Cooperative Robotics: Eye in the Sky
ELEC4000 – Fall 2006 Senior Design
Dr. Thaddeus Roppel
12 December 2006

The Blimp Team
Sydney Alamy
Clay Askew
Nick Cotton
Jim Cunningham
Blake Lane
John Macker
Chris Wilson

Table of Contents . . . . page 2

1 Abstract

This report describes a senior design project in the field of cooperative robotics. Two autonomous vehicles – one in the air and one on the ground – equipped with various types of sensors are to locate and acquire a predetermined target in a specified coordinate system.

2 Introduction

2.1 Project Overview

This project consists of an autonomous aerial vehicle and an autonomous ground vehicle linked by a laptop computer. The aerial vehicle – a blimp – is equipped with a wireless video camera, which points directly down at a grid-based coordinate system at all times. The video stream is transmitted back to the laptop computer where it is digitized and processed in order to determine the current location of the blimp with regard to the coordinate system as well as the orientation of the blimp. In addition to determining the location and orientation of the blimp, the image processing software simultaneously searches the video stream for a target – in this case, an orange cone. Once the target has been positively identified by the image processing software on the laptop, the coordinates are then transmitted to the ground vehicle. The ground vehicle then uses an array of sensors, including an infrared line counter, a digital compass, and an optical mouse, to navigate the coordinate system and locate the orange cone while using sonar to detect obstacles and modify its path accordingly.

2.2 Team Members

The Blimp Team consists of seven students in the Electrical and Computer Engineering Department of Auburn University's Samuel Ginn College of Engineering. These members and their roles in this project are detailed below.

Sydney Alamy (ECE) was responsible for the PC end of the blimp control interface. In addition to the blimp control interface, he was responsible for writing the autonomous control algorithm that was to be used to control the blimp during its flight pattern while searching for the target.

Clay Askew (WIRE) was responsible for the physical design and construction of the ground vehicle. He was also responsible for testing various components, including the digital compass and Bluetooth serial communications link, and for the overall integration of the various components.

Nick Cotton (EE) served as team leader and was involved in many aspects of the project, including the design of the blimp control interface, the overall ground vehicle control logic, the debugging of various sensors, and the debugging LCD controller. Nick was also responsible for establishing a data connection between the various PIC microcontrollers on the ground vehicle using the I^2C serial communications protocol.

Jim Cunningham (WIRE / Physics) was responsible for the design of the controller for the optical mouse, including the analysis and recreation of its PS/2 communications protocol. He was also responsible for using the information from the optical mouse and the digital compass to establish and maintain the ground vehicle's absolute location. Jim also contributed to the design of the overall ground vehicle control logic.

Blake Lane (WIRE) was responsible for the design and construction of the blimp's payload gondola. Blake was also responsible for the maintenance of the grid coordinate system.

John Macker (EE) was responsible for the design of the controller for the sonar modules used for obstacle detection and avoidance. He was also responsible for the infrared line counter controller, which was to be used in conjunction with the optical mouse for keeping track of the ground vehicle's location within the coordinate system.

Chris Wilson (WIRE) was responsible for all of the image processing done on the video blimp's live video stream. The image processing software determines the blimp's location and orientation while also searching for the target orange cone. Chris also designed a GUI that reads serial debug data from the ground vehicle and displays it in human-readable format and allows the operator to send control commands directly to the ground vehicle.

3 Usage

In order to use the "eye in the sky" senior design project, the following items are necessary.

1. Blimp and wireless controller
2. Ground vehicle
3. Laptop computer running Windows XP
4. Video camera and wireless transmitter
5. USB TV tuner
6. Blimp control interface
7. Grid-based coordinate system

After installing all software and drivers, the following steps are necessary to operate the "eye in the sky" project.

1. Connect the USB TV tuner to the laptop computer.
2. Connect the video camera's wireless receiver output to the input of the USB TV tuner.
3. Connect the the blimp control interface to the laptop computer's serial port and the blimp's wireless controller.
4. Place the blimp directly above the grid and note its starting coordinates.
5. Place the ground vehicle at location (1,1) in the grid coordinate system.
6. Place the target cone somewhere on the grid.
7. Run the image processing software and enter the blimp's starting coordinates into specified text boxes in the GUI, shown below in Figure 1.
8. Apply power to the blimp, wireless controller, and ground vehicle.
9. Initialize the image processing software.

10. The laptop computer should then begin sending commands to the blimp in response to the data provided by the image processing software. The blimp will eventually locate the target and send the coordinates to the ground vehicle, which will then also locate the target.

Select Device: myAvi

☑ Process Angles
☑ Process Grid
☑ Process Target

Angle Offset 359

	Row	Column
Blimp Start Grid	1	1
Blimp Current Grid	1	1
Target Grid		
Transmitted Target Grid		

Figure 1 – The image processing GUI

4 Design

The project is comprised of two main systems, the blimp and the ground vehicle, each made up of various subsystems.

4.1 Blimp System

The blimp's overall purpose is to fly an autonomous search pattern over the grid and transmit video back to the laptop so that the video stream can be processed in order to locate the target within the coordinate system.

The blimp system includes the blimp itself, the wireless controller, the video camera and USB TV tuner, and the image processing software on the laptop. The video camera on the blimp sends a video signal to the USB TV tuner, which digitizes and converts the video data into a format the image processing software can use. The video stream is processed and, based on the results of the image processing, commands are sent out through the blimp control interface. These commands are converted to a standard Futaba control signal and sent to the blimp's wireless controller, where the Futaba signal is converted into the RF signal that the blimp receives and interprets to control its motors.

4.1.1 Image Processing

One of the main obstacles encountered during the design process was determining a way to establish the locations of the vehicles. The project was to be done indoors due to the nature of the vehicles, so GPS was not an option.

After considering several options, it was decided that the blimp would rely on the video feed and image processing for location tracking. The image processing software takes the streaming video from the wireless video camera, extracts the grid lines from this feed, and uses this grid system to determine the location and heading of the blimp relative to its starting position. Additionally, the image processing concurrently searches the video stream for the target orange cone using color matching. Once the targets center of mass is found, the position of the target relative to the blimp's current position is determined, and the absolute position of the target to the accuracy of a specific grid square is output. A block diagram illustrating the overall high level image processing and recognition algorithm is shown below in Figure 2.

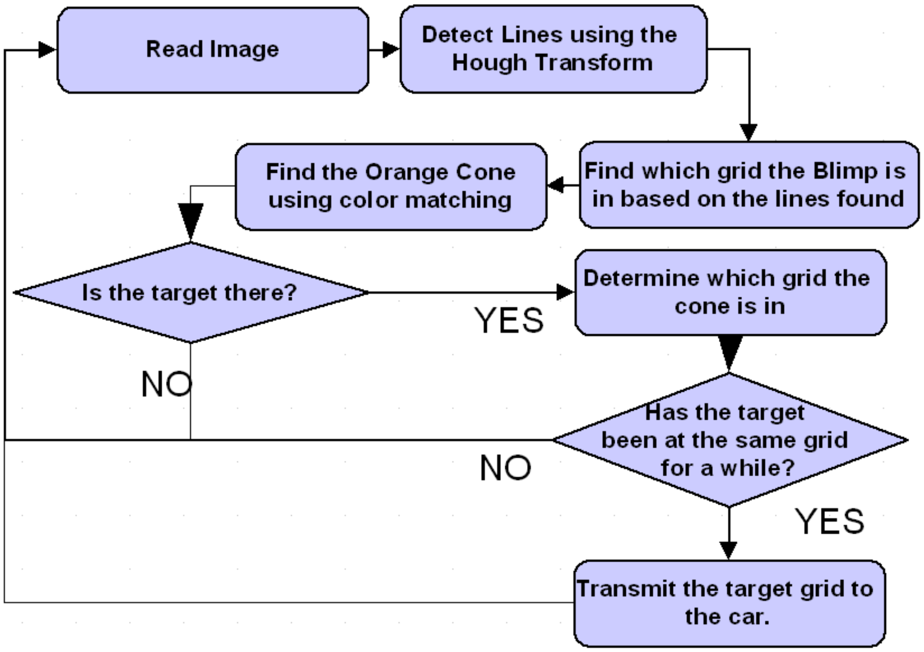

Figure 2 – Block diagram of the image processing algorithm

The extraction of the grid lines is performed by having a priori knowledge that the grid lines are black, thus liberal use of thresholding allows the algorithm to reduce the information that the software must process through the Hough Transform, which is used to determine the lines on the image. Proper thresholding on the transform image also allows further reduction of information to allow processing speeds to increase from 0.5Hz to a maximum of 3.3Hz including simultaneous target processing. Local maxima taken from the transform image are inverted back into the real space to reveal the lines in the captured image. Tracking these lines provides the grid knowledge, and absolute position can be established assuming the starting grid is specified by the operator.

Target detection is performed by changing the RGB color image to a HSV colorspace and then performing thresholding on each color channel to attain the desired range of values for the target. These ranges were previously determined by extensive testing. Reducing the range allows for the reduction of false positives. Computing the center of mass on any detected pixels allows for the determination of which grid the target is in relative to the blimp's current location. Additional filtering is preformed by requiring the same grid coordinate to be reported continually for 30 frames of data, which corresponds to approximately 10 seconds, before the transmission to the ground vehicle takes place.

4.1.2 Blimp Control

In order for the blimp to be completely autonomous, the laptop computer would be required to control the blimp. After originally considering a mechanical option that would physically manipulate the controls of the blimp's standard wireless controller, the manufacturer of the blimp provided a special version of the wireless controller with an input port that takes a standard Futaba control signal which can be converted by the wireless controller into the RF signal that would be sent directly to the blimp's motor.

To control the blimp with the laptop, a communications link must first be chosen. The standard serial communications port was chosen for this task due to the relatively low amount of necessary bandwidth. The standard control signal can have up to eight channels, so the blimp control software needed to be able to format the data in such a way that data for each of the eight channels could be distinguished from one another. If a single byte were used for this, three bits would be required to represent eight channels and only five bits would remain for data, leaving only a maximum of 32 different values to represent the entire range of control in each channel. This resolution is much too low, so a proprietary multi-byte format was established. Each channel is represented by two bytes. The format for each byte is *AAANDDDD* where *A* represents an address bit, *N* represents upper or lower nibble, and *D* represents one of eight total bits of data. For example, if the the two bytes, 00100011 and 00111001 were sent across the serial port, they would be interpreted as a value for channel 1 of 00010011, or 147 in decimal. The eight data bits allows for a data value ranging from 0 to 255. Although the blimp control interface is capable of controlling up to eight channels, the blimp used in this project uses only three channels. Channel 0 controls left and right rotation. Channel 1 controls vertical ascension or descension. Channel 2 controls the throttle.

Because the wireless blimp controller cannot process simple RS-232 serial data, the completed blimp control interface also consists of a MAX202 RS-232 transceiver chip that takes an RS-232 serial signal (+/- 12V) and converts it to a TTL signal (0-5V). The TTL signal is then sent to a PIC microcontroller. This serial data is used to generate the standard Futaba control signal, which is essentially a pulse width modulated (PWM) signal with multiple pulses of varying widths, each representing one of up to eight channels. This Futaba signal is then sent to the wireless blimp controller via a modified RJ-11 cable where it is converted into the RF signal.

4.2 Ground Vehicle System

The ground vehicle's overall purpose is to accept the coordinates of a specific target in the grid via the Bluetooth wireless serial link and navigate to that grid while detecting and avoiding obstacles.

The ground vehicle system includes the two servo motors, three sonar modules, an infrared line counter module, a digital compass, an optical mouse, a Bluetooth wireless serial link, and a "master" microcontroller for overall control. The digital compass, optical mouse, and line counter subsystems are used for navigation while the sonar is used for obstacle detection and avoidance, and the Bluetooth wireless serial link is used for two-way communications with the laptop. The overall structure of the components and subsystems of the ground vehicle is shown below in Figure 3.

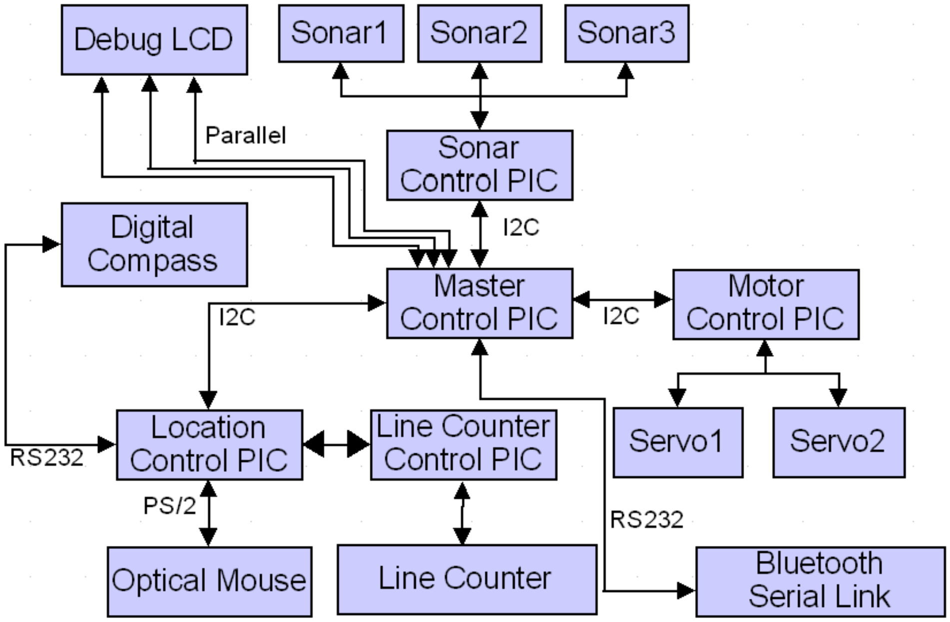

Figure 3 – Block diagram of the ground vehicle's various components

The master controller receives a set of target coordinates via the Bluetooth link, and the master controller determines the distance in each direction the ground vehicle must drive in order to reach the target. The digital compass is used to find the appropriate heading, and the ground vehicle rotates until it matches that heading. Then the motors drive forward while the optical mouse keeps track of absolute positioning within the grid. When the first waypoint is reached, the ground vehicle rotates, again using the digital compass as a guide, to match the new desired heading, and moves forward again until the next waypoint is reached. This process is repeated until the target is reached. If an obstacle is encountered along the way, the ground vehicle maneuvers around the obstacle and reestablishes the path to the target.

4.2.1 Sonar

The ground vehicle is equipped with three sonar modules for obstacle detection and avoidance. One sonar is mounted facing directly forward while the other two modules are mounted on either side of the front-facing module at an approximately 45-degree angle. To get a reading, a short pulse is sent to each module. In response to this pulse, the sonar module sends out a short, ultrasonic "ping" and then listens for its return. The sonar module then outputs a pulse whose width is directly proportional to the round trip time of the "ping" signal and so also the distance to any object in front of the sonar within its operating range. The width of this pulse can then be translated into a physical distance. A PIC16F88 is used to drive these sonar modules.

4.2.2 IR Line Counter

The ground vehicle is also equipped with a Lynxmotion infrared LED line tracker. The line tracker is used in conjunction with a PIC12F629 to keep track of the lines crossed by the ground vehicle in an attempt corroborate the ground vehicle's location with the other components of the navigation system.

4.2.3 Digital Compass

One of the main obstacles for tracking the location of the ground vehicle was the ability to know the vehicle's heading. In the end, it was determined that a digital compass would be able to directly provide heading information. The HMR3100 Digital Compass was selected for this task. This particular digital compass is capable of returning a heading with regard to the earth's natural magnetic field at a resolution of half a degree and a rate of 2Hz using a simple serial communications protocol. The digital compass is interfaced with a PIC16F88 microcontroller that polls the compass for data and receives it in the original three-byte format.

4.2.4 Optical Mouse

The data obtained from the digital compass is used largely in conjunction with the data provided by the optical mouse. The optical mouse provides precise information about two-dimensional movement, which is combined with the heading information provided by the compass to predict the overall absolute location of the ground vehicle within the coordinate system. Because information from the mouse and compass components work so closely together, they are both controlled by the same PIC16F88 microcontroller. The optical mouse uses the PS/2 protocol, which is a common but older communications protocol used by computer peripherals such as mice and keyboards. Because the PIC microcontrollers used in this project do not natively support the PS/2 protocol, support for the protocol had to be developed from the ground up.

The first step was to define the low level protocol of the individual input and output wave packets. Using documentation on the protocol, PIC subroutines were created to read and write serial wave packets and send them to and from the optical mouse. The input and output information is stored on in an input and output byte in the PIC memory allowing similar functionally to the natively supported RS-232 protocol.

The next step was to develop the high level protocol for the mouse interface. For the output functions, this was accomplished by creating subroutines that define the output byte according to the numerical code for different commands to the mouse. The low level transmit subroutine takes care of the actual sending of the code. The input portions of the high level protocol are essentially functions that would call the receive subroutines and make determinations based on the byte received. These subroutines were then meshed together in the main routine, creating initialization and data collection segments of code.

After retrieving data from the mouse, this data is combined with the information retrieved from the compass to determine what cardinal direction the car is facing and increment or decrement the corresponding X or Y coordinate depending on the direction of the compass and the distance determined by the mouse. This gives a system by which absolute X and Y position of the ground vehicle can be predicted.

4.2.5 Motor Control

The ground vehicle is propelled by two simple servo motors modified to turn continuously, which enables them to effectively turn wheels. The nature of the servo motors means that the wheels can only turn at one rate or not at all. This limits the motion of the car to basic forward, stop, and backward control options. Turning is accomplished by turning each servo motor and wheel in opposite directions, resulting in the ground vehicle turning in place. The degree and speed of the turn is determined by the degree of difference between the desired heading and the heading as currently indicated by the digital compass. A heading significantly different from the desired heading results in fast, substantial turning motions while a heading closer to the target heading results in slow, incremental turns. The motors are controlled by one PIC16F90 microcontroller.

4.2.6 PC Interface

Since the ground vehicle is designed to be “blind,” it cannot feasibly locate the target by itself. Therefore, it relies on the blimp to find the target and on the laptop computer to relay information about the location of the target. In order to communicate with the laptop computer, the ground vehicle must be equipped with some wireless communications link. The Spark Fun BlueSMiRF Bluetooth wireless serial cable replacement was chosen for this task.

The BlueSMiRF is interfaced directly with the ground vehicle's master controller using a standard serial communications protocol. The wireless link allows the ground vehicle to both send and receive information to and from the laptop computer. The ability to receive data allows the master controller to receive information about the location of the target from the laptop computer, which has been determined by the blimp and the image processing software on the laptop. At the same time, the ability to send data back to the laptop provides very useful real-time data about the status of each of the ground vehicle's subsystems. This drastically simplifies the debugging process. The ability to send data to the ground vehicle also allows manual control of the vehicle by sending direct commands. The GUI of the software that sends and receives data to and from the ground vehicle is shown below in Figure 4.

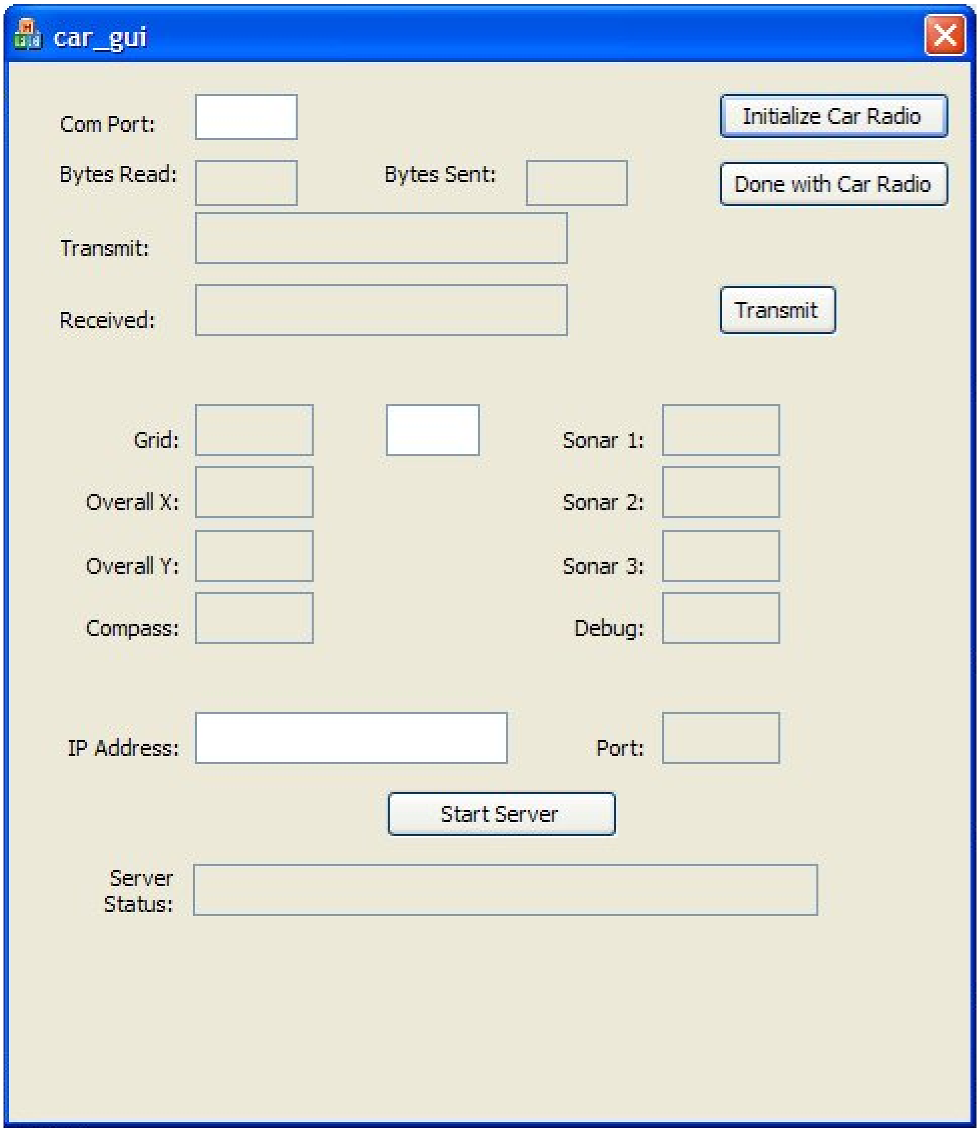

Figure 4 – The ground vehicle debug and control GUI

4.2.7 Master Controller

With so many different sensors and subsystems, the ground vehicle needs an overall “brain” to gather all the various types of information and govern the ground vehicles actions accordingly. The first and probably most significant requirement for the master controller is the ability to communicate with all of the other microcontrollers in the ground vehicle system. After some investigation, it was determined that the Inter-Integrated Circuit, or I^2C, protocol should be used. The PIC16F87A was chosen for the role of the master controller for its overall power and native support for the I^2C protocol.

The master controller is responsible for establishing and maintaining communications between itself and the various microcontrollers and for processing the data it receives from these systems in order to determine the appropriate actions to be carried out by the ground vehicle. The master controller is also responsible for communications with the laptop computer via the Bluetooth wireless serial communications link. For debugging purposes, the master controller is also directly connected to a small LCD screen, which is capable of displaying data from the various sensors. The master controller is also able to use Bluetooth communications to send this debugging data back to the laptop computer to be processed and displayed in a more convenient and human-readable format.

5 Status

As the project stands at the completion of the semester, the image processing software is completed, and the location and orientation of the blimp can be fairly accurately determined. The blimp is fully capable of being controlled by the laptop computer, but testing on the blimp's autonomous control algorithm is largely incomplete. All of the components of the ground vehicle work as designed except for the digital compass. Late in the design process, the digital compass was determined to be significantly more susceptible to stray magnetic fields in Broun Hall than initial testing had indicated. Because the digital compass is such an important component of the ground vehicle's navigation system, the ground vehicle generally has difficulty navigating the coordinate system to reach the target location as determined and specified by the image processing software. Because the ground vehicle follows a heading provided by the compass, the car tends to follow stray magnetic fields instead of correctly navigating the coordinate system.

6 Conclusions

It was clear from the beginning that this cooperative robotics project would be a significant undertaking, and in the end it proved to be too complex to be completed entirely in one semester. However, vast amounts of experience in many areas were gained throughout the semester, including teamwork, communication, and systems design and integration, both low level and high level.