall metric of the performance of the system as the spring and damper constants vary. The free length error plots are seen in figures 4-10 and 4-11.

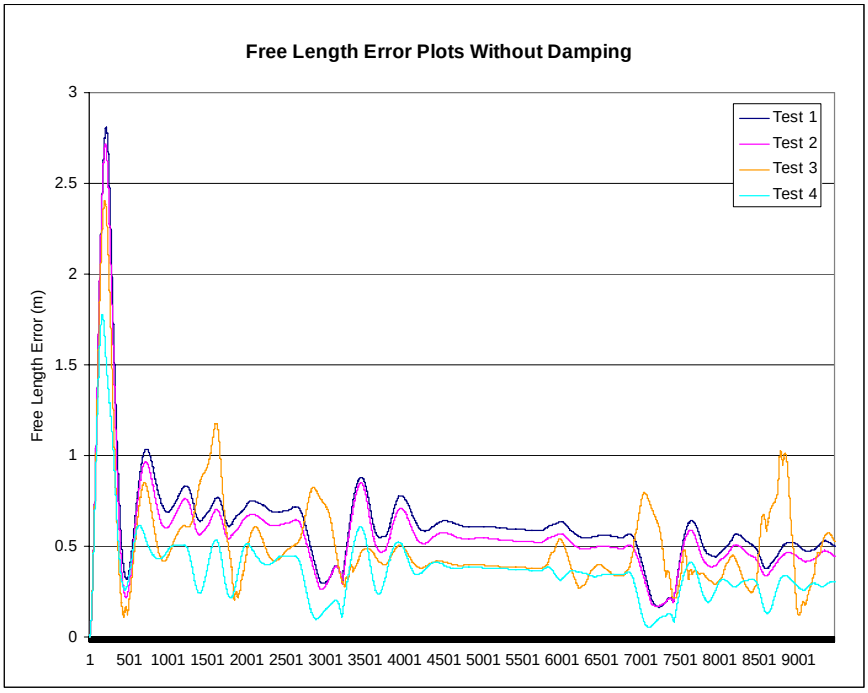


Figure 4-10: Free length error plot comparisons across each test without damping.

The error in the free lengths of the linear springs reveals that the error is lowest with the addition of the torsional springs. Again, there are spikes in the plot and this can be due to rapid changes in direction of the trajectory. For both the potential energy data and the free length error data the addition of damping to the system tends to smooth the data and decreases the overall amplitude of the data. From figure 4-10, test eight has a good amount of oscillation. This can be attributed to the choice of the spring constants. This was a "lopsided" experiment in that the spring between the leader and follower one was stiffer than all of the other springs in the

system. This resulted in a harmonic behavior between the robots.

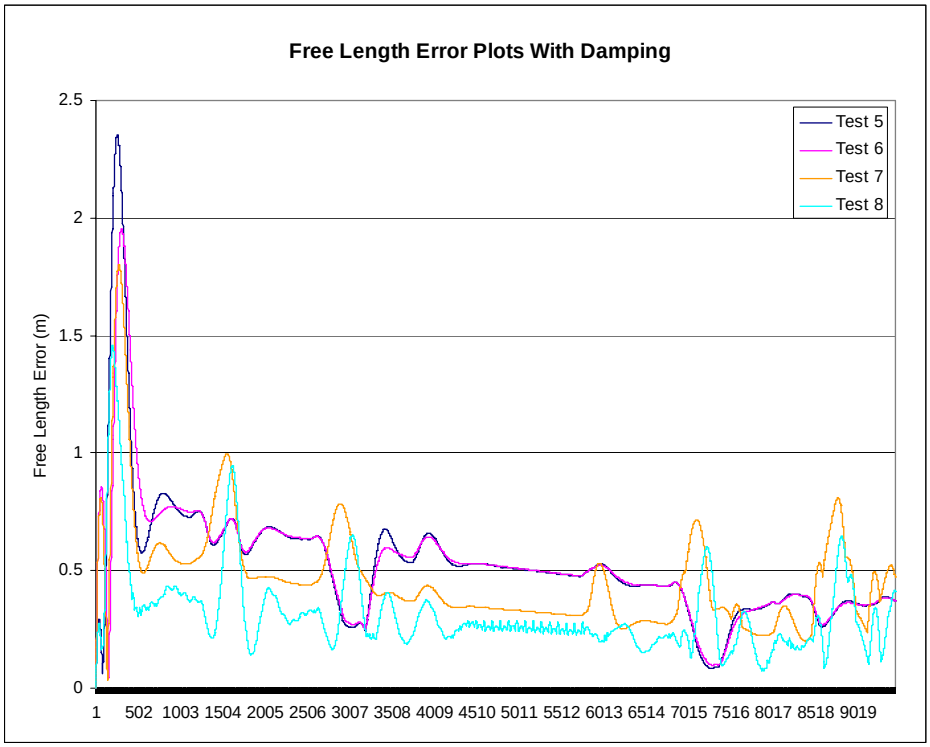


Figure 4-10: Free length error plot comparisons across each test with damping.

CHAPTER 5
VEHICLE SIMULATION AND VIRTUAL ENVIRONMENT

Development of the compliant formation control algorithm was performed in a virtual environment using simulated robots. Using the C++ programming language and OpenGL graphics capabilities a virtual world was created for algorithm development and testing. A custom graphical user interface (GUI) was designed to allow for testing of multiple vehicles in a variety of formations. This virtual environment provided an ideal developing and testing platform for this research.

The simulated robots were modeled using the vehicle kinematics and dynamics equations presented in the previous chapter. In order to provide the most realistic results the simulated robots were modeled after an actual ground vehicle in the Center for Intelligent Machines and Robotics. This research attempts to provide a realistic approach to formation control of a multi-vehicle system by establishing a model that combines theory with first-hand field experience.

Vehicle Simulation

As stated in the previous chapter, the robot geometry is modeled after a front-steered vehicle. Research performed in this field typically models the robots using a differential steering model or simply as point models. These approaches greatly reduce the complexities associated with vehicle kinematics and dynamics. In an effort to develop a "realistic" robot simulation, this research incorporates the kinematics and dynamics of a typical ground vehicle. The front wheels can turn to the right or left but are constrained by a maximum turning angle as shown in figure 5-1. Limiting the range of motion of the front wheels provides a more realistic vehicle model.

The robot formation consists of a "lead" robot and "follower" robots. The lead robot is assumed to know where it is globally. The positions of all follower robots are referenced from the lead robot. The lead robot is able to navigate to a series of waypoints or traverse a given

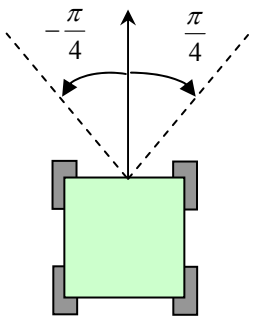


Figure 5-1: Steering angle constraints on front wheels.

path it receives. This differs from the follower robots in that their motion (velocity) is based on the forces calculated due to the virtual springs and dampers. From a programming point of view, two separate classes were developed. One class was designed to handle the lead robot: `class LeadRobotModel` and another were developed to handle all of the follower robots: `class FollowerRobotModel`. These classes contained all of the variables necessary to describe the motion of the lead robot or the follower robots. In the case of the follower robots, several instances of this class can be called depending on the desired number of follower robots in the system. Each instance of the robot model is updated in a timer that runs every ten milliseconds– this updates the forces, positions, velocities, and accelerations of each vehicle.

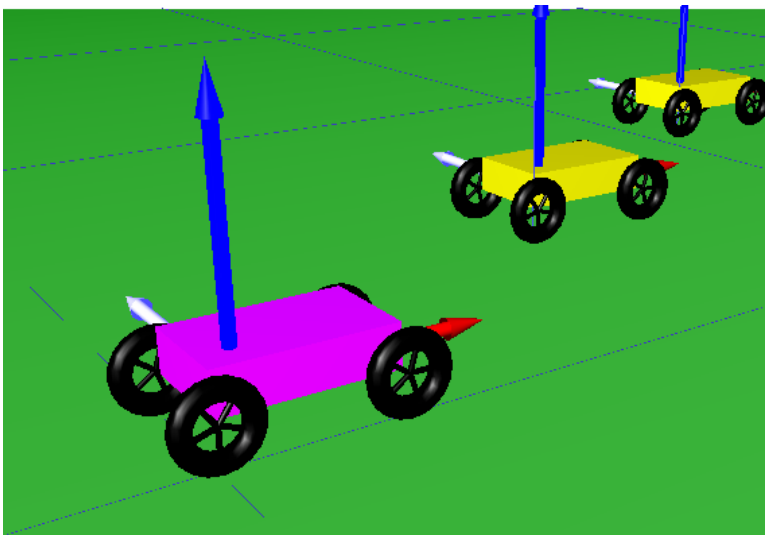


Figure 5-2: The lead robot in the compliant formation control simulator environment.

Virtual Environment

Development of a virtual environment was essential for performing initial experiments to validate the equations of motion derived previously. It also was the test bed for developing the compliant formation control algorithm. The interface was completely designed using C++ and the non-commercial version of the Qt 3 software. OpenGL was used to create the world, the coordinate systems, and construct the robots. As the vehicles move throughout the virtual world the front wheels are mapped to steer in the desired direction and all four wheels are able to rotate. The simulator software had to meet several objectives:

- Allow the user to choose a file of waypoints
- Generate a trajectory through the waypoints
- Draw the waypoints and the trajectory
- Allow the user to switch views
- Run the simulation
- Time the simulation
- Collect all data for each simulated vehicle.

All of these objectives were realized in the compliant formation control simulator shown in figure 5-3. A five robot system is shown in a wedge formation with the lead vehicle indicated by pink and the four follower robots by yellow.

This software is designed to provide the researcher with a flexible interface that allows for ease of experimentation with different trajectories, formation shapes, and various numbers of robots. The individual robot classes are designed to encapsulate all of the necessary dynamics and control of each robot. Once the robot models are tuned properly, these classes should operate in a transparent manner for the user. The virtual environment consists of a ground plane and the three-dimensional ground vehicles and is also encapsulated in its own class.

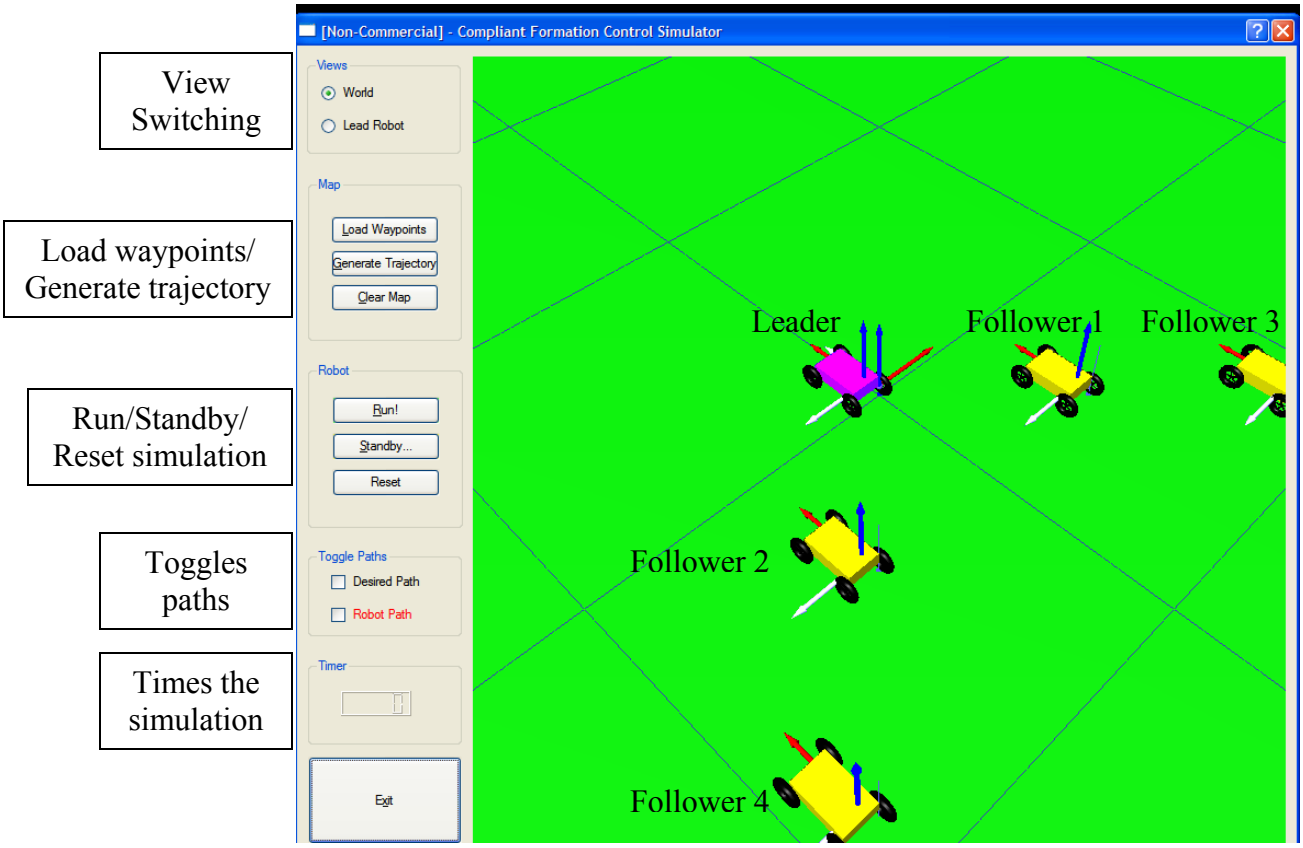


Figure 5-3: Compliant formation control simulator in a 3D world view.

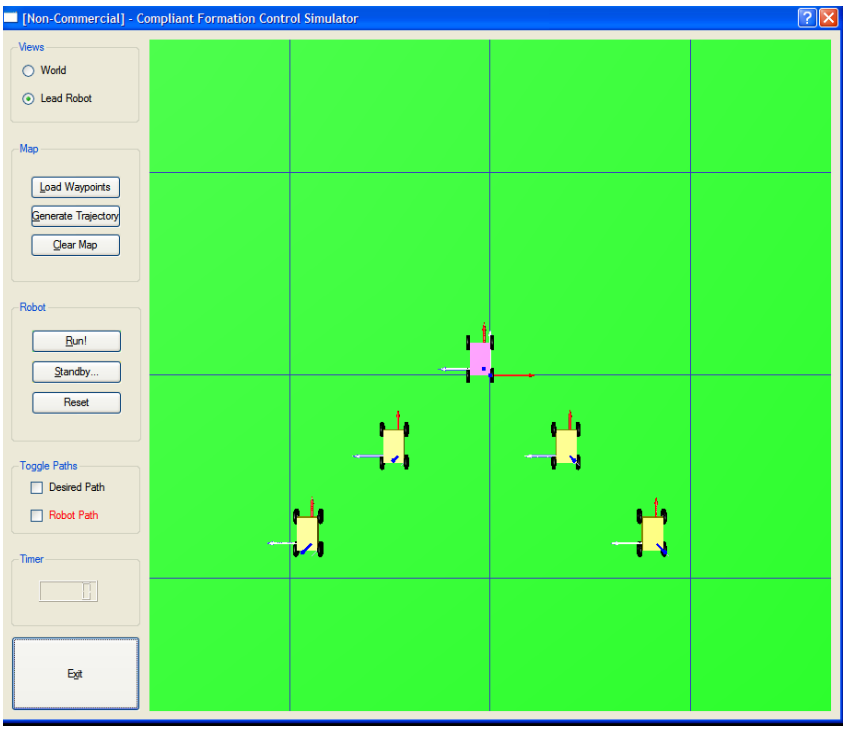


Figure 5-4: An example of a lead robot, top down view.

The top down view keeps the lead vehicle in the center of the map. As the robots travel along the path in formation, the virtual camera moves with the lead vehicle. This is very useful when trying to determine if the vehicle dynamics are responding accordingly and when observing the formation as the vehicles travel along the path.

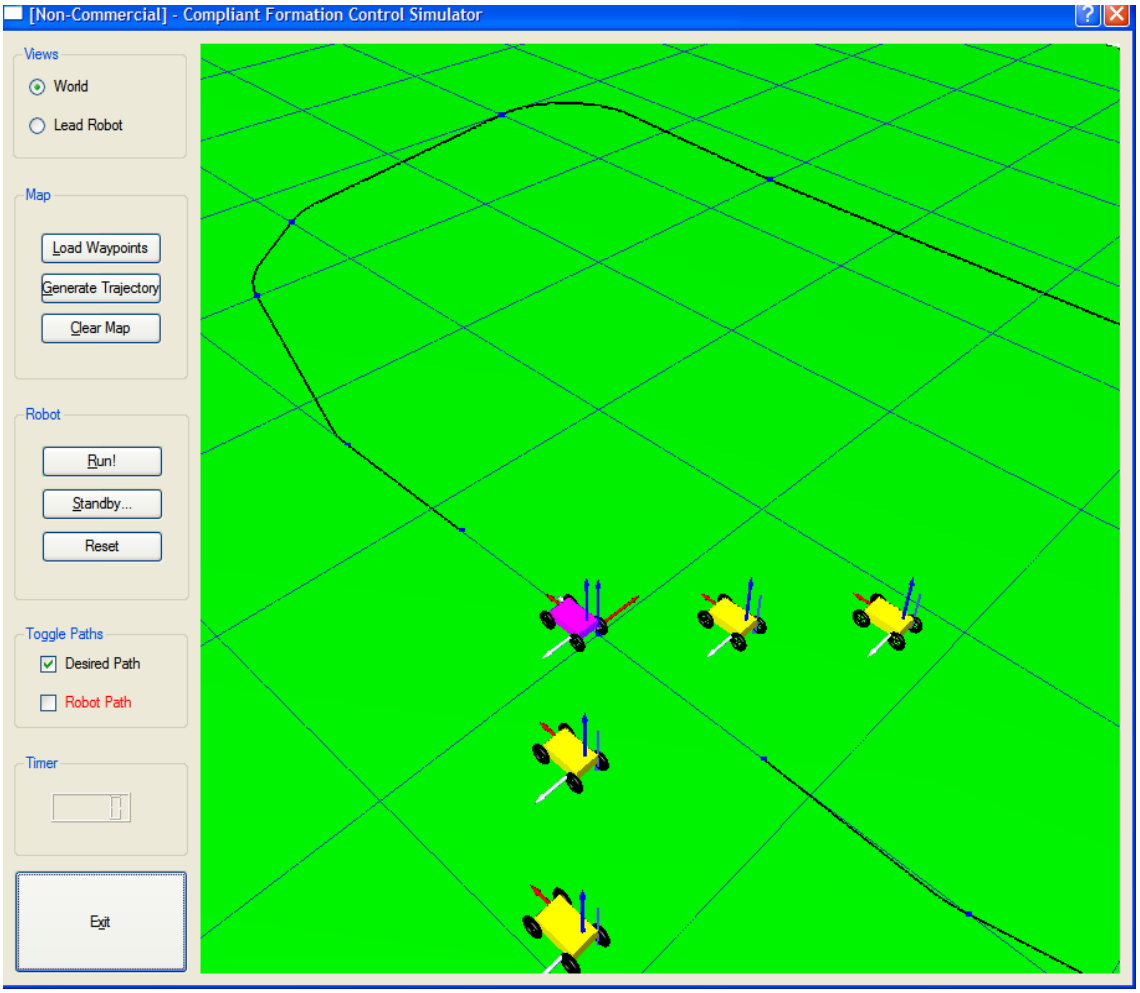


Figure 5-5: An example of a trajectory in 3D world view.

The user is able to open a file containing a series of waypoints (x, y). Once these waypoints are loaded into the simulator clicking on the "generate trajectory" button will result in a smooth path that passes through each of the desired waypoints. Choosing the "desired path" checkbox will draw the trajectory to the screen. Derivation of the trajectory generation algorithm is described in detail in the next section.

Trajectory Generation Algorithm

Previous research involves autonomous navigation of a ground vehicle to a series of waypoints (latitude, longitude). For this research, instead of simply navigating to a point, a path is generated through a set of waypoints and the lead vehicle traverses along this path. This provides a continuous path rather than a set of discrete points in space. The algorithm is able to generate a continuous path through any given set of points using arcs and straight lines segments.

The first trajectory segment is a straight line segment that proceeds from the first waypoint to the second. A direction vector, $\bar{u}$, from the first waypoint to the second waypoint is calculated.

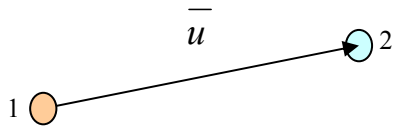


Figure 5-6: Straight line segment from first waypoint to second, also $\bar{u}$.

Each successive trajectory segment is composed of an arc and a straight line segment. For each arc, a center point is calculated using the maximum turning radius of the vehicle.

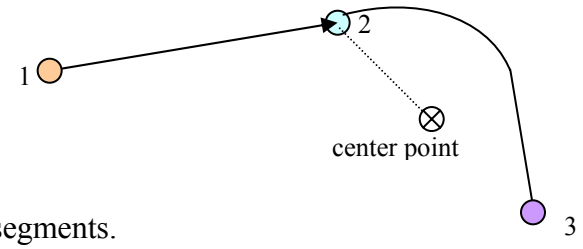


Figure 5-7: Arc and straight line segments.

A cross product is used to determine the center point's location relative to the waypoint. In other words, the center point can be to the right or left of the waypoint. In figure 5-8 the vectors $\bar{u}$ (from 1→2) and $\overline{PR}$ (from 2→3) are shown and for this example the center point would lie to the right of waypoint two.

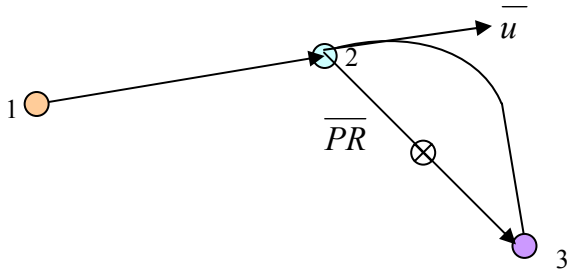


Figure 5-8: Vectors $\overline{u}$ and $\overline{PR}$.

The center point location relies on the sign of the cross product as well as a vehicle constraint. The turning radius of the vehicle is considered when determining the location of the arc's center point. A turn that is too sharp may not be physically realizable by the ground vehicle and therefore that portion of the path cannot be generated.

In order to construct the arc, the center point and the sweep angle are necessary. Calculation of the sweep angle is a multi-step process that requires additional angles and vectors. A detailed drawing in figure 5-9 describes the vectors and angles that are required. Vector $\overline{w_{2C}}$ goes from a waypoint (the second waypoint in this example) to the center point. The sweep angle is denoted by α and is calculated using intermediary vectors and angles.

$$\varphi = \cos^{-1}\left(\frac{\overline{w_{2c}} \bullet \overline{w_{23}}}{\left|\overline{w_{2c}}\right|\left|\overline{w_{23}}\right|}\right), L_1 = \left|\left(\overline{w_{23}} - \overline{w_{2c}}\right)\right|$$

$$\theta = \sin^{-1}\left(\frac{R_{\min}\sin\varphi}{L_1}\right)$$

$$\gamma = \cos^{-1}\left(\frac{R_{\min}}{L_1}\right), \alpha = \pi - \theta - \varphi - \gamma$$

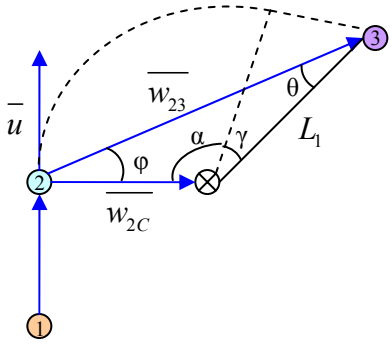


Figure 5-9: Vectors and angles required to determine the sweep angle.

The sweep angle is further resolved depending on which quadrant it is located in. The sweep angle is either added or subtracted from π depending on its location in the Cartesian coordinate system. This calculation is detailed in the following equations.

$$\overline{TV_1} = \frac{\overline{w_{23}} - 2\overline{w_{2c}}}{\left(\left|\overline{w_{23}} - 2\overline{w_{2c}}\right|\right)} \tag{5-1}$$

$$\overline{TV_2} = \overline{w_{2c}} \times \overline{TV_1} \tag{5-2}$$

$$TD = \overline{w_{2c}} \bullet \overline{TV_1} \tag{5-3}$$

$\overline{TV_1}$ and $\overline{TV_2}$ are temporary vectors that are used in these calculations. The value of the z-component ($TV_2[2]$) of $\overline{TV_2}$ and TD are considered when determining the quadrant that α lies in. The following cases are checked:

- If the center point lies to the left of the current waypoint:
    - Quadrant 1 and 2 are checked:

```
// Quadrant 2 check
if(TD < -1.0e-10 && TV2[2] > 1.0e-10)
      α = α + π ;
// Quadrant 1 check
else if(TD > 1.0e-10 && TV2 [2] > 1.0e-10)
      if (α < π /2.0) α = π - α ;
```

- If the center point lies to the right of the current waypoint:
    - Quadrant 3 and 4 are checked:

```
// Quadrant 3 check
if(TD < -1.0e-10 && TV2[2] < -1.0e-10)
      α = α + π ;
// Quadrant 4 check
else if(TD > 1.0e-10 && TV2 [2] < -1.0e-10)
      if (α < π /2.0) α = π - α ;
```

After sweeping through the arc, α, the path segment is completed with a straight line segment that begins at the endpoint of the current arc and continues to the next waypoint in the path. This method was tested for various sets of waypoints. The generated trajectory provided a baseline for comparison of the navigational abilities of the ground vehicle formation between experiments. Plots of the waypoints and the corresponding planned trajectory are shown in the figures that follow.

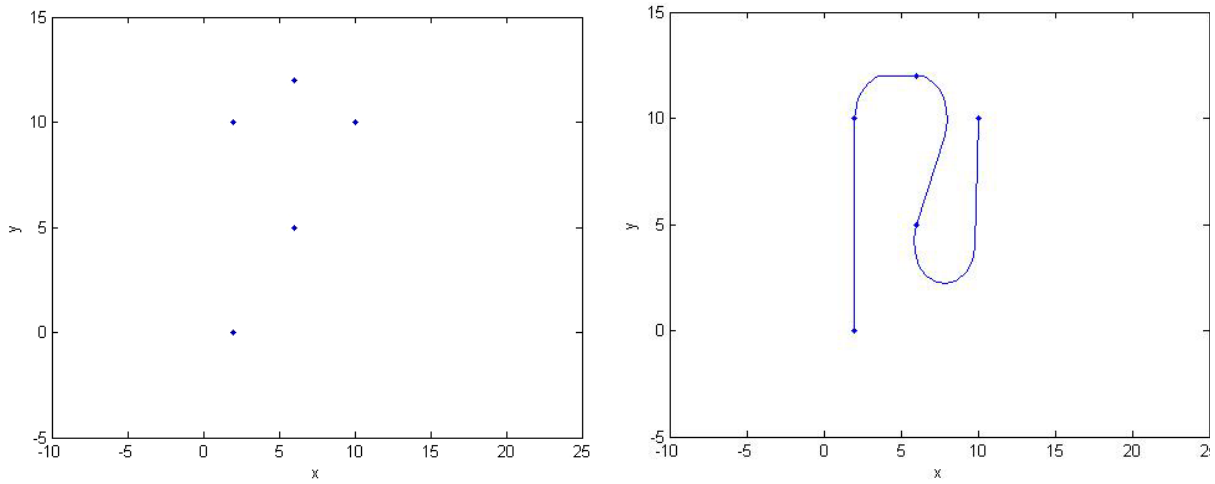


Figure 5-10: A plot of a simple set of waypoints and the corresponding trajectory.

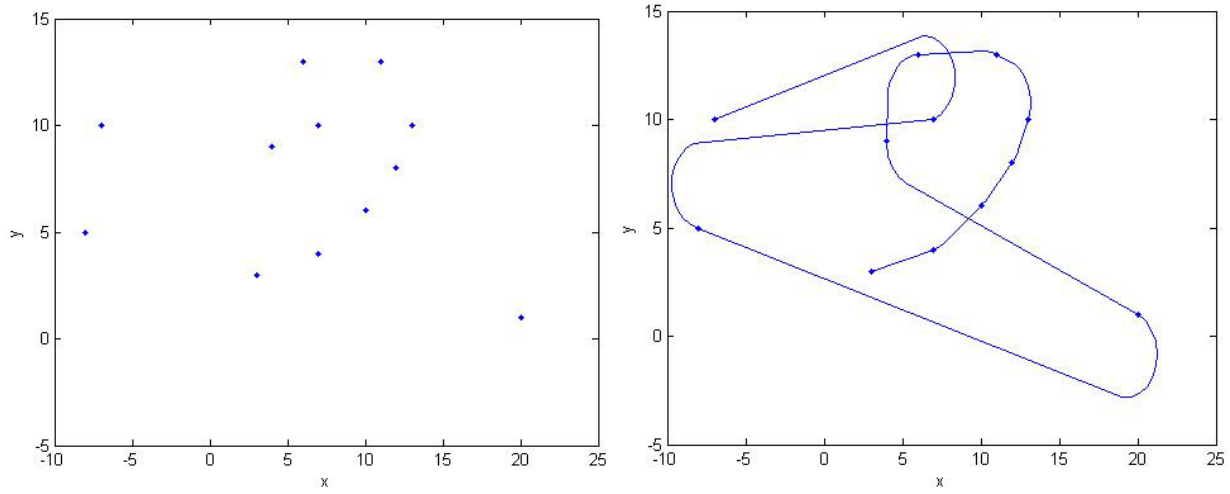


Figure 5-11: A plot of a set of complex waypoints and the corresponding trajectory.

The trajectory that is generated is given directly to the lead vehicle to traverse. With the implementation of the compliant formation control algorithm, the follower vehicles are able to navigate in formation relative to the lead vehicle.

CHAPTER 6
EXPERIMENTS AND RESULTS

The derivation and development of the compliant formation control algorithm has been presented thus far. Discussion of how virtual springs are used to keep formation among the robot system has become instrumental in the *compliant* formation control algorithm. This methodology treats the robot system as a structure that can translate and rotate throughout a two-dimensional environment.

The program used to develop this algorithm is also used to perform experiments that test the feasibility of the compliant formation control algorithm. The experiments are designed to test the validity and robustness of the algorithm. It is necessary to have a "simple" experiment that is used to verify that the algorithm is functioning. After collecting a valid set of data for the "simple case" a more complicated experiment could be attempted. This includes a more complex trajectory and addition of more follower robots to the system.

The formations are defined by the follow angle and follow distance that is referenced from the lead robot. The formations used in these experiments are the vee shape and wedge shape. These shapes are commonly found in nature especially when observing birds flying in formation (see figure 6-1).

Figure 6-1: A formation of Canada geese (Photo by: Russell Link)

Data is collected for a three-robot system and a five-robot system in the two different formation shapes and for two different trajectories. The trajectory provides the leader with a

path to navigate along and provides a method to quantify the navigation ability of the robots. Also, data will be presented for the same robot formations but with the addition of virtual dampers to the system.

For all experiments the robots are initially in formation with some spacing errors and have no heading errors. This was done in order to minimize the errors when switching between vehicle states. The vehicle parameters are detailed in table 5-1.

Mass of vehicle	75 kg
Maximum force engine exerts	50 N
Maximum velocity of vehicle	5 m/s
Vehicle length	1 m
Dynamic friction coefficient (k_d)	0.2

Table 5-1: Vehicle parameters.

All experiments used the same values for the spring and damper constant with the exception of the vee formation experiments. This formation required higher gains due to the unstable nature of the formation. All of the constants used are seen in tables 5-2 and 5-3.

			5 Robot Case	
			k_{12}	1.0
			k_{1L}	1.0
3 Robot Case			k_{2L}	1.0
k_{12}	1.0		k_{13}	1.0
k_{1L}	1.0		k_{24}	1.0
k_{2L}	1.0		k_{34}	1.0
k_t	3.0		k_t	3.0
c_{12}	0.2		c_{12}	0.2
c_{1L}	0.2		c_{1L}	0.2
c_{2L}	0.2		c_{2L}	0.2
			c_{13}	0.2
			c_{24}	0.2
			c_{34}	0.2

Table 5-2: Spring and damper constants used in simulations of wedge formation.

			5 Robot Case	
			k_{12}	2.0
			k_{1L}	2.0
3 Robot Case			k_{2L}	2.0
k_{12}	2.0		k_{13}	2.0
k_{1L}	2.0		k_{24}	2.0
k_{2L}	2.0		k_{34}	2.0
k_t	6.0		k_t	6.0
c_{12}	0.2		c_{12}	0.2
c_{1L}	0.2		c_{1L}	0.2
c_{2L}	0.2		c_{2L}	0.2
			c_{13}	0.2
			c_{24}	0.2
			c_{34}	0.2

Table 5-3: Spring and damper constants used in simulations of vee formation.

Experiment #1: Straight Line Trajectory

The straight line trajectory was chosen because it is the simplest case that can be encountered in terms of path planning and navigation. It also provides the opportunity to troubleshoot the compliant formation control algorithm. That is, if the algorithm is not functioning for the simple case then it most likely will not perform well for a complicated situation. This also provided the opportunity to tune controller gains for velocity control and adjust the magnitudes of the virtual spring constants. The balance between the stiffness of the structure and the responsiveness of the robots had to be determined in order to maintain the formation. The spring and damper constants were tuned in the same manner as controller gains.

Three-Robot System Without Damping

All of the data collected for each robot is stored to a data file and contains the robot's position, velocity, acceleration, and steering and heading angles. For example, the velocity data is used to tune controller gains for the velocity control of each vehicle. After proper gain selection and tuning the position data was collected and analyzed for each robot and compared

with the generated trajectory. The positioning plots are shown in figures 6-2 for the wedge and vee formation and in 6-3 for the wedge and vee formation.

As the robots proceed along the path in formation there is a slight disturbance seen in the plots. This occurs as the vehicle state changes from static to forward velocity. Due to the spring-like nature of the robot-system the follower robots are pulled or pushed along until they reach equilibrium.

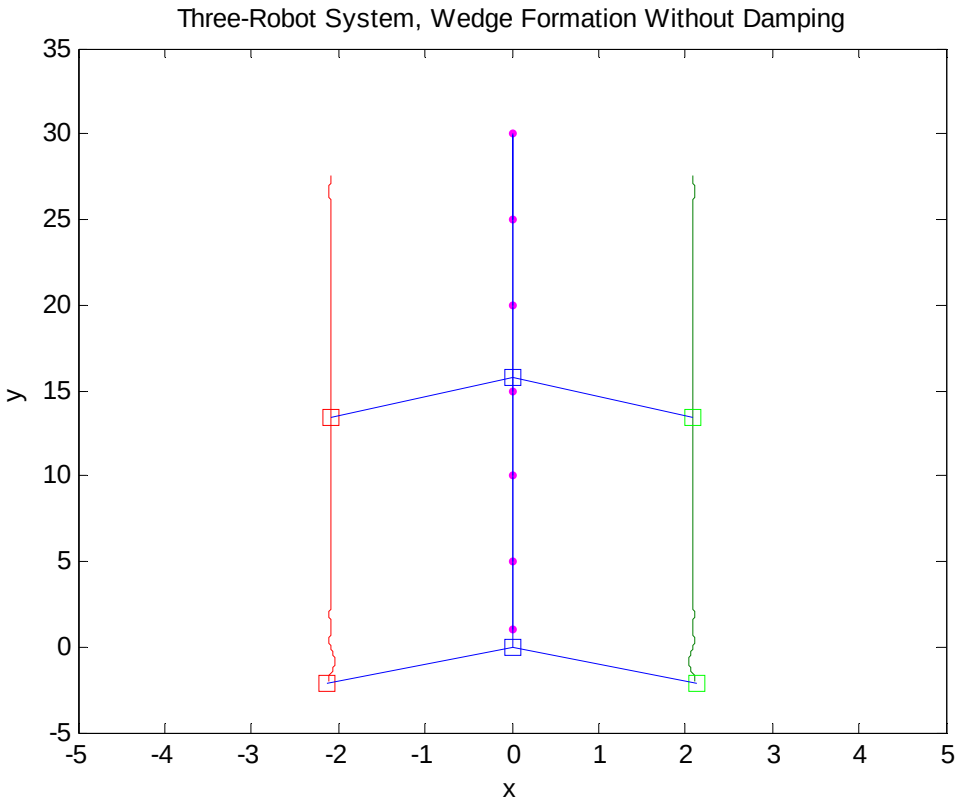


Figure 6-2: Results for a three-robot system in a wedge formation (blue:leader, green:follower 1, red:follower 2).

In order to avoid large discontinuities and direction changes, the robots are initially in the desired formation at the beginning of the experiment. Previous experiments showed instability in the formation control when there were large initial disturbances in the formation. The follower robots in the vee formation have small disturbances that can be attributed to the formation shape.

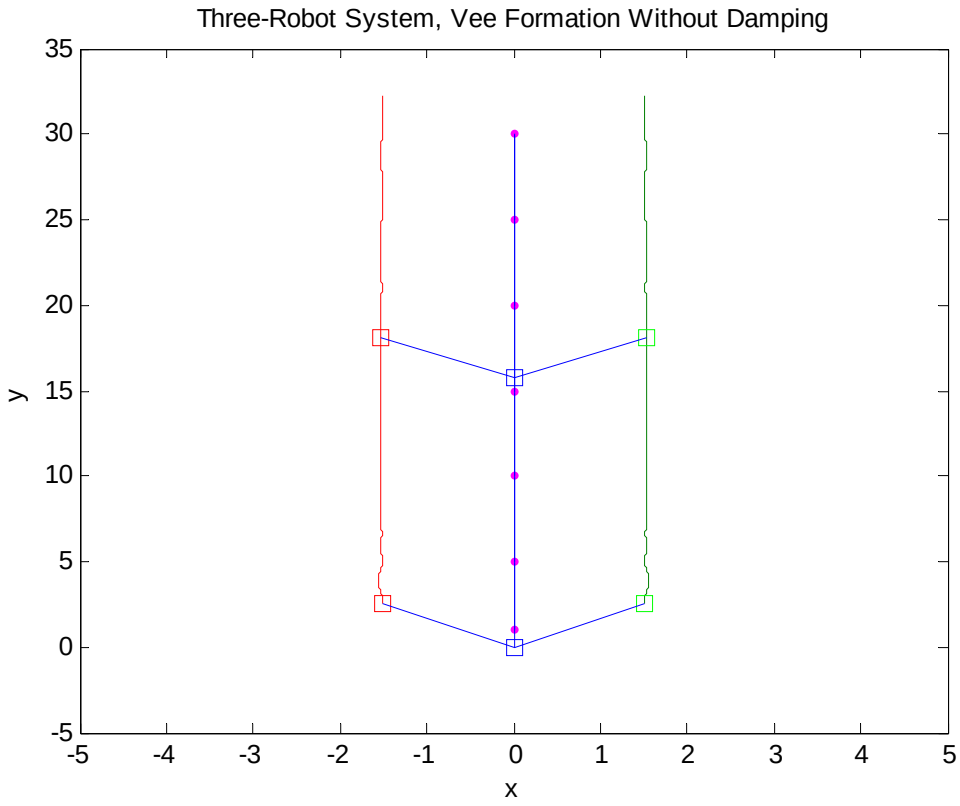


Figure 6-3: Results for a three-robot system in a vee formation (blue:leader, green:follower 1, red:follower 2).

The follower robots are ahead of the lead robot – this situation can be likened to an inverted pendulum problem. An inverted pendulum is inherently unstable and must be actively balanced to remain upright. With the follower robots ahead of the leader, slight disturbances are introduced and can be observed in figure 6-3.

Three-Robot System With Damping

Another series of experiments were performed that include not only virtual springs but virtual dampers as well. It is hypothesized that addition of the dampers to the formation of robots will result in less oscillation of the robots as they travel in formation. The same formation shapes are used for this series experiments. The results for the wedge and vee formations are shown in figures 6-4 and 6-5.

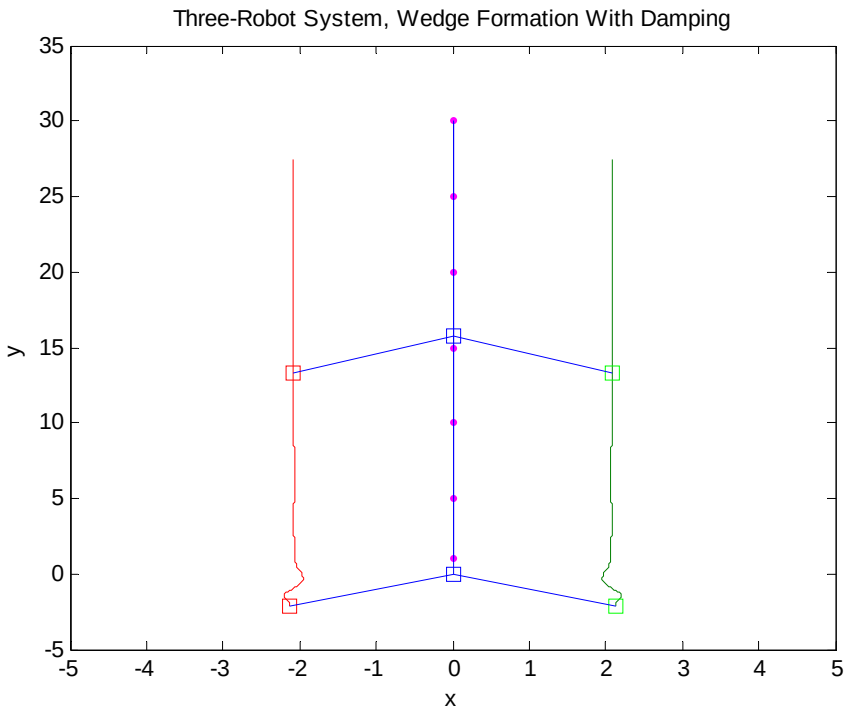


Figure 6-4: Results for a three-robot system with damping in a wedge formation (blue:leader, green:follower 1, red:follower 2).

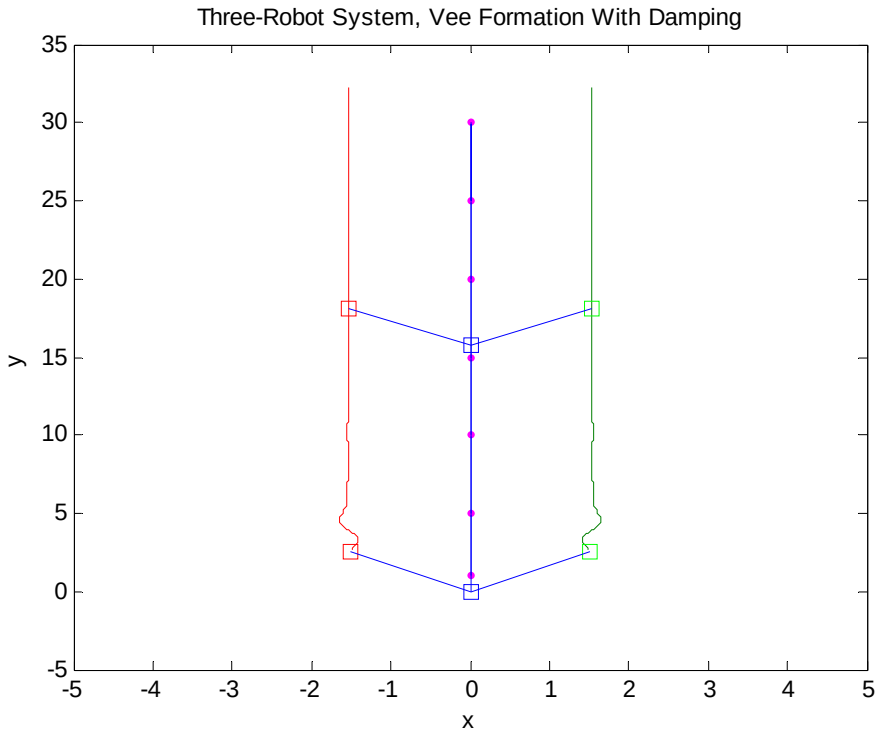


Figure 6-5: Results for a three-robot system with damping in a vee formation (blue:leader, green:follower 1, red:follower 2).

Initial oscillations occur however, they are quickly eliminated due to the effect of the dampers that are incorporated into the system. The responsiveness of the follower robots is reduced due to addition of the dampers. This actually provides a benefit in that, the follower robots are less likely to overshoot their desired positions within the formation.

Five-Robot System Without Damping

Scalability of the algorithm is important in order to perform realistic experiments in simulation. Once the three-robot case was functioning additional robots were added to the formation. The addition of more robots to the formation required a few more virtual springs and the associated equations. The five-robot system was modeled as follows:

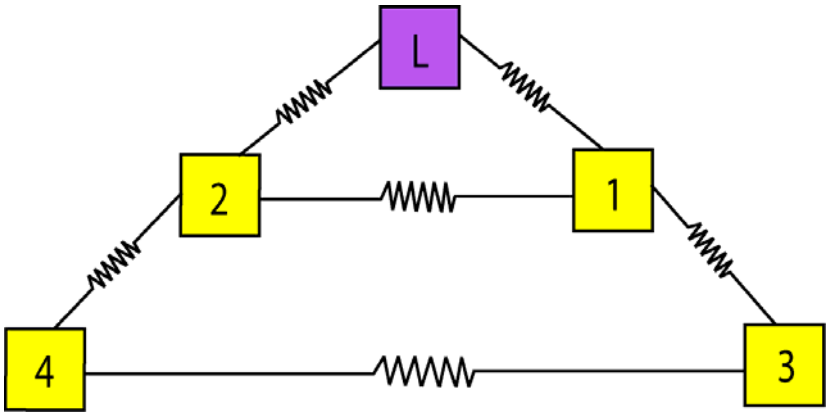


Figure 6-6: Five robots in a wedge formation.

The positioning plots corresponding to each formation is shown in the next two figures. The vee formation has slightly more disturbances when compared to the wedge formation which corresponds to data presented for the three-robot system. The initial disturbances can be attributed to the lead vehicle's transition from rest to forward motion while the follower vehicles' try to travel with the leader. The lead vehicle incrementally increases its velocity until it reaches the desired velocity. The equations of motion regulate each vehicle's velocity so that the commands received are incremental rather than instantaneous.

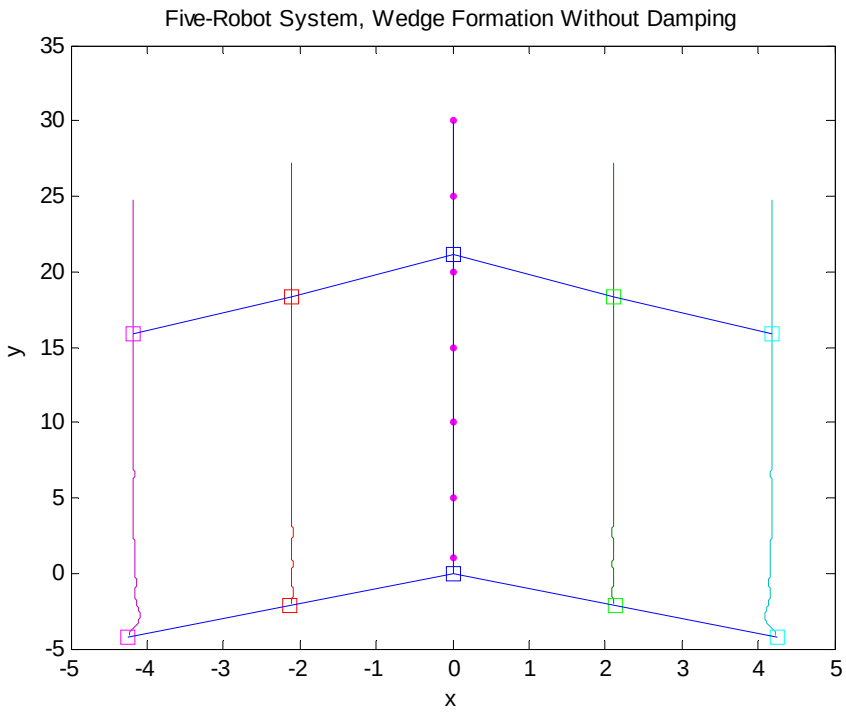


Figure 6-7: Results for a five-robot system in a wedge formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

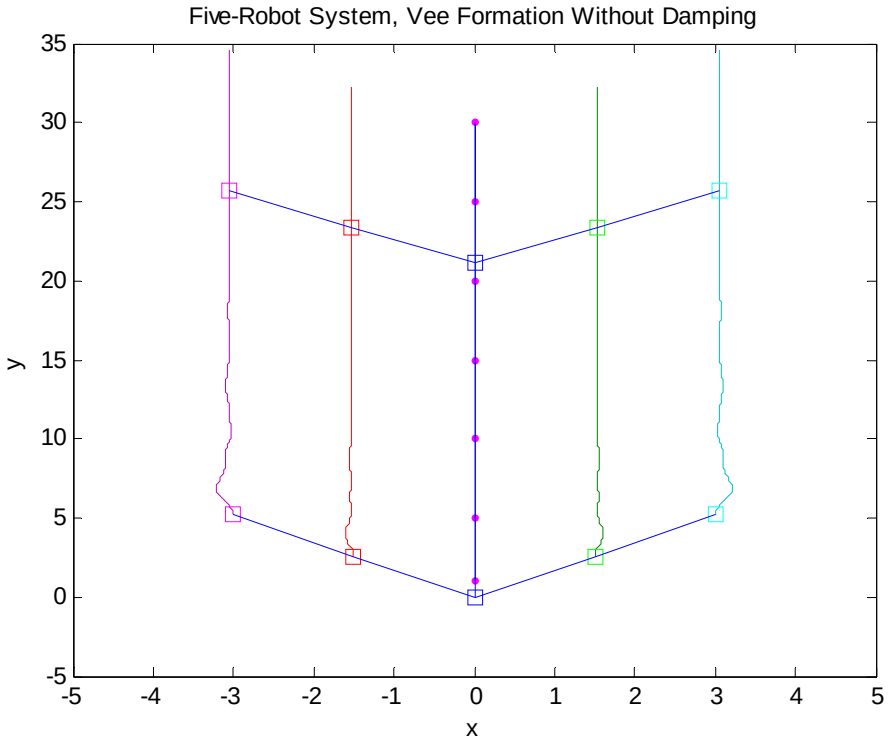


Figure 6-8: Results for a five-robot system in a wedge formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

Five Robot System With Damping

Damping was added to the five robot system as well. A model of the damped five-robot system is shown in figure 6-9. As the size of the robot system increases small oscillations propagate from one robot to the next and can result in an unstable or oscillatory system. Addition of damping to the system will minimize the oscillations of the system.

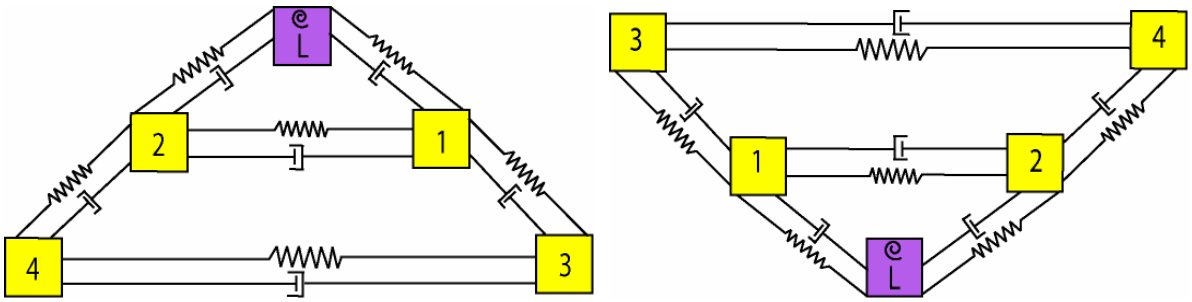


Figure 6-9: Five robots in a wedge and vee formation with damping.

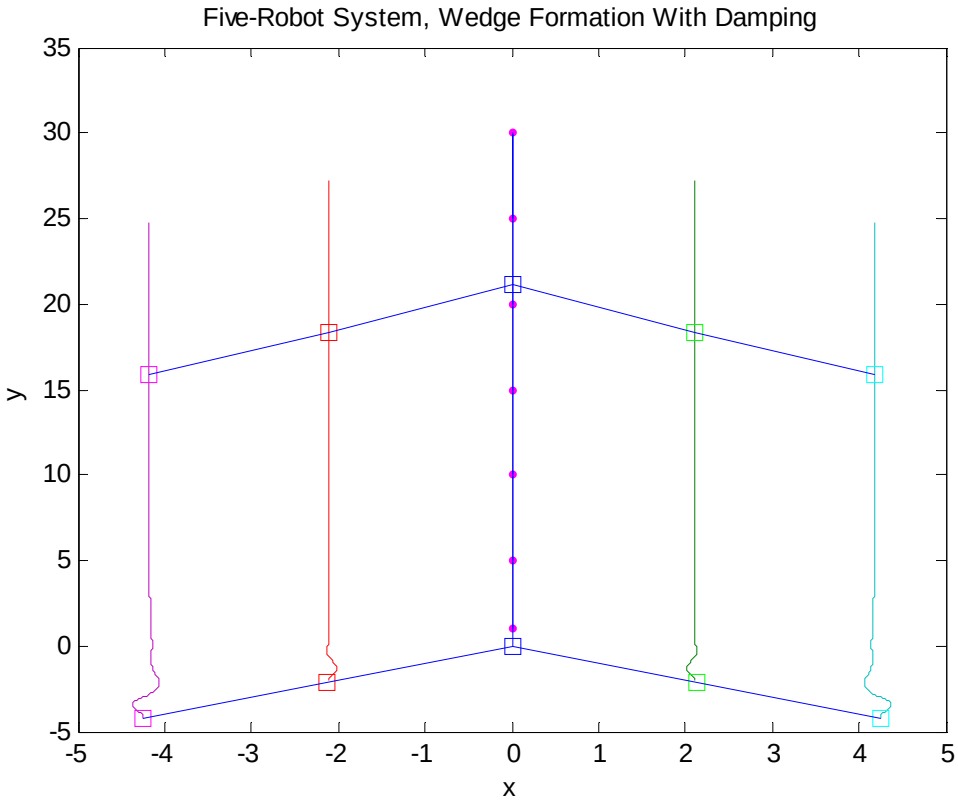


Figure 6-10: Results for a five-robot system with damping in a wedge formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

Similar to the three-robot system with damping, there are initial oscillations which are quickly eliminated. In contrast, the five-robot system without damping has oscillations that propagate for a longer duration especially with follower 3 and follower 4.

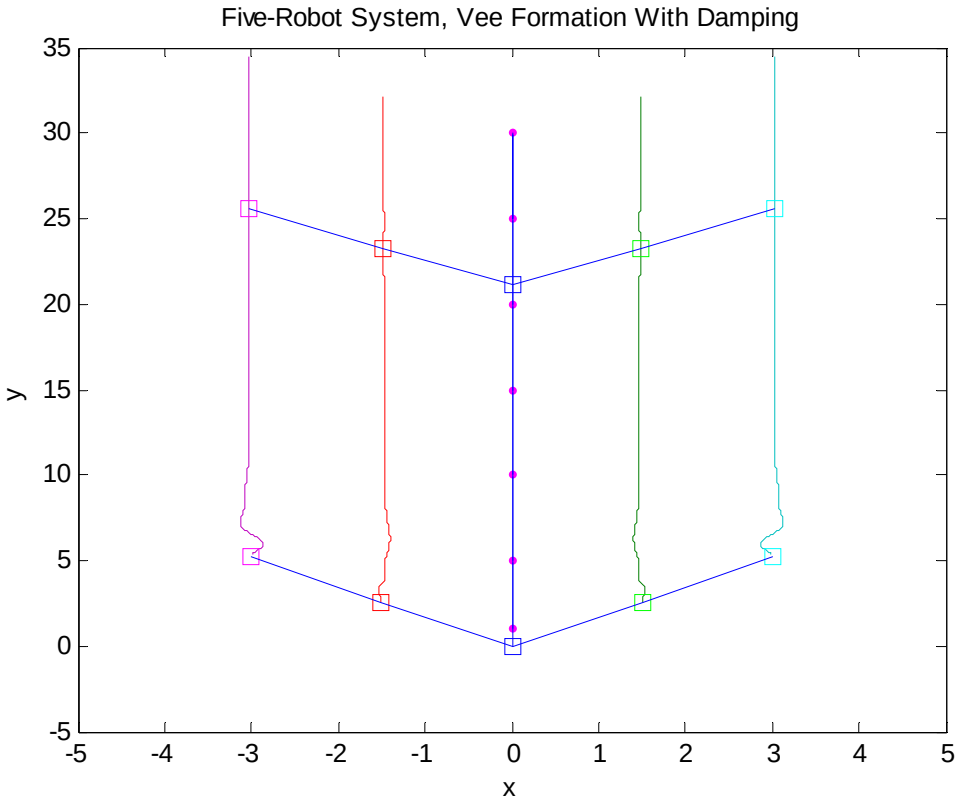


Figure 6-11: Results for a five-robot system with damping in a vee formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4)

Experiment #2: Racetrack Trajectory

After extensive experiments with different formations and numbers of robots on a simple path it was necessary to test the robustness of the compliant formation control algorithm. A trajectory was created that had straight-aways, slight curves, and sharp turns (see figure 6-12). The sharp turns are just at the physical limits of the vehicle's turning radius. This type of trajectory will push the limits of the formation control algorithm due to disturbances from direction changes. As the lead vehicle proceeds along the trajectory without hesitation the

follower vehicles will be adjusting their velocities to follow as commanded. The same formations are evaluated for the three-robot and five-robot system.

Similar to the three-robot experiments, a series of experiments were performed that incorporate virtual damping into the force equations. Introducing damping into the compliant formation control algorithm will minimize the oscillations of the robot system.

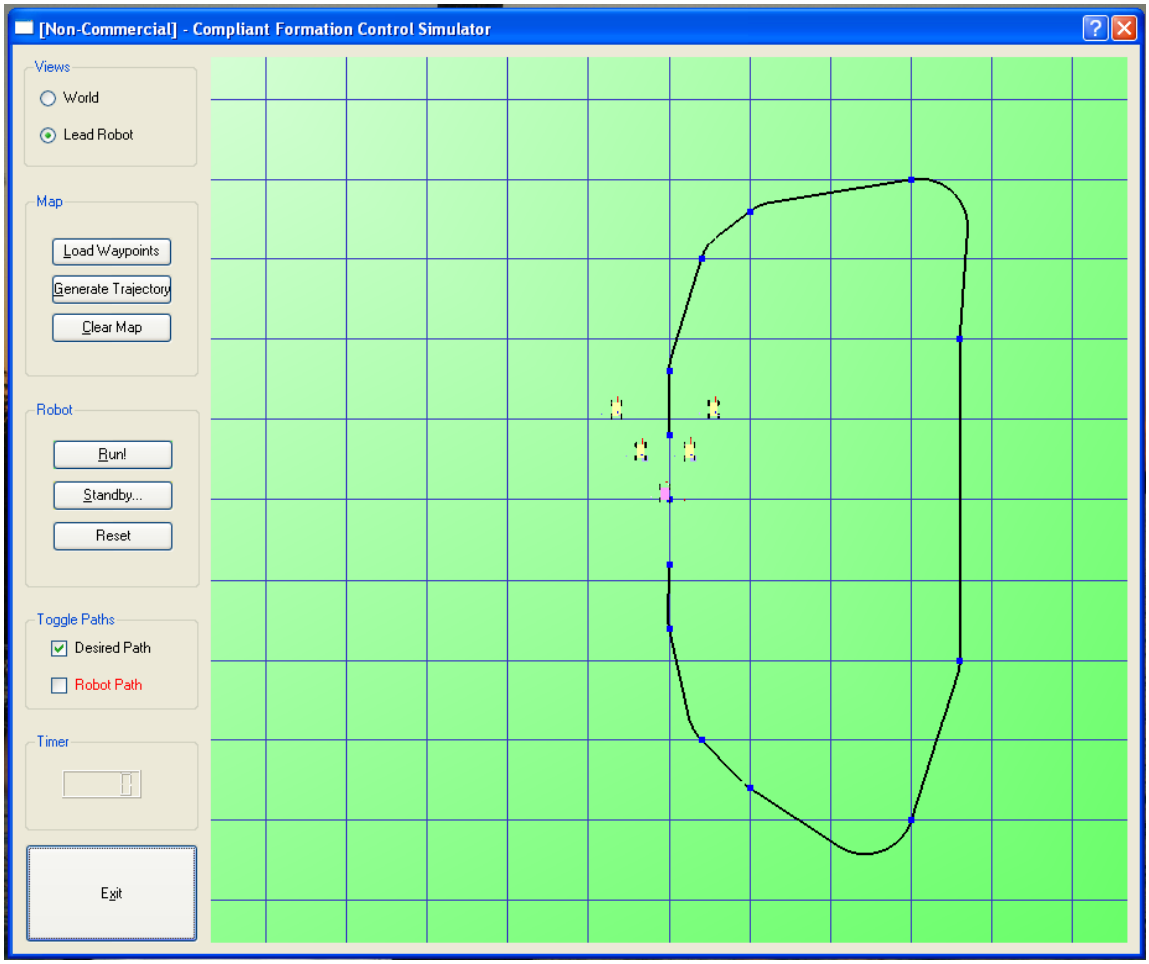


Figure 6-12: Top-down view of racetrack trajectory.

Three-Robot System Without Damping

Again, the robots are initially in the desired formation at the beginning of the experiment. At this point, the velocity controller and the virtual spring constants are adjusted based on initial data. The position plots for all three robots are shown in figures 6-13 and 6-14.

For both formations the follower 1 robot has to make the most extreme turns whereas follower 2 is required to make more gradual turns. However this does not affect the ability of the robots to maintain the wedge or vee formations. The disturbances are slightly greater for the vee formation.

The lead vehicle navigates along the planned trajectory while the follower vehicles are commanded to maintain formation. In other words, at all times the follower vehicles are trying to drive the inter-robot forces to zero by maintaining formation.

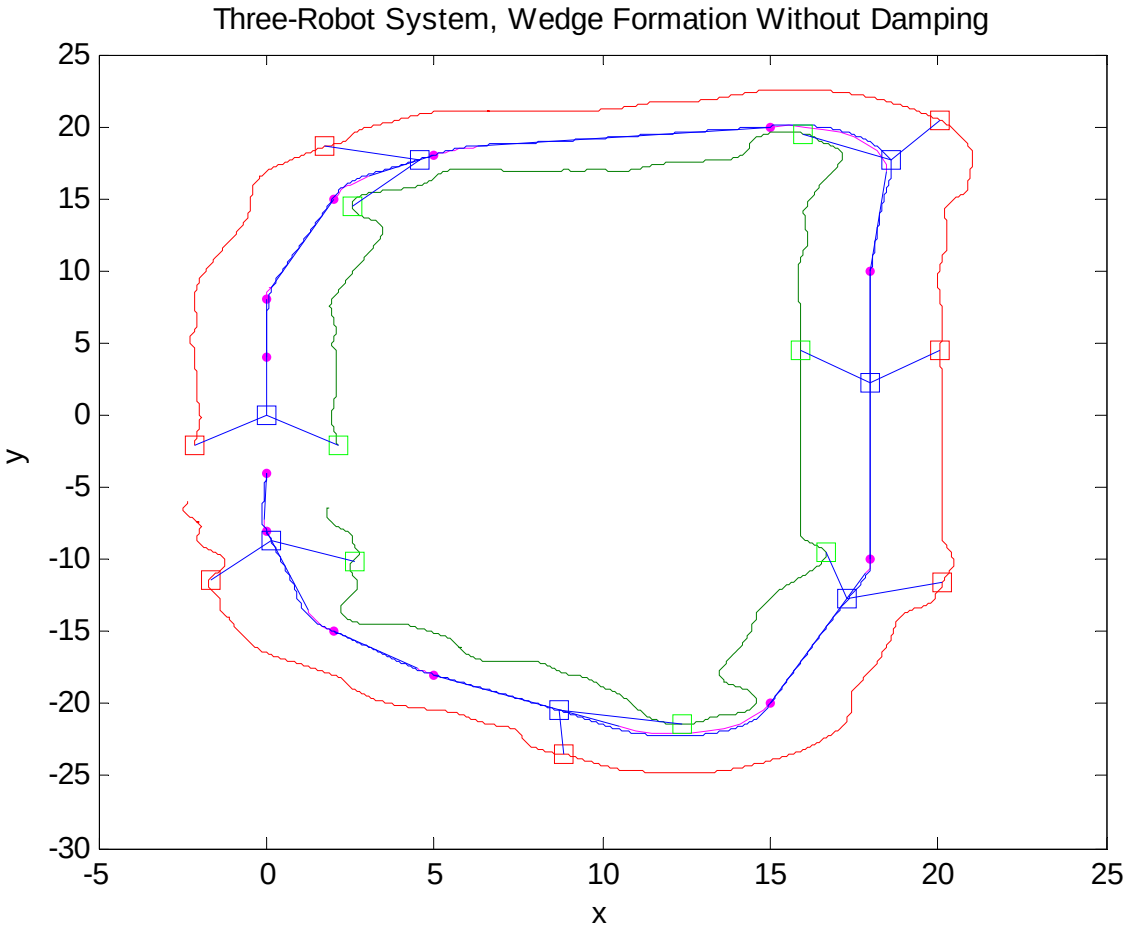


Figure 6-13: Results for a three-robot system in a wedge formation (blue:leader, green:follower 1, red:follower 2).

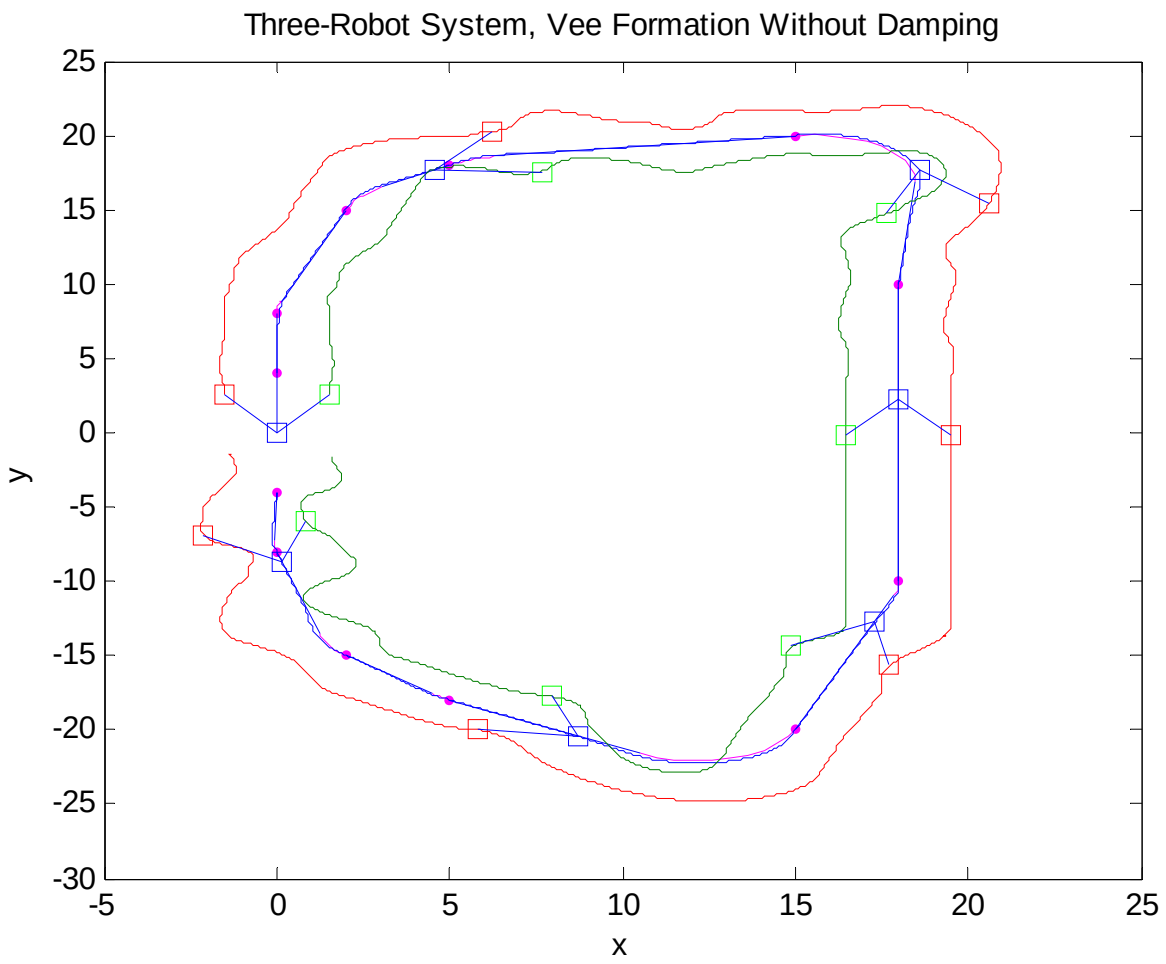


Figure 6-14: Results for a three-robot system in a vee formation (blue:leader, green:follower 1, red:follower 2).

Three-Robot System With Damping

The complicated nature of this trajectory contributes to oscillations in the robot formation. However, the robot system is able to maintain the desired formation regardless. Again, the damping slightly reduces the oscillations of the follower robots. For both cases, the formations are maintained as the robots traverse the given trajectory. Even though the vee formation is inherently unstable, it is still able to maintain the desired formation shape.

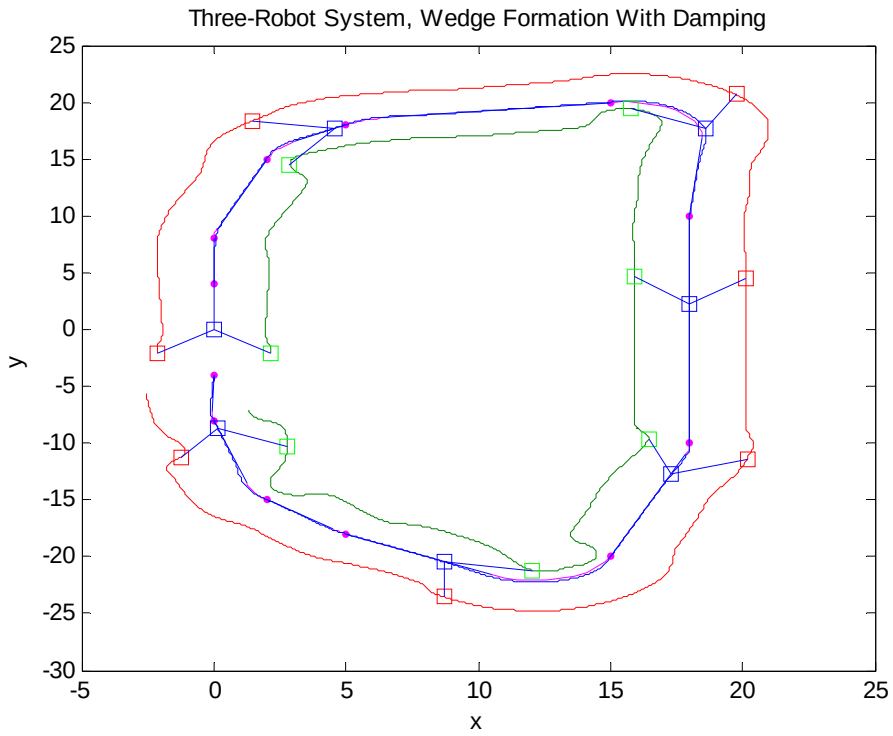


Figure 6-15: Results for a three-robot system with damping in a wedge formation (blue:leader, green:follower 1, red:follower 2).

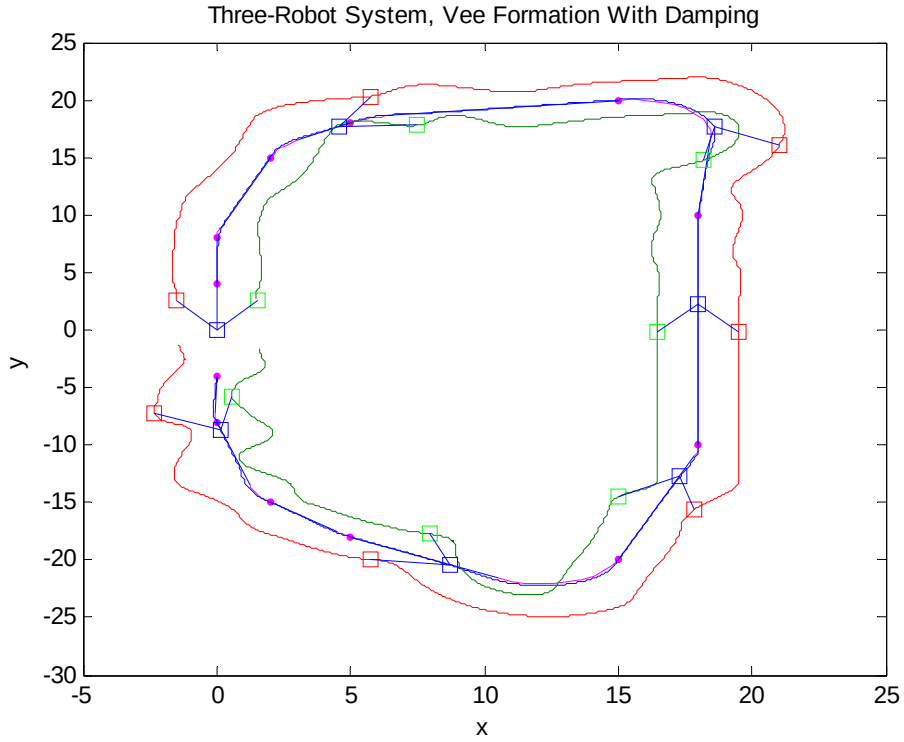


Figure 6-16: Results for a three-robot system with damping in a vee formation (blue:leader, green:follower 1, red:follower 2).

Five-Robot System Without Damping

The final experiment that is performed uses the racetrack trajectory and a five-robot system. The system of robots is expected to have results similar to those of the straight-line trajectory experiment. The results are presented in figures 6-17 and 6-18.

The inside robots – followers 1 and 3 – appear to have a more difficult time when maneuvering through sharp turns. The errors that follower 1 robot makes propagate to follower 3. This is apparent for the vee formation experiment. Like the inverted pendulum problem, the vee formation is inherently unstable. However, for both cases the robots are able to navigate the entire path, in formation most of the time.

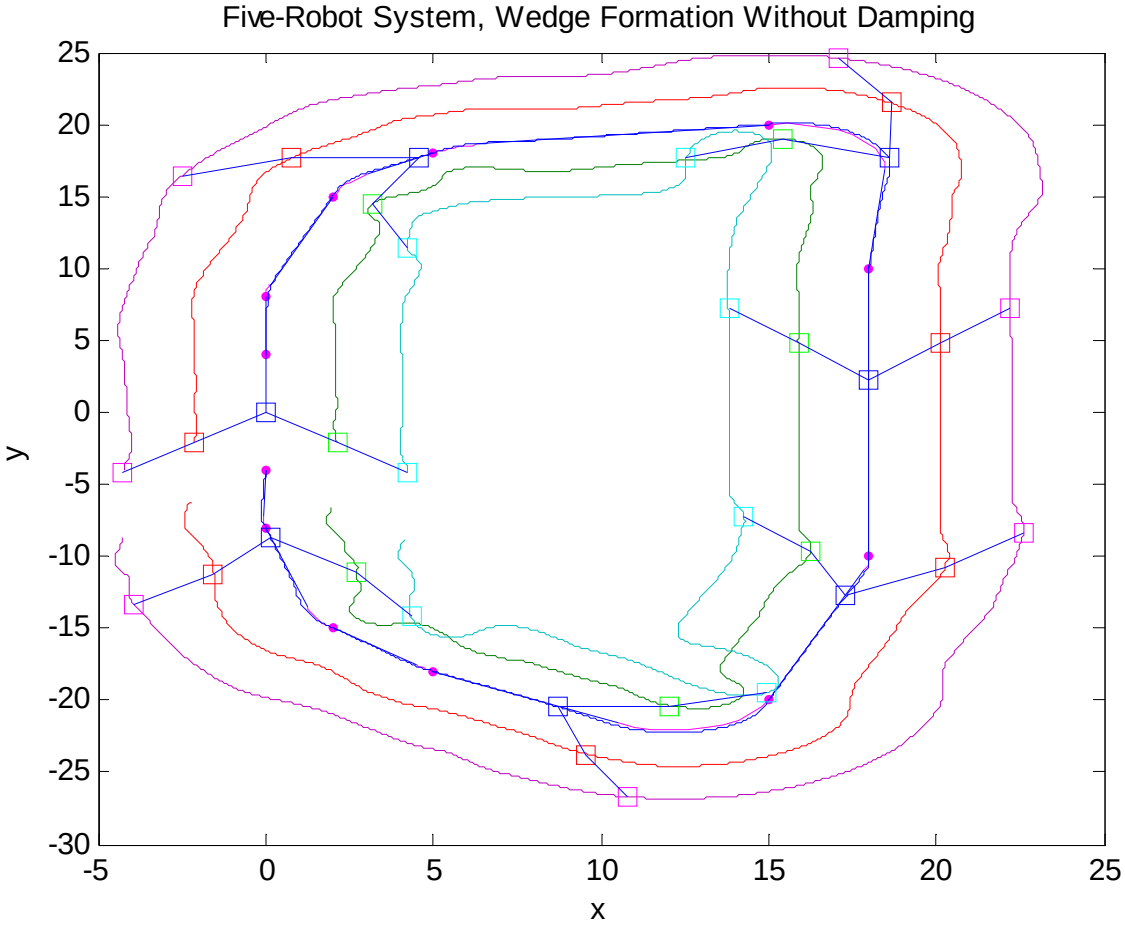


Figure 6-17: Results for a five-robot system in a wedge formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

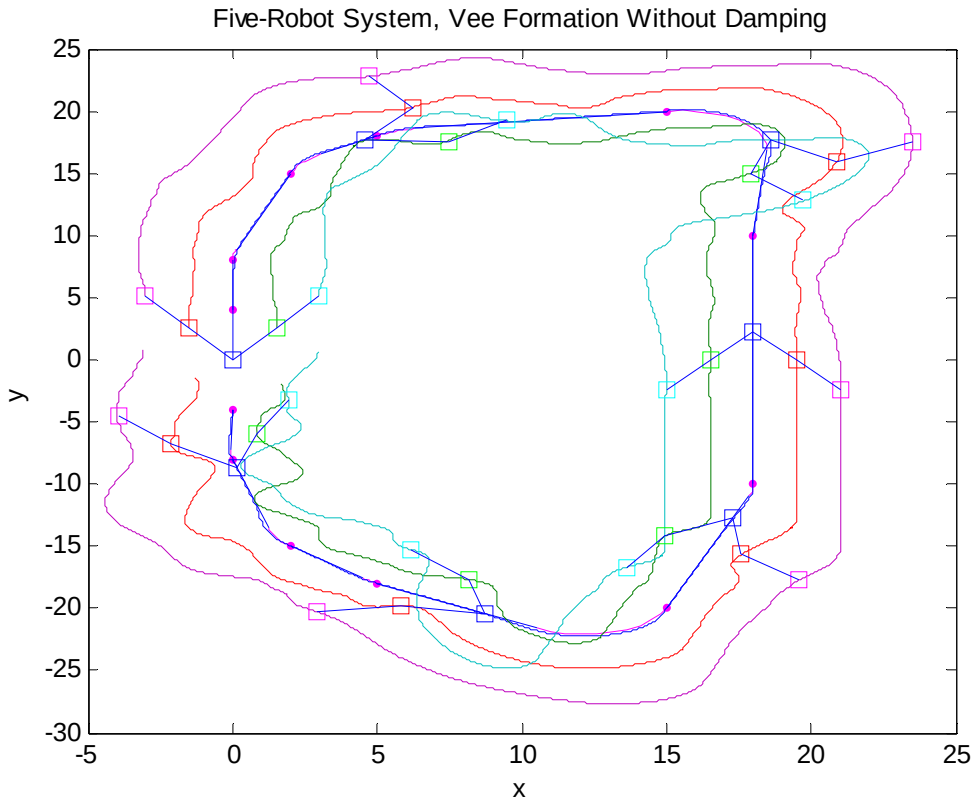


Figure 6-18: Results for a five-robot system in a vee formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

From figure 6-18 the vee formation has difficulty on the sharp turns encountered along the trajectory. However, it is able to return to formation afterward. Of all of the experiments this is the most extreme situation.

Five-Robot System With Damping

The final set of experiments involves the five-robot system on the racetrack trajectory that incorporates damping into the force calculations. The resulting positioning plots have similar results when compared to the system without damping. The results obtained from the wedge formation are slightly smoother when compared to results without damping. However, the results obtained from the vee formation did not dramatically reduce the system oscillations. This can be attributed to the inherently unstable nature of this type of formation.

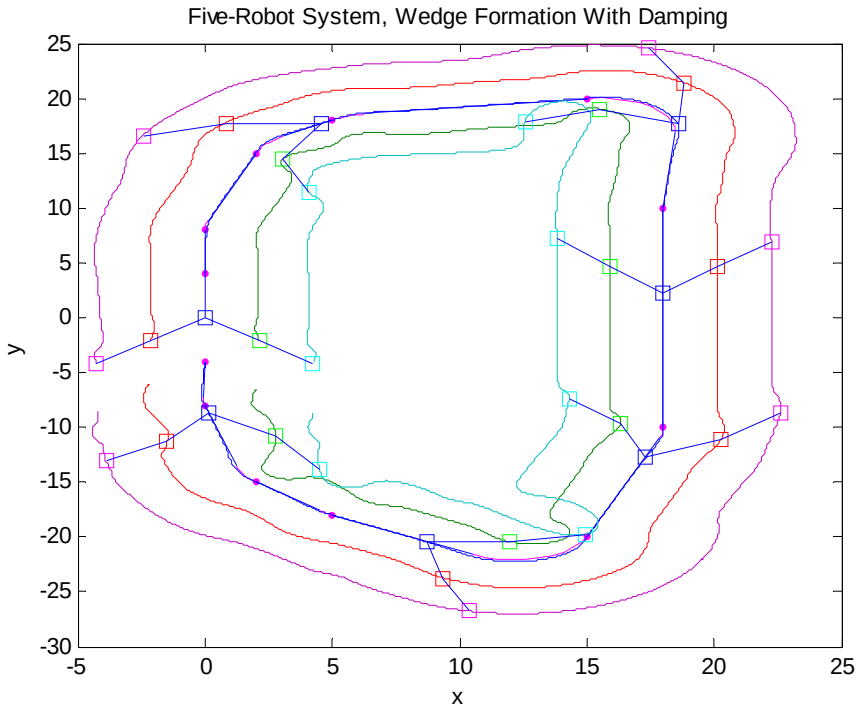


Figure 6-19: Results for a five-robot system with damping in a wedge formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

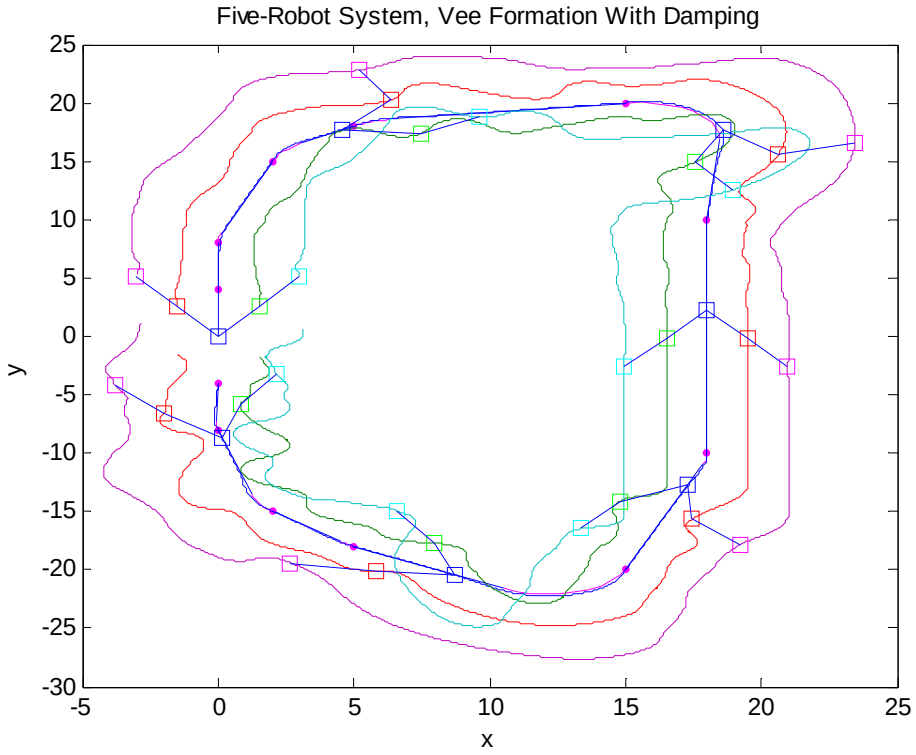


Figure 6-19: Results for a five-robot system with damping in a vee formation (blue:leader, green:follower 1, red:follower 2, cyan:follower 3, purple:follower 4).

Potential Energy

The data presented thus far has been position plots of the robots along two different trajectories. This data provides an insight into whether or not the formation is being maintained. In order to further validate the compliant formation control algorithm, it is necessary to observe the potential energy. The robot system is modeled in terms of (virtual) springs, dampers, and masses (the robots). Therefore the same basic energy principles can be applied to the robot system. The equations presented in an earlier chapter are used to calculate the potential energy of the entire system. This calculation is performed throughout the simulation and saved in a data file for analysis. An indication that the system is at equilibrium occurs when the potential energy reaches zero. Therefore, as the robots file into formation the system reaches equilibrium and the potential energy will tend toward zero.

An initial spike in the potential energy occurs due to the change in the robots' state. The potential energies are almost identical for both the vee and wedge formations for the three-robot system. In general, the five-robot systems have the highest potential energy with the vee formation having the highest for the straight line trajectory. The combination of two extra robots and an inherently unstable formation causes the increase in the potential energy for this case. Overall, the potential energy is approaching zero.

There is an initial spike in the potential energy for all of the different cases. In general, the energy peaks can be attributed the complexity of the trajectory and the drastic changes in direction. The wedge formation has a higher potential energy for the five-robot case for the racetrack trajectory when compared with the vee formation for this case. This can be attributed to the fact that the spring constants had to be adjusted in order to maintain the vee formation.

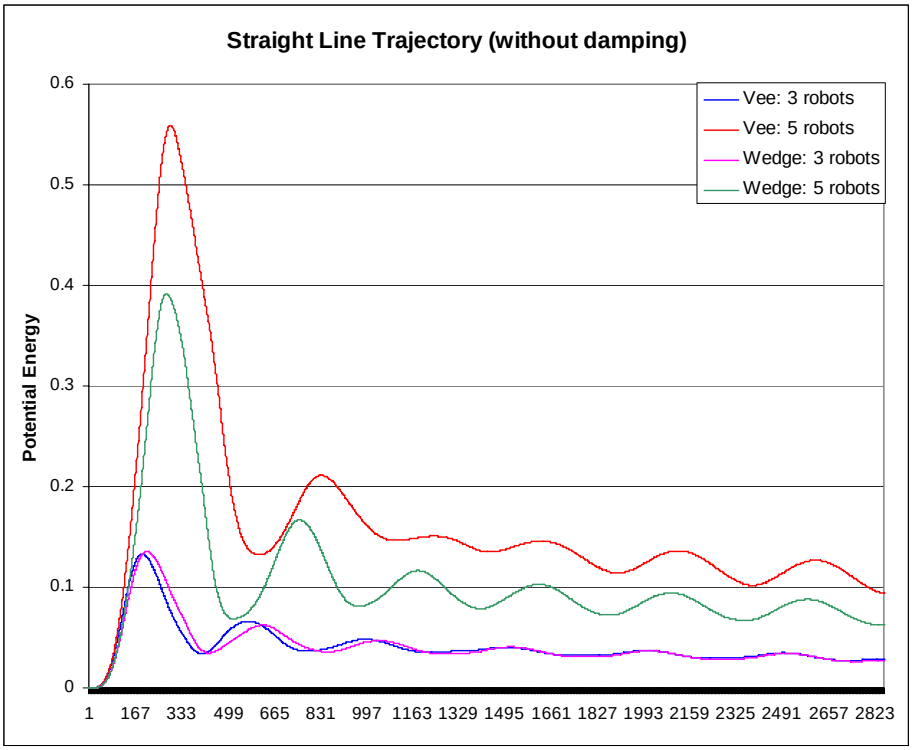


Figure 6-20: Potential energy plots for the straight line trajectory without damping.

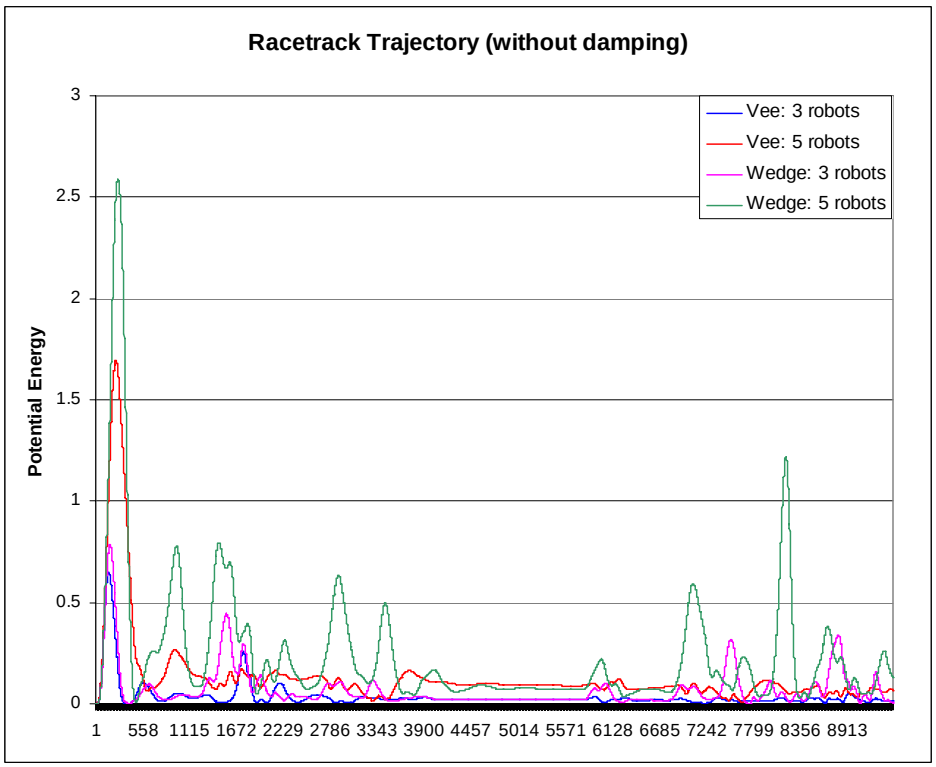


Figure 6-21: Potential energy plots for the racetrack trajectory without damping.

The system of robots could not maintain a vee formation using the spring values that were used for the wedge formation. In particular, the torsional spring constant was almost doubled in order to produce a viable vee formation.

The potential energy was collected for the experiments that include virtual damping, as well. The damping inadvertently affects the potential energy. The damping appears to smooth the potential energy plots which indicate there is an overall reduction in the oscillation of the robot system.

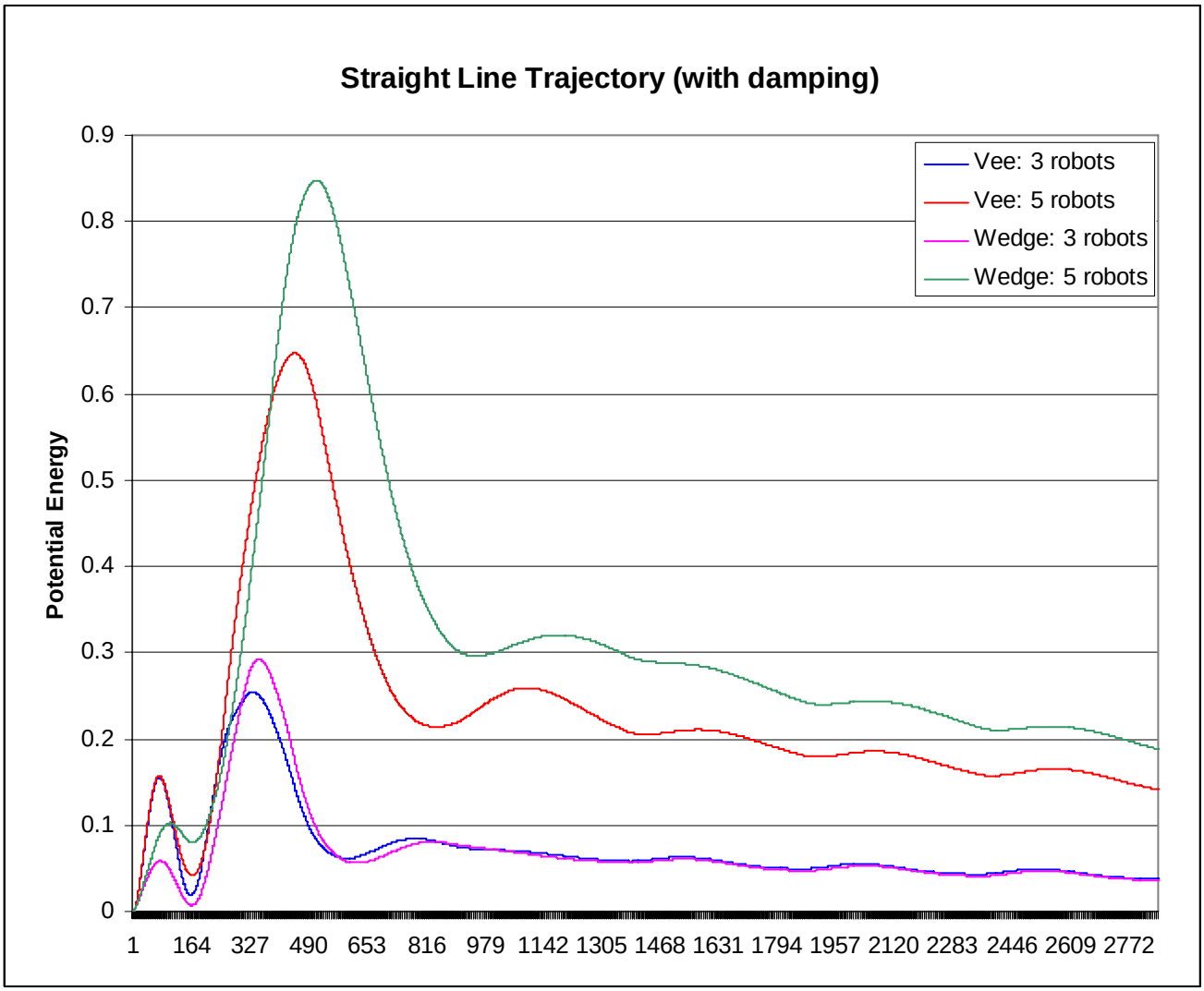


Figure 6-22: Potential energy plots for the straight line trajectory with damping.

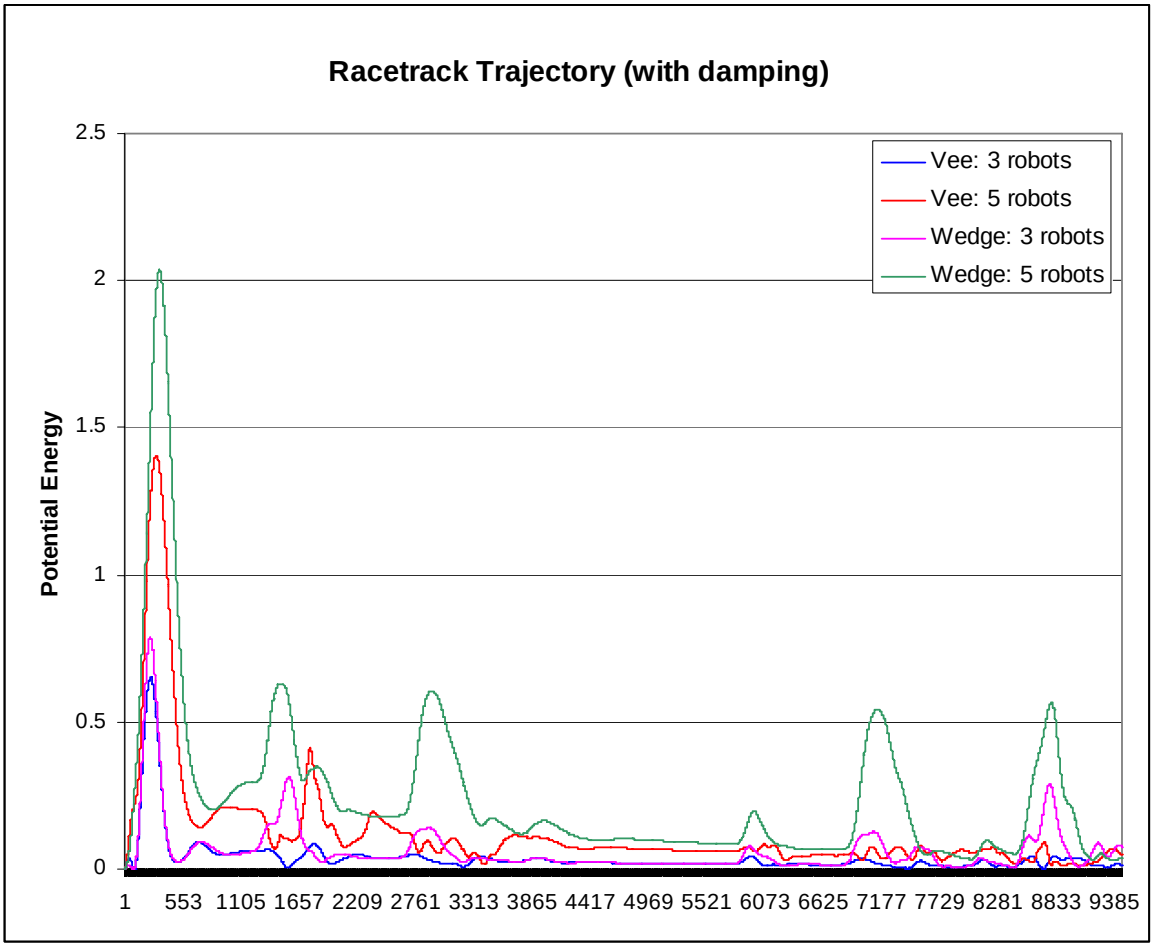


Figure 6-23: Potential energy plots for the racetrack trajectory with damping.

The results presented in this chapter provide evidence to support the premise of the compliant formation control algorithm. Positioning data was presented for three and five robots trying to maintain either a vee or wedge formation. Finally, an analysis of the potential of the system was presented for the different trajectories. Potential energy plots indicate that the system of robots is in formation as the potential energy tends toward zero.

Graphical User Interface Examples

The examples are taken from the GUI designed for testing and development of this algorithm. The images show the robots in formation as they navigate along the trajectory.

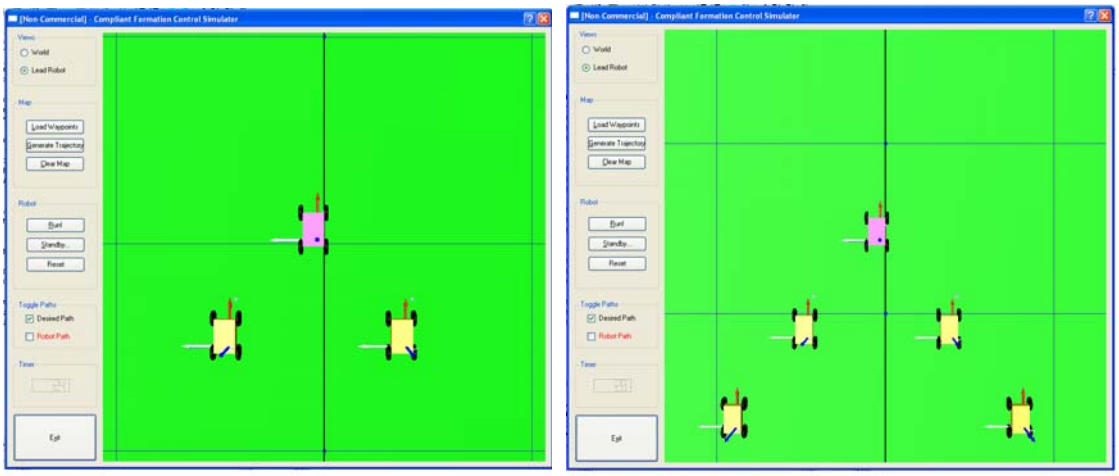


Figure 6-14: A three-robot and five-robot system in wedge formation.

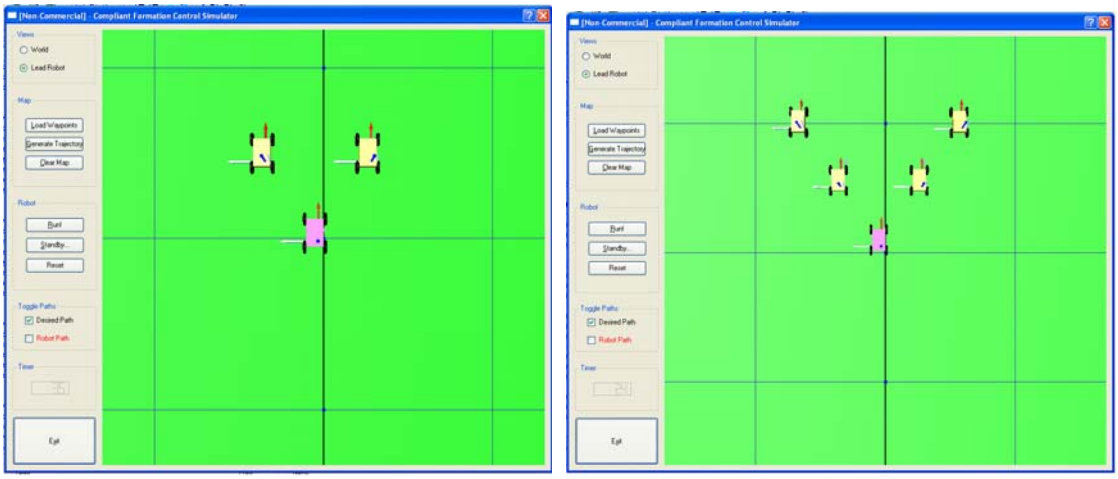


Figure 6-15: A three-robot and five-robot system in vee formation.

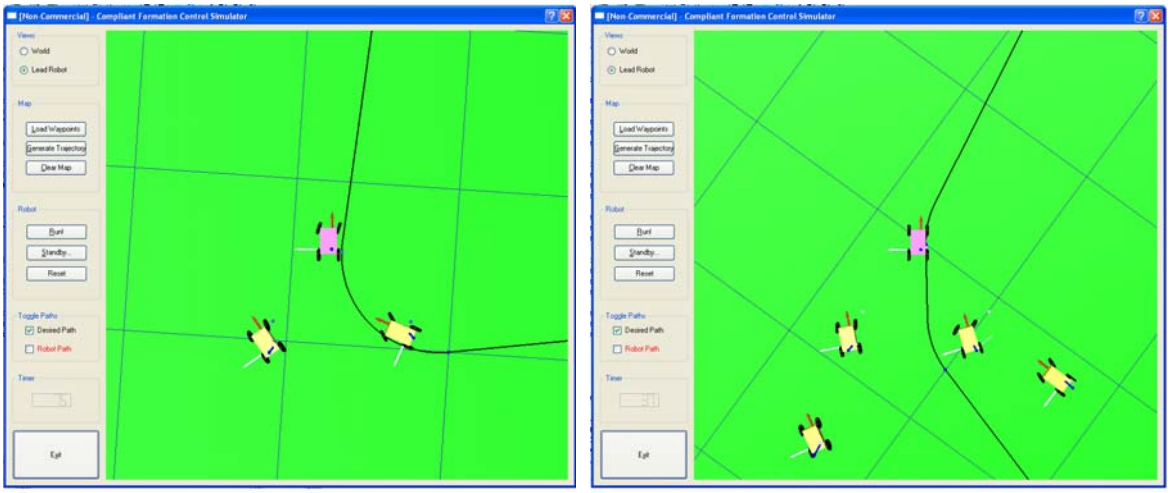


Figure 6-16: A three-robot and five-robot system negotiating a turn in wedge formation.

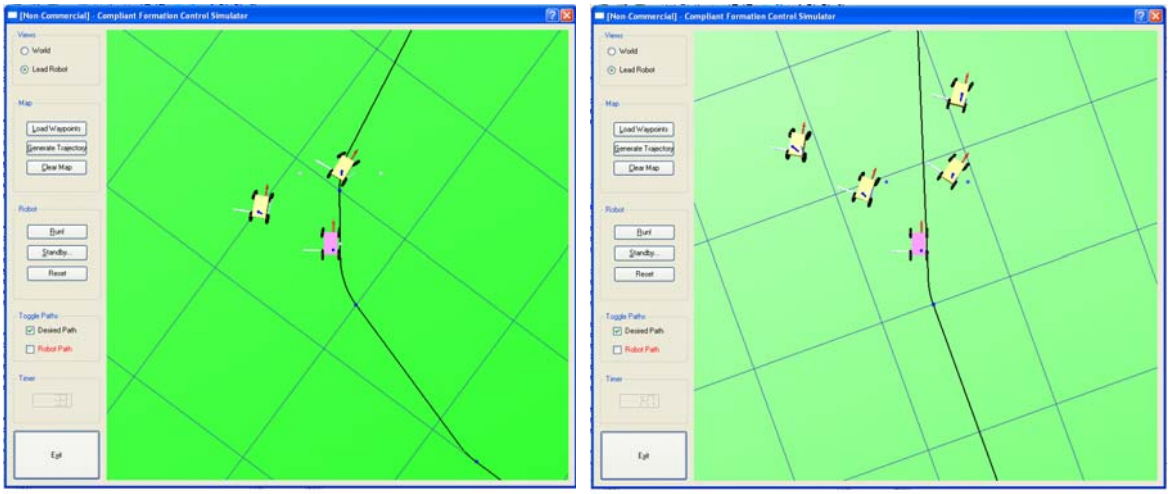


Figure 6-17: A three-robot and five-robot system negotiating a turn in vee formation.

CHAPTER 7
CONCLUSIONS

This research has presented a unique and novel algorithm for formation control of a multiple vehicle system. This coordination technique is fast and robust enough to maintain a formation of a group of unmanned ground vehicles while in motion. The formation control methodology combines the principles of vehicle dynamics and structural dynamics into a versatile, scalable algorithm.

The compliant formation control algorithm uses linear and torsional virtual springs to maintain the formation. The torsional spring is essential in order to maintain the proper formation orientation while the linear springs are mainly used to maintain the robot-to-robot distances. An unforeseen benefit of using virtual springs became apparent during simulation experiments. As the robots traveled along a trajectory, it was observed that the robots were prevented from colliding with one another. This phenomenon provides an indication that the algorithm was functioning properly.

The intricacies of the algorithm were discussed in detail and then implemented in custom software. The graphical user interface was used to perform several series of experiments that demonstrate the algorithm's ability. The GUI was also used to create a trajectory through a series of waypoints. This trajectory was an integral piece of the simulation environment, in that it provided the lead vehicle with a traversable path. The trajectory information was used to gauge the progress of the robot system.

Results were presented for three-robot and five-robot cases that were commanded to maintain two different formations. The results indicated that the robots were able to navigate the trajectory in formation for three or five robots. The vee formation proved to be the most unstable

formation for both the three-robot and five-robot cases. However, the formation was still maintained even with disturbances due to the inherent instability of the formation shape.

The potential energy results presented an additional method to quantify the robot system's ability to maintain the formation. The formation control algorithm makes use of virtual springs to impose a constraint on each vehicle. Therefore, the potential energy of the overall system was calculated. The plots of the potential energy indicate that as the system reaches equilibrium (i.e. is in formation) the potential will tend toward zero.

The compliant formation control algorithm can be further developed to account for obstacles present throughout the environment. Assuming that the lead vehicle has the ability to sense obstacles, upon detection, the obstacle would emit a repelling force based on its location relative to the leader. This force would act in a negative sense on the system of robots and the formation shape would be modified to account for avoidance of the obstacle.

This research has provided a viable methodology for formation control of a system of unmanned ground vehicles. The simulation incorporated a realistic vehicle model, trajectory generation, and animation of the vehicles along the trajectory. The results presented reinforce the validity of the compliant formation control algorithm.

LIST OF REFERENCES

1. Gould, J., Gould, C., The Honey Bee, Scientific American Library, New York, 1988.

2. Beni, G., Wang, J., Swarm Intelligence, NATO Advanced Workshop on Robots and Biological Systems, Il Ciocco, Tuscany, Italy, 1989.

3. Krieger, M., Billeter, J.B., Keller, L., Ant-like Task Allocation and Recruitment in Cooperative Robots, Nature, 2000, 406:992-995.

4. Holldobler, B., Wilson, E., The Ants, The Belknap Press, Cambridge, 1990.

5. Dorigo, M., Maniezzo, V., The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996, 26:1-13.

6. Arkin, R., Behavior -Based Robotics, MIT Press, Cambridge, 1998.

7. Arkin, R., Cooperation Without Communication: Multiagent Schema-Based Robot Navigation, Journal of Robotic Systems, 1995, 9(3):351-364.

8. Powers, M., Balch, T., Value-Based Communication Preservation for Mobile Robots, Presented at the 7th Annual International Symposium on Distributed Autonomous Robotic Systems, Toulouse, France, 2004.

9. Mataric, M., Coordination and Learning in Multirobot Systems, IEEE Intelligent Systems, March/April 1998:6-8.

10. Parker, L., A Case Study for Life-Long Learning and Adaptation in Cooperative Robot Teams, In Proceedings of SPIE Sensor Fusion and Decentralized Control in Robotic Systems II, 1999, 3839:92-101.

11. Castelpietra, C., Iocchi L., Nardi, D., Rosati, R., Coordination in Multi-Agent Autonomous Cognitive Robotic Systems, In Proceedings of 2nd International Cognitive RoboCup Workshop, Amsterdam, Netherlands, 2000.

12. Mäkelä, H., Kaarmila, P., Koskinen, K., Convoy Navigation, In Proceedings of 3rd Symposium on Intelligent Autnomous Vehicles, Madrid, Spain, 1998, 1:31-36.

13. Canudas-de-Wit, C., NDoudi-Likoho, A., Nonlinear Control for a Convoy-Like Vehicle, Automatica, 2000, 36:457-462.

14. Belkhouche, F., Belkhouche, B., Modeling and Controlling a Robotic Convoy Using Guidance Laws Strategies, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 2005, 35(4):813-825.

15. Schneiderman, H., Nashman, M., Wavering, A., Lumia, R., Vision-Based Robotic Convoy Driving, Machine Vision and Applications, 1995, 8:359-364.

16. Stella, E., Lovergine F., D'Orazio, T., Distante, A., A Visual Tracking Technique Suitable for Control of Convoys, Pattern Recognition Letters, 1995, 16:925-932.

17. Chaumette, F., Rives, P., Espiau, B., Positioning of a Robot with Respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing, In Proceedings of IEEE International Conference on Robotics and Automation, Sacramento, CA, 1991, pp. 2248-2253.

18. Marchand, E., Bouthemy, P., Chaumette, F., Moreau, V., Robust Real-Time Visual Tracking using a 2D-3D Model-Based Approach, IEEE International Conference on Computer Vision, Kerkira, Greece, 1999, 1:262-268.

19. Capparella, F., Freda, L., Malagnino, M., Oriolo, G., Visual Servoing of a Wheeled Mobile Robot for Intercepting a Moving Object, In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada, 2005, pp. 2737-2743.

20. Tsakiris, D., Rives, P., Samson, C., Extending Visual Servoing Techniques to Nonholonomic Mobile Robots, The Confluence of Vision and Control, Eds. Hager, G., Kriegman, D., and Morse, S., Lecture Notes in Control and Information Systems (LNCIS), Springer-Verlag, 1998.

21. Vicsek, T., Czirok A., Jacob, E., Cohen, I., Schochet, O., Novel Type of Phase Transitions in a System of Self-Driven Particles, Physical Review Letters, 1995, 75:1226-1229.

22. Jadbabaie, A., Lin, J., Morse, A., Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules, IEEE Transactions on Automatic Control, 2003, 48:988-1001.

23. Desai, J., Kumar, V., Ostrowski, J., Control of Changes in Formation for a Team of Mobile Robots, In Proceedings of IEEE International Conference on Robotics and Automation, Detroit, MI, 1999, 2:1556-1561.

24. Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H., Coordination for Multi-Robot Explorations and Mapping, In Proceedings of the National Conference on Artificial Intelligence, Austin, TX, 2000, pp. 852-858.

25. Howard, A., Mataric, M., Sukhatme, G., An Incremental Deployment Algorithm for Mobile Robot Teams, Presented at IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.

26. Nguyen, H., Pezeshkian, N., Raymond, M., Gupta, A., Spector, J., Autonomous Communication Relays for Tactical Robots, Presented at IEEE 11th International Conference on Advanced Robotics, Coimbra, Portugal, 2003.

27. Desai, J. P., Ostrowski, J., Kumar, V., Controlling Formations of Multiple Mobile Robots, In Proceedings of IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998.

28. Jongusuk, J., Mita, T., Tracking Control of Multiple Mobile Robots: A Case Study of Inter-Robot Collision-Free Obstacle Avoidance, In Proceedings of IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 2885-2890.

29. Ögren, P., Leonard, N. E., Obstacle Avoidance in Formation, In Proceedings of IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003, pp. 2492-2497.

30. Lewis, M. A., Tan, K., High Precision Formation Control of Mobile Robots Using Virtual Structures, Autonomous Robots 4, 1997, pp. 387-403.

31. Crane, C., Duffy, J., Kinematic Analysis of Robot Manipulators, Cambridge University Press, Cambridge, 1998.

32. Bouguet, J., Camera Calibration Toolbox for MATLAB®, http://www.vision.caltech.edu/bouguetj/calib_doc/, January 2004.

33. MacArthur, E., MacArthur, D., Crane, C., Use of Cooperative Unmanned Air and Ground Vehicles for Detection and Disposal of Mines, Presented at SPIE Optics East Conference, Boston, MA, 2005.

BIOGRAPHICAL SKETCH

Erica Zawodny MacArthur was born on July 6, 1978, of parents Joseph F. and Yvonne D. Zawodny in Miami, Florida. She graduated from MAST Academy High School in June 1996 and shortly thereafter began attending college at the University of Florida. She received a Bachelor of Science in Mechanical Engineering and graduated with honors in August 2000. She began studying for a Master of Science degree in the fall of 2000 and received her degree in May 2003. She continued her Ph.D. work at the University of Florida in the Department of Mechanical and Aerospace Engineering in the field of robotics with an emphasis on collaborative robotic control. Erica is married to Donald MacArthur and they currently reside in Gainesville, Florida.