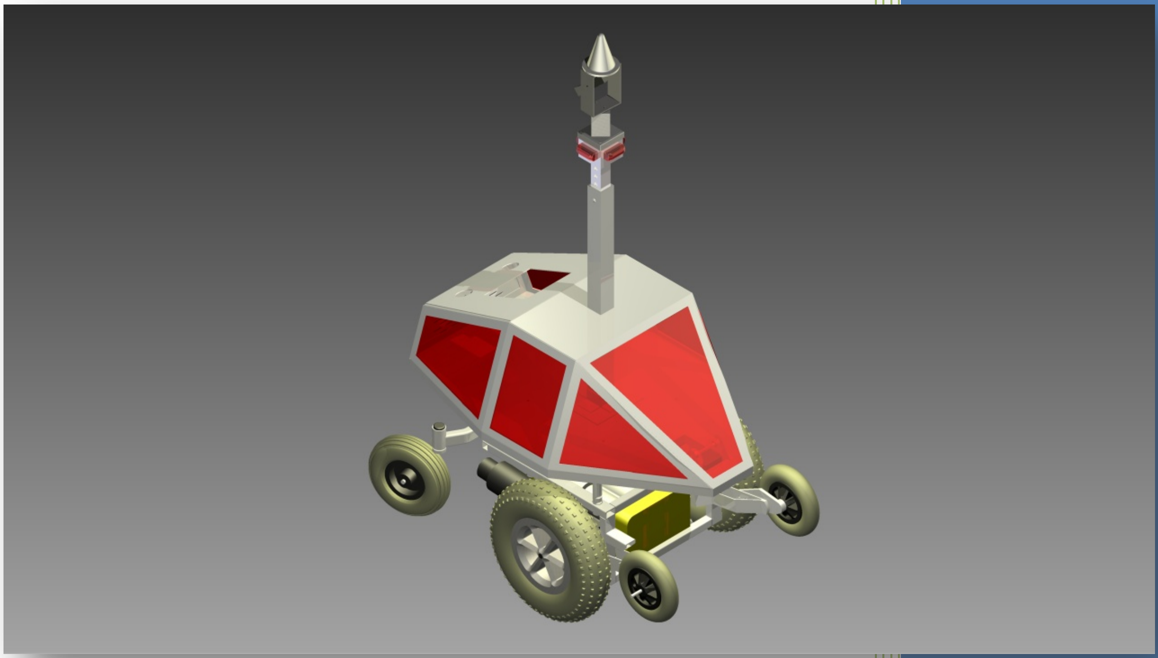
IGVC 2012

Design Report



Faculty Advisor Statement:
We certify that the engineering design of *ARES* has been significant and each student, undergraduate only, has earned at least five semester hours credit for their work on this project.

Dr. Riggins/Dr. Ozyavas

1. Introduction

For the 2012 Intelligent Ground Vehicle Competition (IGVC), Bluefield State College (BSC) presents its newest IGVC robot ARES. The BSC team created ARES by taking advantage of the proven previous successes of BSC’s IGVC robots over the past twelve years. Many successful engineering feats have been accomplished by combining proven components from other projects. This year’s team has created ARES by building on that principle. ARES uses a proven wheelchair base that Anassa, one of BSC’s previous IGVC robots, used in IGVC 2009. ARES also uses the highly successful body frame from Vasilius, also one of BSC’s previous robots, used in IGVC 2004. The superior camera and GPS mount comes from Archon, BSC’s IGVC 2011 contestant. By using these components from robots of successful teams in the past, and making modifications and improvements on these components, ARES will be BSC’s strongest contender to date. The success of developing this year’s robot is attributed to the dedicated, multi-disciplinary team of undergraduate students who all share the same goal of designing an autonomous robot that will meet the challenges of this year’s IGVC.

Figure 2.1: Anassa (2008-2009)
Archon (2009-2011)

2. Design Process

2.1. Design Method

Designing an entry for an annual performance-based competition such as the IGVC is an exercise in continuous improvement based on the lessons learned from years past. For example, BSC’s IGVC 2008/2009 robot, known as Anassa (Figure 2.1 top), was BSC’s most successful autonomous robot to date, placing third overall in IGVC 2009 and first in the autonomous challenge in IGVC 2008. In 2009, the base of Anassa was a modified rugged outdoor 1170 Jazzy wheelchair, including the original wheelchair motors and controller. Because of the success of that base, the ARES team decided to use a modified form of Anassa’s base in this year’s competition robot. Another example is the body BSC used on the 2004 IGVC robot Vasilius. That body was well-designed, having ample internal real estate for components, allowing easy access for maintenance, using a convenient control panel for operation, and sporting a nice sleek look. Consequently the ARES team decided to use a modified version of that body on ARES. A third example is the mast and GPS antenna and camera mount used on the 2011 IGVC robot Archon. That mast was also well-designed and so the ARES team decided to use it as well on ARES.

The adoption of successful traits from past IGVC robots is complemented by the rejection of not-so-successful traits. For example, over the past three years, the previous IGVC teams have entered different versions of Archon. The Archon teams completely redesigned and rebuilt new bases each year. Last year's version of Archon (Figure 2.1 bottom) had a base using a two-wheel drive design and special industrial servos for locomotion. Although this design was revolutionary for the robotics team, the complications of the internal software of the industrial servos which drove the robot made it very difficult to synchronize and operate correctly when under a load or high-torque situations such as in an outside environment. Last year's design also consisted of a higher center of gravity, which made the robot more unstable and more likely to tip over under abrupt turning scenarios. Consequently, the ARES team this year avoided these pitfalls by using proven methods, such as the much more reliable, heavier, and lower base.

2.2. Team Organization

The team is composed of undergraduate students from three engineering disciplines and one non-engineering discipline: electrical, mechanical, computer science, and nursing. Each discipline has a certain set of skills that helps bring our team together and helps provide us with a project that is complete in every aspect. Figure 2.2 shows the role of each individual and the structure of the team. The team logged roughly 1500 hours over the past year, and each student received five credit hours of course instruction over the past year at BSC in robotics.

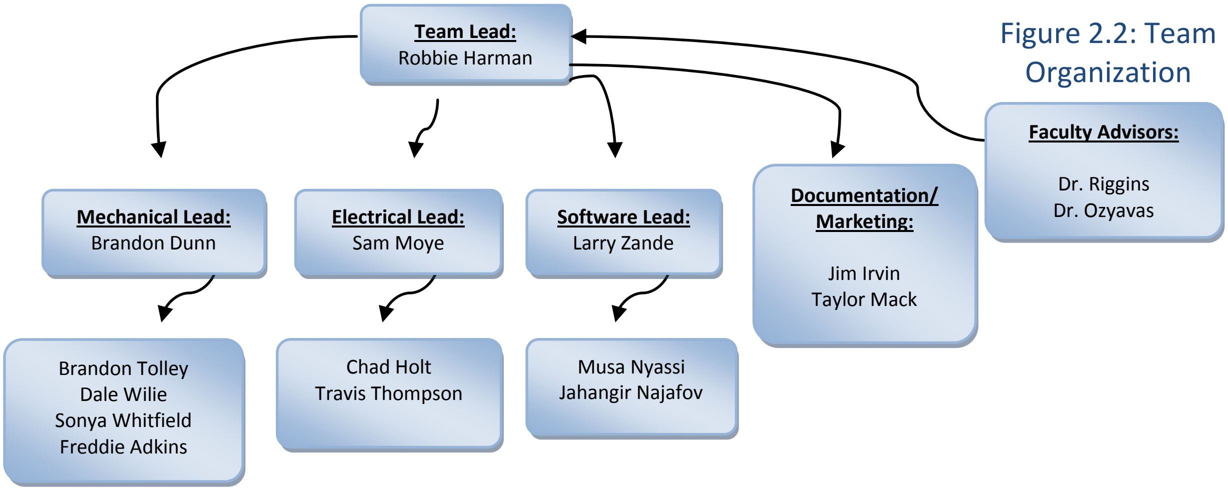


Figure 2.2: Team Organization

The following list names each team member and their academic department and class.

- Robbie Harman, Senior, Mechanical
- Sam Moye, Junior, Electrical
- Larry Zande, Senior, Computer Science
- Brandon Dunn, Junior, Mechanical
- Brandon Tolley, Sophomore, Electrical
- Chad Holt, Senior, Electrical
- Travis Thompson, Senior, Electrical
- Sonya Whitfield, Senior, Mechanical
- Dale Wilie, Senior, Mechanical
- Jahangir Najafov, Junior, Computer Science
- Musa Nyassi, Junior, Computer Science
- Freddie Adkins, Senior, Mechanical
- Jim Irvin, Sophomore, Electrical
- Taylor Mack, Freshman, Nursing

3. Design Innovations

Most of the design innovations for this year's team consist of making the robot physically more modular and easier to disconnect/connect and access various integral components. The top portion of the robot was the body frame used by BSC's Vasilius robot, but modifications on ARES makes it very modular and maintainable. For example, it is much easier to separate the body from the base, and to remove and access the two 12-volt batteries in ARES' base. The design of how the team members access, change, or repair the batteries was also improved with a sliding tray to easily and quickly install and remove the batteries (See Figure 3.1). The team also designed a very functional way to house the Hokuyo (a Laser Measurement System, or LMS), a pivotal component of our robot's guidance. The design allows the Hokuyo to be easily lowered during operation and raised during

Figure 3.1 Fabricated Battery Tray

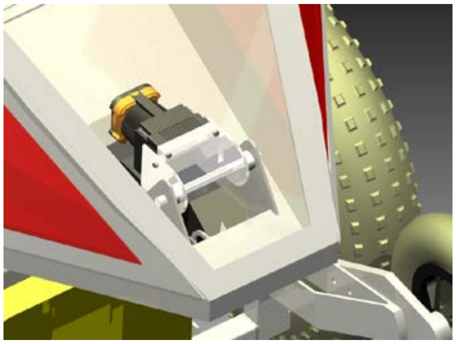


Figure 3.2 Hokuyo Mount

transportation of the robot to ensure its safety from anything striking the unit. Figure 3.2 shows the final design of the Hokuyo mount.

Figure 3.3 Payload Mount

Another design innovation developed by members of the mechanical design team was a payload mount (see Figure 3.3). This mount will allow easy installation and removal of the payload. The mechanical design team was able to successfully design, fabricate, and mount the payload mount with just enough space available to clear the Hokuyo. The position of the payload is low for keeping the center of gravity close to the ground and therefore frees the top of the robot for other components.

ARES also employs a design innovation in its software. Although ARES' navigation logic is very close to Anassa's navigation logic (Anassa was the BSC 2009 IGVC robot), the software for ARES is being completely redesigned and used in the C++/Qt environment. Qt allows ARES' software to be implemented easily in many different platforms (e.g., Macs, PCs, Linux, Windows, etc.).

4. Vehicle Design

4.1. Mechanical Systems

The mechanical design of an autonomous vehicle is very crucial, perhaps the most important design decision that the team has to make based upon resources. Without a great mechanical design, the robot would always be limited. Once again, the team decided to utilize proven components instead of undergoing the huge task of completely redesigning the robot from the ground up.

4.1.1. Chassis

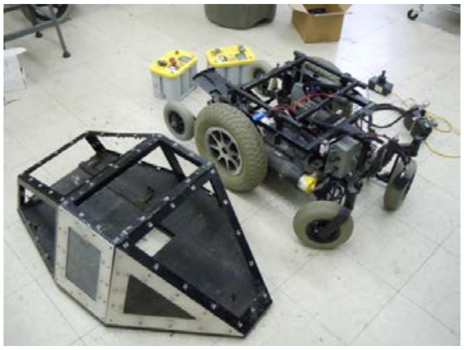


Figure 4.1 Wheelchair Base and Top Frame

The chassis consists of a wheelchair base and a top portion that were used on previous designs. Figure 4.1 shows the bare components before any modifications were made.

ARES is 26.25 inches wide, 70.5 inches tall (including the camera mast), and 40 inches long, and including the 20-pound payload weighs approximately 200 pounds when fully loaded. ARES meets IGVC specifications, but these dimensions are at the lower limit, keeping ARES as small as possible. ARES' top frame was designed by previous

BSC robotics team and fabricated by a local business.

Overall, this team has developed a very creative and successful way of combining these components of previous robots into one versatile robot. The mechanical dynamics of this year's robot rivals or exceeds the most successful performances of past BSC robots. Figure 4.2 shows the final three dimensional rendering of Ares on the right and a photo of this year's competition robot near completion on the left.

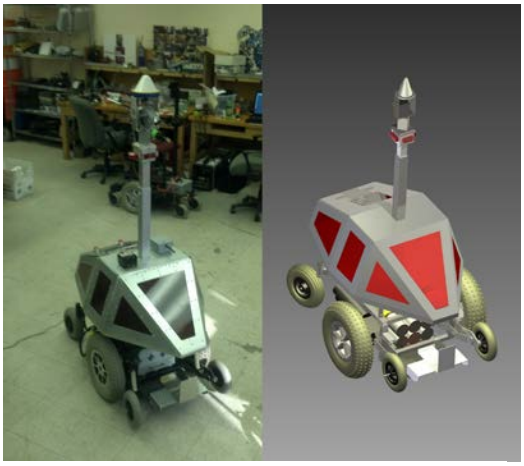


Figure 4.2 Robot near completion (left). Final three dimensional design (right).

4.2. Electrical Systems

4.2.1. Power Distribution

ARES uses two 12-VDC batteries rated at 30 amp-hours each. Under normal operating conditions on smooth and flat ground, this battery will allow the vehicle to be operated for 58 minutes at full speed. The 24-VDC supply provides power for two motors, a "Pilot" wheelchair controller, and for one 24-to-12-volt dc-to-dc converter. All of ARES' devices except the laptop derive their power from this converter. Figure 4.4 shows how the power is distributed throughout the robot.

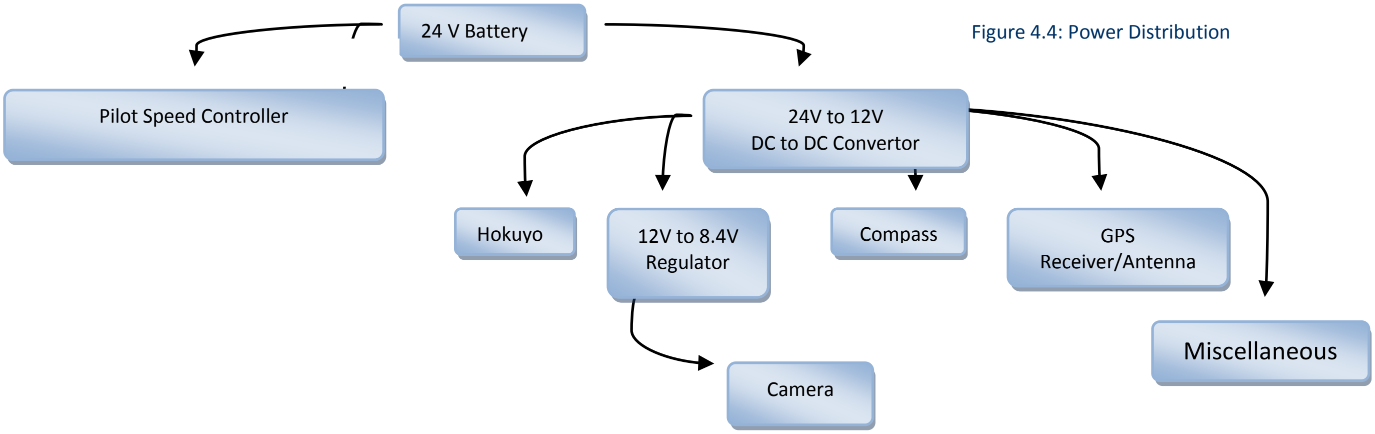


Figure 4.4: Power Distribution

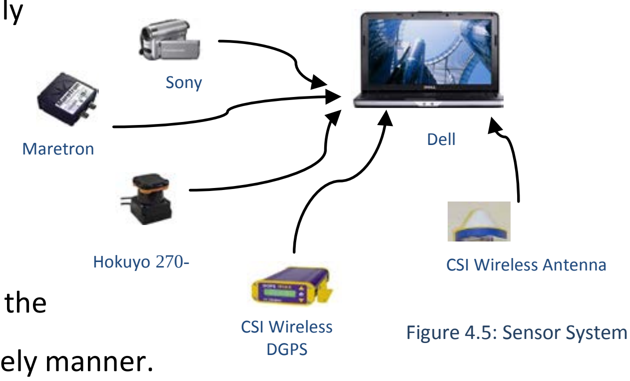


Figure 4.5: Sensor System

To ensure that ARES is safe, reliable, durable, and easily serviceable, several special features have been incorporated into the power distribution system. We redesigned ARES' frame to simplify the battery replacement process through the use of a tray. The team fabricated a battery tray for easy battery access. The batteries can now be removed easily from the front of the base and replaced with another in a quick and timely manner. Its location toward the bottom of the robot contributes to a safer low center of gravity.

4.2.2. Sensor System

ARES incorporates four sensors into its compact design: a camera, a DGPS, a Hokuyo, and a digital compass. The mounts for each sensor are designed to facilitate their easy removal for maintenance or replacement if it becomes necessary. The following is a brief description of the sensors that are used by ARES as shown in Figure 4.5.

Camera: The team selected the Sony HandyCam camcorder camera as the vision sensor for this vehicle. This camera is easy to use and very effective for ARES because it uses automatic lighting and focusing feedback and also has usb video streaming and can be used with firewire. The Sony HandyCam camcorder's progressive scanning and high frame rates minimize motion blurring. The camera has a 0.3x wide angle lens creating a 110° field-of-view. The wider angle field-of-view increases the effective image area and makes our navigation algorithm's mapping more complete (see Section 5).

Hokuyo: ARES uses the Hokuyo laser measurement scanner for obstacle detection. The unit is capable of collecting data in a 270° field-of-view in 0.25° increments with a range of 30m. The Hokuyo connects directly to a usb port on the on-board laptop.

DGPS & Antenna: To obtain positioning data in both the autonomous and navigation challenges, ARES uses the CSI DGPS system. The DGPS antenna is mounted to the top of the vehicle's mast while the receiver is securely positioned inside the top chassis.

Compass: A Maretron Solid State Compass helps determine vehicle heading. This compass provides a heading accuracy of 0.1° and updates at 10 Hz. This rate is sufficient for the vehicle's desired performance.

4.2.3. Wireless Remote Control and E-Stop Systems

Although ARES is fully autonomous, incorporation of a wireless remote control facilitates manual operation of the vehicle. The wireless remote control uses a VEX 75 MHz transmitter and receiver. ARES can operate in one of two modes, autonomous or manual; the autonomous mode uses the on-board laptop for the command source and the manual mode uses the transmitter/receiver pair for the command source. In addition to the wireless remote control ARES also has a wireless E-Stop system (a team-made 2.4 GHZ transmitter and receiver.) This E-Stop receiver uses a relay in parallel with the other E-Stop momentary switches to kill power to the wheelchair motor controller.

5. Software Development and Navigation Logic

5.1. Software Development Overview

The ARES software team embarked on two development paths; to reevaluate and redesign the previous Visual C++ software and to convert the Visual C++ code into Qt which the Bluefield State robotics team envisages to be the software of choice in the future. Qt allows the team to use this software across a wide range of platforms. Work on the software was divided along these lines; a group worked on making the C++ software adhere to sound OOP Software development principles, while another worked on Qt in the same vein. ARES's autonomous navigation system entails map generation, sensor fusion, map augmentation, goal selection, path-planning and control decisions; each of these components is executed every 60 milliseconds.

5.2. Map Generation

ARES's map is made up of a matrix of 6400 blocks arranged in an 80x80 grid; each block is approximately 4x4 inches. The upper left node and lower right nodes have a location designation of (0, 0) and (79, 79) respectively. ARES always occupies block (40, 60), encompasses a forward-looking range of 20', a rearward-looking range of 6'8" and a range to each side of 13'4". During map creation, a weight of 1000 is assigned to all blocks.

5.3. Sensor Fusion

Hokuyo LMS, camera, compass and GPS are the four main forms of sensor data used by ARES. Each of these provides data in a unique format. LMS and camera data are converted to distance and vector information before being integrated and placed on the map. Compass and GPS data are used for path-planning and are not integrated until that time.

Fig. 5.1 illustrates an example of LMS data relative to the robot, denoted by “R”. LMS data consists of a 1080-element array representing a 270° field of view in ¼° increments. Accuracy of the LMS is limited by the resolution of the map grid. Obstacles detected by the LMS are assigned a value of 3500, which represents a high level of confidence due to the accuracy of hardware.

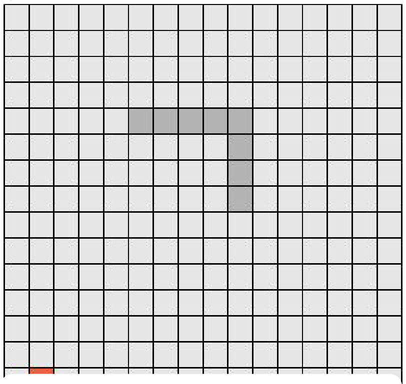


Figure 5.1: Map after LMS data added

Fig. 5.2 shows the addition of camera data. Camera data comes to the laptop as a wide-angle video feed (using a 0.3X fish-eye lens), distorted both radially and geometrically. The image is processed to generate a 181-element array. Obstacles are detected by an image analysis algorithm which recognizes a number of user-defined color values. Specifically, the image analysis algorithm will look for white boundary lines used to denote the course, for potholes, flags, and for any object where color is important. Distance measurements to obstacles are fed into the camera array index corresponding to the vector of the detected obstacle. Camera-detected obstacles are deemed to be of lower confidence than LMS-detected obstacles and are assigned a lower weight of 3000.

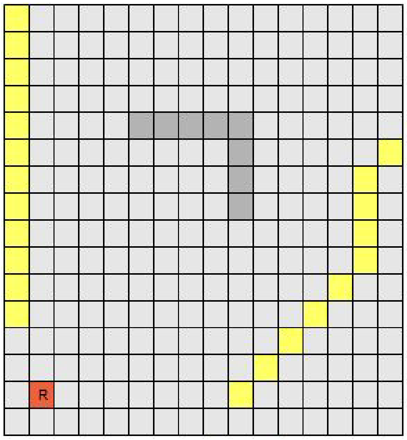


Figure 5.2: Camera data added

5.4. Map Augmentation

As illustrated in Figure 5.3 the program adds extra blocks around detected objects. These extra blocks around obstacles act as buffer zones which repels ARES away as it gets near. This layer around the obstacles, denoted as “fat” is user-adjustable with typical values of 3 to 6. A layer of 4 provides a minimum usable gap of 32” between detected obstacles. During path planning, map squares denoted as fat are treated as obstacles. ARES has a frame width of 26 inches. A fat layer of 4 provides a 16-inch clearance on each side of a detected

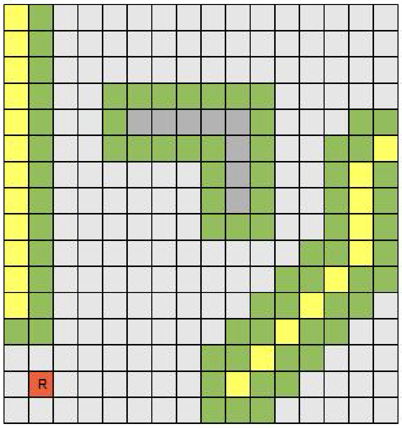


Figure 5.3: Fat layer added

obstacle. Therefore, since ARES has a width of 26 inches, then theoretically ARES should have 3 inches of clearance on each side. Should a different clearance be desired, the fat layer can be adjusted in the field, repelling it further away from or moving it closer to detected obstacles.

5.5. Goal Selection

ARES calculates a destination by a process referred to as "goal selection". As its map is only an 80x80 grid, a provisional goal is selected if the desired goal falls out of the current map's boundary. The provisional goal is selected through a dynamic weight equation (see Equation 1 below) using five parameters: straight, distance, gap, slant, and final destination (user supplied waypoints). Any of these weights, including the waypoints, can be disabled without impeding the basic functionality of ARES. All parameters are user-defined and can be adjusted in the field by a human operator. This allows for dynamic adjustment of ARES's goal selection preferences. Parameters for setting the goal are defined in Figure 5.4.

Characteristic	Definition
$\boldsymbol{d}$	A measure in meters of the distance between the robot and the possible goal node
$\boldsymbol{a}$	A measure in degrees of how in line a possible goal node is with the actual direction of the GPS waypoint with respect to the robot
$\boldsymbol{\beta}$	A measure in degrees of how in line a possible goal node is with the straight ahead direction of the robot
$\boldsymbol{S}$	A measure of how in line the node is with the slant of the map (slant is discussed below)
$\boldsymbol{G}$	A measure of the gap on either side of the possible goal node if obstacles are present. (gap is discussed below)

Figure 5.4 Definition of goal node characteristics

$$\boldsymbol{Weight} = (\boldsymbol{d} \times \boldsymbol{P_d}) + (\boldsymbol{a} \times \boldsymbol{P_a}) + (\boldsymbol{\beta} \times \boldsymbol{P_\beta}) + (\boldsymbol{S} \times \boldsymbol{P_S}) + (\boldsymbol{G} \times \boldsymbol{P_G})$$

Equation 1: Weight Equation With User-Determined Parameters

In selecting a goal, "Slant", denoted by "S", allows the robot to stay on course despite intermittent lines. In a path that tends to bear to the northwest, ARES will notice the tendency and tend to favor a path following the same bearing.

"Gap", denoted by "G", measures the amount of space on each side of a potential goal node. ARES will be inclined (according to the gap parameter) to go through the biggest gap that is less than or equal to a predefined width. Since the obstacle course at IGVC has a known maximum width and known minimum width, ARES can be instructed to prefer paths that fall within these bounds.

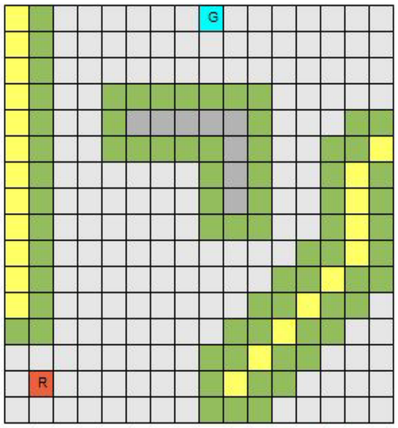
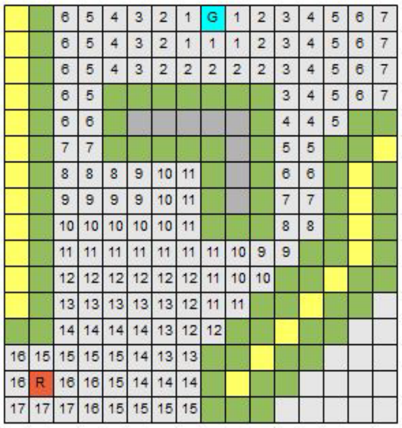


Figure 5.5: Goal Node added

At this point the goal is selected, denoted as “G” in Figure 5.5. The goal selection procedure begins at the robot's node and iterates through all allowed nodes resulting in a path-- an allowed node being one which is not occupied by an obstacle. As the search progresses through each row for gaps, the center node of each gap is evaluated using the weight equation. When an evaluated node has a higher weight than the current candidate goal node, it becomes the new candidate goal node. The procedure continues until either all rows have been evaluated or there is no path to the next row. Consequently at this point a new candidate goal node has been found, and it gets a weight of zero (all other nodes at this point in the algorithm has weight 1000 or higher.)

5.6. Path Creation

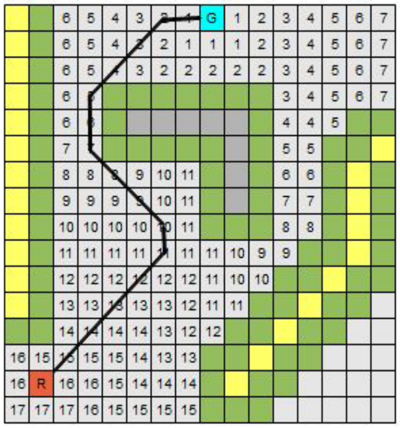
The path selection process entails three sub-processes: ripple algorithm, waterfall algorithm and smoothing algorithm. These processes pave the way for an optimal path to the goal node. The goal selection algorithm described above ensures that at least one path from ARES to the goal exists.



		6	5	4	3	2	1	G	1	2	3	4	5	6	7
		6	5	4	3	2	1	1	1	2	3	4	5	6	7
		6	5	4	3	2	2	2	2	2	3	4	5	6	7
		6	5								3	4	5	6	7
		6	6								4	4	5		
		7	7								5	5			
		8	8	8	9	10	11				6	6			
		9	9	9	9	10	11				7	7			
		10	10	10	10	10	11				8	8			
		11	11	11	11	11	11	11	10	9	9				
		12	12	12	12	12	12	11	10	10					
		13	13	13	13	13	12	11	11						
		14	14	14	14	13	12	12							
16	15	15	15	15	14	13	13								
16	R	16	16	15	14	14	14								
17	17	17	16	15	15	15	15								

Figure 5.6: Map with ripple algorithm

The ripple algorithm assigns a weight to all map nodes not occupied by an obstacle beginning with the goal node by employing a breadth-first search strategy. A recursive algorithm carries on checking nodes adjacent to nodes already assigned a value until all valid adjacent nodes are checked. This process flows backwards from goal to ARES and is displayed in Figure 5.6.

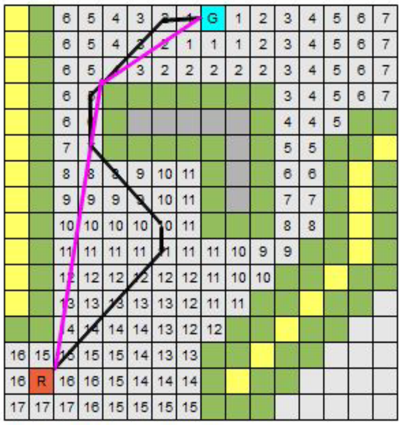


		6	5	4	3	2	1	G	1	2	3	4	5	6	7
		6	5	4	3	2	1	1	1	2	3	4	5	6	7
		6	5	4	3	2	2	2	2	2	3	4	5	6	7
		6	5								3	4	5	6	7
		6	6								4	4	5		
		7	7								5	5			
		8	8	8	9	10	11				6	6			
		9	9	9	9	10	11				7	7			
		10	10	10	10	10	11				8	8			
		11	11	11	11	11	11	11	10	9	9				
		12	12	12	12	12	12	11	10	10					
		13	13	13	13	13	12	11	11						
		14	14	14	14	13	12	12							
16	15	15	15	15	14	13	13								
16	R	16	16	15	14	14	14								
17	17	17	16	15	15	15	15								

Figure 5.7: Path Selection

The waterfall algorithm is based on the well-known A* algorithm, which finds the best possible optimal path from ARES forward to the goal node. Starting with ARES's position, all nodes surrounding the current step in the path selection process are evaluated and the node with the lowest value is chosen as the next node in the path. In case of a tie, a cost equation decides which node to use. With this algorithm, the selected path is optimal in terms of the number of nodes in the selected path. Figure 5.7 shows the path selected. A special case algorithm has been implemented if a flag has been detected on the map between ARES and its next designated waypoint. If this special case is detected, path selection will always pass on the proper side of the detected flag.

Figure 5.8 demonstrates the path smoother. Path selection of the Waterfall algorithm is restricted to 45º and 90º turns; a limitation overcome by ARES with a smoothing algorithm to "cut corners." Thus a smooth path will never guide ARES into a node occupied by an obstacle, allowing ARES to take a straight-line path wherever possible. This results in an optimal path as measured in real-world distances.



🟨	🟩	6	5	4	3	2	1	G	1	2	3	4	5	6	7
🟨	🟩	6	5	4	3	2	1	1	1	2	3	4	5	6	7
🟨	🟩	6	5		3	2	2	2	2	2	3	4	5	6	7
🟨	🟩	6	5	🟩	🟩	🟩	🟩	🟩	🟩	🟩	3	4	5	6	7
🟨	🟩	6	6	🟩	▨	▨	▨	▨	▨	🟩	4	4	5	🟩	🟩
🟨	🟩	7	7	🟩	🟩	🟩	🟩	🟩	▨	🟩	5	5	🟩	🟩	🟨
🟨	🟩	8	8	8	9	10	11	🟩	▨	🟩	6	6	🟩	🟨	🟩
🟨	🟩	9	9	9	9	10	11	🟩	▨	🟩	7	7	🟩	🟨	🟩
🟨	🟩	10	10	10	10	10	11	🟩	🟩	🟩	8	8	🟩	🟨	🟩
🟨	🟩	11	11	11	11	11	11	11	10	9	9	🟩	🟩	🟨	🟩
🟨	🟩	12	12	12	12	12	12	11	10	10	🟩	🟩	🟨	🟩	🟩
🟨	🟩	13	13	13	13	13	12	11	11	🟩	🟩	🟨	🟩	🟩	
🟩	🟩	14	14	14	14	13	12	12	🟩	🟩	🟨	🟩	🟩		
16	15	15	15	15	14	13	13	🟩	🟩	🟨	🟩	🟩			
16	R	16	16	15	14	14	14	🟩	🟨	🟩	🟩				
17	17	17	16	15	15	15	15	🟩	🟩	🟩					

Figure 5.8: Smoother algorithm

5.7. Control Decision

After path planning, ARES implements its control decision. Speed is dynamically adjusted based on the length of the first straight line segment of the planned path. This enables it to move at a lower speed when faced with obstacles, and eventually increase speed on longer segments of clear space up to ARES' maximum speed of 6 mph. This strategy ensures sufficient time is available for it to react in tight quarters. ARES's turn angle is determined by the direction of the first straight line segment of the planned path. The speed and turn commands are then processed by a separate team-built micro-controller, which in turn provides analog signals to the "Pilot" controller.

5.8. Simulation

Webots, a development environment used to model, program and simulate mobile robots was acquired by BSC. The software allowed for testing ARES's behavior in physically realistic worlds. By deploying ARES in these virtual environments it allowed for the monitoring of its goal-setting and path-finding algorithms on independent platforms. All data used by ARES in its decision making process can be captured and analyzed in an easier-to-read format. Thus it allows the team to carry out seamless testing without the hassle and complications of physical outdoor testing, saving the team a lot of development time as it continued to test and refine ARES even when the robot itself is not available.

5.9. Software Innovation

The largest advancements in software innovation this year have come in the form of adherence to principles of object-oriented design. This entailed restructuring the design for a much more closed-for-modification and open-for-extension software design.

The software team designed well-defined logical classes which do single tasks and moved away from overloaded classes which carry out multiple tasks. Therefore, ARES' software has a more task-distributed structure, delegating tasks to classes which are known to carry out particular functions.

A major stride was also made in moving towards Qt, a platform-independent software. Development in Qt was carried out in parallel with Microsoft Visual Studio C++. Qt looks very promising and ARES' software team feels it will be the software of choice for future BSC robots.

6. Predicted Performance

6.1. Speed

Given the vehicle's 15-inch wheels and 7:1 gear ratio, ARES' motors are capable of theoretically driving the vehicle at 6 mph (Jazzy 1170 wheelchair specifications.) Vehicle testing has yielded results close to this estimate. In accordance with IGVC regulations, this maximum speed of ARES is well within the limit of 10 mph.

6.2. Ramp-climbing ability

According to IGVC regulations the vehicle must be able to climb a ramp-like structure with an angle of 15%. Given the motor specifications and the robot's weight, ARES will be capable of climbing the 15% incline at 6 mph. ARES' base has been tested on a 25% incline with little or no reduction in maximum speed. ARES' ramp-climbing ability while fully loaded will be tested before IGVC.

6.3. Reaction time

The software program cycles through all procedures in less than 0.06 seconds. Theoretically, the overall reaction time of the robot could be very fast. At the time of this report, the actual reaction times have not been set and measured, but will be by the time of IGVC 2012.

6.4. Battery life

Table 1 lists the power consumed by the vehicle components under normal as well as worst-case operating conditions. Using these values, it is expected that the vehicle will be able to run for approximately 58 minutes under normal operating conditions and 55 minutes under the worst-case conditions.

Device	***Normal Operating Conditions***			***Worst Case Conditions***		
	Voltage (V)	Amps (A)	Watts (W)	Voltage (V)	Amps (A)	Watts (W)
Hokuyo	12	.7	8.4	12	.7	8.4
CSI Wireless DGPS Receiver and Antenna	12	1	12	12	1.5	18
Sony HandyCam Camcorder	8.4	.2	1.68	8.4	.6	5.04
Maretron Solid State Compass	12	.15	1.8	12	.15	1.8
Motors/Controllers	24	30.28	1453.44	28	31.92	1532.16
Total (Watts)			***1477.32***			***1565.40***

Table 1: Power

6.5. Distance at which obstacles are detected

The vehicle's Hokuyo unit is configured for a range of 30 meters (98.4 ft). The camera is set up for a somewhat shorter range of 7 meters to eliminate glare and horizon effects.

6.6. Accuracy of arrival at navigation waypoints

With a differential beacon, the CSI Wireless DGPS gives an accuracy of two feet 67% of the time. This accuracy will most likely come within the one-meter circles at IGVC.

7. Safety, Reliability, and Durability

As with any product, it is not enough to perform well. One must also provide a strong and durable product that is capable of operating safely and reliably. ARES includes several features that not only contribute to its performance, but also increase its safety, reliability, and durability. Two independent hard-wired E-Stop systems on the rear of ARES are implemented to ensure that the vehicle can be stopped safely, quickly, and reliably using the on-board push buttons. These are called the "soft" and "hard" E-Stops. The soft E-Stop maintains power to all the devices but stops the motor controller while the hard E-Stop cuts power to everything. Also, two independent wireless remote E-Stops are used on ARES: one on the 69MHz to 89MHz VEX manual controller, and one using the team-built 2.4GHz transceiver.

At the time of writing this report, the team plans to cover the base layer of the top frame with a non-conductive matting to protect the electronic components from any possible shorts to the frame. All electrical circuits are carefully fused to prevent electrical damage. Furthermore, each device is fastened securely in order to ensure that no device becomes dislodged while the vehicle moves.

8. Costs and Sponsorships

Table 2 breaks down the team cost and Table 3 shows a list of sponsors with the service provided.

Description	***Retail Cost***	***Actual Cost***	***Comments***
Frame/Body	***$1000.00***	***$0.00***	***Wheelchair Base and Top Frame***
Two 12V Batteries	***$380.00***	***$280.00***	***Purchased New at Cost***
Camera/Lens	***$320.00***	***$0.00***	***Previously Used***
Hokuyo	***$5600.00***	***$5600.00***	***Purchased New***
DGPS & Antenna	***$3000.00***	***$1800.00***	***Previously Used***
Compass	***$700.00***	***$700.00***	***Previously Used***
Laptop	***$2200.00***	***$2200.00***	***Previously Used***
Extra Wires/Components	***$300.00***	***$200.00***	***Previously Used***
Total	***$13,500***	***$11,060***	**Savings of $*2440.00***

Table 2: Team Cost

Sponsors	**Materials**
Smith Services	Aluminum
Conn-Weld	Donation of steel
CART Inc.	Funding of trips and materials

Table 3: Sponsors

9. Conclusion

With the additional amount of time available due to the smaller amount of fabrication time as compared to years past, it has allowed this year's team to conduct more testing, plan software strategies, and improve the overall design of each component. At the time of writing this report, we plan to utilize the remaining time to conduct further testing. This year we are very optimistic and satisfied with the operation of our robot thus far and look forward to a successful placement in this year's IGVC competition.