Fall 2007

Vision based leader-follower formation control for mobile robots

Gerard Sequeira

VISION BASED LEADER-FOLLOWER FORMATION CONTROL FOR MOBILE ROBOTS

by

GERARD SEQUEIRA

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2007

Approved by

Sanjeev Agarwal, Advisor

R. Joe Stanley

Michael Nelson

ABSTRACT

Creating systems with multiple autonomous vehicles places severe demands on the design of control schemes. Robot formation control plays a vital role in coordinating robots. As the number of members in a system rise, the complexity of each member increases. There is a proportional increase in the quantity and complexity of onboard sensing, control and computation. This thesis investigates the control of a group of mobile robots consisting of a leader and several followers to maintain a desired geometric formation. The group considered has several inexpensive sensor-limited and computationally limited robots that follow a leader robot in a desired formation over long distances. This situation is similar to a search, demining, or planetary exploration situation in which several deployable/disposable robots are led by a more sophisticated leader. Complex sensing and computation are performed by the leader, while the followers perform simple operations under the leader's guidance. The architecture consists of two main components: (i) a model-based vision system and (ii) a control algorithm. The model-based vision system can recognize and relatively localize the follower robots using markers mounted on the leader robot. A Bézier trajectory based mechanism is selected to enable a group of follower robots to follow the leader. The following control method is mathematically simple, easy to implement, and well suited for long distance navigation. The algorithm only requires knowledge of the leader-follower relative distance and bearing angle. Both types of data are computed using measurements from a single camera, eliminating the need for a more sophisticated stereo system.

ACKNOWLEDGMENTS

I would like to express my heartfelt thanks to my advisor, Dr. Sanjeev Agarwal. Without his support and guidance this thesis would not have been possible. He provided a good mixture of guidance and independence, allowing me to explore and develop my own ideas. I learned a lot of research skills from him.

I am very thankful to Dr. Joe Stanley and Dr. Michael Nelson for agreeing to be the members of my thesis committee and for taking the time to read my work and provide valuable suggestions. I would also like to thank my colleagues who have made my stay in the ARIA Lab very enjoyable. My heartfelt thanks goes to my lab mate Shivakar Vulli for his immense help and patience in getting me acquainted with coding languages and Linux.

My parents, Jerome and Geraldine Sequeira, and my brother Joel deserve thanks for giving me the love and support that enabled me to do this work. Lastly, I wish to thank my four very good friends – Shubhika, for all her loving care, Amit, whose cooking will always miss, Pravin, for all his joking around, and Divya, for her care, support and understanding.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

1. INTRODUCTION

1.1. MULTI ROBOT-SYSTEMS

The prospects of multi-robot systems have been increasing in recent years. This is because the advantages such systems offer over a single robot including greater flexibility, adaptability, scalability, and affordability. Having a group of robots move in formation would be beneficial in many real world applications, such as search and rescue, demining in military missions, transporting large objects, and convoying. It is possible for one user to control an entire group of robots without having to specify explicitly the commands for each one. Figure 1.1 shows a situation in which only the lead truck in a supply convoy needs to be manned. In an alternative scenario, the lead robot may be endowed with more sophisticated sensors and computational capabilities for overall planning and navigation, while other robots in formation are simple and specialized. The de-emphasis of one large and expensive robot reduces the chance of a catastrophic mission failure. The use of several mobile robots in a coordinated manner enables the achievement of complex tasks. The robots do not have to be very complex in structure, since each one can be specialized for a particular task. Reducing the number of sensors on each robot in a multi-robot system plays a major role in reducing the complexity and cost.

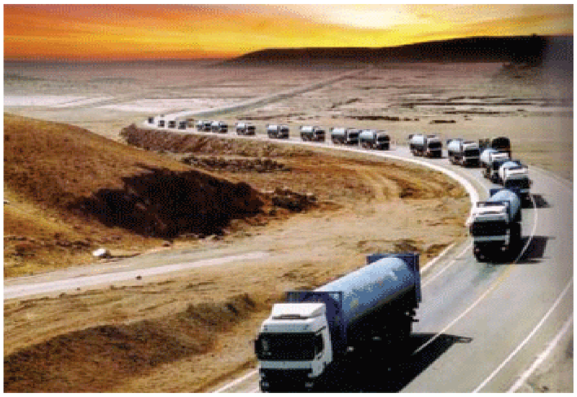


Figure 1.1. Convoy of Supply Trucks.[1]

[1] Picture courtesy: http://www.stockinterview.com/News/03222006/china-coal.html

One potential advantage of multi-robot systems is the reduction of human involvement in dangerous situations including search and rescue, mining areas, battlefields, and planetary missions. Such applications subject the robots to high damage and failure rates. Single units fitted with expensive equipment raise unacceptable economic concerns. The use of multiple robots equipped with low cost components, among which tasks are distributed are more suitable for such situations. Task distribution includes sensing, computation, and control for the group as a whole. Simple robots can be considered disposable to the overall mission, so loosing a few units does not result in mission failure. In a leader-follower framework, the leader is given the task of navigation, including path planning, and obstacle avoidance, whereas the followers' tasks involve tracking the leader, gathering data, and handling communication. These groups can be easily expandable to accommodate more units for a larger sensor coverage or troop movement.

This thesis addresses the problem of designing a leader-follower framework with sensor-limited robots. Given a leader robot that moves about in an unknown trajectory, an attempt was made to maintain the robots following the leader at a certain distance behind, by using only visual information about the position of the leader robot. The robots were designed to be as simple as possible in structure. The units are not equipped with expensive sensors such as laser or sonar rangefinders. Global Positioning System (GPS) receivers are subject to satellite availability and sometimes lose a signal if there is no clear view of the sky. Sensors for keeping track of robot movement, such as wheel encoders, are subject to accuracy issues and drift, making information gathered from such sensors unreliable. For these reasons, in the current work, all the sensing was done via vision. The team consisted of three or more units equipped with only a forward facing camera for gathering data.

1.2. RESEARCH MOTIVATION

Traditionally, the control design for mobile robots relies on measurement from dead reckoning sensors such as wheel encoders which provide odometer or position data. However, these measurements render themselves completely unreliable after a few meters of navigation due to the encoder's low accuracy and drift. Wheel slippage, uneven surfaces, and electrical noise render readings imprecise. Acoustic sensors such as radars and sonars are expensive and their readings are susceptible to stray reflections. They are also unable to recognize and distinguish between objects of similar shape, type and color. However, due to the effectiveness and low cost of vision sensors and their relative cheapness in computing power, the current trend is to design systems that use vision as their primary sensor. Image processing in software obviates the need for complex systems using odometry, sonar or laser sensors. In this thesis, an effort was made to acquire localization and sensory information through vision. Although this method results in increased computation, better algorithms can be applied to reduce this concern. This research entailed getting a group of mobile robots with inexpensive sensors to follow a leader with a desired geometric formation. A single forward facing camera was the only sensor being used for setting up and maintaining formation. These robots are significantly less powerful and complex in comparison to other robotic systems in terms of sensors and computational power.

1.3. RELATED WORK

1.3.1. Visual Tracking. Localization using vision sensors in robot formations is also known as the visual tracking problem. Fiala, (2004) presented a vision based system for controlling multiple robot platforms in real time using imagery from a top view video camera. Lowe, (2004) described a method that involved extracting distinctive features by matching individual detected features to a feature database of known objects using a fast nearest neighbor algorithm. Chen et al. (2005) developed a monocular-camera based visual servo tracking controller for a mobile robot subject to nonholonomic motion constraints using Lyapunov-based techniques. By comparing corresponding object target points from two different camera images, geometric relationships are exploited to derive a transformation that relates the actual position and orientation of the mobile robot to a reference position and orientation. This transformation is used to synthesize a rotation and translation error system from the current position and orientation to the fixed reference position. Han and Hahn, (2005) used a single forward facing camera to determine the relative position of a robot with respect to another robot or object. Since the shape of the target in the image frame varies due to rotation and translation of the target, they suggested a tracking scheme which uses the extended snake algorithm to extract the contour of the target and updates the template in every step of the matching process. In this work, the follower robots forward facing camera has an unobstructed view of the leader robot. The leader has a colored pattern mounted behind it. This helps in determining the position and orientation of the leader robot with the follower robot as a reference. Perspective geometry is used to transform real world 3D coordinates to a 2D image plane.

1.3.2. Formation Control. A variety of approaches have been proposed for robot formation and control. One of the first approaches was developed by Balch and Arkin, (1994) who proposed a behavior-based approach for formation of a team of military unmanned ground vehicles as scout units equipped with GPS, sonars, and vision sensors. Tan and Lewis, (1996) applied the concept of virtual rigid structure for formation maintenance. Their algorithm assumed that all robots had global knowledge and iteratively fit the virtual structure to the current robot positions, displaced the virtual structure in some desired direction, and updated the robots' positions. Vidal et al. (2003) translated the formation control problem into a separate visual servoing task for each follower, which is also the same approach used in this thsis. The follower robot uses vision to estimate the position and velocities of its leader. However, this was accomplished using omni-directional cameras. Although using distributed camera sensors in these approaches requires intense computation at each robot, implementing better algorithms for vision processing reduces the need for higher processing power. Chiem and Cervera, (2004) proposed an efficient method to control robot formations using Bézier trajectories. Each follower robot uses a forward facing color-tracking vision camera to estimate the relative pose of the leader. Then, a local Bézier trajectory is created and followed. The work presented in this thesis uses Bézier curves for trajectory generation on simple lines, because it is computationally simple and is well suited to the simple follower robots. Work by both Fredslund and Mataric, (2002) and Michaud et al. (2002) involved the followers panning their cameras to center the leader in the camera's field of view. Their works also proposed methods in which the robots initialize and determine their own positions in the formation, in addition to formation control. However each member of the team of homogeneous robots, is equipped with sonar, laser, a camera, and a radio link for communicating with the others. Parker et al. (2004) presented a control approach for heterogeneous robots in which a more capable leader assists simpler follower robots that have no onboard capabilities for obstacle avoidance or localization and only minimal capabilities for kin recognition. The leader controls navigation using a “chaining” formation method, forming a sensor network. Das et al. (2001) presented a paradigm for switching between centralized and decentralized controllers that allows for changes in formation. Like the leader, followers can also take part in obstacle avoidance. A single omni-directional camera is used in all robots and a host computer is used as a centralized processing unit. The host computer receives video from all robots and calculates relative velocity and pose between each follower and its leaders. Desai et al. (1998) used methods of feedback linearization for controlling formations that utilize only local sensor-based information in a leader-follower motion. First, the lead robot is given a motion plan. Each robot has the ability to measure the relative position of other robots that are immediately adjacent to it. Once the motion for the lead robot is given, the remainder of the formation is governed by local control laws based on the relative dynamics of each of the follower robots and the relative positions of the robots in the formation. Akella and Hutchinson, (2002) address the task of coordinating the motions of multiple robots when their trajectories (defined by both the path and velocity along the path) are specified, used primarily to avoid collisions between units. In Hutchinson’s work, each follower robot is equipped with a forward facing camera. Visual tracking the leader robot gives its orientation and position with the follower robot as a reference. A Bézier curve is then generated between the two robots, enabling the follower to track the leader robot using velocity control.

1.4. ORGANIZATION

This thesis is organized as as described below. Section 1 provides an introduction to research topics related to this work. Multi-robot systems are introduced, then related work on various visual tracking and formation control methods is reviewed. Finally, the motivation behind this work is discussed.

In Section 2, the whole system setup is described in detail. The physical and electrical configuration of the robots is presented and the visual capture algorithm is discussed. The technique used to measure and estimate the robot position and orientation using parameters from the visual algorithm is discussed, followed by a description of the software used.

Section 3 describes the trajectory tracking control algorithm, which is based on Bézier Trajectories.

Section 4 presents the experimental results of the formation control law, using the mobile platforms in the laboratory. The experimental setup is described in detail followed by results with different formations.

Section 5 summarizes the contributions of this work and identifies areas in which future work could improve and extend the methods developed here.

2. DESIGN METHODOLOGY

2.1. SYSTEM OVERVIEW

Each robot in a team is capable of localizing and following the leader robot using vision. The system structure of such a robot is shown in Figure 2.1. It consists of six units: main unit, power unit, locomotion, sensing, communication and the host computer units. The main unit, an ARM7 CPU, is described in Section 2.2.1. The power unit is a single 8V lithium-polymer battery along with 3.3V and 5V voltage regulators for powering the main control board. The drive motors run directly on the 8V. An onboard CMOS camera captures JPEg images and sends them over serially to the ARM7 controller. However, in the research for this thesis, a USB camera was used due to camera firmware issues. A Zigbee radio handles communication between the ARM7 controller and the host computer. All the image processing is accomplished by the host computer and control commands are sent back to the robots locomotion module.

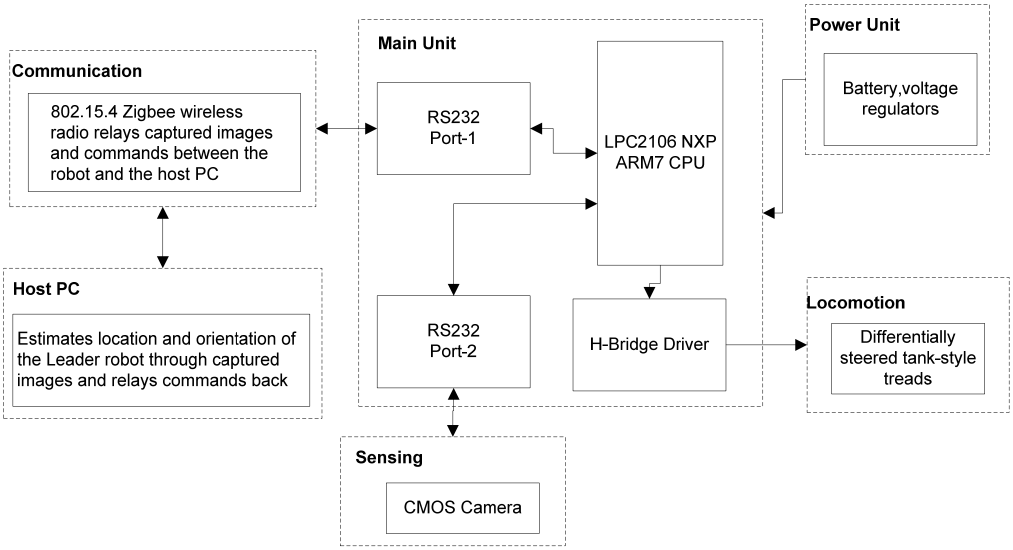

Communication
802.15.4 Zigbee wireless radio relays captured images and commands between the robot and the host PC
Main Unit
RS232 Port-1
LPC2106 NXP ARM7 CPU
RS232 Port-2
H-Bridge Driver
Power Unit
Battery,voltage regulators
Host PC
Estimates location and orientation of the Leader robot through captured images and relays commands back
Locomotion
Differentially steered tank-style treads
Sensing
CMOS Camera

Figure 2.1. Robot System Overview.

2.2. ROBOT HARDWARE

The robots, based on Surveyor™ Corporations[2] SRV-1 platform, are palm sized and feature two independently driven, differential-steered tank-style treads run via two DC gear motors. An ARM7 controller handles processing and manages onboard peripherals. The platforms are equipped with infrared sensors for detecting near ranged obstacles and a forward facing camera. All communication between the robots and the host computer is conducted serially via Zigbee wireless radios. Figure 2.2 illustrates key platform parts. The system is powered by an 8V lithium polymer battery. Aside from the drive motors, the platform runs at 3.3V to 5V.

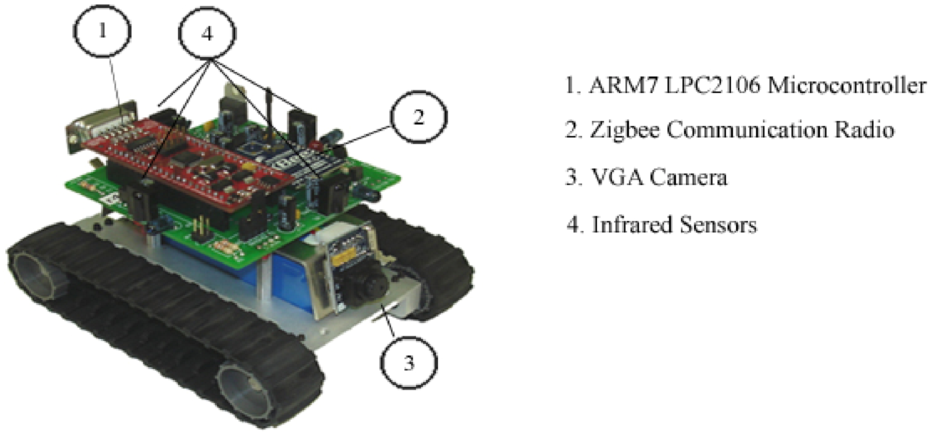

1
4
2
3
1. ARM7 LPC2106 Microcontroller
2. Zigbee Communication Radio
3. VGA Camera
4. Infrared Sensors

Figure 2.2. Key Parts of the Robot Platform.

2.2.1. Processor.

The platforms are powered by a 60MIPS 32bit ARM7TDMI-S LPC2106 processor from Philips NXP with 64kB on chip static RAM and 128kB on chip flash memory for storing code. The ARM7TDMI-S core is a synthesizable embedded RISC processor that provides system designers with the flexibility necessary to build embedded devices requiring small size, low power, and high performance. The processor employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions or applications in which code

[2] Surveyor™ Corporation, http://www.surveyor.com/

density is an issue. The key idea behind Thumb is a super-reduced instruction set. Essentially, the ARM7TDMI-S processor has two instruction sets including the the standard 32-bit ARM set and the 16-bit THUMB set.

The Thumb sets 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM’s performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Due to the huge code size of this work, all code has been written in Thumb mode.

Peripherals for the LPC2106 include two Universal Asynchronous Receivers Transceivers (UARTs). One UART provides a full modem control handshake interface; the other provides only transmit and receive data lines. The implemented I2C-bus supports bit rates up to 400 Kbit/s (Fast I2C-bus). Six single edges and/or double edge controlled PWM outputs are available for motor control.

As shown in Figure 2.3, the processor is based around a development board from Embedded Artists.

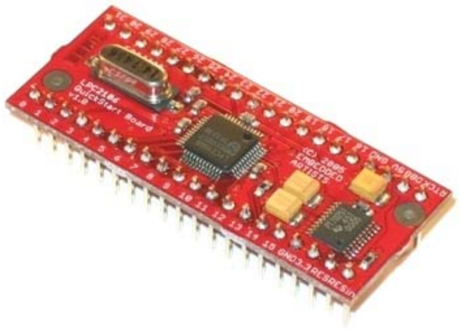


Figure 2.3. The ARM7 LPC2106 Controller from Philips.

The board runs at a voltage of 3.3V and has all of the processor's pins routed to header for easy interface. Downloading of new firmware onto the chip is handled by the chip's onboard boot loader using a serial channel.

2.2.2. Wireless Communication. Each SRV1 communicates with the base station via Zigbee 802.15.4 compliant wireless radios. These are full fledged radio modems capable of speeds up to 250,000bps. They have a transmitting power of 100mW with an indoor range of about a 100m each. Data is transferred between the robots and the base station at 115200bps. The radios are used for telemetry, as well as for remotely downloading code onto the robots. All SRV-1 data and control commands, including camera images are sent via these radios. Figure 2.4 shows one such radio module. Figure 2.5 shows the base station, which consists of a similar radio connected via Universal Serial Bus.

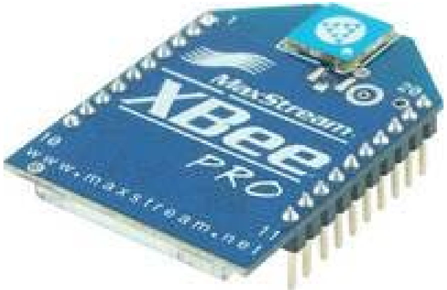


Figure 2.4. The Zigbee Radio Modem.

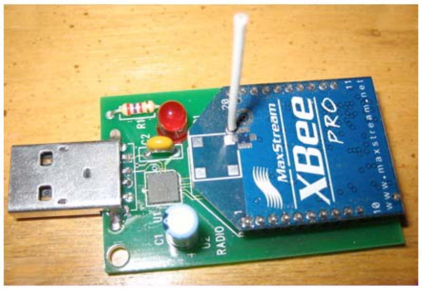


Figure 2.5. USB to Serial Radio Adaptor.

2.2.3. Camera. The SRV-1 robots are equipped with a C328 JPEG compression module which performs as a video camera or a JPEG compressed still camera (OV7640 sensor). The included lens has a focal length of 4.63mm with a 57 degree field of view.

As show in Figure 2.6, the camera consists of a lens, an image sensor (Omnivision’s OV7640), and a compression/serial-bridge chip (Omnivision’s OV528). The OV7640 is a low-voltage CMOS image sensor that supports various image resolutions (VGA, CIF, SIF, QCIF, 160×128, 80×64) as well as various color formats (4 gray/16 gray/256 gray/12-bit RGB/16-bit RGB). It provides complete user control of image quality, formatting, and output data transfer. The OV528 Serial Bridge is a controller chip that implements both a JPEG compression engine and a serial (RS-232) interface to a host controller. The OV528 implements a set of 11 initialization commands, including taking a snapshot, getting a picture, and setting the packet size.

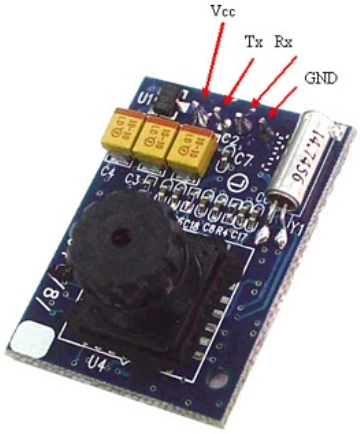


Figure 2.6. CMOS Camera Module.

Commands to the camera are issued via a serial RS-232 interface at 921,000bps. Snapshot commands from the ARM7 controller capture a full resolution single-frame still picture. The picture is then compressed by the JPEG engine (OV528) and transferred to the host which is further relayed to the base station for processing. The camera is capable of taking JPEG snapshots at 80x64, 160x128, 320x240, 640x480 resolutions. The 160x128 resolution is used due to radio bandwidth issues. The camera provides a frame rate of about 1 to 2 frames per second. Figure 2.7 shows a JPEG image as seen by one of the robots.

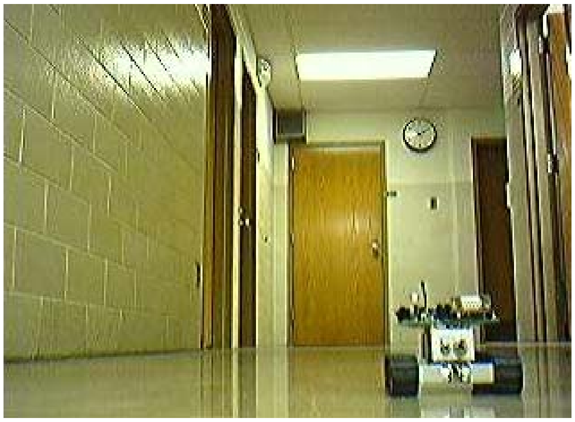


Figure 2.7. Follower View of the Leader Robot

Due to firmware problems of the camera module, however, it was not possible to obtain the required frame rate of 15-20 frames per second. This reduced frame rate resulted in a lot of lag and miscalculations in sending commands to the robot. Hence, in this research, a low cost CCD camera was fixed and used instead of the onboard camera. The webcam is a Philips 900NC model capable of grabbing images at 30 frames per second. Figure 2.8 shows one of the modified robots fitted with the webcam. Images are captured and analyzed at a rate of 15 frames per second by the vision processing program.

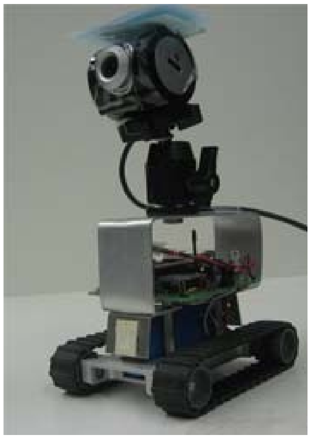


Figure 2.8. Follower Robot Fitted With a CCD USB Camera.

2.3. VISUAL SYSTEM

The visual tracking problem is divided into target detection and pose estimation. Target detection, explained in this section, involves capturing an image from the camera and processing it to detect features of interest. Estimation of pose is explained in Section 3.0. The vision processing can be divided into the tasks illustrated in Figure 2.9.

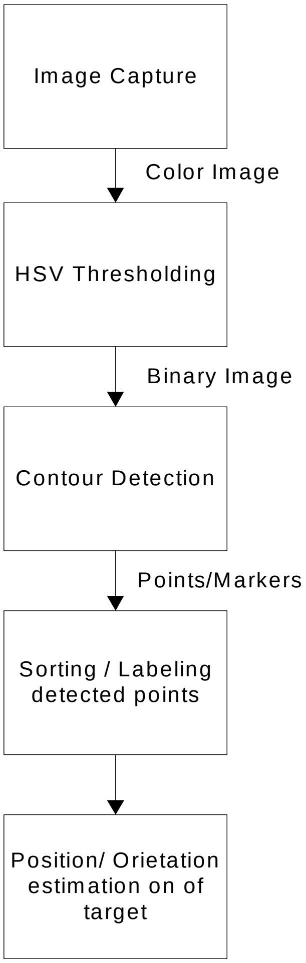


Figure 2.9. Vision Processing Flow Control.

2.3.1. Image Capture. The CCD camera captures images of the leader robot in a JPEG format. Captured Images are the converted into OpenCVs IplImage format for processing. Images are captured at the rate of 15 frames per second.

2.3.2. HSV Thresholding. One of the main tasks for the vision system is to detect the leader using colored blobs mounted on the leader robot. Detection is done through color segmentation in Hue Saturation Value (HSV) space. While the Red Blue Green (RGB) color system is used widely by most digital and capturing devices, it is not suitable for use in recognition tasks. When the value of a separate channel (red, blue, green) changes, the color presentation of the entire image is affected greatly whereas in the HSV space, the Hue value represents color while saturation indicates intensity of the color and value contains information about how bright the pixel is. Unlike the RGB space where colors are mixed up from three different color channels, using the HSV color system to detect a feature of a particular color is much satisfactory.

Figure 2.10 shows the HSV thresholding sequence where a three channel IplImage captured into OpenCV which is converted from RGB image space to HSV as shown in Figure 2.10b. The HSV image is further stripped into three individual hue, saturation and value channels. The hue channel is then individually thresholded to detect a specific color. The saturation and intensity values are thresholded depending on lighting conditions. Since the HSV thresholded IplIimage as shown in Figure 2.10c is a one channel image, it does not show any color.

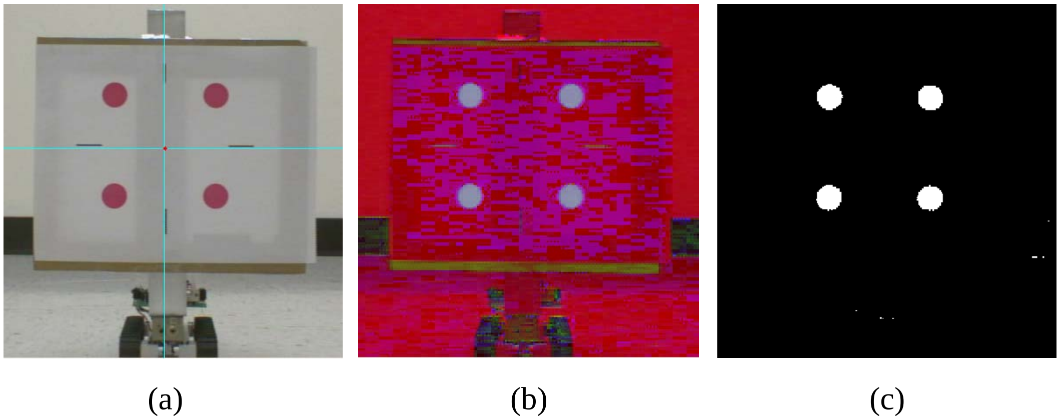
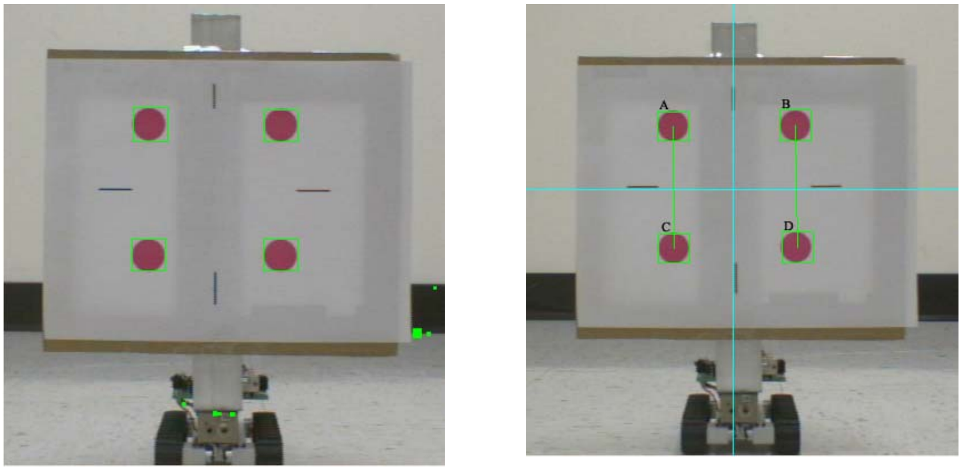


Figure 2.10. HSV Thresholding Sequence (a) Color Image, (b)the HSV Image, and (c)the HSV Thresholded Image.

2.3.3. Contour Extraction and Selection. The thresholded images are run through an OpenCV implementation of contour extraction based on the algorithm developed by Suzuki and Abe, (1985) due to its simplicity, robustness and fast speed. The Suzuki-Abe algorithm retrieves contours from the binary image by raster scanning the image to look for border points. Once a point that belongs to a new border is found, a border following procedure is applied to retrieve and store the border in the Freeman chain format[3]. During the border following procedure, visited border points are marked with a special value. The algorithm outputs a list of contours using Freeman chain code. Figure 2.11 shows the detected contours. However, since the algorithm detects contours of any shape and size, a few false detections are generated, as illustrated in Figure 2.11a. Since the markers on the leader robot are of a known area range, the detected contours are area thresholded. Figure 2.11b shows the detected markers after filtering.



(a) (b)

Figure 2.11. Detected Contours (a)False Detection, (b)False Detections Filtered.

[3] OpenCV Reference Manual – http://www.sourceforge.net/projects/opencvlibrary/

2.4. SOFTWARE

All code used in this work was developed using GNU open source tools. Code for the ARM7 microcontroller was written in C/C++ using an ARM port of GCC, which is a GNU compiler. OpenCV, an open source computer vision library developed by Intel was used to process images. This library provides functions for image capture and tracking, as well as processing. OpenCV is an image processing library developed by Intel™ specifically for their processors. It makes use of both the multimedia and streaming Single Instruction, Multiple Data (SIMD) extensions (MMX and Streaming SIMD Extensions) that Intel have introduced into their Pentium range, resulting in image manipulation speeds of up to 25fps.

Figure 2.12 is a flow diagram illustrating how captured images are sent wirelessly to a host computer for processing. Features are then extracted and used by the visual servoing algorithm for generating velocity feedback commands. These commands are then sent back to the robots via the same radio link.

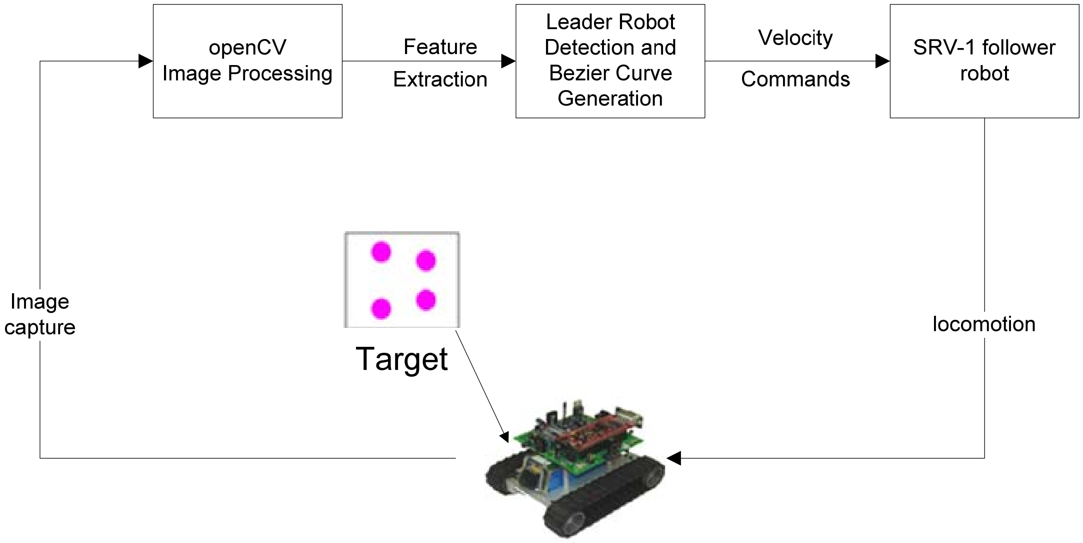


Figure 2.12. Vision Processing Flow Diagram.

3. ROBOT FORMATION CONTROL

3.1. ROBOT MODEL

The two treaded robot can be described by the following kinematics equations:

$$\dot{x} = v\cos\varphi \tag{1}$$

$$\dot{y} = v\sin\varphi \tag{2}$$

$$\dot{\varphi} = \omega \tag{3}$$

where (x,y) are the Cartesian coordinates of robot position, φ is the robot orientation angle, and v and ω are the robot linear and angular velocities. v_1 and v_2 denote linear speeds of the left and the right wheels, respectively. The linear (v) and angular (ω) velocities are expressed below:

$$v = \frac{(v_1 + v_2)}{2} \tag{4}$$

$$\omega = \frac{(v_1 - v_2)}{L} \tag{5}$$

where L represents the distance between the two driving wheels. Figure 3.1 illustrates the geometric model of the robot.

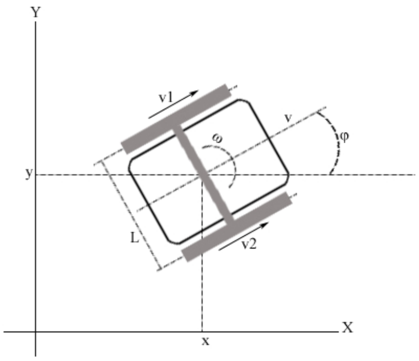


Figure 3.1. Robot Geometric Model.

3.2. VISUAL MEASUREMENT OF TARGET POSTURE

This section presents a vision-based framework for mobile robot detection and tracking using off-the-shelf cameras mounted on the robot. Target detection and pose estimation are performed from single frames using markers as key elements. The method consists of tracking a rectangular shaped structure behind each robot. Determination of the position and orientation of the leader can be achieved by estimating its distance and relative orientation with regard to the followers. As shown in Figure 3.2, only the position (x, y) and orientation θ needs to be estimated.

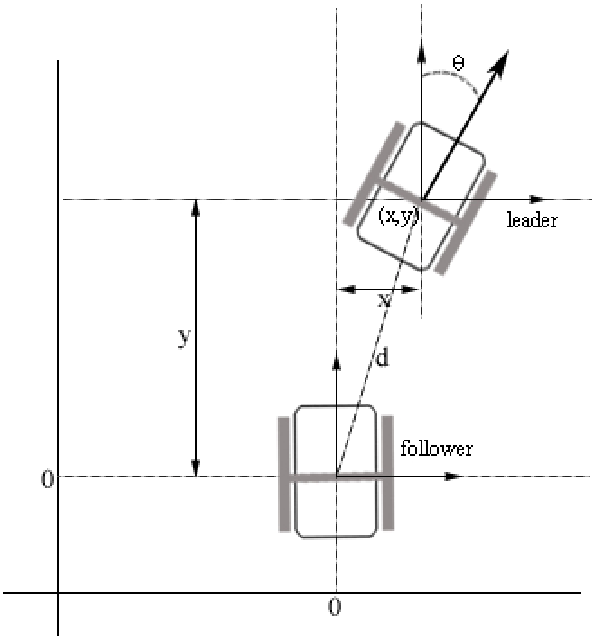


Figure 3.2. Position and Orientation of the Leader in the Followers Frame of Reference.

Pose and orientation estimation refers to the issue of obtaining relative position and orientation between two or more mobile robots. The camera captures a pattern mounted on the leader robot, as shown in Figure 3.3.

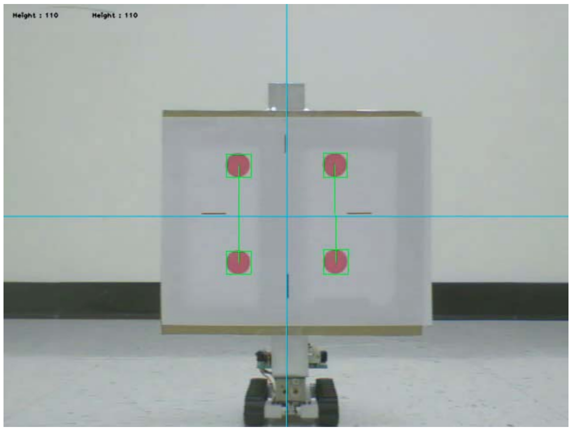


Figure 3.3. Figure Showing the Follower Robots Camera Looking at the Leaders Pattern.

The pattern as shown in Figure 3.4, features four circles at each corner of a square of known length E(mm). The two segments AC and BD provide an estimate to the distance between the follower and the leader based on their perceived and real heights. The difference between the perceived heights of the two segments gives an estimate of the orientation of the pattern with respect to the follower robot. The suggested pattern on the leader robot is illustrated in Figure 3.4a. Figure 3.4b. Illustrates the pattern as observed in the follower robots image plane (X_{image},Y_{image}). h_L and h_R are the heights of the two segments AC and BD. The positions of the pattern's marks on the image are expressed in pixels as (X_i, Y_i) where *i = A,B,C and D*.

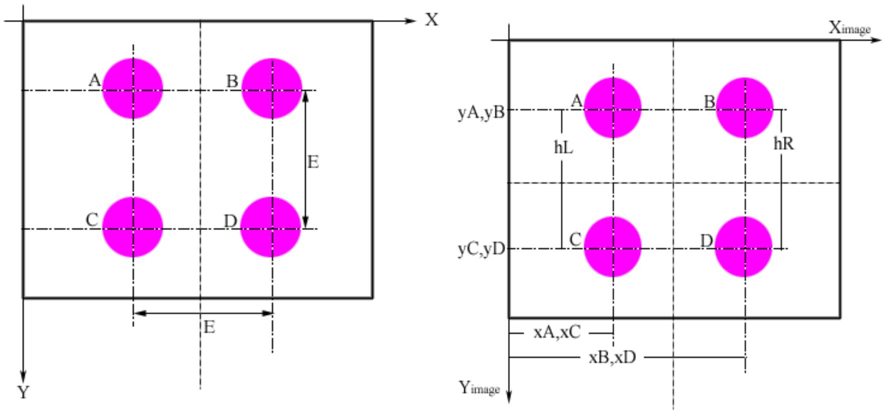


Figure 3.4. Detection Pattern (a) Leader Robot, (b) Pattern on the Followers Camera Image.

Figure 3.5 illustrates the horizontal projection of the vision system, showing the posture of the leader vehicle on a coordinate system ($X_{C,}$ Z_C) associated with the camera. '*f*' is the focal length of the camera being used. Points A, C and B, D represent segments AC and BD when viewed from the top.

NO !!!!!!!!

NO !!!!!!!

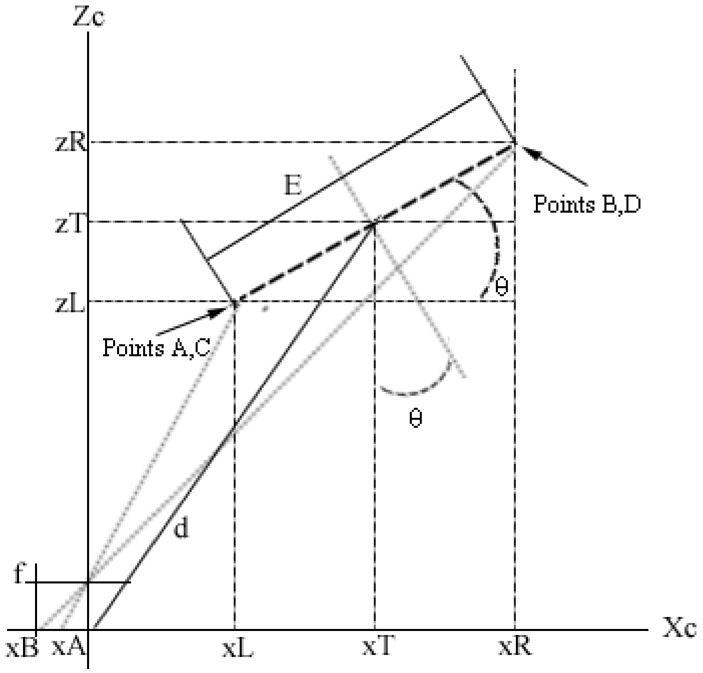


Figure 3.5. Horizontal Projection of the Visual System.

The projected pattern on the robot's camera image appears with a projection distortion as represented in Figure 3.6. The heights h_L and h_R now change as the leader changes orientation. From these image features, it is possible to compute the posture of the target vehicle (x_T, $z_{t,}$ θ).

The reverse perspective model is used to project a 3D point in the 2D camera image plane, resulting in the following parameters, which correspond to Figure 3.5.

$$x_L = \frac{(z_L - f)}{f} x_A, \qquad x_R = \frac{(z_R - f)}{f} x_B \tag{6}$$

$$z_L = f\left(\frac{E}{h_L}+1\right), \qquad\qquad z_L = f\left(\frac{E}{h_R}+1\right) \tag{7}$$

How to calculate E ???

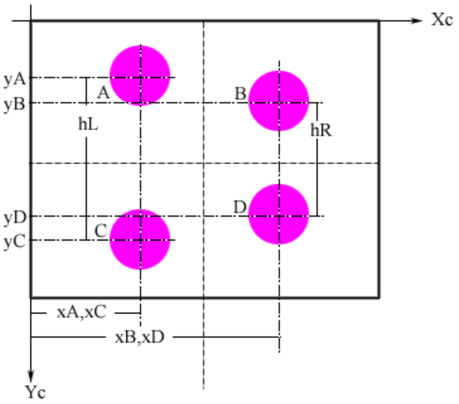
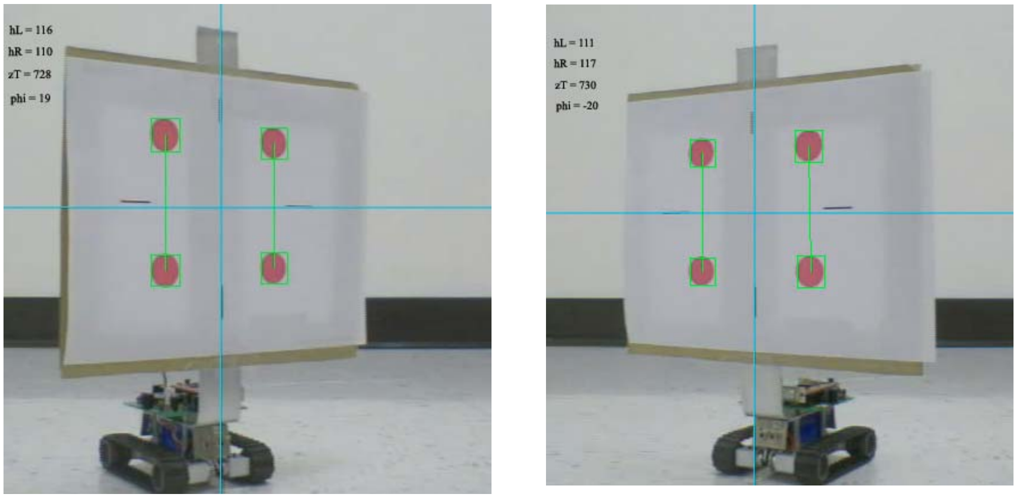


Figure 3.6. Projected Pattern of the Leader on the Followers Camera Image.

$$x_T = \frac{x_R + x_L}{2}, \qquad\qquad z_T = \frac{z_R + z_L}{2} \tag{8}$$

$$\theta = \tan^{-1}\left(\frac{z_R + z_L}{x_R + x_L}\right) \tag{9}$$

Figure 3.7 shows frames captured by the follower robot when the leader robot is straight in front of the follower at a distance of 730mm in different orientations. Figure 3.7a shows the leader oriented at a 20 degree angle to the left whereas Figure 3.7b shows the leader with the same orientation to the right. Heights h_L and h_R change in value as the pattern orientation is altered.



(a) (b)

Figure 3.7. Distortion in the Followers Image Plane when the Leader Changes Orientation, (a) Left and (b) Right.

3.3. BÉZIER TRAJECTORY GENERATION

3.3.1. The Bézier Trajectory Principle. A Bézier curve in its most common form is a simple cubic equation used for curve fitting. Originally developed by Pierre Bézier, who used it to design the Body of a Renault Car in the 1970s, it has only recently been used in robotics. Figure 3.8 illustrates a simple Bézier curve.

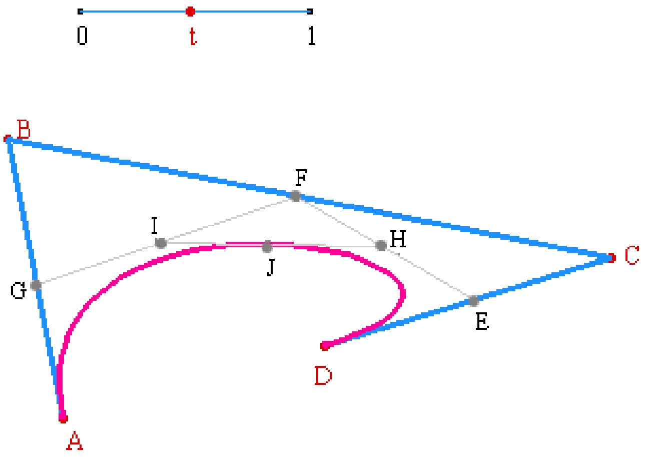
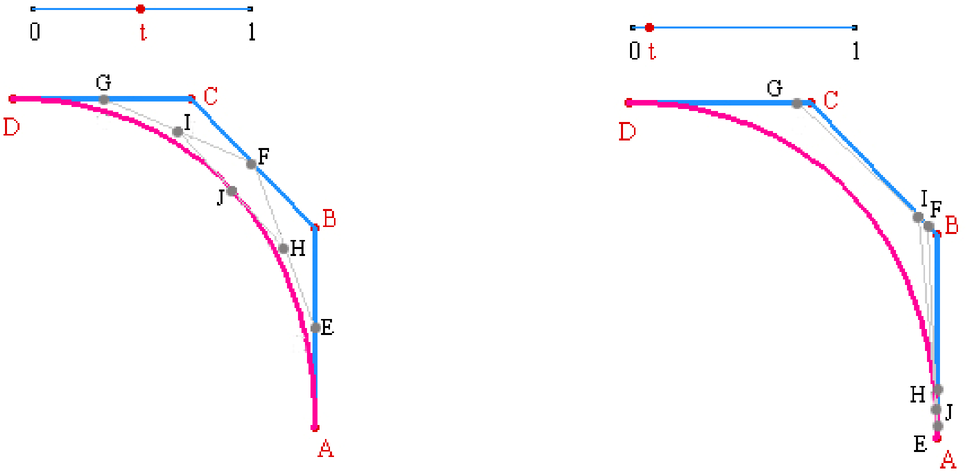


Figure 3.8. A Simple Bézier Curve.

A Bézier curve is a curve which is exactly determined by a set of control points. Each point of the curve is calculated from a parametric mathematical function which uses the coordinates of the control points as parameters.

Given the four points A, B, C, and D, and a value *t* between 0 and 1, the points E, F, and G are constructed a *t*-fraction of the way along the segments AB, BC, and CD, respectively; points H and I are then placed a *t*-fraction of the way along the segments EF and FG; and finally, J is constructed a *t*-fraction of the way along the segment HI. The locus generated by J as *t* goes from 0 to 1 is the generated curve. Points A and D are the *end points*, points B and C are the *control points*. The curve can be changed in shape by changing the distance between the segments AB and DC. In reference to the Leader-Follower model, point A represents the follower and D represents the leader robot. Point J on the curve varies as *t* varies from 0 to 1. As illustrated in Figure 3.9a J lies at the midpoint of the Bézier curve with a value of *t*=1/2 whereas in Figure 3.9b J moves down to point A as *t* approaches 0.



(a)

(b)

Figure 3.9. Point *J* on the Bézier Curve (a) $t = \frac{1}{2}$, (b) $t \to 0$.

Given N+1 control points P_k with k=0 to N, the Bézier parametric curve is described by $P(t)$ as follows:

$$P(t) = \sum_{k=0}^{N} P_k \frac{N!}{k!(N-k)!} t^k (1-t)^{N-k} \left.\vphantom{\frac{N!}{k!}}\right\} 0 \le t \le 1 \tag{10}$$

Bézier curves are parametric curves that, when applied independently to the x and y coordinates for a 2D curve, give:

$$P(t).x = \sum_{k=0}^{N} (P_k.x) \frac{N!}{k!(N-k)!} t^k (1-t)^{N-k} \left.\vphantom{\frac{N!}{k!}}\right\} 0 \le t \le 1 \tag{11}$$

$$P(t).y = \sum_{k=0}^{N} (P_k.y) \frac{N!}{k!(N-k)!} t^k (1-t)^{N-k} \left.\vphantom{\frac{N!}{k!}}\right\} 0 \le t \le 1$$

The equation for a Bézier curve is a polynomial of degree N (one less than the number of control points). As the number of control points increases, degree N rises, so it becomes expensive in terms of processing power to draw the curve. However, most curves can be drawn utilizing only four control points. The polynomial degree is then three (thus the name "Cubic Bézier" curve).

Given four control points- P_0, P_1, P_2, and P_3, the mathematical formula for a Cubic Bézier curve is as follows:

$$P(t) = \sum_{k=0}^{3} P_k \frac{3!}{k!(3-k)!} t^k (1-t)^{3-k} \left.\vphantom{\frac{3!}{k!}}\right\} 0 \le t \le 1 \tag{12}$$

$$P(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3$$

$$P(t) = t^3(-P_0 + 3P_1 - 3P_2 + 3P_3) + t^2(3P_0 - 6P_1 + 3P_2) + t(-3P_0 + 3P_1) + P_0$$

$$P(t) = t^3(P_3 + 3(P_1 - P_2) - P_0) + 3t^2(P_0 - 2P_1 + P_2) + 3t(P_1 - P_0) + P_0$$

Let,

$$P(t) = at^3 + bt^2 + ct + d \;\} 0 \le t \le 1 \tag{13}$$

where,

$$
\begin{aligned}
c &= 3(P_1 - P_0) \\
b &= 3(P_2 - P_0) - c \\
a &= P_3 - P_0 - c - b \\
d &= P_0
\end{aligned}
\tag{14}
$$

3.3.2. Bézier Curve Length. The length of the Bézier curve *s*, which is an integral is**:**

$$\partial s = \sqrt{\partial x^2 + \partial y^2}$$

$$s = \int \partial s = \int \sqrt{\partial x^2 + \partial y^2}$$

$$s = \int_{t=0}^{1} \sqrt{\left(a_1 + 2a_2 t + 3a_3 t^2\right) + \left(b_1 + 2b_2 t + 3b_3 t^2\right)}dt \tag{15}$$

The length of s is calculated by integration from t=0 to t=1. a_0, a_1, a_2, a_3 and b_0, b_1, b_2, b_3 for *x* and *y* can be calculated from Eqn. (14).

3.3.3. Bézier Trajectory Generation. The follower robots use a Bézier curve to generate a trajectory to the leader robot. This curve is the weighted sum of four control points (P0, P1, P2, and P3), as shown in Fig. 2.12. D is the distance between points P_1-P_0 and P_3-P_2.

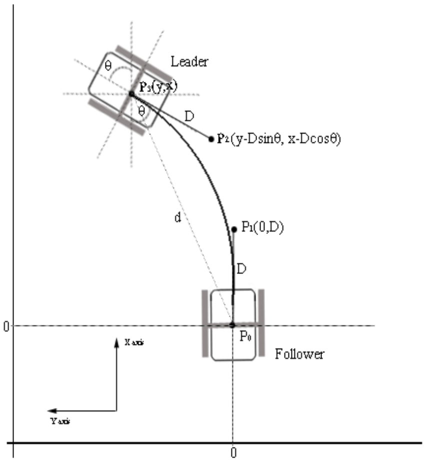


Figure 3.10. Bézier Curve Between the Leader and Follower Robots.

Endpoints P0 and P3 define the positions of the follower and leader, respectively. Control points P1 and P2 are determined by the orientation of the robots. The four points are defined in the follower robot's reference frame:

$$P_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad P_1 = \begin{pmatrix} D \\ 0 \end{pmatrix}, \qquad P_2 = \begin{pmatrix} x - D\cos\theta \\ y - D\sin\theta \end{pmatrix}, \quad P_3 = \begin{pmatrix} x \\ y \end{pmatrix} \tag{16}$$

The cubic Bézier curve from Eqn. (13) given by:

$$P(t) = at^3 + bt^2 + ct + d, \forall t \in [0,1] \tag{17}$$

where the vectors a, b, c are defined as in Eqn. (14).

Values of x, y and θ are obtained from the vision system. The theory of Bézier curves states two properties regarding the endpoints:

- The curve passes through the endpoints themselves, and
- The curve is tangent to the vectors P_1-P_0 and P_3-P_2 at the endpoints

The control points can be arbitrarily chosen anywhere in the space between the two robots. The shape of the curve can be easily deformed by modifying the distance D between points P_1-P_0 and P_3-P_2. However, placing the points in different locations results in different curves or trajectories, as illustrated in Figure 3.11.

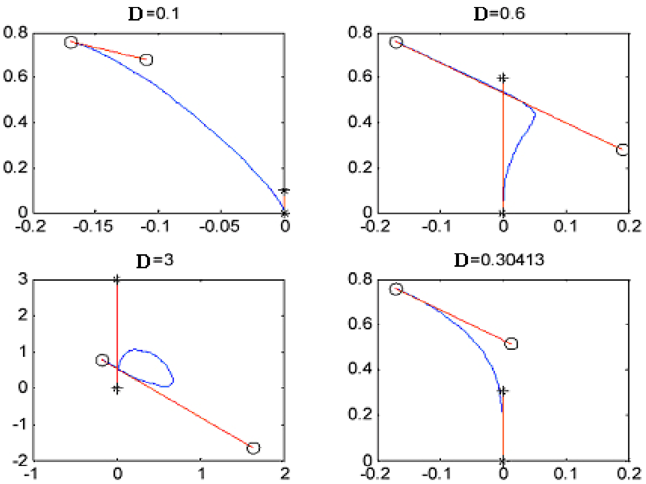


Figure 3.11. Bézier Cubic Curves with Different Values of Scale Factor D.

The follower robots need to keep a set distance away from the leader robot. This is done by selecting an appropriate value of D. This D is proportional to the distance between the robots, and the same D is used for both pairs of endpoints and control points such that

$$D = \left\| P_1 - P_0 \right\| = \left\| P_3 - P_2 \right\| \tag{18}$$

By experimental experience as shown by Chiem and Cervera, (2004), and from Figure 3.11.d, the value of D which makes the Bézier curve exactly the same as a 90-degree arc is:

$$\begin{aligned} D &= \left(\frac{2}{3}\right)\sqrt{2}\left(\sqrt{2}-1\right)\left\| P_3 - P_0 \right\| \\ &= \left(\frac{2}{3}\right)\sqrt{2}\left(\sqrt{2}-1\right)\sqrt{x^2+y^2} \\ &= 0.39 * \sqrt{x^2+y^2} \end{aligned} \tag{19}$$

Although the leader robot is set on a predefined trajectory with a constant velocity *v*, the velocity may change by small amounts. The follower robot has to take into account these changes in velocities. This is done by keeping track of the length of the Bézier curve, *s*. The controller is proportional in nature, increasing the followers speed if it is too far from the leader and decreasing speed if it is too near. The change in linear velocity Δv is given by:

$$\Delta v \propto \left(s - s_0\right) \tag{20}$$

where s_0 is the desired length of the curve.

Also, the follower robots need to vary their angular velocity, ω, which is computed as follows, to keep up with the leader robot. This angular velocity must correspond to the curvature of the Bézier trajectory at P_0:

$$\omega = \left(\frac{v}{R}\right) = v\kappa \tag{21}$$

where R and κ are the radius and curvature of the trajectory, respectively. The curvature of any parametric curve is:

$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \tag{22}$$

From Figure 3.10, since the follower robot is at the origin (0,0) and it is only necessary to compute the curvature at t=0,

$$\begin{aligned} \dot{x} &= v \\ \dot{y} &= 0 \end{aligned} \tag{23}$$

so that the curvature at this point is:

$$\kappa = \frac{\ddot{y}}{\dot{x}^2} = \frac{2(y - D\sin\theta)}{3d^2} \tag{24}$$

where D is obtained from Eqn. (19) and the desired angular velocity is computed from Eqn. (21). Individual wheel velocities are then calculated from Eqn. (4) and Eqn. (5).

The camera runs at a frame rate of 25 frames per second. When a frame is captured, the new position of the leader and the length of the Bézier curve is estimated. Using these parameters, the Bézier points are calculated and the new linear velocity v and angular velocity ω are computed.

3.4. MULTIPLE ROBOT FORMATION

In this research, the follower robot maintains a position relative to a leader robot. Using the same leader-follower philosophy, multiple robots can be made to follow the same leader with the help of *virtual points*. These points are the position of the leader displaced a certain distance away. The only drawback of such a formation is that every follower robot has to have the leader in view.

Virtual destinations are assigned to each follower to maintain a geometric formation. These points are the position of the leader robot moved perpendicular to and a certain distance away from the leaders y-axis. Then, cubic Bézier trajectories, as described in Section 3.3.2, are defined between the follower and these virtual destinations to allow the robots to follow the leader. The trajectory is updated in real-time because the virtual destination varies as the leader robot moves. Figure 3.12 illustrates two follower robots tracking virtual points V1 and V2 displaced by certain amount in different directions from the leader.

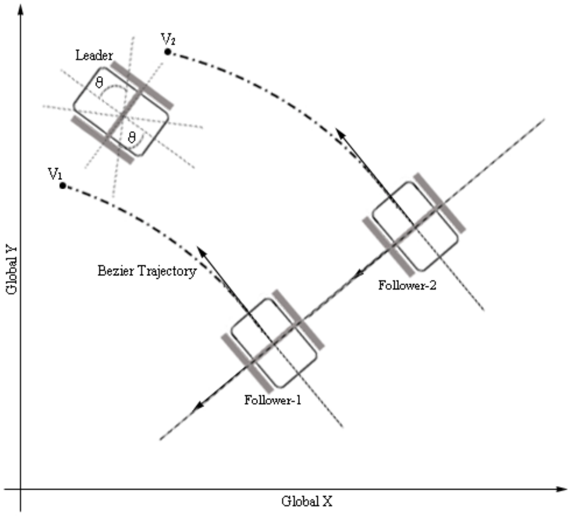


Figure 3.12. Multiple Robot Formation.

4. EXPERIMENTAL RESULTS

4.1. EXPERIMENTAL SETUP

The setup consisted of an overhead camera covering a 7.0 x 5.3 sq. feet area. The camera was a low-cost CMOS vision sensor connected via USB to a computer mounted on the laboratory ceiling. The leader and the follower robots had colored markers mounted on their top. These served as robot identification for the program analyzing the area. The area was covered with white boards to ensure minimum disturbances from other colors and easy detection for the image processing program. Colors for the two robots were chosen so that they stand out against the background. Figure 4.1 shows the setup with the leader colored red and the follower colored blue.

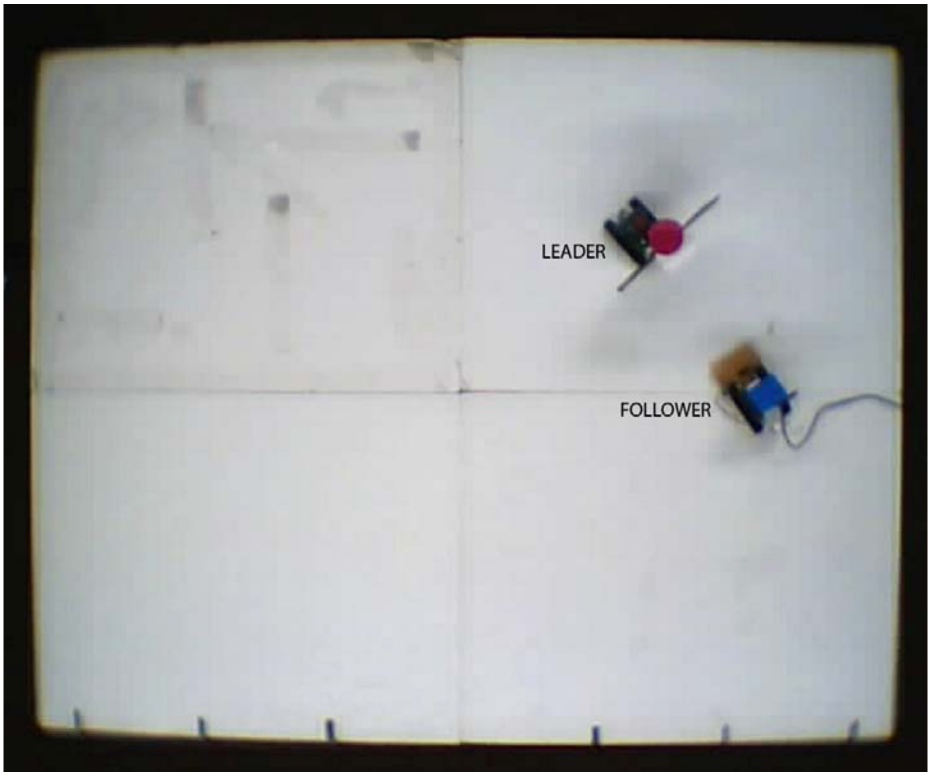


Figure 4.1. Overhead View of the Leader/Follower Robots.

OpenCV was used to capture and filter images and detect the markers. The colors were detected using the HSV threshold algorithm as described in Section 2. For a particular run the camera captured a video involving the leader and follower robots at a 640 x 480 resolution at 25 frames per second. The video was then processed through OpenCV for analysis. Verification of the follower robots maintaining a desired formation with the leader was done by making sure that the robots maintain a constant predefined Bezier length between them and visually verifying their trajectories.

4.2. FORMATION MAINTENANCE RESULTS

This experiment demonstrates how the robots attain formation after a certain period of time. The leader and follower robots are separated at a distance of about 1000mm at startup. The required distance between the robots is 500mm. This distance is maintained by calculating the length of the Bézier curve. Figure 4.2 Illustrates results for three independent runs. It is seen that the separation between the two robots converges to the desired value (500mm) after a certain period measured in frames.

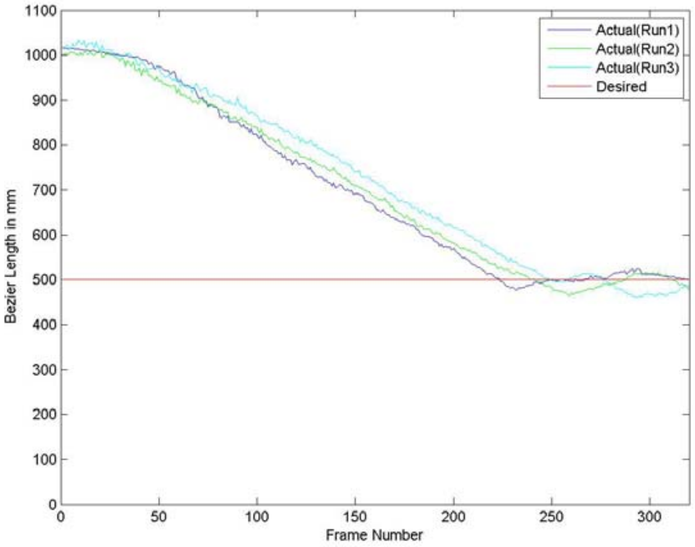


Figure 4.2. Evolution of Bézier Length Between the Follower and Leader.

Figure 4.3 shows the follower keeping up right behind the leader. The red trail represents the leader while the blue is the follower. The follower started its run at an angle to the leader. After a certain period, the follower maintained its position right behind the leader at the desired distance.

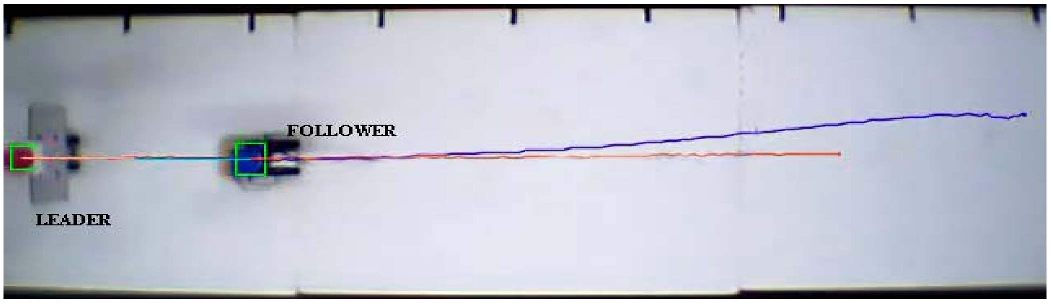


Figure 4.3. A Follower Tracing a Straight Line Path Defined by the Leader.

Figure 4.4 shows the two robots maintain a separation of 500mm between them. The oscillations from the desired value are due to a purely proportional controller.

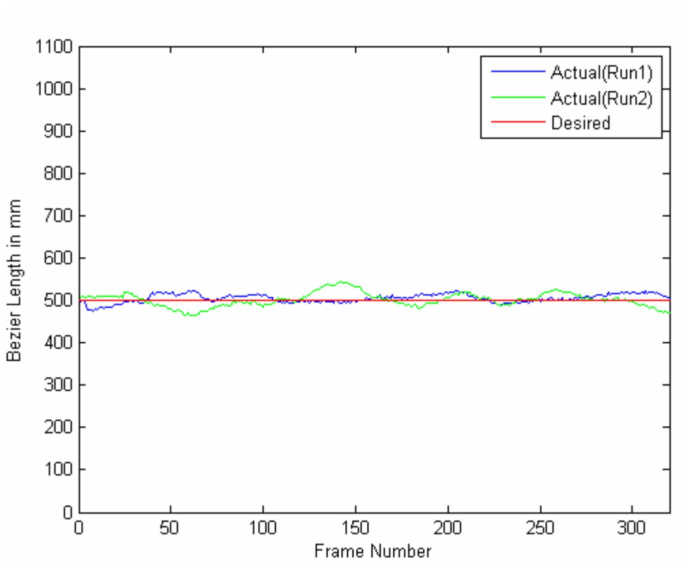


Figure 4.4 Separation vs Frame Number for Straight Line Formation.

In another experiment, a follower robot followed the leader in a circular arc motion as in Figure 4.5. During arc motions, although the follower got behind the leader, the vision system takes some amount of time before it could get the follower right behind the leader. Figure 4.6 shows the how the follower tried to maintain a distance of 500mm behind the leader. During curve motions there is a larger deviation from the desired separation as compared to linear motions. This discrepancy is due to a poor velocity controller model.

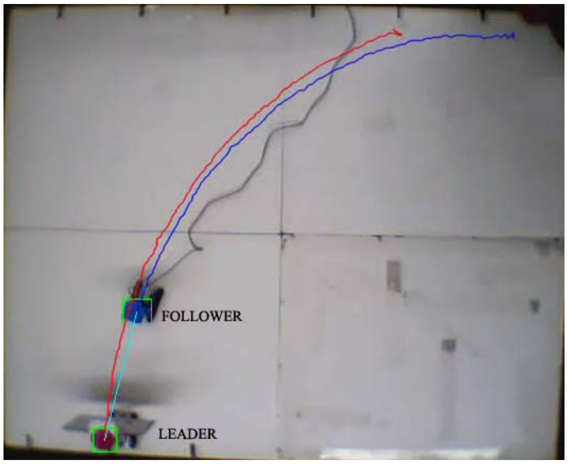


Figure 4.5. Follower Tracing a Curve Generated by the Leader.

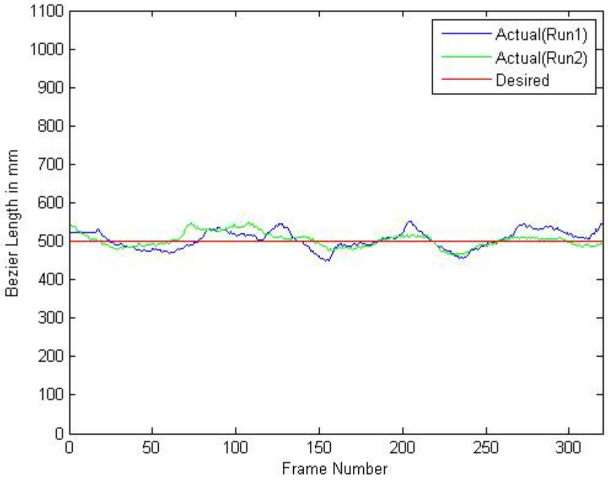


Figure 4.6. Separation vs Frame Number with the Follower Tracing a Curve Generated by the Leader.

However, on curves with a larger radius, these deviations seem to reduce. Figure 4.7 Illustrates the follower robot tracking a virtual point to verify multiple robot formations. Figure 4.8 shows the separation between the two robots for two independent runs. Figure 4.9 is the same experiment, but along a curve. As discussed before there is a much more deviation in the separation values when the robots move along a curve. The separation between the two robots is illustrated in Figure 4.10.

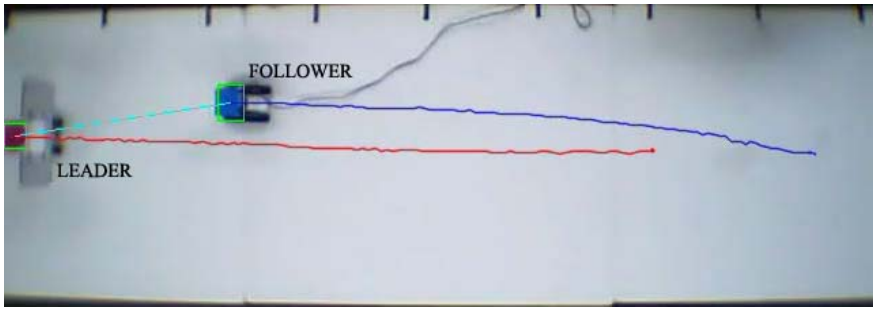


Figure 4.7. Overhead View of the Follower Robot Tracking a Virtual Point.

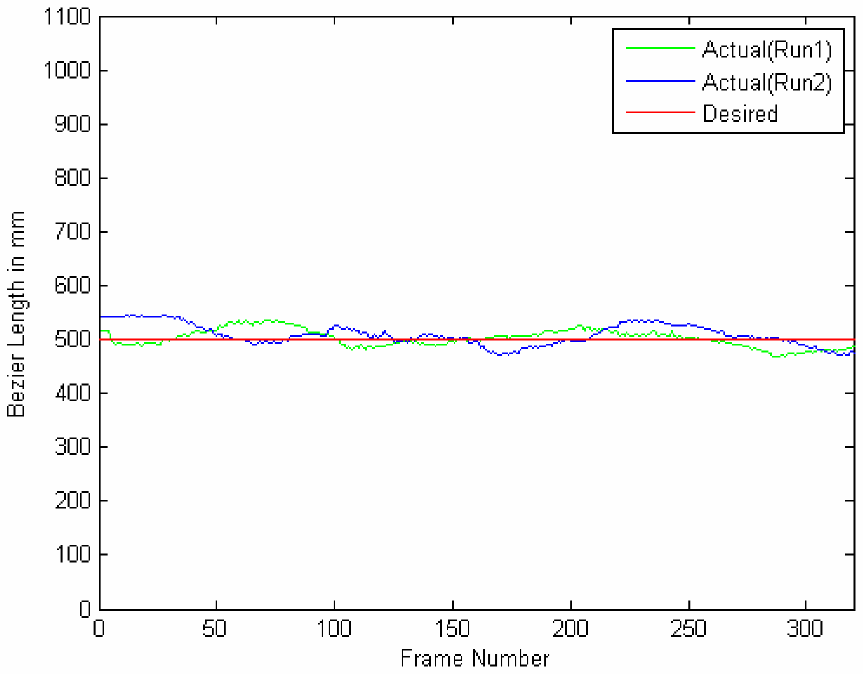


Figure 4.8. Separation vs Frame Number with Virtual Point Formation.

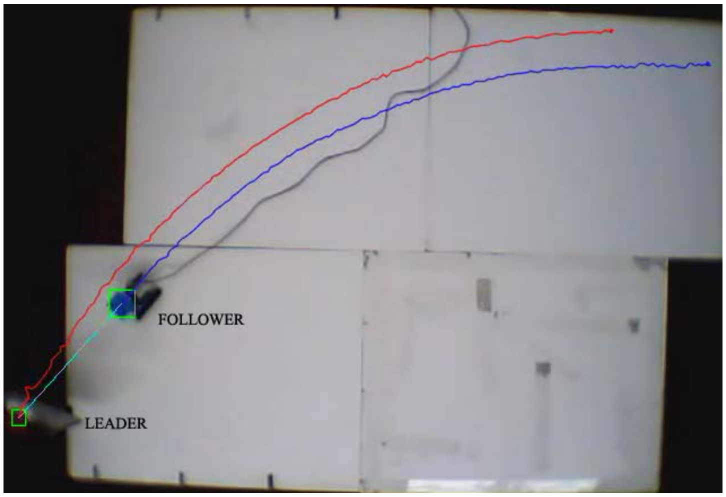


Figure 4.9. Overhead View of the Follower Robot Tracking a Virtual Point Along a Curve.

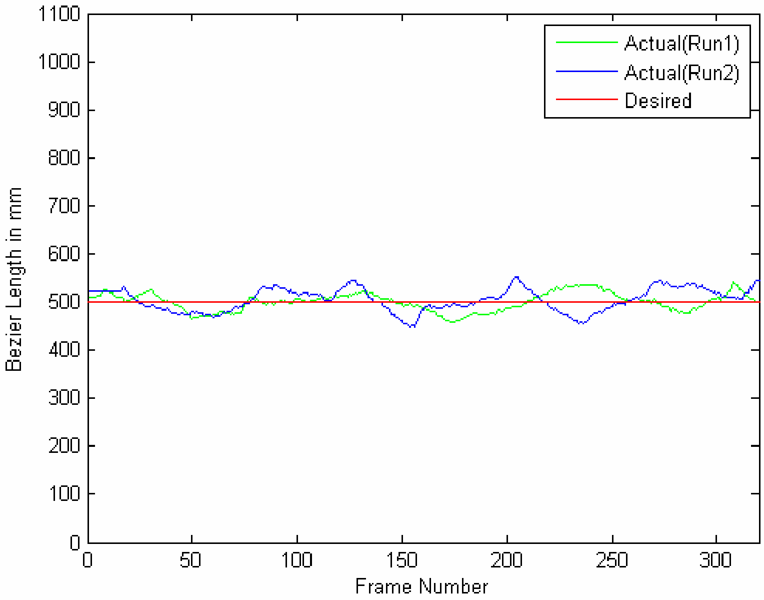


Figure 4.10. Separation vs Frame Number with the Follower Robot Tracking a Virtual Point along a Curve.

5. SUMMARY AND FUTURE WORK

5.1. SUMMARY

This thesis presents a framework for vision-based control for multi-vehicle coordination using the leader-follower approach. Unlike the leader, the followers are designed to be equipped with low cost sensors, primarily vision sensors. The leader is a more sophisticated vehicle with potential navigation, ranging and obstacle avoidance capabilities. For current work, the leader is programmed with a predefined trajectory. This framework consists of a vision based model and a formation control algorithm for the follower robots.

The follower requires only information about the position and orientation of the leader to follow. The vision system uses markers for identification of the leader robot. It can estimate the relative pose (distance and angle) between two robots from single images. The vision system is divided in two main components, image processing and pose estimation. A HSV thresholding approach is used for processing the captured images for markers detection. Using the model of perspective geometry, the position and orientation of the leader robot is deduced.

The formation control algorithm is based on generating a Bézier trajectory between the leader and follower. The Bézier trajectory is defined according to the relative configuration between the leader and the follower and is updated in real-time as the leader navigates. This type of control method is simple and does not require complicated computation in the followers. Experimental results show that the algorithm performs well with both straight line and virtual point following. While negotiating curves, the follower robots find it a little difficult to keep up with the leader. This can be solved by developing a better higher order velocity controller for the followers.

5.2. FUTURE WORK

The vision based pose estimation is precise enough so that the followers do not need accurate positioning systems. Only the leader robot carries such a system to command the entire formation. The proposed formation method is limited in that the followers need to maintain line-of-sight contact with the leader. If for any reason a follower looses sight of the leader, a “search for leader” mechanism is run. Once visibility is recovered, the follower can catch up with the leader. A better solution would be to use a wide angle camera or an omni-directional camera. An omni-directional vision sensor consists of a conical lens and a camera which give a 360 view of the environment. The follower would not need to search for the leader because it would always be in view.

The robots in this research try to keep the leader in view by modifying their angular and linear velocities. However, since this work stresses the need for low cost and the smallest number of sensors, velocity feedback could be calculated from the vision sensor using optical flow algorithms.

Formation switching between the robots and obstacle avoidance is also important. There are times when the formation may need to form a single line. In such a scenario, each follower robot becomes a leader to the one behind it. Future work could focus on how to produce a centralized control strategy for teams of follower robots that switch between different formations in reaction to environmental changes or to avoid obstacles. However, such formations may increase communication bandwidth between the robots. Using omni-directional cameras as described above could also provide flexibility in achieving different formations.

APPENDIX

COEFFICIENTS OF A CUBIC BEZIER CURVE

A Cubic Bézier curve, in a 2-D plane, in parametric terms is :

$$x = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$y = b_0 + b_1 t + b_2 t^2 + b_3 t^3$$

The four control points P$_0$, P$_1$, P$_2$, and P$_3$ can be expressed as:

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) .$$

$x = x_0$ and $y = y_0$ when $t = 0$

Hence, $x_0 = a_0, y_0 = b_0$ (A.1)

$x = x_3$ and $y = y_3$ when $t = 1$

Hence, $x_3 = a_0 + a_1 + a_2 + a_3, y_3 = b_0 + b_1 + b_2 + b_3$ (A.2)

The other two control points P$_1$and P$_2$ require the derivative of the curve.

As in Figure 3.9, derivative along tangent AB:

$$\frac{dx}{dt} = a_1 + 2a_2 t + 3a_3 t^2, \frac{dy}{dt} = b_1 + 2b_2 t + 3b_3 t^2$$

$\frac{dx}{dt} = 3(x_1 - x_0)$ and $\frac{dy}{dt} = 3(y_1 - y_0)$ when t=0

Hence $3(x_1 - x_0) = a_1, 3(y_1 - y_0) = b_1$ (A.3)

Derivative along tangent DC:

$\frac{dx}{dt} = 3(x_3 - x_2)$ and $\frac{dy}{dt} = 3(y_3 - y_2)$ when t=1

Hence $3(x_3 - x_2) = a_1 + 2a_2 + 3a_3$ (A.4)

$$3(y_3 - y_2) = b_1 + 2b_2 + 3b_3$$

Solving for $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$ from Equations (A.1), (A.2), (A.3), (A.4)

$$a_0 = x_0, b_0 = y_0, a_1 = 3(x_1 - x_0), b_1 = 3(y_1 - y_0)$$

$$a_2 = 3(x_0 - 2x_1 + x_2), b_2 = 3(y_0 - 2y_1 + y_2)$$

$$a_3 = x_3 - x_0 + 3(x_1 - x_0), b_3 = y_3 - y_0 + 3(y_1 - y_0)$$

BIBLIOGRAPHY

Agrawal M., and Konolige K. (2006), "Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS," *The 18th International Conference on Pattern Recognition*, vol. 3, pp. 1063-1068.

Akella S., and Hutchinson S. (2002), "Coordinating the Motions of Multiple Robots with specified Trajectories", *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 624 – 631.

Armesto L., Tornero J. (2006), "Robust and Efficient Mobile Robot Self-Localization using Laser Scanner and Geometrical Maps," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 3080-3085.

Balch T., and Arkin R. (1994), "Motor Schema-based Formations Control for Multiagent Robot Teams," *Proceedings of the First International Conference on Multiagent Systems*, pp. 10-16.

Balch T., and Arkin R. (1998), "Behavior-based Formation Control for Multi-Robot Teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939.

Belta, C., and Kumar, V. (2001), "Motion Generation for Formations of Robots: A Geometric Approach," *IEEE International Conference on Robotics and Automation*, Korea, vol 2, pp. 1245- 1250.

Brooks R. (1986), "A Robust Layered Control System for A Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, No. 1, pp. 14-23.

Bruce J., Balch T., and Veloso M. (2000), "Fast and Inexpensive Color Image Segmentation for Interactive Robots," *Proceedings of International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2061 – 2066.

Carelli R., Soria C., Nasisi O., and Freire E. (2002), "Stable AGV Corridor Navigation with Fused Vision-Based Control Signals," *Conference of the IEEE Industrial Electronics Society*, Sevilla, Spain, vol.3, pp. 2433- 2438.

Chen J., Dawson M., Dixon W., and Behal A. (2005), "Adaptive Homography Based Visual Servo Tracking for Fixed and Camera-in-Hand Configurations," *IEEETransactions on Control Systems Technology*, vol 13, pp. 814- 825.

Chiem S., and Cervera E. (2004), "Vision-Based Robot Formations with Bezier Trajectories," *Proceedings of Intelligent and Autonomous Robots*, pp. 191–198.

Das A., Fierro R., Kumar V., Ostrowski J., Spletzer J., and Taylor C. (2002), “A Vision-based Formation Control Framework,” *IEEE Transactions on Robotics and Automation*, Vol. 18, pp. 813- 825.

Das V., Fierro R., Kumar V., Southall B., Spletzer J.,Taylor C. (2001), “Real-time Vision-Based Control of a Nonholonomic Mobile Robot,” *International Conference on Robotics and Automation*, Seoul, Korea, pp. 1714–1719.

Desai J., Ostrowski J., and Kumar V. (1998), “Controlling Formations of Multiple Mobile Robots,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, vol. 4, pp. 2864-2869.

Fredslund J., and Mataric J. (2002), “A General Algorithm for Robot Formations using Local Sensing and Minimal Communication,” *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp. 837-846.

Fredslund J., and Mataric J. (2001), “Robot Formations Using Only Local Sensing and Control,” *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA-01),* Banff, Alberta, Canada, pp. 308-313.

Fiala M. (2004), “Vision Guided Control of Multiple Robots,” *Proceedings of the First Canadian Conference on Computer and Robot Vision*, Washington DC, USA, pages 241-246.

Forsyth, and Ponce, “Perspective Projection,” in *Computer Vision : A Modern Approach.* Prentice Hall, 2002, pp. 4-5.

Han Y., and Hahn H. (2005), “Visual Tracking of a moving Target using Active Contour based SSD Algorithm,” *Robotics and Autonomous Systems.*

Huang J., Farritor S., Qadi A., and Goddard S. (2005), “Localization and Follow-the-Leader Control of Heterogeneous Groups of Mobile Robots,” *ASME International Mechanical Engineering Congress and Exposition.*

Jiangyang H. (2007), “Localization and Follow-the-Leader Control of a Heterogeneous Group of Mobile Robots”, Ph.D. dissertation, University of Nebraska.

Larsen T., Bak M., Andersen N., and Ravn O. (1998), “Location Estimation for Autonomously Guided Vehicle using an Augmented Kalman Filter to Autocalibrate the Odometry,” *Fusion 98, First International Conference on Information Fusion*, Las Vegas, USA.

L. E. Parker, B. Kannan, F. Tang, and M. Bailey (2004), “Tightly-Coupled Navigation Assistance in Heterogeneous Multi-Robot Teams,” in Proc. IEEE Int. Conference Intelligent Robots Syst., Sendai, Japan, vol. 1, pp. 1016–1022.

Lowe D. (2004), "Distinctive Image Features from Scale-Invariant Key points," *International Journal of Computer Vision*, vol. 60, pp. 90-110.

Martinelli A., Tomatis N., Tapus A., and Siegwart R. (2003), "Simultaneous localization and Odometry Calibration for Mobile Robot," *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1499-1504.

Michaud F., Letourneau D., Guilbert M., and Valin J. (2002), "Dynamic Robot Formations Using Directional Visual Perception," in Proceedings of *IEEE International Conference of Intelligent Robots and Systems*, EPFL, Lausanne, Switzerland, pp. 2740-2745.

Nguyen D. (2005), "Robust Tracking of Input Props with Webcams," M.S. thesis, University of Amsterdam.

Rodrigo R., and Samarabandu J. (2005), "Monocular Vision for Robot Navigation," Proceedings of the *IEEE International Conference on Mechatronics and Automation,* pp. 707-712.

Shao J., Xe G., Yu J., and Wang L., "Leader-following Formation Control of Multiple Mobile Robots," in Proceedings of the *IEEE International Symposium on Intelligent Control*, pp. 808-813.

Suzuki S., Abe K. (1985), "Topological Structural Analysis of Digital Binary Image by Border Following," *CVGIP*, vol. 30, pp. 32-46.

Tan K. and Lewis M. (1997), "Virtual Structures for High-Precision Cooperative Mobile Robot Control," *Auton. Robots*, vol. 4, pp. 387-403.

Vassallo F. (1998), "Visual navigation: Combining Visual Servoing and Appearance Based Methods," *Int. Symp. on Intelligent Robotic Systems,* Edinburgh, Scotland, pp. 137.

Vidal R., Shakernia O., and S. Sastry (2003), "Formation Control of Nonholonomic Mobile Robots with Omni directional Visual Servoing and Motion Segmentation," *IEEE International Conference on Robotics and Automation*, vol. 1, Taipei, Taiwan, R.O.C., pp. 584–589.

VITA

Gerard Sequeira was born on September 23, 1982 in Mumbai, India. In June 2004, he received his Bachelor’s degree in Instrumentation Engineering from the University of Mumbai, India. He joined the Master of Science program in Electrical Engineering at the University of Missouri—Rolla in Fall 2005. While pursuing his graduate degree, he was supported by the Department of Electrical Engineering as a Graduate Research Assistant. He received his Masters degree in Electrical Engineering from the University of Missouri—Rolla in December 2007. His research interests include embedded systems, controls, and robotics.