THE UNIVERSITY of TULSA
ELECTRICAL ENGINEERING

Technical Overview of
HURRICANE ALBERTO

The University of Tulsa Autonomous Vehicle for
The 8th Annual International
Autonomous Unmanned Vehicle System

Submitted to:

Mr. William G. Agnew
3450 31 Mile Road
Washington, Michigan 48095

Submitted by:

Mohamed Al-Najjar
Martin Bean
Sarah Billingsley
Sigfried Falkenburg
Earl Hammon, Faculty
Kaveh Ashenayi, Faculty Advisor
Email: kash@utulsa.edu

1. Introduction

The University of Tulsa entry in the 8th annual Autonomous Unmanned Robotics Systems (AUVS) competition is named "Hurricane Alberto". "Alberto" is the seventh generation vehicle, built using the same chassis as the previous year's model. It was designed with the intent of using "off-the-shelf" equipment. This approach allows for simple construction and facilitates quick replacement of damaged parts. In addition, it provides for an easy upgrade path.

"Alberto" was designed and built with the primary goal of competing in the Autonomous Challenge Competition. However, its low cost and high maneuverability as well as its high load carrying capability presents an opportunity to the industry as an inexpensive, flexible, effective and highly reliable robotic vehicle.

This year's team consists of a faculty advisor, Kaveh Ashenayi; a faculty member, Earl Hammon; a graduate student, Mohamed Al-Najjar; a senior, Martin Bean; an exchange student, Sigfried Falkenburg; and a freshman, Sarah Billingsley. Sarah Billingsley was selected as a member of the team to assure continuity for the next year's team.

Hurricane Alberto was built from ground up specifically for the unmanned robotics competition. At its most basic level, Alberto is simply a physical drive unit with a guiding intelligence. Alberto uses an analog RGB video camera as its only sensory input. An onboard PC processes the information from the camera and makes a decision. The decision is then implemented via a custom designed ISA expansion card.

The vehicle can move at either 0 or 4.6 miles an hour. Decisions about speed and direction are made at a rate of up to 30 frames per second. There is an emergency stop to terminate vehicle operation and a joystick so that it can be driven manually when needed. Differential steering provides the dual advantages of simplicity and maneuverability.

This paper covers the design of the car, including both the mechanical design and the electrical design, such as the vision system, the ISA controller card, and the power system. The cost of the vehicle is also included. Time and effort of each team member is also presented.

2. Mechanical Design

For the purposes of this paper, mechanical design refers to the parts of the vehicle that have to do with its physical motion, except for the electrical control systems. The motors are included in the description of the mechanical design of the car. The car is a refurbished model from one of the prior AUVS teams. One of the main design premises was to make the car as simple as possible. Therefore, the chassis is a simple steel rectangle with a high mount for the camera and mounts for the other devices. The vehicle does not exceed the height of 5 feet at its highest point. Chassis is 3' wide and is 5' long.

The drive train consists of two electric motors driving (via chains) four 5.5-inch wide tires. This provides for significant footprint to support the high torque available from the motors and provides more surface area to prevent it from sinking in the sand. The car maneuvers using differential steering like a tank. Since the motors are controlled by relays, there are only three states for the motors: forward, stop and backward. The relays are located in the relay box. The ISA card, a part of the computer, controls the relays.

The motors used are 24-volt permanent magnet DC electric wheel-chair motors. Each motor is rated at one-quarter horsepower with a full rotational speed of approximately 450 RPM. They both have built in mechanical brakes that engage when electrical power is removed.

3. Electronics

The electronics are any part of the car that deals with information gathering and control of the vehicle. This consists of an analog video camera, a frame grabber, a Pentium II central processing unit (CPU), a Microchip PIC 17C44, the batteries, and all of the relays used for motor control, as well as their supporting electronics. The three main categories of the electrical design of the car are the management of its vision, the ISA control card, and power distribution.

3.1 Vision

The most important subsystem is the vision system. It consists of two major components – the hardware and the operating software. The hardware is simply the RGB camera, the frame grabber and the computer. The software is a completely integrated developing environment with a graphical user interface (GUI).

The camera is a JVC 1270 color RGB camera equipped with a Fujinor 4.8 mm wide-angle auto-iris lens. It provides distinct red, green, and blue signals, as well as a separate composite sync. The camera sends its signal to the computer (via the frame grabber), which analyzes it and sends the processed information on to the ISA card. The ISA card controls the steering and is considered a separate part of the vehicle.

3.1.1 Premises

There were several basic premises (all based on the rules of the competition) that were used to simplify development of the vehicle.

- We assume that all obstacles and lines can be detected by their colors. This simplifies the problem by allowing us to ignore the possibility that an obstacle is the same color as the background. Under the rules of the competition, this seems to be a reasonable assumption.
- We assume that all objects are flat on the ground. This will cause objects to appear to have more substance “behind” them than is truly the case. However, this will not be a problem as the near side would block this area from the current path in any case.

- We assume that the ground in front of the robot (for about 5 to 10 feet) is a flat surface. This causes some distortion, but will not significantly reduce the algorithm's effectiveness.
- We assume that staying on the path is the same as avoiding the lines that define the path. This simplifies the program because both staying in the path and avoiding the obstacles can be contained in the same algorithm.
- We assume that each pattern of obstructed and clear regions requires the same steering decision every time it is encountered. The easiest way to justify this assumption is to prove the opposite wrong. We could not see any situation where two identical patterns would require different steering decisions.

3.1.2 Integrated Editor

A complete integrated editor has been developed to make the vision system simpler and quicker to use. It is a complete environment for designing, debugging, testing, and running the vehicle's vision. From its Graphical User Interface, the user specifies all aspects of the intelligence. First, the user specifies the region of the ground visible to the camera, and from this the computer calculates the camera parameters. The user then specifies the set of zones on the ground in front of the vehicle that the image is divided into. After that, the user creates a list of color sets that the vehicle must recognize. Each color set is a grouping of one or more fuzzy colors that are treated identically in the rules. The user next creates the rule patterns, in which each rule specifies the desired color of each zone in a matching input pattern and the importance of each zone. After the user creates these properties, the state machine handles the actual state transitions and decisions. These steps will be discussed in greater detail later in the paper.

3.1.3 Vision Algorithm

Image processing in the vision algorithm passes through three stages: image capture, color/object detection, and move selection. Image capture is performed by the frame grabber hardware, and can be done while the computer is still processing the previous frame. This effectively doubles the speed at which the images are processed.

Image processing involves several steps. First, we divide the captured video image from a single color camera into a set of zones. The individual pixels in each zone are then compared to all the pre-determined fuzzy colors within each color class, flagging each color class as either “present” or “absent” in that zone. Next, the program finds the rule that best matches the pattern found. This matched pattern is used to determine the next state in the state machine. Lastly, the current state of the state machine is used to decide the output.

3.1.4 Dividing the Image into Zones

Two distinct classes of space are defined for the manipulation of the image – camera space and world space. Camera space is the rectangular array of pixels that we get from the camera. World space is the trapezoidal region of the ground that the camera-space maps onto, using the assumption that the terrain is flat.

Zones are rectangles that can be edited in either camera space or world space. The two spaces can be mixed; you can have some zones that are rectangular in camera space, and other zones that are rectangular in world space. The editor allows the user to alter size/position of any zone, and to create new zones.

3.1.5 Recognizing Colors

Three factors complicate object detection based on color. The first is that ambient lighting conditions can change constantly as clouds block or uncover the sun, or if shadows cover part of the course. The second is that there is some noise inherent to the digital camera, which also may change its output levels as it warms up. The final challenge is that color is not a strictly definable concept. For example, there is no discrete point where red ends and red-orange begins.

The first problem (changing light conditions) and part of the second (noise while the camera warms up) are handled by dynamic recalibration and color normalization. This concept is identical to that employed by the Mars rover for adapting to Martian lighting. The program simply finds the black and white regions in a known part of the image and uses those values to ensure that the maximum and minimum limits for each color component are accurate. It then builds and uses a look-up table to convert each color into a standardized range. Alternatively, it can be set so that the individual components of each pixel are scaled to make the maximum of each of the three color components constant among all of the pixels. This will eliminate the effects of shadows, but will also make it impossible to distinguish different shades of the same color.

The challenges presented by the camera's noise and inconsistency, the difficulty of exactly defining colors, and the possibility of changing light levels can be handled by using "fuzzy colors." A "fuzzy color" is a set of three fuzzy membership functions – one for each color component. The shapes of the membership functions are completely user-defined, and can be different for each of the three components of a fuzzy color. A pixel's membership to a fuzzy color is calculated by taking the fuzzy AND of the membership of each pixel component to its corresponding component membership function. The pixel is identified as the color with the greatest membership value; however, if the greatest membership value is below a user-defined threshold, the color is unrecognized.

Each fuzzy color's membership function is applied to each pixel in each zone. This gives a fuzzy match for that color to that pixel. If "enough" pixels in a zone match "well enough," that zone is said to contain the color class that goes with the fuzzy color. The color classes have a collective meaning for all the constituent colors, such as "blocked" or "safe." A color class is present in a zone if any of its component colors are present in the zone; otherwise, it is absent. Present is represented as a positive one, while absent is represented as a negative one.

As mentioned previously, the rules depend on color sets rather than individual colors. Each fuzzy color, including the “unrecognized” color, belongs to exactly one color set. Pixels are scanned on a per-regions basis. At the start of a region scan, all color set counts are zeroed; as a color is recognized, the corresponding color set’s count is incremented. When any color set’s count exceeds its independent threshold, the region is flagged as containing an object belonging to that color set. This builds the pattern of obstructed and clear regions.

3.1.6 Pattern Matching

We use a pre-trained neural network to match the best rule pattern to the current combination of present and absent color classes. Each zone has a signed weight for each color. A positive weight indicates a color class that must be present, and a negative weight indicates a color class that must be absent. A weight of zero indicates a color class that does not matter. In this zone, each color class match level, either positive or negative one, is multiplied by its weight and added to the total sum for the rule. A rule-dependant bias is added to each rules match level, and then the resulting match level is normalized to the range [0,1]. Basically, this means that no rule is more likely to be found correct simply because more or less of it lies in zones that do not matter. After all of this is done, the rule with the highest final match level is picked.

3.1.7 Neural Network

The color/object recognition stage gives a unique pattern that represents the arrangement of the obstacles and course lines from each input image. Since the number of possible patterns is prohibitively large, some technique must be used to limit the number of rule patterns. It was decided that neural network pattern recognition offered an ideal solution. The weights mentioned above are simply the connection weights for the neural network that recognizes an individual rule. The complete neural network is a set of sub-networks for each rule pattern linked together by a maxnet, so that only the closest-matching pattern is activated. Since the neural network activates the closest solution, it is possible to combine two similar rules into one. Additionally, setting the connection weight to zero can create “don’t-care” regions, which further reduces the necessary rule size. The final advantage of the neural network approach is that it can be trained using standard techniques either while the user is driving the car manually or by using a set of stored images. In other words, the computer can automatically generate an appropriate vision algorithm. Of course, the user can still edit the weights and add or remove rule patterns.

3.1.9 State Machine

As hinted earlier, the neural network merely activates a rule pattern. A state machine transition makes the final decision of what move to make. A finite state machine was chosen to give the vehicle a rudimentary memory. In this way, the vehicle can perform actions that require more than one frame from the digital camera. The major advantage is that these actions can be defined from the editor, so that absolutely no recompilation is necessary to build a completely different yet complex driving algorithm.

The output of the device is controlled by a tweaked version of a Moore state machine. The next state is decided by the current state, the currently selected rule, and the minimum transition delay. The only nonstandard feature of this state machine is the requirement that it stay in the current state for at least N decision cycles, where N is the user-edited transition delay. While N is always one for traditional state machines, this is not always the case for the state machine we use. Note that N must be greater than or equal to one to make physical sense. Once this criterion has been dealt with, the vehicle’s finite state machine is similar to other state machines, such as the current output depends only on the current state. One nice feature is that the state machine, like the rest of the vision algorithm, can easily be edited through the integrated editor.

3.2 Control System

The Industry Standard Architecture (ISA) control card is a new feature that has been added to the car for this year’s competition. In the past, we were faced with the problem of the external components receiving interference from the motors and relays. We also had a problem when the computer sent the command for the E-stop to be activated. The power to the relays was then terminated, but the car rolled for two feet before the back-EMF brakes engaged. This problem arose when the incline was added to the course. To solve these two main problems, an ISA card was designed.

The ISA card is directly plugged into the CPU bus. The CPU bus provides power to the ISA card, which eliminates the extraneous signals from external sources and minimizes the need for some of the external components. Creating a delay between terminating the power to the motors and terminating the power to the relays solves the E-stop problem. The delay is adjustable and is currently set at about 0.25 seconds.

The ISA card, in its basic use, emulates a parallel port. It is based on a 20-bit addressing scheme with an 8-bit data bus, therefore an 8-bit comparator is used. The ISA card works as 8 banks of 8-bit registers. Bank ‘0’ is the most important and used as data area. The ISA card was designed to provide additional features. The first feature is that it takes information/data directly from the bus and, as a backup, we can take information from the CPU parallel port as well. Secondly, once we get the information/data from the CPU via the parallel port or bus, we can send it directly to the relays, which control the motors. Third, the ISA card supports an external power supply. This allows us to run the car manually using the joystick without depending on the CPU to supply power. We used a 12-Volt to 5-Volt inverter inside the relay box to achieve this. The ability to switch the power supply allows us to expand the life expectancy of the batteries that provide power to the CPU. In addition, it will allow us to manually drive the car even if the PC is dead. An LED is used to indicate which power source the ISA card is using. Fourth, The ISA card supports an external joystick with an E-stop and a reset push button. The joystick overrides the ISA bus when it is plugged in. Lastly, optoisolators are used to protect the ISA card electrically, in case the cable is cut, damaged, or shorted.

3.2.1 Microprocessor

The microprocessor is a high-performance, CMOS, fully static, 8-bit micro controller that employs an advanced RISC architecture. The PIC17C44 has enhanced core features, 16-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard Architecture allow a 16-bit wide instruction word with a separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single cycle. Branch instructions are the single exception, requiring two cycles. A total of 58 instructions are available in this device. Furthermore, a large register set gives some of the architectural innovations used to achieve a very high performance. PIC17C44 device has up to 454 bytes of RAM and 33 I/O pins. The microprocessor also has four timer/counters, two capture inputs, two PWM outputs and a Universal Synchronous Asynchronous Receiver Transmitter (USART). These special features reduce external components required which enhances system reliability and reduces power consumption. The PIC17C44 can execute programs from internal memory, external memory, or both. The PIC17C44 also can address its register files or data memory directly or indirectly. All special function registers, including the Program Counter and Working Register, are mapped in the data memory.

3.2.2 Control Software

The program for the microcontroller is written with the Microchip development tool MPLAB. MPLAB is a Windows-based Integrated Development Environment (IDE) for the Microchip Technology Incorporated PIC micro controller family. It is a very powerful and useful tool to write and test (simulate) programs with.

There were several requirements for the microcontroller program

- If the joystick gets plugged in, it will override the control of the relays and motors.

- As stated previously, if an E-stop button is pressed, power is cut off to the motors; then, after a delay routine, power is terminated to the relays. No joystick or computer commands get accepted and the computer gets the massage “error has occurred”. This state is retained until the reset button is pressed. The routine has to be an interrupt routine.
- Illegal input combinations must be ignored.

3.2.5 Emergency Stop

"Alberto" was designed and constructed with safety devices as an integral part of the design process. We are using mechanical auto-lock motors, so that when there is no power supplied to the motors they lock in place guaranteeing that the car will not move. The Emergency Stop (E-stop) was designed so that the vehicle could be stopped at any time by any of three methods. The primary E-stop trigger is the Radio Frequency (RF) that, when applied, stops the vehicle. Due to the time-consuming and complex nature of designing an RF E-stop, a pre-assembled transmitter/receiver pair was purchased from Linx Technologies to meet the range and stability criteria. The device provides a range of 500 feet, operates at 315 MHz, and can transfer 5,000 bits/second of data. The transmitter is mounted in a small casing for weather protection and ease of use; the receiver is mounted inside the vehicle for the same reasons. The second method is a round push-pull button approximately 2 inches in diameter. The third method is a normally open button (on the joystick) that, when pushed, activates the E-stop sequence. This last method is only effective when the vehicle is running off the joystick. A manual restart is required after the E-stop sequence is activated.

3.2.5 Joystick

The joystick allows direct, independent control of each motor. Each motor must be in one of three states: forward, stop, or reverse. We used two self-centering tri-state switches to directly send the motors to these states. The switches are connected logically; the right switch controls the right side, and the left switch controls the left side. Pushing the switch forward causes that motor to go forward, and pulling the switch back causes it to go in reverse. The center position stops the motor. Self-centering is an important safety feature, because we want the vehicle to stop immediately in the event that the operator has to let go of the joystick. Additionally, the joystick includes two buttons for manual E-stop and reset. Lastly, the vehicle is designed to stop immediately in the event that the joystick becomes accidentally unplugged.

3.3 Power Distribution

System protection was an important consideration when the vehicle's electronics were designed. The vehicle's systems are electrically separated from one another with optical isolators to prevent the failure of one portion of the system from destroying the remainder of the system. The electronics can be classified into two categories. The first category consists of how the power is distributed all through the car. The second category is what is needed to drive the motors. The following is a brief description of the above categories.

Four 12-volt DC gel cell batteries power the vehicle. Two of the batteries are dedicated to the motors and electronics, and the other two are dedicated to the CPU and camera. Two batteries dedicated to the motors are connected in series because the motors require 24 volts for proper operation. The motors are placed in-line with an H-Bridge (made of four relays) and a fuse rated at 70 amps, as a safety apparatus, on the 24 volts side of the battery plant.

The two batteries for the CPU and camera are positioned in parallel to provide longer operation periods. The CPU requires 120 volts, 60 Hz AC power, so an inverter is connected to the 12-volt side of the battery plant. The camera operates at 12 volts and is linked directly to that side of the battery plant. In addition, the relay control circuit, the joystick and the ISA card (via a 12 to 5 volt converter) are attached to the 12-volt side of the power distribution.

4. Organization and Serviceability

In the construction of Alberto, organization and serviceability were major concerns. Without considering these issues it would be virtually impossible to find and fix any problems that could occur. To address these issues we made sure that the wires were all marked and color-coded, to allow them to be simple to trace. Additionally, the vehicle uses different connectors in different places to avoid the possibility of a piece being accidentally connected in the wrong place or the wrong way. We are using ribbon cable where it is possible so that we can avoid the problem of wires becoming loose and tangling around each other. Any wires that could not be replaced with ribbon cable are bundled together and placed in conduits to avoid this problem.

5. Manpower and Resources Required

As it was stated previously, the team consisted of four students. The following will outline their contribution and their efforts. In addition, we will review the cost of this unit.

5.1 Time and Effort

Name	Major	Classification	Hours
Mohamed Al-Najjar	EE	Graduate Student	223 hours
Responsibilities: ISA Card Design, Relay Integration, Power Distribution, Documentation, Construction, Wiring, Modeling.			
Martin Bean	EE	Senior	279 hours
Responsibilities: ISA Card Design, Power Distribution, Testing, Construction Supervision, Wiring, Relay Integration, Modeling, Motor Control.			
Sarah Billingsley	EE	Freshman	124 hours
Responsibilities: Documentation, Testing, Vision, Team Supervision.			
Sigfried Falkenberg	EE	Exchange Student	217 hours
Responsibilities: ISA Card Design, Joystick design, Software, Testing.			

Hammon, Earl	Faculty	100 hours

Responsibilities: Vision, Camera Field Testing, Supervising.

Ashenayi, Kaveh, PhD.	Faculty Advisor

Responsibilities: Overall project supervision.

Total person-hours worked in the design and construction: 843.

5.2 Project Budget

The project budget was as follows:

PC	$1,500.00
Frame Graber	$ 700.00
Camera	$ 300.00
Chasis	$ 200.00
ISA Control Card	$1,000.00
Misc. Parts	$ 500.00
Total Cost	$4,200.00

6. Conclusion

This paper described the conceptual design and working components of The University of Tulsa, Hurricane Alberto. First, the mechanical components of the vehicle were discussed. This included the description of the steering mechanisms, drive train, speed and frame design. Second, electronic components of the system were discussed. This included a description of the power distribution, motor switching control, motor speed control, microprocessor units, video and emergency stop.