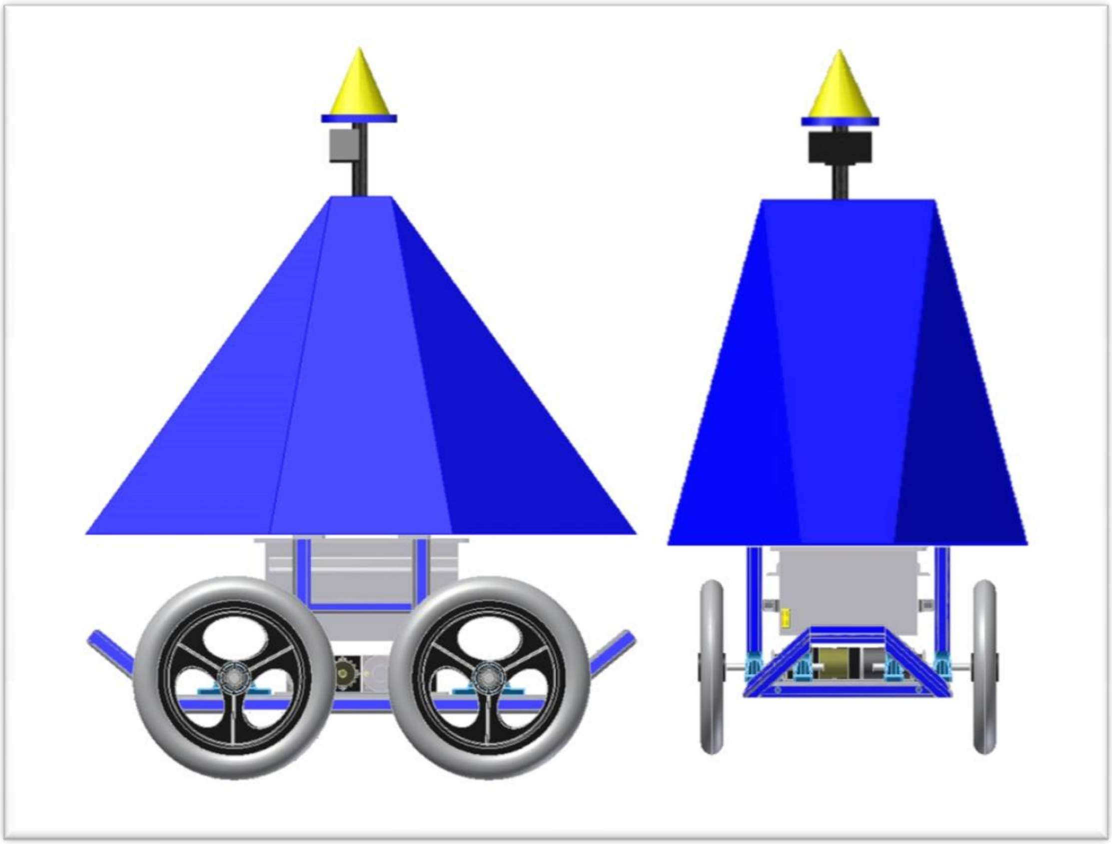


ARCHON DESIGN REPORT 2009

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

I, Dr. Robert Riggins, Professor of the Department of Electrical Engineering Technology Department at Bluefield State College do hereby certify that the engineering design of Archon has been significant and each team member has earned at least two semester hours credit for their work on this project.

Signed, Date

Phone: (304) 327-413 E-mail: briggins@bluefieldstate.edu

1.0 Introduction

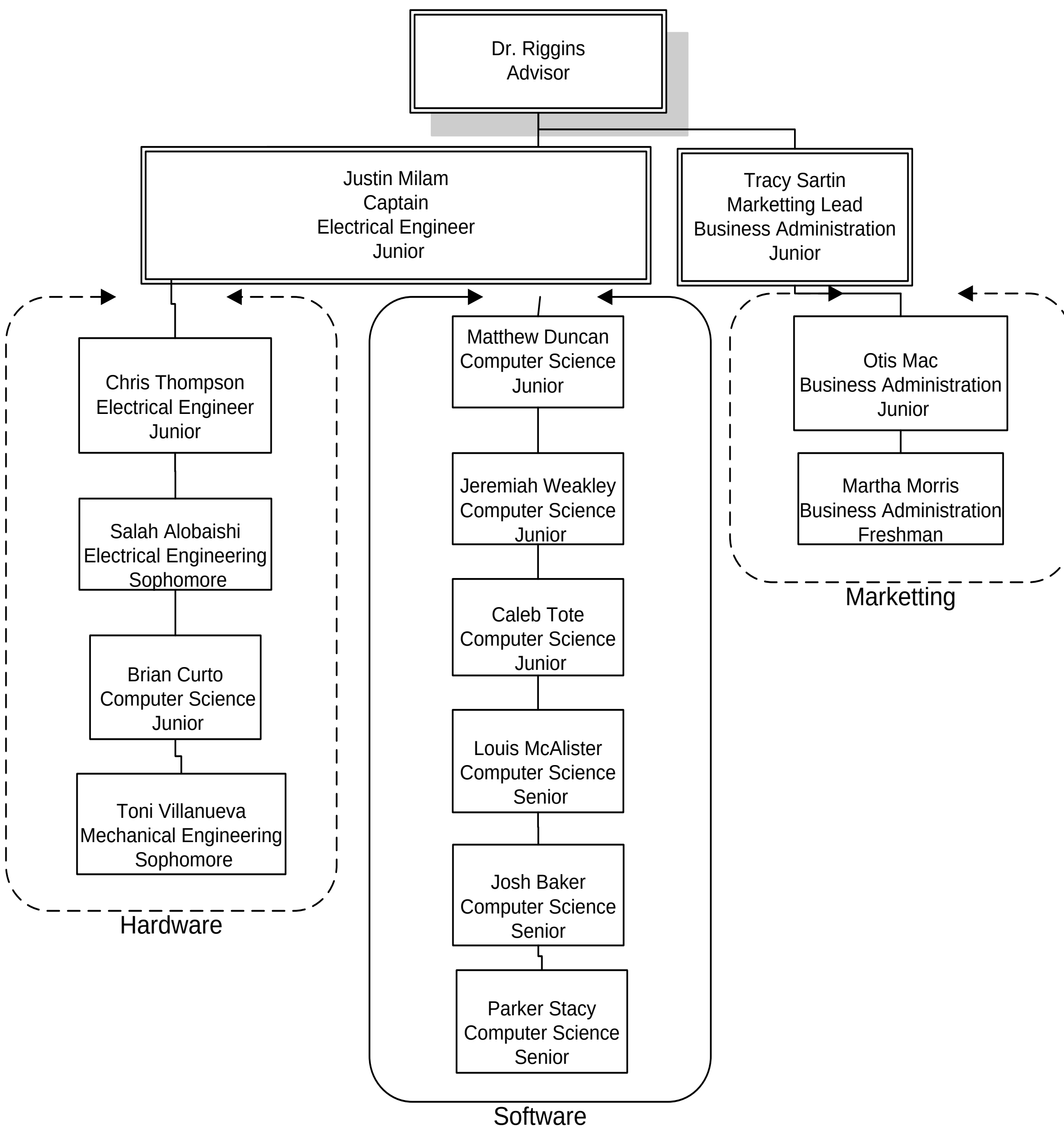
The Bluefield State College (BSC) Robotics Team is pleased to announce the creation of our newest robotic system called Archon. Archon was created with the aim of addressing the limitations of our past robots while using what worked to its fullest potential. We have named our robot Archon, which means prince in Greek; we believe it is a worthy successor to our previous robots known as the king and queen.

This latest project has been designed from the ground up. Unlike past iterations based on modifying preexisting devices such as electric wheelchairs, Archon has been designed completely by the BSC robotics team. This allows for features and controls not available from existing commercial vehicles as well as the ability to repair and service the new robot easily. Truly Archon is a unique machine, and this report will describe this amazing robot and how it came to be.

Fig 2.1 Team Organization

2.0 Team

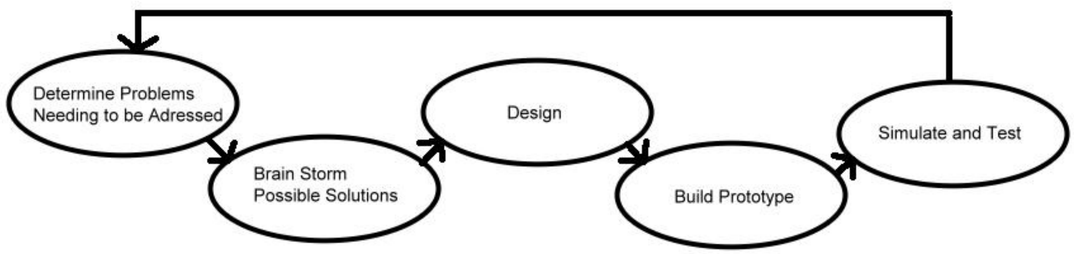
The robotics team contains a large variety of students from every engineering and computer science degree offered at BSC. While many decisions are made collectively as a group there does exist a team captain who has the final say on any changes made to Archon. For a full list of the students that helped in the creation of our newest robot, please see Figure 2.1.



3.0 Design Process

When we began to plan the design of Archon we knew there would be many challenges ahead. We would be designing a robot from the ground up for the first time, but thanks to our experience with past robotics systems we would have a good foundation on which to build. As we try to advance the robotic systems to the next level, it is crucial that we look at what issues arose from past generations of IGVC robots. This allows the former generations to act as a proving ground for many of the functions used in this newest model. This design process is depicted in Figure 3.1.

Fig 3.1 Design Process



3.1 Evaluation

After IGVC 2008, the team conducted a complete analysis of our 2008 robot Anassa IV. We determined what worked well and what could be improved. This led to a good idea of what issues should be addressed while designing Archon and its systems.

3.2 Issue Definition

Now we will describe the first two blocks of the Design Process in Figure 3.1 above. As a result of our evaluation, there were a number of issues to address when planning the new robot. In this section we address the most important of these issues.

3.2.1 Reverse functionality

While our previous robots did possess the ability to move in reverse, they never could move as easily backwards as they did forwards. This problem led to difficulties maneuvering through certain obstacles. We sought to make it just as natural for Archon to move in either direction, which would allow for smoother movements and easier control. What resulted was the idea of symmetry, having no noticeable physical difference in the robot from front to rear.

3.2.2 Complex Control Interface

The overall control scheme for our former robots has been complex from a mechanical point of view as well as from a programming point of view. To initialize the system, a precise sequence of switches would have to be pressed in a set order. That was followed by the use of many separate computer programs; each needed to ready the robot for use and to run the full array of sensor and control operations.

The solution to this was to allow the computer to control more of the electrical system through a single integrated program. Through this system all the necessary components can be controlled from a single keyboard, with no need to jump from one program to another to completely control the system.

3.2.3 No Internal Diagnostics

The prior generations of BSC robots possessed no way to diagnose a problem within their systems, and no way to check the status of sensors or other external devices. This led to problems during testing as the team would not know an error existed until a full test run was made. There were also no checks against external forces that could influence the robot's performance.

This problem could be addressed with the added control given to the computer that was discussed above. Allowing the computer to control the system and monitor its function would allow for a system of checks and balances to look for errors almost instantaneously. Thus we are creating a system that will know when it is running at peak efficiency or if an issue has occurred.

3.2.4 Limited User Communication

Once the previous robots were running the IGVC course, the only way for the team to know the robot's actions and intentions was to simply observe and/or record its behavior. This would lead to those in the team scratching their heads on the sidelines when the robot did something unexpected.

By adding a more versatile user interface the observers of Archon will not be as "in the dark" about what Archon is thinking and doing during operation. With new features such as a multi-faceted light system and a sound system capable of speaking to indicate status, it will be far easier to discover and address problems that may arise during operations.

3.2.5 Outdated Control Code

While the original control codes for our robotic systems is quite a feat of engineering from previous teams, years of adding and removing pieces has caused a build-up of code that is no longer streamlined and efficient. The Archon team is in the process of replacing older programming code with a new code started from the ground up. This will allow for only the essentials to be included and new features to be easily integrated.

3.2.6 Physical Limitations

Many past BSC autonomous vehicles have been based on an electric wheelchair design. This design is extremely functional but has certain limitations such as the non-symmetry of the base, which causes an inability to turn 360 degrees with ease, and places the center of gravity too far forward. By designing and creating our own base for Archon, we can have a drive system that better suits our unique needs (see Section 4.0, Mechanical Design).

3.3 Development of Design

Once it was determined what changes would need to be made to create a better robot, the team began to determine exactly how to best address these issues in order to establish Archon's design (third block of the Design Process of Figure 3.2.) The earliest steps taken in design were simple brain-storming sessions among members of the BSC robotics team. We discussed possible solutions for the problems identified above, and tossed around ideas for what we would like to see implemented in the new robot. As we talked and debated, we gradually settled on a design plan that eventually lead to an early draft of the new robot's appearance and functionality.

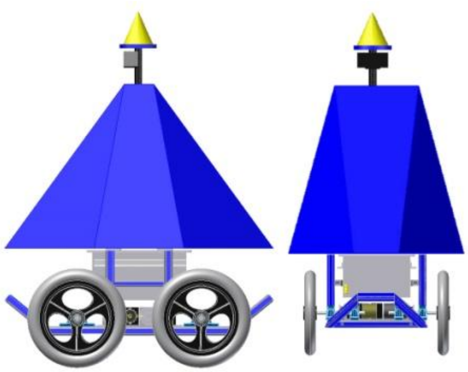


Fig 4.1 Archon Frame Computer Design

4.0 Mechanical Design

Archon's new physical design is radically different from our previous IGVC robots. It combines form and function to maximize the ability and versatility of its systems. We set out to create a very compact robot that still possesses ample space for all needed equipment. Many different designs were discussed and this led to a pyramid-like shape being used as the basis for the Archon frame. The pyramid shape allows maximum stability despite a 6-foot high mast, allows maximum maneuverability in tight situations, allows the sensors to see all around the robot, and allows us to use the same software for both forward and backward motion.

4.1 Computer-Aided Design

Once a first-iteration design decision was reached for Archon's shape and functionality, the team used computer design programs such as AutoCAD and Autodesk Inventor to better show how the robot would appear from different viewpoints. This also allowed the mechanical engineering students to do stress and vibration analyses on different body shapes to find ones that could take the most strain, while satisfying all our physical design objectives.

4.2 Vehicle Frame and Chassis

The most notable feature of Archon is the symmetry of the entire frame. This complements the drive system by allowing the robot to move fluidly forward or backward. It places the center of gravity low and exactly in the horizontal plane center. Archon is now capable of making a 360° stationary turn with its centralized wheel system. Due to the robot's camera/GPS mast being located at exact center, Archon can now move its camera to any horizontal angle and have no part of the chassis blocking the line of sight. Archon can see backwards with an identical view as what it sees moving forward.

Our new body design is also completely modular. This makes it easy to separate the top camera mast from the control section as well as to separate the control section from the wheel base. This new modularity also makes it possible to change the actual wheel base. We currently have many wheel base designs that can function differently to suit whatever challenge we may face. These include four wheeled systems, track systems, two wheeled systems, and systems based on a wheelchair design. For the 2009 IGVC we plan to use a 4-wheel drive base using skid-steer. We designed the gears to make Archon's top speed to be 5 miles per hour, and designed enough torque to drive archon's 250-pound projected weight up a 20-degree incline at full speed.

4.3 Materials

Archon is constructed from a mix of materials to fit each unique piece. The control module at the top as well as the camera mast is made from aluminum. Using a light yet incredibly durable material like aluminum is perfect for providing a protective shell around Archon's more intricate parts. As aluminum is non-magnetic, it also lowers any chance of the frame interfering with electro-magnetic sensors such as the compass.

In order to take the added strain and weight of the motors, the wheel base has been constructed of steel tubing. This allows the frame to support the weight of the heavier components such as the battery and to take the torque of the motors with ease. The added weight of the steel keeps the center of gravity low on the robot to prevent tipping. Rubber bushings are also being utilized between the control module and wheel base to reduce any vibration.

5.0 Electrical Design

We've taken many of the electrical designs tested in our earlier robots and used them to their fullest potential in the Archon system. The new compact size of Archon's body offered interesting electrical challenges during the design process, but this has led to a very ergonomic and functional design.

5.1 Power System

As mentioned before, Archon's frame is the smallest we've ever created. This means we do not have the same amount of space available for a large battery system. Such a problem led to the need for a new way to look at our power source. We replaced the large two battery system with a smaller single twelve-volt battery. An upgrade has also taken place in the type of battery we use. In the past we have made use of the same kind batteries you would find in any car. Now we use a single commercial grade battery (rated for 75 amp-hours), more often found in large trucks. This allows for a longer battery life.

Going to a new battery type is a definite improvement, but this still is less than the total amp hours we had available in older generation BSC robots which used a two-battery system. To compensate for having less battery life, we designed the frame such that the single battery can be removed easily and quickly to replace it with a fresh battery. A battery rail system was developed for Archon to make this battery replacement process easy. The battery, while protected in an aluminum box, can now be removed quickly from the wheel base.

5.2 Wiring

Archon's wiring system has many new features. As mentioned earlier, we have made an effort to create an intelligent electronics system. Now almost all electronic devices are wired to a microprocessor-controlled relay system. This allows the robot's computer to directly control the power operations. The robot can turn sensors and components on as needed instead of this being performed by an operator with switches. Also included is intelligent battery monitoring, so the robot will know if the battery is getting low before a power loss failure occurs. Like many past generations of BSC robots, we use a quick-disconnect wiring system to make the changing of parts easy.

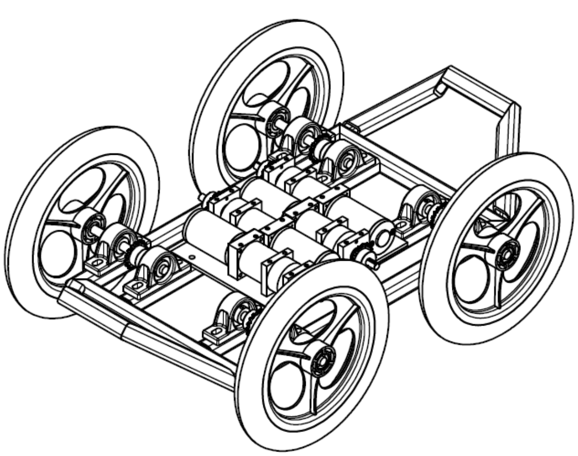


Fig5.1 Wheel Base Design

5.3 Actuators

With the modularity of Archon we can use a large range of different motors. Our current configuration make's use of four twelve volt DC motors operating in a skid steer set up. A system of motor encoders are used to detect the speed and direction of each motor, allowing the robot to compensate for slipping or other motor issues. We have calculated that this motor design allows the robot to perform at five miles per hour.

5.4 Safety

Safety is always a major issue with our robotic systems and with Archon this is no different. With both hardware and software methods of disabling the robot, we believe Archon is an incredibly safe machine. A new key-based hard E-Stop system is the most definitive way of shutting down the system by manually cutting the batteries from the system. The key-based system also adds a level of safety to the maintenance level of the robot, allowing anyone working on the electronics to shut down the system and remove the key, thus preventing the system from energizing. This was inspired by the “lock out, tag out” standard for working on electrical systems. There is another easily accessible manual E-Stop called the soft E-Stop which cuts communication from the control system to the drive control, effectively stopping the robot while retaining power to the system. A radio-controlled (RC) wireless E-stop is also incorporated into the control system to allow an operator to stop the robot from a long distance. Large numbers of fuses and breakers have been placed to prevent any harm to equipment or team members.

Archon has the ability to self-detect many system errors, and control the robot appropriately according to those detected errors. Archon uses two methods to do this. One method is the software program in the main computer that continuously monitors the performance of all of Archon’s components. The second method stems from distributed computing with various microcontrollers that Archon uses. The microcontroller-based drive structure has the capability to evaluate the commands coming from the computer and stop the robot if the commands are no longer recognizable due to a computer error.

5.5 Instrument Panel for Monitoring and Control

Archon’s system for monitoring and control is a large shift from the switch-based control systems used in our previous robots. Through a touch screen monitor we are able to control and evaluate the entire robot using a versatile software system that can easily adapt to any changes or new features. In previous years, to add a new component you would have to drill holes, mount switches, and run new wiring to the switches on the control panel, but now it is as simple as a few clicks of a mouse. This new system allows for more information to be given to an operator as well as many more control options.

5.6 Other Communication methods

Archon possesses an unprecedented number of indicators to correctly convey information back to an observer. A complex light system is being implemented to convey such information as direction, mode, errors, and the power state of assorted devices. With the use of a Bluetooth headset it is also possible for the computer to directly speak to an observer, conveying information such as battery level and computer errors.

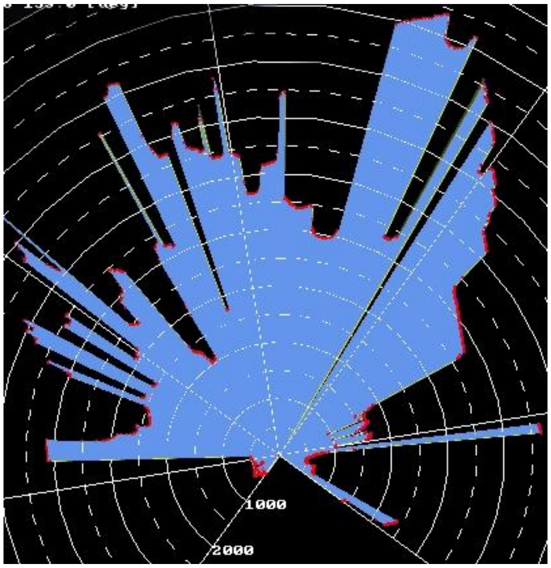


Fig5.2 LMS Sweep

5.7 Sensors

The following is a list of Archon's sensors and a brief description of their function:

CSI Wireless DGPS receiver and antenna: Retrieves the latitude and longitude.

Two Hokuyu 270-degree LMS: Sweeps 270 degrees for object avoidance.

Maretron Solid State Compass: Determines Archon's heading.

Sony HandyCam Camcorder: Readily available and pre-engineered vision system

Archon's sensors are placed in strategic positions around the robot so that they provide a full field of vision. A 270° Laser Measurement System (LMS) is placed on each end of Archon, giving the robot complete LMS coverage. The camera direction can now move to point anywhere on the 360-degree horizontal circle, giving vision coverage all around the robot. These allow for Archon to have complete sensor coverage.

5.8 Computer

The computer in Archon is small yet very functional. A HP TouchSmart tablet PC is used due to the small size, allowing it to fit well in Archon's control module. Even with its small size there is more than enough performance to meet our goals. The built-in tablet functions also allow easy integration of an additional touch screen.

6.0 Software Design

Our software design for Archon has been based on previous generations' successful algorithms. This allows for a high level of familiarity with the code process. The program is created in Microsoft Visual Basic 6.0 because it is easy for any student to learn and runs well in our PC computer environment.

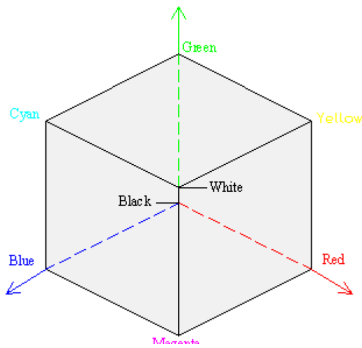
6.1 Inputs

The inputs to our software program are the data sent from the external devices which includes the compass, camera, GPS unit, and twin LMS units. Archon makes all of its decisions based on the data that it receives from these devices. This data must be interpreted into a form that Archon can use to make decisions concerning the environment around it.

The GPS unit sends its data to the computer in the form of GPS “sentences”. Archon reads this data via a RS232 cable. The robot interprets this information in order to get the latitude and longitude data for the current location of the robot.

The compass also sends positional data. This data contains the heading of the robot in degrees with a precision of 1°**.** This data is interpreted in the same way GPS data is evaluated.

The LMS sends its data as an array of integers. The array contains 270 elements which correspond to the 270° view of each Laser Measurement System. The number that is in each array position is the distance in millimeters to an observed obstacle at that degree.



<u>Fig 6.1 Color Vectors</u>

How camera data is brought to the program is the most difficult data to interpret compared to data from other devices. We first had to install a separate program that will retrieve frames from the camera and save them as images so that we can analyze the content. The program uses a process we call “color vectors” to analyze camera data. See Figure 6.1. Certain colors correlate to different obstacles. For example, the lines that mark the boundary of the course are white, so our program recognizes white as a barrier that cannot be crossed. The robot also recognizes different colors as either okay or restricted based on the rules set by the IGVC.

Once Archon has all the data pertaining to obstacles, the information is used to select the best path around them. In order for Archon to work with this data, it must be converted from its varying formats into one consistent style that can be used to generate a map of the robot’s surroundings.

6.1.1 Mapping of Inputs

In our program we use a “map” to represent the obstacles logically. This map is a two-dimensional array of integers that measures 80 nodes by 80 nodes. Each node in the array represents about four square-inches of real-world space. This size was chosen because it is approximately the size of the smaller features on the IGVC course. After the robot gets the input from each of the sensors, Archon places the information on this map. The computer then stores obstacles as a cost value on the map. If a node has a high cost, the robot knows that it cannot go through that node. The cost assigned to a node depends on what sensor brought that data in. For instance, the LMS data has a higher cost than the camera data because LMS is more accurate than camera data. Once all the data has been placed on the map, Archon begins the process of choosing a path that will get it around the obstacles most efficiently.

6.2 Autonomous Navigation

Once all of the data has been mapped, the process of deciding which path to take begins. We are currently using two slightly different algorithms to make these decisions; one for autonomous navigation and one for GPS navigation. Each algorithm is used in the GPS navigation or autonomous course navigation at IGVC so that the best algorithm is utilized on each course. We will discuss the algorithm for lane-following autonomous navigation first.

6.2.1 Parameters

The program that we created for planning a path involves several different tasks to analyze the map and make the best decision. The first task is to set a goal that tells the robot what point it should try and move towards. Archon use's several different parameters to make this decision. These include: distance, slant, path width (which is called gap), confidence, and straightness. Before we discuss the process, we feel that it is important to understand each of these parameters.

The distance parameter tells the machine 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, Archon moves slower. We will discuss later why this is the case.

The slant parameter gives Archon 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, you can get a good idea which way the road will turn ahead of you by looking at the edges of the road. On the edges of the road, there are white 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 tell which way the road curves.

This slant parameter and confidence parameter work together so that Archon will know whether or not to trust the slant parameter. The confidence parameter is determined by the total slant of the map. If the majority of the objects on the map are slanted in one direction then the confidence will be high. However, if the objects seem to be slanted different ways or there are more "blob-like" objects on the map then the confidence will be low.

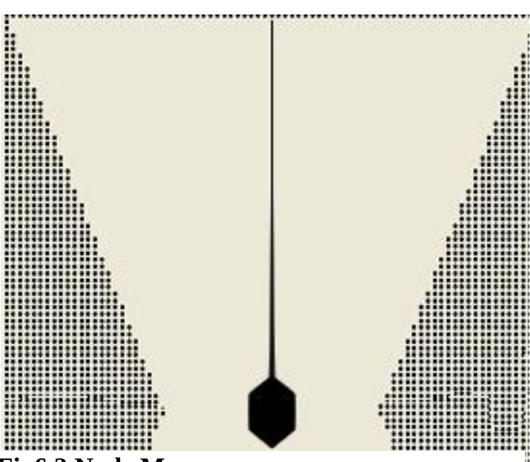


Fig6.2 Node Map

Gap is how the program compares multiple candidate paths. If there are two paths that Archon can take, the gap parameter tells Archon to prefer the wider path.

Straightness is a measure of how straight a node is in front of the machine. Archon prefers to go as straight as possible. By giving a value to straightness, Archon keeps from veering off course or possibly choosing a sub-optimal path.

6.2.2 Control Decisions

Now the decision process begins. To set a goal node the robot must examine the nodes in its map to decide which one is the best candidate to be the goal node based on the parameters described above. However, to examine every node on the map requires a large amount of processing time. In order to save time we limit the nodes that are examined. Since we always want Archon to move forward, it does not examine some of the nodes that are on the edges of the map. Instead, the nodes it examines form a trapezoid in front of the robot. See Figure 6.2. The most important part of setting the goal node is that the goal will never be where the robot doesn't have a path. Each node in the trapezoid is examined starting at the bottom left corner and moving to the top right corner. Each node whose value is not representative of an obstacle is then entered into the following cost equation.

Equation 6.1 Cost Equation for Autonomous Navigation

$$Cost = 1/\left\{A * B * \left[1 + C * e^{(-.05*(D-E)-(F/G))}\right]\right\}$$

Thus the first open node is made the candidate to be the goal node. If any subsequent node produces a lower cost than the current candidate it is then made the goal node. This process continues until the robot recognizes that either the path is blocked or it has reached the edge of the map. Archon knows that the path is blocked if every node across the trapezoid contains a cost that represents an obstacle.

6.3 GPS Navigation

The program that we created for planning a path to a GPS waypoint is very similar to the autonomous program with a few key differences. The differences are the parameters used to assign a cost to each node and the cost equation. Archon uses the following parameters for GPS Navigation: straightness, waypoint pull, obstacles, and range. The straightness parameter is the same as the straightness parameter from the normal autonomous program. We use this parameter in GPS navigation so Archon would tend to move with the least amount of turning.

The waypoint pull parameter is used to keep the robot going toward the current waypoint. This factor is a measure of how in line a node is with the waypoint direction. The more in line a node is with the waypoint direction the higher the value.

The range parameter is what is used to cause the robot to go the farthest distance it can on its way to the waypoint. This parameter is also used to cause the robot to temporarily "forget" about going to the waypoint and instead focus on getting around the obstacles in its way.

Due to traps on the course 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.

The cost equation that we use is very similar to the one used in the autonomous program. Each node is evaluated based on the parameters listed above. The number of nodes is again limited to save on processing time. The robot continues examining nodes at the bottom left and finishes at the top right corner of the trapezoid to select the goal node. The equation for GPS navigation is shown below in Equation 6.2 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.

Equation 6.2: Cost Equation for GPS Navigation

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

Once the goal node is assigned, the program uses the same method to determine the path and then calculate the angle and speed to make the next move toward the goal.

6.4 Path Following

Archon's next task is to decide what path the robot must take in order to reach the goal. Since the program will not select a goal node that is unreachable, this part of the program must determine the best path to get to that goal.

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

Fig6.3 Goal Weights

The first task performed by the path planner is a process we call "wave front". This is an action where Archon assigns weights to the nodes starting at the goal node. The first group of nodes that surround the goal node are assigned 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 6.3, G represents the goal node.

The second task is to decide which path to take. This is done by using what we call a "waterfall" algorithm. There will be a high weight at the robot, and a weight of zero at the goal node. Starting at the robot, the path planner looks for the next lowest weight. There are situations where there are multiple nodes with the same weight. In these situations, our algorithm will choose the node that is more in line with the direction to the goal node.

At this point, the algorithm has determined a path from the robot to the goal node. Now it makes this path as smooth as possible. For this we have written code that will smooth the corners. Normally, the path created is very choppy because most of the next node decisions will involve 90 and 45 degree changes. This does not hurt the functionality, but we want Archon's movement to be smooth and fluid. Our program will smooth this path if it can. It will not cut the corner into an obstacle. By smoothing the path, we add another level of intelligence to Archon's programming.

6.5 Outputs

Once a goal has been selected and a path has been decided on, all that is left to do is to start moving. Our program calculates the speed and direction that is necessary to move to the first node on the path to the goal. Speed is calculated based on how close the robot is to any objects. If Archon is in a situation where there are many obstacles close to it, then it will travel slowly to improve maneuverability. If there are no obstacles, then Archon will choose to go as fast as possible toward the goal. The direction is calculated by interpreting compass data with environmental sensor data such as the LMS. Once both speed and direction are calculated, the computer sends the information to the motor controller which in turn sends the correct voltage to the motors.

.

6.6 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 Archon to perform the desired task.

7.0 Design Innovations

This new robotic system is innovative from top to bottom. Every aspect of Archon - from mechanical, electrical, and computer systems - have been revolutionized. Many of these innovations are discussed above. What follows is a list of some of the most important innovations integrated into Archon.

7.1 Symmetry

As mentioned earlier, the robot is externally symmetrical to allow an identical sensor viewpoint from both the front and rear. With the combination of twin laser measurement systems and the new camera swivel system, Archon can easily travel in any direction. This also adds an aesthetic layer to Archon's form proving how something can look good while being functional.

7.2 Modularity

With the modular system there are a near limitless number of combinations of parts and systems. Changing the wheel base module allows Archon to travel through nearly any terrain. Changing camera angles in two dimensions while running allows the robot to see the best possible information in a myriad of ways.

The modularity also adds a level of ease to the maintenance and replacement of parts. We no longer need to solder or weld in replacements, rather we just quickly add them using simple connectors. This greatly cuts time and costs to maintain the robot. With this level of modularity, Archon has been created in a way that it is very upgradeable.

7.3 Controls

The control system has perhaps changed the most from our earlier robotic models. Previously we were largely dependent on prefabricated wheel chair controls that we could not repair ourselves. Furthermore, our control interface to the wheelchair controller was complicated and non-standard. Now the controller is completely built and programmed by BSC students. This allows us to troubleshoot the control process with an intimate knowledge of every part of the system.

Archon also uses full wireless control. An RC receiver is built straight into the control module allowing a wireless alternative to operate the robot without starting the computer. There is also a voice feature that allows not only a wireless, but a completely hands-free way of driving the robot. It’s as simple as audibly telling Archon where to go and how fast you want it to get there.

Even the manual control interface has been greatly improved with the touch screen-based computer system. Now a single screen replaces a panel full of control switches while also providing meaningful output to the operator.

7.4 Mobile Sensors

Archon’s ability to turn its head seems like a simple addition, but it leads to massive improvements. Combined with our 270° laser measurement systems the robot can now see in any direction. This allows for Archon to turn and look in the direction it is headed or to track an obstacle or line to better avoid it.

7.5 Drive System

Our four-wheel drive system now allows Archon to move in a way not seen in our previous robots. The ability to control each drive wheel independently allows for a turning radius and speed that was previously unattainable. With the inclusion of encoders measuring motor speed, we also know exactly what the motor is actuallydoing, instead of assuming they are acting as we predict.

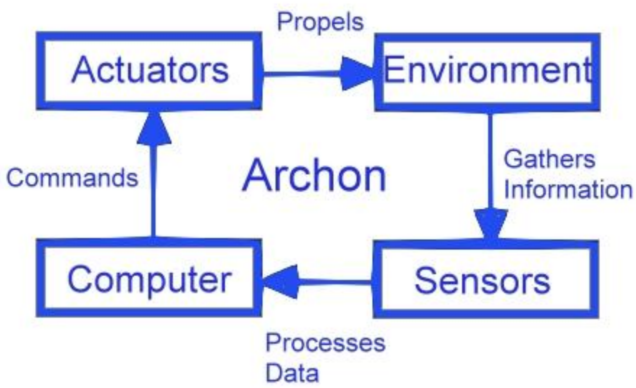


Fig 8.1 System Cycle

8.0 System Integration

As discussed throughout this report we use a large number of sensors to gain information, processors to evaluate that information, and actuators to then use that information to effect the environment. How it all comes together is the heart of our robotic system.

Data from the sensors is carried to the computer by communication wires using USB or serial ports. This is then brought into the robot's software and processed to render the appropriate maps. This map data is next evaluated by Archon's program, and a path through the environment is determined. Information is then sent once again through communication wires from the computer to the motor controller, where it is evaluated and compared to previous commands and current motor status. Once the correct actions are determined, the controller sends the commands out as pulse width modulations to the actuators. Movement of the actuators causes the robot to move in the desired direction. This changes the environment, which affects what the robots sensors sense. The whole process starts over and continues until a goal is achieved. Thus the robot acts in a cycle of sensing, assessing, and action.

9.0 Conclusion

The creation of Archon has been a year-long process with countless iterations and prototypes existing before the result discussed above was finally reached. Archon is still and will probably always be a work-in-progress. As technology and the skills of the Bluefield State Robotics Team improve, Archon will be able to do things far beyond its current state. What we have created is a very strong and capable foundation on which to grow. Thank you for taking your time to read our report on "the prince" Archon. I hope you came out of it with an idea of what we are trying to accomplish and perhaps even learned something new.