Bluefield State College

Anassa V

Team Members

Matthew Duncan, Chris Thompson, Justin Milam, Caleb Tote, Jeremiah Weakley, Tracy Sartin, Martha Morris, Otis Mac, Andrew Scott, Louis McAllister, Toni Villenueva, Brian Curto, Parker Stacie, Josh Baker, Salah Alobaishi

Advisor

Dr. Bob Riggins

Faculty Advisor Statement

I certify that the engineering design of the vehicle described in this report has been significant, and that the student effort is equivalent to a senior design capstone project.

_____

Bob Riggins, Electrical Engineering Technology, Bluefield State College

Table of Contents

1. Introduction

1.1 Overview

The Bluefield State College (BSC) Robotics Team is pleased to present Anassa V for entry in the 17th Annual Intelligent Ground Vehicle Competition (IGVC). The name Anassa means "Queen" in Greek. Our team is confident that Anassa V will continue to live up to her legacy and become Queen of the IGVC 2009. This is the fifth year that an Anassa-series robot has entered IGVC. Although Anassa IV was the Autonomous Event champion in 2008, we have added new design innovations in both hardware and software to Anassa V. This paper is comprehensive in scope, covering Anassa V's design in its entirety. In addition, the new design innovations will be emphasized in each section and summarized in Section 5.

This year our team was fortunate to have some members from the BSC Marketing Division. Through their outreach efforts, we were able to acquire necessary funding for Anassa V. In the past, our local businesses have had a great relationship with the robotics team, and they have often hired team members before and after graduation. Local industries have also donated parts, discounted our purchases, or provided assistance in some way that has greatly benefited our projects. For Anassa V, these industries include Pemco, Inc., ConnWeld Industries, Wal-Mart (Bluefield, VA.), Advanced Auto Parts, Bucyrus, Miller Associates, and Charlatte America. In addition to the sponsors listed above, BSC and the Center for Applied Research and Technology (CART) also provided funds for Anassa V.

1.2 Anassa V Specification Overview

Table 1.1 lists Anassa V's physical specifications.

Weight	250 lbs (excluding 20lb payload)
Horizontal Center of Gravity	24 inches to rear, 20 inches to front
Vertical Center of Gravity	11 inches to ground
Height, Length, Width	6 feet, 45 inches, 26 inches
Wheel Diameter	15 inches front, 8 inches rear
Wheel Base, Ground Clearance	25 inches, 4.25 inches

Table 1.1

2. Design Process

2.1 Overall Design Methodology

The process we used to design Anassa V began with a fully documented "Post-IGVC-2008 Analysis" of Anassa IV. This document is hereafter referred to as Post-IGVC Analysis. After last year's

IGVC, the team examined Anassa IV's performance and noted the areas that needed improvement in both hardware and software. These results are discussed in other sections of this report.

The Post-IGVC Analysis includes a list of items that need improvement. Once the Post-IGVC Analysis is documented, the BSC team divides into hardware and software groups to develop a set of proposed solutions for each item in the analysis relating to their respective group. A decision about which solution to implement is made through the hierarchical team structure (See Section 2.3). At this point, we test each solution in a simulation, real life, or both, and analyze the results. If the results prove that our solution is successful, we proceed to the next item on the analysis. Otherwise we re-analyze the proposed solution. Figure 2.1 shows an abstract view of our design process.

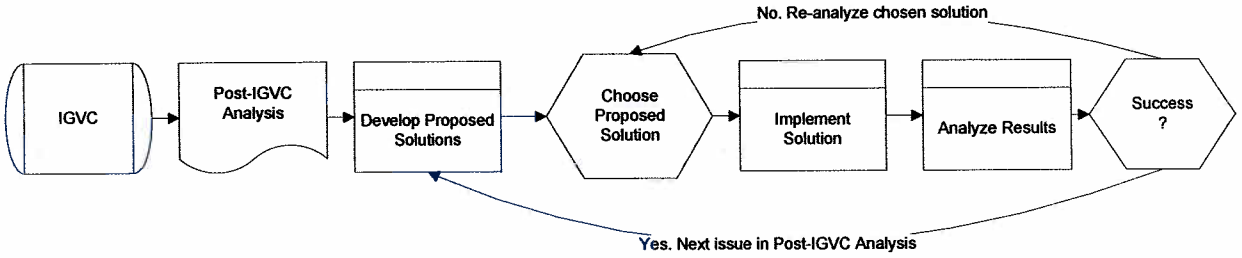

Figure 2.1

2.2 Software Design Methodology

Our software design methodology is based on the object-oriented style of programming and a combination of the throwaway prototyping and parallel methods of software development. Depending on the difficulty of the task and the number of available programmers, sub-groups are assigned to work on one item each from the Post-IGVC Analysis. After the code is written, a prototype is made by combining the sub-projects. The prototype is then tested and analyzed. This process continues until a final version of the software is created (See Figure 2.2).

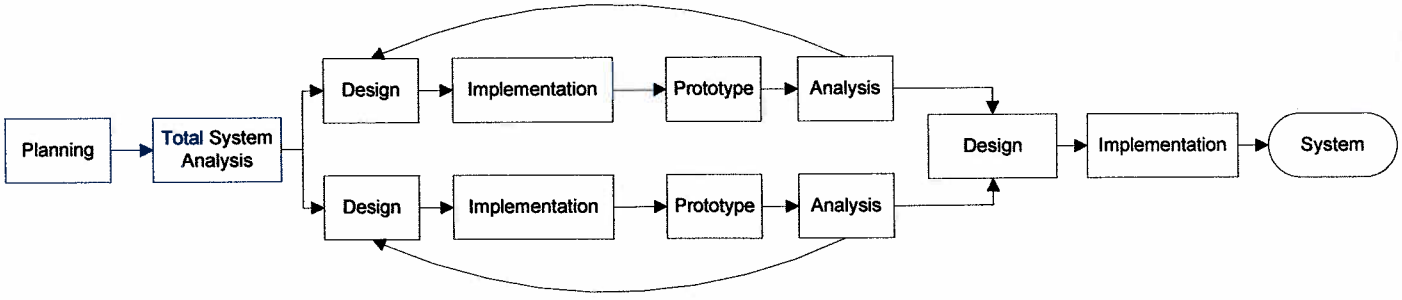

Figure 2.2

2.3 Decision Making Process and Team Structure

Our team is comprised of undergraduate students from the Electrical Engineering Technology, Computer Science, and Business Divisions at BSC. Each member contributed to all or part of

the design and fabrication of Anassa V. We estimate that our team spent 1,200 hours designing and fabricating Anassa V. The team structure and members are listed in Figure 2.3.

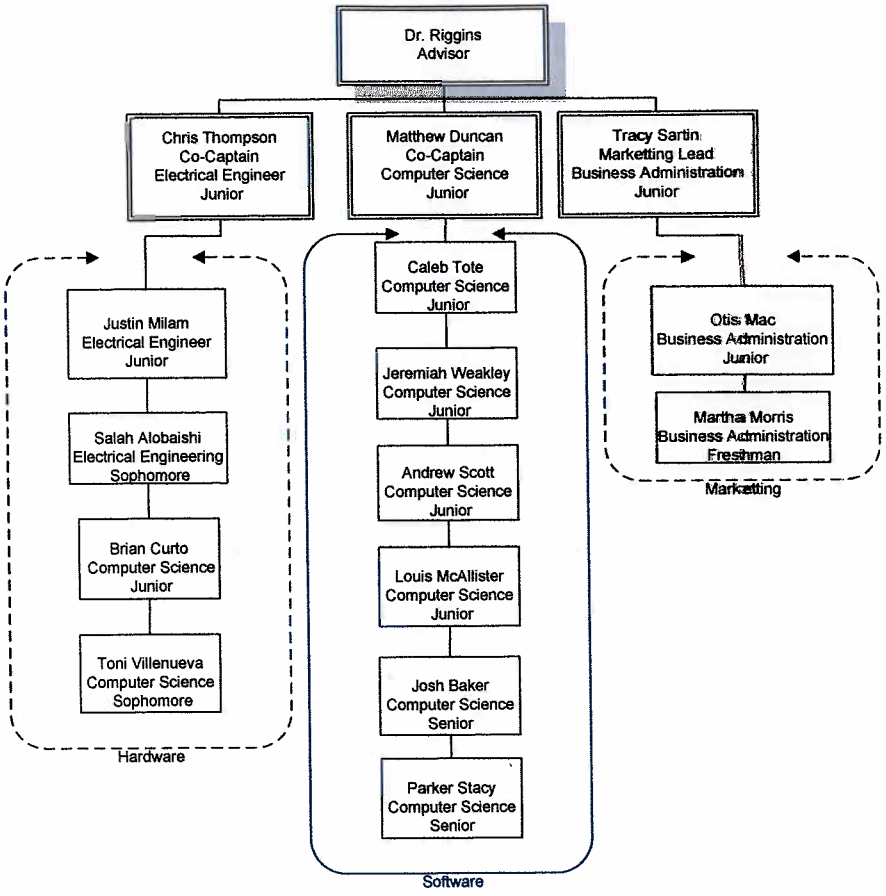

Figure 2.3

3. Hardware Design

The hardware design of Anassa V is divided into two sections: mechanical design and electrical design. Table 3.1 lists the replacement cost and the team cost of each component and the total cost of Anassa V.

Description	Actual Cost	Team Cost
Desktop computer	$1,300	$1,300
Sony camera	$700	$700
2x 180 degree LMS(SICK)	$10,000	$6,000
DGPS-w/antenna/cables	$3,000	$2,100
Aluminum platform	$500	$0
Wireless E-stop	$100	$100
Compass	$700	$700
24 to 12 volt dc to dc	$200	$200
24-volt charger	$300	$300
24-volt pc power supply	$250	$250
Microcontrollers	$25	$25
Miscellaneous	$278	$198
Jazzy wheelchair	$5,977	$1,037
Total	$23,330	$12,910

Table 3.1

3.1 Mechanical Design

3.1.1 Top

The top of Anassa V was designed so that it would be sturdy and require no more material than we deemed necessary to fit all of our equipment inside. The top is made

of aluminum and tinted lexan. By using these materials, not only is the top very durable, but it is also very light. We wanted the top to be lighter than the bottom in order to keep Anassa V's center of gravity as low as possible. We used tinted lexan for two reasons: first, to allow us to see inside Anassa V without having to remove the panels, and second, to reflect sunlight in order to keep the inside of Anassa V as cool as possible. Anassa V's aluminum body also reflects the sun's energy very well.

3.1.2 Bottom

The bottom of Anassa V is the modified chassis of a Jazzy 1170XL electric wheelchair. The frame is constructed of fourteen-gauge milled square steel tubing that is built to carry a payload of more than four hundred pounds. Both the frame and chassis components are weatherized with a primer coating and multiple layers of enamel paint. The chassis components include an active-track suspension system that allows all wheels of the drive system to travel independently as the vehicle moves across uneven terrain. The vehicle is elevated by fifteen-inch pneumatic tires (the drive axle), adjustable eight-inch rear articulating caster wheels, and adjustable six-inch anti-tip wheels for stability. The arrangement of the wheels provides a four and one quarter-inch ground clearance which generates a low center of gravity. This, coupled with the active-track suspension system, gives the robot a curb-climbing height of six inches. The overall robot base of twenty-two and one-half inches wide by forty-five inches long provides a tight turning radius of twenty-three inches.

3.1.3 Efficient Use of Materials

We decided to use the frame of an electric wheelchair for three reasons. First, with a ready-made frame, we saved both time and money designing and fabricating the bottom of Anassa V. Second, the Jazzy wheelchair was designed over the course of twenty years to be strong enough to carry a 400-pound person safely. This makes it strong enough to carry all of our equipment and the payload. Third, the rugged design of the base makes it reliable and durable.

3.2 Electrical Design

3.2.1 Power

Anassa V uses two deep-cycle 12-volt batteries to supply power to all of its electronic devices. Since all devices are connected to a single power source, we do not have to charge multiple sources, possibly at different times. Each battery is rated at 55 amp-hours. By dividing the total amp-hours of the batteries by the total amps consumed by all the devices, we calculate that the total battery life of Anassa V is 1.6 hours under normal operating conditions and 60% operating efficiency. Our observations concur with this maximum run time. See Table 3.1 for a complete list of devices and current draw.

3.2.2 Sensors

The sensors on Anassa V are used for obstacle detection, positioning, and communication. The following list describes the sensors that Anassa V uses to interact with its environment.

Component	Average Current Draw (Amps)
LMS x 2	3
GPS	1
Compass	.15
Computer	6
Camera	.5
Wireless E-Stop	1
Monitor	2
Motor x 2	20
Controllers	2
Wireless Router	.5
Miscellaneous	5
Total	41.15

Table 3.1

Two SICK Laser Measurement Systems (LMS) – measure distance between robot and objects. Maximum Range: 8 meters. Resolution: 1°. Sweep Angle: 180°. Precision: one millimeter.

Maretron Solid State Compass – determines heading in degrees. Precision: 0.1°.

CSI Wireless DGPS Receiver and Antenna – determines latitude and longitude of the robot as well as velocity. Precision: 2 feet and 0.01 feet at one Sigma (67% of the time).

Sony Handycam – captures still images of Anassa V's surrounding environment. Resolution: 640x480 pixels.

Linksys Wireless Router – communicates via Wi-Fi 802.11b/g. Range: up to 100 meters. Bandwidth: 54Mbps.

3.2.3 Computer Systems

The computer in Anassa V is a high performance machine with a dual core processor, two gigabytes of ram, and a one gigahertz front-side bus. We use a wireless keyboard and mouse to operate and program Anassa V. In addition to the main computer, the robot also uses several microcontrollers to handle various tasks. For example, an 8-bit microcontroller controls Anassa V's motors. This distributed computing allows the main computer more time to perform the sensor integration and path planning described in later sections.

3.2.4 Actuators

The drive train consists of two twenty-four volt DC motors that provide vehicle locomotion. The motors are connected through a speed reducer to the drive axle. The speed reducer

converts the high speed, low torque of the motors to the low speed, high torque required to move a four-hundred-pound payload on level ground at top speed. This produces a top speed of 4.9 mph for Anassa V. It also allows Anassa V to climb up to a 30° incline. This calculation was predetermined by Pride Mobility Products Corporation for the Jazzy wheelchair.

3.2.5 Efficient Use of Power

In order to provide power to all of Anassa V's devices, we only have to charge two batteries. To ensure that the batteries are charged when we need them, we have two ways to charge them: fast charge and trickle charge. The primary method is the on-board trickle charge. Trickle charge takes up to eight hours to charge Anassa V to full charge, but it helps to extend the life of the batteries. When we need to get a full charge quickly, we use the secondary method, the off-board fast charge. The fast charge can recharge Anassa V's batteries in 30 minutes or less. However, repeated use of the fast charge method will decrease the life of the batteries.

3.3 Safety

Anassa V incorporates several safety features. First, the hardware utilizes multiple Emergency Stops (E-Stops) to provide several levels of control. Anassa V now has three ways to manually E-Stop the robot, and three wireless ways. Second, software programs continually monitor the system for errors in control, communications, and battery charge levels. Third, "Heartbeat" (fault monitoring) signals are monitored at critical points in the system. Finally, the software has the inherent "fail-safe" ability to abort the current mission and shutdown the vehicle in the event an error is detected. Anassa V has both a Soft E-Stop and a Hard E-Stop located on the rear instrument panel. The on-board Hard E-Stop switches all power on/off. The on-board Soft E-Stop and the wireless E-Stop both switch the controller on/off, but they do not affect the main power. These two systems are independent of each other for additional safety. The wireless radio-controlled (RC) E-Stop has been installed to extend our control range to fifteen hundred meters. In addition to the E-Stops, Anassa V is also wired for safety. All electrical connections are made using correct wiring gauges. Furthermore, each device is connected to its power source through a fuse box, and the fuse for each device is the correct size to allow for the maximum safe current. Anassa V is also safe while charging. When the chargers are connected to Anassa V, the motor controller will not allow Anassa V to move using manual control methods or computer control.

3.4 Post-IGVC Analysis and Solutions

The Post-IGVC Analysis contained a couple items relating to hardware: add wireless support for remote communication and completely rewire and rearrange the inside of Anassa V. To add wireless support for our vehicle, we put a Linksys wireless router in the body of Anassa V. Our team removed all

components and wiring from Anassa V and arranged it in such a way that now the most commonly modified devices are the easiest to access.

4. Software Design

4.1 Signal Processing

The data collected from our external devices, including the compass, camera, DGPS unit, and the LMS units, is sent to the computer, becoming the inputs to our software program. Anassa V makes its decisions based on that data so the signal received from each device must be processed into a form that Anassa V can use. For example, the DGPS unit sends its data in the form of GPS "sentences." The data is sent in the form of a string, which the software must parse to get the latitude, longitude, and velocity. The compass also sends its data in the form of a string. This string contains the heading of the robot in degrees with a precision of 0.1°. We interpret this data in the same way we interpret GPS data. The LMS sends its data as an array of 373 binary numbers. We convert the binary numbers into an array that contains 181 elements which correspond to one degree each. The number that is in each array position is the distance in millimeters to the object at that position.

How the camera data is brought into our program is more process-intensive than how we bring in data from any other device. In order to use images from the video camera, we must use a software frame grabber which retrieves frames from the camera and saves them as images so that the software can analyze the content. We use a process we call "color vectors" to analyze camera data. There are certain colors and magnitudes that correspond to different obstacles on the IGVC course. For example, the lines that mark the boundary of the course are white. To determine if an object is white, we analyze each pixel's red, green, blue (RGB) content. Every color in the spectrum is made of a certain amount of red, green, and blue. Our software analyzes the RGB vectors to determine if the pixel's vector is close enough to the hypothesized vector of an object or marking (See Figure 4.1). Our software recognizes colors as either passable or restricted based on the rules set by the IGVC. One of our design innovations is to use a process called adaptive color vectoring. Adaptive color vectoring allows Anassa V to learn and modify the hypothesized color vectors of features of interest.

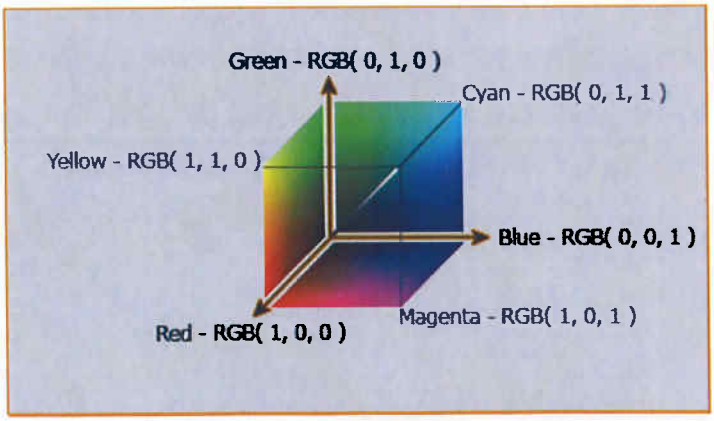

Figure 4.1

Color vectors can change based on environmental variables such as full sun or shade. For Example, white lines have a different intensity and RGB content in the shade than in full sun. Once Anassa V has received all the data pertaining to obstacles that it is able to

perceive, the data must be converted to a common geometry before any decisions can be made about obstacle avoidance and path planning.

4.2 Mapping the Inputs

Anassa V uses a "map" to represent obstacles logically inside the software. This map is a two-dimensional array of integers that measures 80 by 80 nodes. Each node in the array represents about four inches of real-world space. We chose this size because it is approximately the size of the smallest feature on the courses at IGVC, the width of the white line. After the computer receives the data from each of the sensors, it places that data onto the map. An obstacle is stored as a cost on the map. If a node has a high cost, the robot knows that is should not go through that space. The cost assigned to a node depends on what sensor reported that obstacle. For instance, the LMS data has a higher cost than the camera data because the LMS is more accurate than the camera. Once all the data has been placed on the map, Anassa V can begin the process to choose a path that will avoid the obstacles.

4.3 Decision Process

In our software, we use two very similar sub-programs, one for each challenge at IGVC. These sub-programs are similar in that they use the same basic path-planning algorithm. The difference between them is in the cost equation that each one uses for determining the desired destination. First we will discuss the basic path-planning algorithm that is common to both sub-programs.

4.3.1 Basic Path-Planning Algorithm

In our basic path-planning algorithm, the first task is to decide on what the desired destination is. We call this the goal node. To select a goal node, the software analyzes the nodes on the map based on a set of parameters and a special cost equation. The parameters and cost equation for selecting the goal node are the key differences between the two sub-programs and will be discussed in detail in Sections 4.3.2 and 4.3.3. Regardless of which method of selecting the goal node is used, one condition remains the same: the goal node can never be placed somewhere that is unreachable. The process of analyzing the nodes in the map is the same for both sub-programs. However, analyzing every node in the 80 by 80 map requires too much time to process. In order to save on cycle time, we designed the program to analyze only certain nodes on the map. These nodes form a trapezoid (See Figure 4.2). In the figure, the black dots represent the space that is not examined by the goal node selection process. The software starts at the robot and begins to analyze the nodes in the trapezoid. Each node whose value is not representative of an obstacle is then entered into the cost equation. By following this process, the first non-obstacle node becomes the

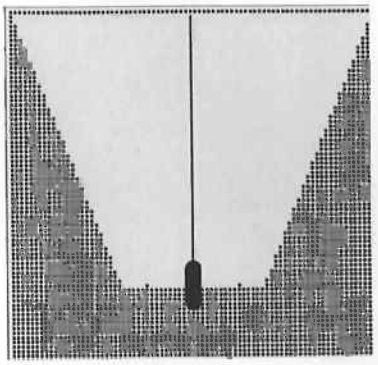

Figure 4.2

candidate goal node. If any node examined after that has a lower cost than the current candidate, it becomes the new candidate goal node. The node with the lowest cost after all nodes have been examined becomes the goal node. This process repeats on every cycle (approximately 10 to 50 ms per cycle) until the robot recognizes that the path is either blocked or it has reached the edge of the map. Anassa V knows that the path is blocked if every node across the trapezoid contains a cost that represents an obstacle. If the software does determine that a path is blocked, it will back up and try to find another path.

The next task in our algorithm is to decide what path the robot must take in order to reach the goal node. Since the program cannot select a goal node that is unreachable, the program is guaranteed to find a path to the goal node. In order to do this we use a process called the "wave front." The wave front algorithm assigns weights to the nodes starting at the goal node. The first group of nodes that surrounds the goal node, and are not obstacle nodes, are given a weight of one. The next group that surrounds the first group is given a weight of two. This continues until the algorithm reaches the position of the robot. In Figure 4.3, "G" represents the goal node.

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	G	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

Figure 4.3

The final task in our algorithm is to find at least one path from Anassa V to the goal. This is done by using a "waterfall" algorithm. There will be a high weight on the node that represents Anassa V's location and a weight of zero at the goal node. Starting at the robot, the waterfall algorithm looks for an adjacent node with the next lowest weight. There are situations where there are multiple nodes with the same weight that are adjacent to the current node. In these situations, the waterfall algorithm will choose the node that is more in line with the direction to the goal node.

The path created by the waterfall algorithm is not a smooth path. The reason for this is that most of the next node decisions made by the waterfall algorithm involve a 45° or a 90° change in direction as seen in Figure 4.3. This does not hurt Anassa V's functionality, but we want the path to be smooth so that Anassa V's movements will be smooth, fluid, and fast. For this reason, we created an algorithm to smooth the jagged edges of the path. This process cuts the corners if possible, but it will not cut corners into an obstacle. By cutting corners, Anassa V is able to move close to objects without hitting them, thereby increasing its overall speed. As you can see, by smoothing the path, we have added another level of intelligence to Anassa V's programming. Figure 4.4 gives an overview of Anassa V's path planning algorithm.

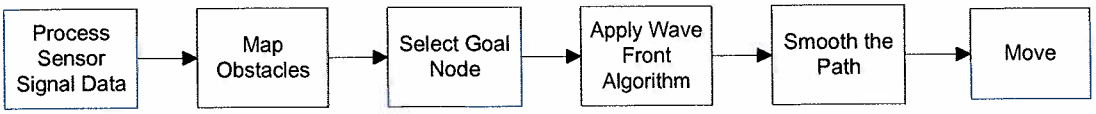

Figure 4.4

4.3.2 Decision Process Based on Lane Following

As stated earlier, the key difference between the two sub-programs is the parameters that emphasize the importance of different aspects of navigation and the cost equation. Our program for autonomous navigation based on lane following uses the following parameters to decide which node should be the goal node: distance, slant, gap, confidence, and straightness. Before we discuss the equation, we feel it is important to understand what each of these parameters mean.

Distance tells us how far a node is from the robot. It is not efficient to always choose a node that is close to the robot's current location as the goal node because when the goal node is close, Anassa V moves slower, and it becomes easier for Anassa V to get trapped.

Slant gives Anassa V an idea as to which way the path is likely to go. The best way to understand this parameter is to think about how a person driving a car knows which way a road turns. When a person drives down the road, he can get an idea which way the road will turn ahead of him by looking at the edges of the road. On the edges of the road, there are lines, tree-lines, and other such markings that tend to run parallel with the road. By noticing the trend that these lines follow, we can anticipate which way the path will go.

Gap is how Anassa V compares multiple candidate paths. For example, if there are two paths that the robot can take, the gap parameter tells Anassa V to prefer the path that has a reasonable gap with markings or obstacles on both sides with a separation of 1 to 3 meters between the robot and the markings or obstacles.

Straightness is a measure of how straight ahead a node is from Anassa V. The robot prefers to go as straight as possible. By including straightness as a parameter, Anassa V chooses the route to a destination with the least amount of turning.

The decision process is the process of finding the optimal goal node based on minimizing a cost function containing these parameters. The cost equation for autonomous navigation based on lane following is shown in Equation 4.1 where A is Distance, B is Gap, C is Confidence of Slant, D is Slant, E is Straightness, and F and G are also parameters that define the Confidence of Slant.

$$Cost = \frac{1}{\{AB\left[1 + Ce^{\left(-.5(D-E)-\left(\frac{F}{G}\right)\right)}\right]\}}$$

Equation 4.1

4.3.3 Decision Process Based on GPS Waypoints

As with the lane-following sub-program, the GPS waypoint sub-program has its own specialized set of parameters. These parameters include: straightness, waypoint pull, obstacles, and range. Each of these parameters is described below.

Straightness is the same as the straightness parameter used in the lane-following sub-program. We use this parameter in GPS waypoint navigation for the same reason described in the lane-following section above – that is, how straight ahead a node is from Anassa V.

Waypoint Pull is used to keep the robot going toward the current waypoint. This parameter is a measure of how much a node is in line with the waypoint. The more a node is in line with the waypoint, the lower the cost will be.

Range is what we use to cause the robot to go the farthest distance it can on its way to the waypoint. In the case of traps and heavy obstacles, we also use this parameter to cause the robot to temporarily "forget" about going to the waypoint and instead focus on getting around the obstacles in the robot's way. Without this, if there is a trap between the robot and the waypoint, it could get stuck because the robot wouldn't know to go around; it would want to go straight to the waypoint. For this reason, we added a variable to count the number of nodes that are marked as obstacles. If this variable goes above a certain threshold, we put more emphasis on the range parameter. Otherwise, we put more emphasis on the waypoint pull parameter.

The cost equation for autonomous navigation based on DGPS waypoints is very similar to the one based on lane-following. As with the other algorithm, the purpose is to minimize the cost associated with the goal node. The equation for DGPS waypoint navigation is shown below in Equation 4.2 where D is the actual distance to the node, A is Range, B is Waypoint Pull, C is the angle between node and waypoint directions, E is the angle between the robot and node directions, F is Straightness, and G is the number of obstacle nodes on the map.

$$Cost = \frac{1}{DA + CB + (90 - |E|)F + G}$$

Equation 4.2

4.4 Output

The last task that Anassa V's software must do is to send speed and angle commands to the motor controller. In order to do that, both the desired speed and direction must be calculated based on the direction of the next node in the path. Speed is calculated based on how close the robot is to any objects. If Anassa V is in a situation where there are many obstacles close to it, then it will not go fast so that it can make tighter turns. If there are no obstacles around, then Anassa V will choose to go as fast as possible toward the goal. To calculate direction, the software uses the vector from the robot pointing in the initial direction of the planned path. Angle is then computed using Equation 4.3. The parameters Δx and Δy are determined as shown in Figure 4.5. Once both speed and direction have been calculated, the computer sends the information to the motor controller which in turn sends the correct voltages to the motors.

$$\tan^{-1}\left(\frac{\Delta x}{\Delta y}\right)$$

Equation 4.3

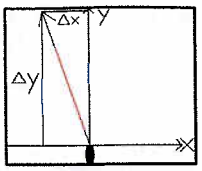

Figure 4.5

4.5 JAUS

Our team is currently studying OpenJAUS and other documentation obtained from www.jauswg.org in order to implement the correct JAUS header. We have implemented the functionality to send and receive commands using UDP packets on port 3794. These commands are interpreted by Anassa V to perform the desired task.

4.6 Safety and Reliability

It is the intent of our team to make Anassa V's code safe and reliable. There are three main ways that the software accomplishes these tasks. First, the code provides safety in that when Anassa V detects obstacles within 0.6 meters, the hazard light will flash. Secondly, in the event the computer malfunctions, the microcontroller that governs the motors will not allow the robot to move until the computer resumes normal function. Lastly, the code is designed to be able to handle any circumstance. For example, if there is an exception thrown by the software, the code is able to catch the exception and the system will not be forced to shut down.

4.7 Post-IGVC Analysis and Solutions

The Post-IGVC Analysis contained three items relating to software listed below.

- Anassa IV's inability to handle glare

Problem - Due to the angle of the sun at certain times, light that was being reflected off objects on the course was filling the camera with white. Since white is the color of boundary lines, the completely white image caused Anassa IV to think that the glare was a forbidden area. This caused unpredictable and undesirable results, such as 180° turn-arounds.

Solution – We implemented two solutions to recognize and handle glare. First, we used the camera's built-in glare removal utility. Second, in the software, we used the knowledge that no obstacle is "big and white." If Anassa V sees such an object, it is ignored.

- Anassa IV's inability to identify 180° turn arounds

Problem – Occasionally, Anassa IV would drive into a situation that would cause it to turn around and go in the wrong direction. When these situations occurred, Anassa IV would not know to turn back around to go in the right direction.

Solution – Anassa V uses GPS markers to identify where the robot has already been. If the robot approaches one of these markers, it will recognize that it is going in the wrong direction.

- Anassa IV's deficiencies in navigation based on GPS waypoints

Problem - Anassa IV was unable to complete the Navigation Challenge due to deficiencies in the GPS waypoint sub-program.

Solutions – Since Anassa IV was very successful in the Autonomous Challenge, we decided to make the waypoint navigation sub-program as similar to the lane-following sub-program as possible.

5. Design Innovations

Throughout this paper we have discussed various design innovations. In this section we will list both hardware and software innovations.

5.1 Software Innovations

- Adaptive color vectoring – Section 4.1
- Autonomous Navigation based on lane following and autonomous navigation based on GPS waypoints only differ in their cost equations for selecting a goal node – Sections 4.3.2 and 4.3.3
- Selecting a goal node by minimizing a cost equation with respect to weighted parameters – Sections 4.3.2 and 4.3.3
- Wave front algorithm defines all paths to the goal node – Section 4.3.1

5.2 Hardware Innovations

- Using pre-engineered materials such as the wheelchair bottom and the video camera – Sections 3.1.3 and 3.2.2
- Centralized power system with only two batteries which are easily charged and/or replaced – Section 3.2.1
- Two independent charge methods, trickle charge and fast charge – Section 3.2.5
- High Performance Computer – Section 3.2.3

6. Systems Integration

6.1 Overall System Feedback Loop

The entire structure of Anassa V's hardware and software is based on a feedback loop. At the start of every cycle, the sensors give the robot information about the environment. Based on this information, Anassa V makes a decision to move in a certain direction. Then at the start of the next cycle, the sensors tell

the robot what actually happened based on the move commands that were executed in the last cycle. Figure 6.1 illustrates this feedback loop.

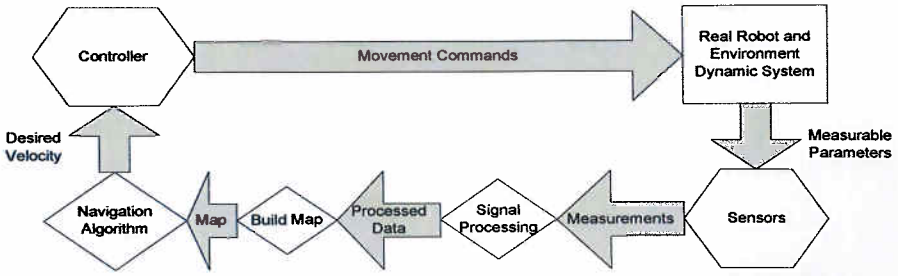

Figure 6.1

6.2 Example

In this section, we will discuss a scenario that will illustrate the systems integration of Anassa V. This example will be based on DGPS Navigation. We chose to do the example this way since our implementations of navigation based on GPS waypoint and navigation based on lane following are similar in almost every respect.

Figure 6.2 was taken from a simulator that was programmed by the BSC Robotics Team in order to test new software ideas. This image represents a global view of a simulated IGVC course. The robot is represented by the blue oval. The red shapes represent obstacles and the black circles represent DGPS waypoints. The program simulates the following devices: one forward mounted LMS unit, a compass, and a GPS unit. The GPS coordinates are determined at run-time based on the GPS coordinates of the waypoints. The simulator uses the same functions that the real program uses to make decisions. The only data that is simulated is the sensor data. Figure 6.3 was also taken from the simulator. It shows what the robot's sensors are reading. The image represents the local map discussed in Section 4.2, Mapping the Inputs. Obstacles are represented by black nodes, free space is represented by grey nodes, and the robot is represented by the black oval. The black line represents the "distance vector." The red node in the upper left-hand corner represents the goal node.

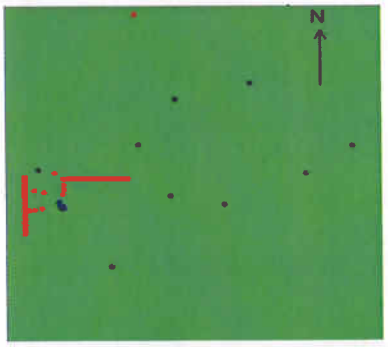

Figure 6.2

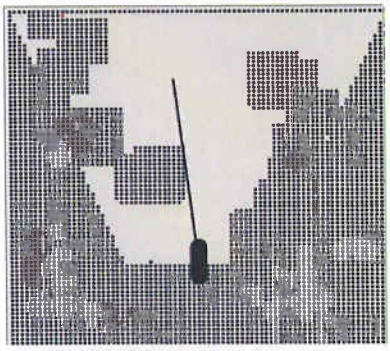

Figure 6.3

From Figure 6.3 we can see that the velocity vector represents the initial segment of the planned path. The smoothing algorithm causes the path to run as close as possible to the obstacle instead of out and away from it. This whole process is repeated every cycle.