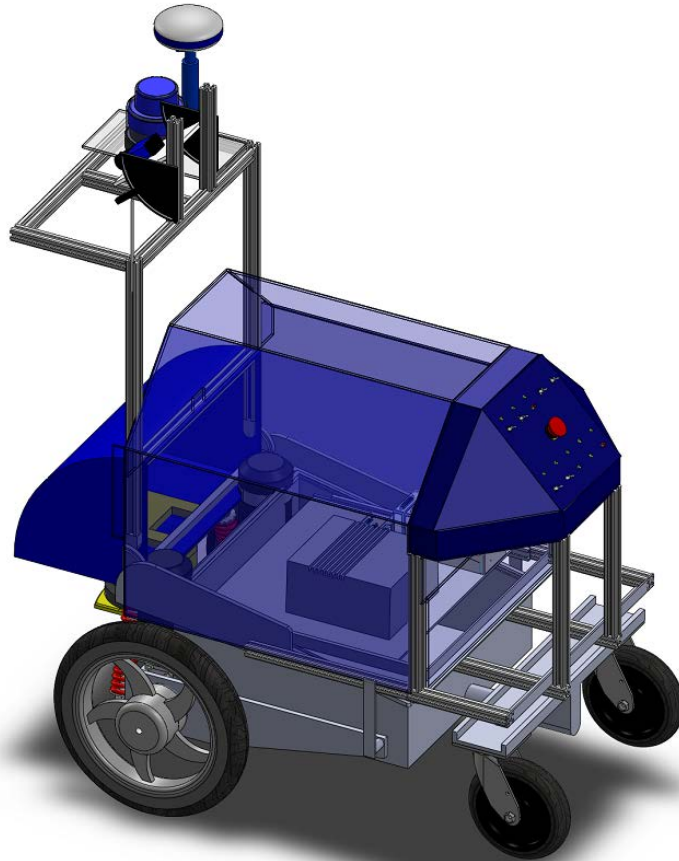


Q

2013 IGVC DESIGN REPORT



Team Members:

Hokchhay Tann '14, Vlad Burca '14, Alex Merchen '14, Junius Santoso '14, Jiajia Zhao '15
Abhishek Khanal '15, Ben Williams '15

Faculty Advisor:

Dr. David Ahlgren

May 10, 2013



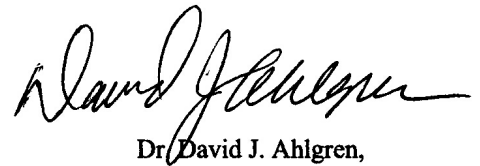
Trinity College
HARTFORD CONNECTICUT

TABLE OF CONTENTS

1. Introduction	2
2. Innovation	2
2.1 Image Processing	2
2.2 Motor Control Feedback	2
2.3 Path Planning	3
2.4 JAUS Upgrades	3
3. Design Process.....	3
3.1 Team Organization	3
3.2 Design Methodology	4
4. Hardware	5
4.1 Chassis and Drivetrain	5
4.2 System Integration	5
4.3 Control Panel	6
4.4 Power Supply.....	7
4.5 Electrical System Upgrades	7
5. Software	7
5.1 Software Architecture	8
5.2 Sensor Interfacing	9
5.3 Communication	9
5.4 Intelligence Algorithms: VFH+	10
5.5 Waypoint Navigation	11
5.6 Image Processing	12
5.7 Motor Control	13
6. JAUS	13
7. Safety	14
7.1 Wireless Emergency Stop	14
8. Vehicle Cost	15
9. Concluding Remarks	15
10. Sponsors	15
11. References	16

Faculty Statement

This is to certify that Q has undergone significant redesign in both hardware and particularly in software from last year's IGVC entry. The Q team members worked on the robot as an Independent Study project and received 1.0 credit (3 credit hours) per semester. This project is significant and has led to many senior design projects in both Computer Science and Engineering.



Dr. David J. Ahlgren,
Karl W. Hallden Professor of Engineering, Trinity College

1. INTRODUCTION

The Trinity College Robot Study Team presents its seventh iteration of Q, an autonomous vehicle for the Intelligent Ground Vehicle Competition (IGVC). Based on an all-terrain wheelchair chassis, Q is a rugged robot. This year the team has continued to improve Q, with a greater focus on reliability during autonomous operation.

2. INNOVATION

The focus of innovation was the improvement of Q's vision system capabilities, as well as improvements in a more robust path planning and motor control system. The 2012 iteration of Q had strong mechanical base and electrical system but a weak vision system which could not properly distinguish white lines from grassy field. Emphasis was primarily put on achieving a working machine, and creative solutions were bypassed for simpler approaches that were easier to program and test. Based on failure analysis from 2012, a new vision system is installed on Q along with newly written software allowing complete feedback loop control over input images and output data. Finally, improvements were made to the motor control system to accomplish a smoother and more reliable navigation reducing the frequency of accelerations and brakes, which could quickly drain out onboard battery.

2.1 Image Processing

The image processing algorithm from 2012 was limited by the use two webcams with narrow fields of vision. The cameras depended heavily on the lighting conditions and contrast of the lines on the course. As a result, the effectiveness of Q's line detection varied drastically based on lighting conditions and greenness of the grass. The 2013 version of Q replaces the two webcams with a Basler Scout camera with wide angle lens locating in the center of the front frame. The new camera allows better resolution images, a wider field of view, and electronic control over gain and shutter speed. Being able to see further down the course allows better path planning and smoother navigation. In additions, electronic control over the camera enables a feedback loop system which would be able to compensate for different lighting conditions and contrast levels. This work is still developing. Finally, parallel processing was included to check for red and blue flags. If detected, a new VFH state would be entered, adding high cost to undesirable paths, such as the slalom gate created by the flags.

2.2 Motor Control Feedback

Previous iterations of Q had open-ended motor control loops, and Q had no means to ensure it was actually traveling at the desired speed. Q's motor controller provides built in encoder functionality, but previously, there was a time delay of approximately five seconds in reading them. This made any attempt to use the encoders in real time completely unrealistic. In 2012, to combat this problem, hardware was designed to directly read the

encoder values and input them through a digital input rather than through the serial port. Two optical encoders with 180 pulse precision now give velocity feedback which can be monitored to ensure that speed constraints are adhered to. Velocity and relative position data are also used as JAUS feedback.

2.3 Path Planning

Last year version of Q was to be equipped with the newly developed Simultaneous Localization and Mapping (SLAM) algorithm. Fallen short on time, SLAM was not properly developed and the algorithm had to be reverted back to the 2011 version. This edition had fixed a major flaw from the past that when Q reached a complex chicane on the track in the autonomous course, it would turn around completely and head towards the starting line. In such an environment, Q decides that the best heading to take is the open path behind the robot rather than the narrower path through the chicane. In this version, a GPS point trail as well as heading history are recorded and stored, giving the general heading over Q's recent path. This heading is used as a heavy weight in Q's VFH cost calculation, giving Q a bias away from the path it has already taken (*see Section 5.4: Intelligence*). In specific situations, such as an area of dense obstacles, Q will stop recording a path history so that the general heading is not skewed by maneuvers around obstacles and the correct bias is preserved. When Q leaves the obstacle field it again begins recording a general heading. Previously to find waypoints in the GPS challenge, Q determined a heading based only upon the nearest GPS waypoint. This heading was inserted into the VFH intelligence algorithm. In 2011 also, Q has added the A* path planning algorithm, thus taking future waypoints into account. This year, major improvements were made to the existing algorithms, VFH and A*, on Q to provide a smoother path planning using data from the new vision system, which allows Q to see much further ahead than the webcam.

2.4 JAUS Upgrades

Last year, JAUS was able to interface seamlessly with Q and the Judge's COP. However, some information was missing when responding to the judge. This year, all required messages are added.

3. DESIGN PROCESS

3.1 Team Organization

The team was organized into small groups based on challenges being faced at IGVC. Each group was charged with solving its main problem and reporting with the team on a weekly basis, during which brainstorming and major decision making took place by the entire team.

- 1. Management:** This group was responsible for leading the team and making all final design decisions. This group also manages all logistics and the planning for IGVC.

2. Image Processing: This team was tasked with upgrading the vision system of Q. They were responsible for any hardware or software modifications related to this challenge.

3. Software: This group was responsible for software management. It ensured the different parts of the program meshed without errors and that the code ran efficiently on the Embedded Vision System (EVS) and cRIO.

4. Simulation Team: This group was responsible for testing the algorithms and performance of Q. The group was also responsible for creating diagnostic programs to expedite debugging.

Members were divided into these groups based on their expertise, interests and workload associated with each group. Most members contributed to more than just one group. All members were required to work together for a minimum of three hours per week as well as attend weekly meetings to give progress reports and discuss their projects.

Since Q is a multi-year project, key members graduate every year and knowledge that was once common to all members is lost. To combat the high turnover rate, each new member works with an experienced mentor on the team who passed on his or her expert knowledge. Furthermore, regular workshops were held on topics such as FPGA programming, Solidworks modeling, and LABVIEW coding to help new members master wide variety of skills.

3.2 Design Methodology

The team followed an iterative design process to improve Q. The process started with a detailed failure analysis of Q's performance in the IGVC 2012. After understanding the faults of Q and analyzing IGVC's new rules, a detailed list of requirements was created. Based on these requirements, strategies were proposed and continually tested to reach a finalized implementation.

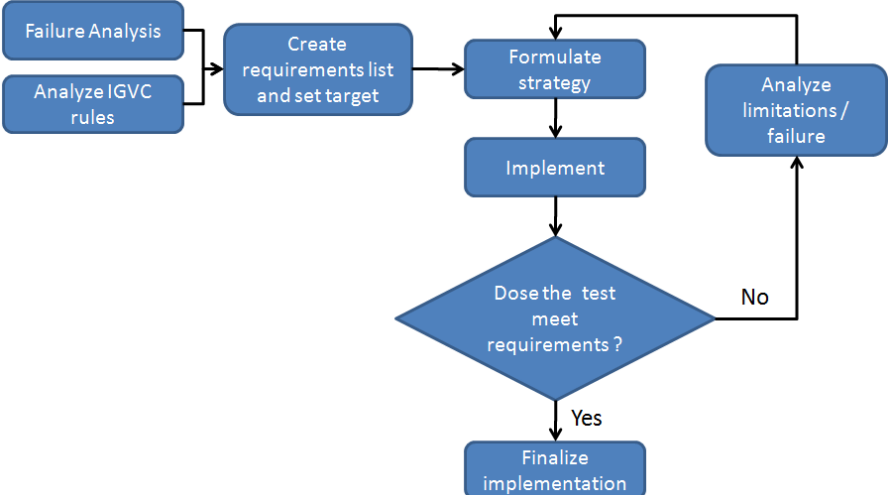


Figure 1: Design Methodology

4. HARDWARE

4.1 Chassis and Drivetrain

The physical platform of Q is a modified PerMobil Trax all-terrain wheelchair. This frame can support a payload of over 250 lbs [1] and has a small footprint of 40" by 26". It features a differential front wheel drive system - a pair of 500W Leroy Somer MBT1141S motors - and a pair of rear mounted casters. The motors are geared with a 25.8:1 ratio, providing 15ft-lb of torque. Additional sensor and payload mounting frames were constructed using 80/20 extruded aluminum channels. The use of these channels allowed for quick and easy component layout without compromising mechanical strength. The camera mount was designed to fold down to reduce Q's volume during transportation.

4.2 System Integration

The cRIO is the central control unit of Q. All motor control, data collection, navigation algorithms and sensor interfacing (with the exception of vision processing and cameras) occurs on the cRIO. This communicates with the NI EVS which interfaces with the two webcams and executes image processing for lane detection, flag detection and ramp detection. The onboard Linksys WRT-54GL wireless router allows wireless monitoring and debugging of the system. The complete system diagram is as follows:

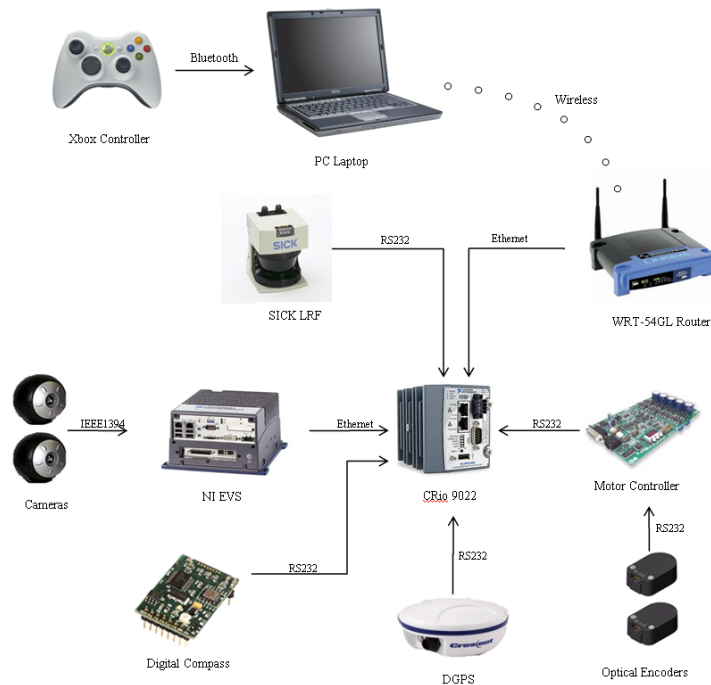


Figure 2: System Overview of Q

4.3 Control Panel

The hardware control panel allows easy control of the robot without the need of a software interface. The primary control panel allows easy operation of the robot when powered on. By simply initializing, turning off the motor safety and then pushing the GO button, the user is able to start a competition run.

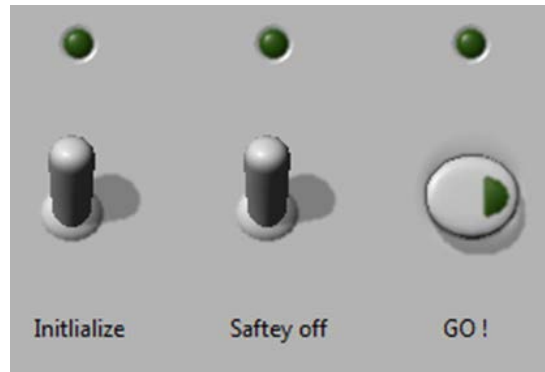


Figure 3: Main Hardware Front Panel

An additional secondary control panel offers the user more options and also gives detailed dynamic feedback of Q's operation. Here, the user is able set competition challenge type and also reset the entire control program without the need to reboot the system. An array of LEDs provides the user with real-time status of the various sensors and the control program. An additional, tri-state LED provides battery health data.



Figure 4: Secondary Hardware Front Panel.

4.4 Power Supply

The power system runs all electrical components (other than the motor) through a single power board. The new power board features separate DC-DC converters for 24V, 12V and 5V power supply. They are rated at 100W, 75W and 75W respectively. In addition, Switchlock circular connectors were added for all major components to allow easy removal and reconnection.

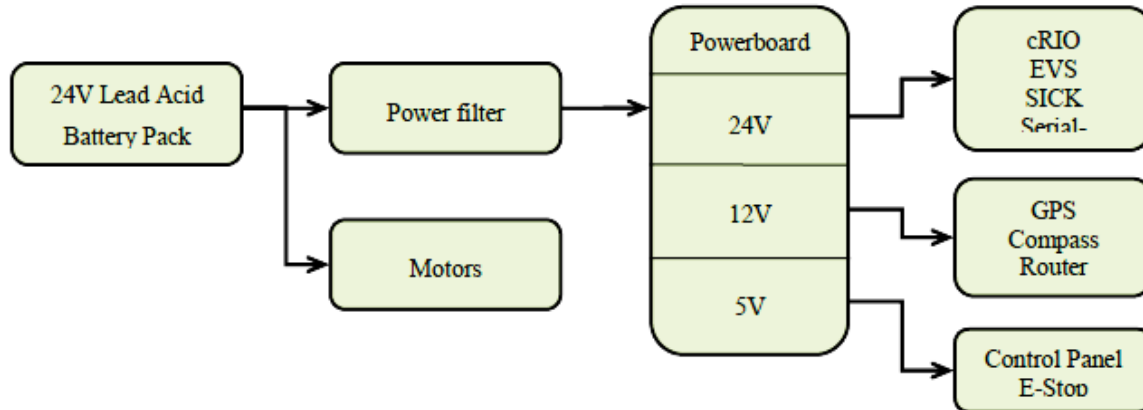


Figure 5: Power Distribution and Electrical System

To secure sensitive electrical components from transients, high frequency feedback, and surges from the motors, a 10A Radius Power Filter model RP220-10-4.7 was added to the motor line. In addition to the filter, slow-blow fuses have been added for all major electrical components to secure them in the case of accidental short-circuits.

4.5 Electrical System Upgrades

The electronics used to regulate the proper voltages on Q, known as the power board, were replaced with newer components. Greater care was taken to properly distribute heat from power board to prevent overheating, a possible cause of failure last year.

5. SOFTWARE

In light of poor software performance in last year's competition, the team has decided to update certain sections of the code running on the Q for better reliability, flexibility and execution speed. For example, the image-processing algorithm has been optimized for better frame rates and the software architecture has been radically redesigned to take advantage of the additional processing power. Additionally, new code has been written for various new system components and features. In the following sections, we will describe the design of the individual software components.

5.1 Software Architecture

The core of Q's control system is located in the cRIO and interfaces with most of the sensors and devices on the Q, including the motor controllers. All movement of the robot, autonomous or remote-controlled, is directed through this main software architecture.

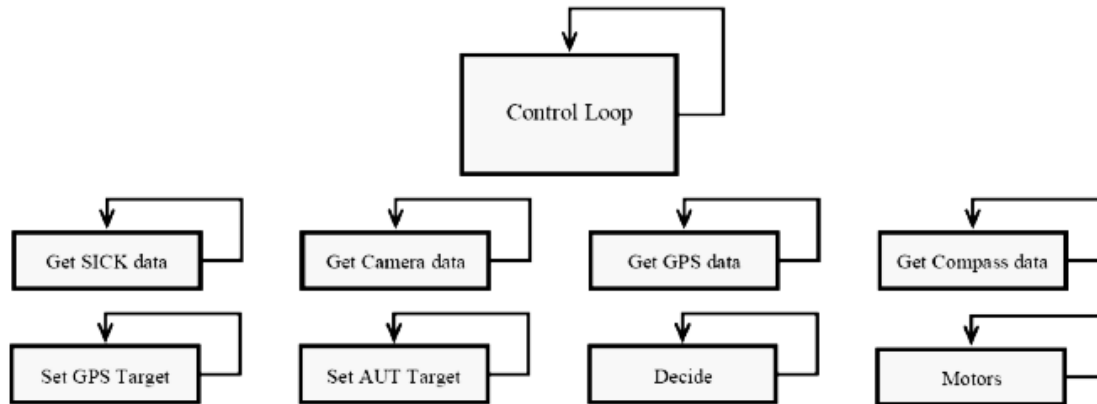


Figure 6: Parallel architecture used for IGVC 2011. Here, each block represents a process running in a loop in separate LabVIEW VIs.

Our goal in designing the main software architecture was to create a system with maximum performance and reliability, but also to make our software as intuitive and as modular as possible. To that end, our new design is based on a parallel architecture, where each component runs relatively independent of other components. Our implementation of this architecture uses parallel loops, each containing code pertaining to one particular component of the robot. For instance, each of the sensors interfaced with the cRIO and the motor controllers is controlled by a separate loop. The sensor loops continuously update data buffers, while the motor controller loop waits for active commands from the main control loop. This is to ensure that there is absolutely no undesired movement of the robot.

The main control loop directs data flow between loops, thereby controlling the behavior of the system. For example, when in the „remote control“ state, motor commands received from the tablet PC are sent to the motor controller loop, and while in the „autonomous“ state, the motor commands are computed by the VFH loop. Communication with the tablet PC and the Embedded Vision System are again achieved by using separate UDP loops.



Figure 7: Control Loop State Machine

The parallel architecture has several advantages, for instance, it allows the overall throughput of the system to be much higher, allowing for smoother control of the robot. The reliability of the software seems to have also improved, as we have seen far less system crashes than last year. Furthermore, the new architecture has proved to be much more intuitive than its predecessor, since the overall structure is much more natural to understand. Having parallel loops has also allowed us to modularize the code and work on separate parts individually, increasing efficiency and reducing the amount of time spent on piecing the various sections of code together.

5.2 Sensor Interfacing

The cameras are connected directly to the EVS's IEEE 1394 ports. LabVIEW provides library functions for capturing images from the cameras. The GPS, compass, and SICK LIDAR interface with the cRIO through the FPGA. Each of them connects to one of the four ports on the cRIO RS232 module, which is directly interfaced with the FPGA.

The code to read the compass data is written entirely in the FPGA. The FPGA examines the bytes coming through the serial port until it sees the line feed character. At this point it begins reading the heading, pitch and roll, which are sent as "heading", "pitch", "roll". One example would be 100,15,33<LF> for a heading of 100 degrees, a pitch of 15 degrees, and a roll of 33 degrees. Each number is extracted as a string, then converted to an integer data type, and passed along to the real time controller.

The code to interpret data from the GPS runs on the real-time controller in the Compact RIO. The FPGA is programmed to act as a serial port. The GPS outputs longitude and latitude following the NMEA format. All latitude and longitude data are preceded by the string `\$GPGGA`, so the GPS interpreter searches for that string, and then parses the following string for the latitude and longitude, which are converted to double floating point values and transmitted to the Compact RIO.

To read data from the SICK LIDAR, some initialization commands must be sent. Once these commands are sent, the SICK sends 360 bytes preceded by a 6 byte header. Those 360 bytes are 180 16 bit distance values in centimeters - one value for each degree, starting at -90 degrees from the heading and going until 90 degrees from the vehicle heading. The FPGA code searches for the 6 byte header and then reads each pair of bytes into some of the dedicated memory in the FPGA.

5.3 Communication

There are several components in the system that must communicate for the purpose of debugging, monitoring, parallel processing, and remote control of the robot. Figure 2 shows how all the processing components in the system are connected.

The EVS and cRIO each have alternate Ethernet ports which are used for direct communication between the cRIO and the EVS. The primary Ethernet ports are used for the wired connection from the EVS and cRIO to the Linksys wireless router. The router is used for wireless debugging, development, and monitoring of the robot. A

LabVIEW application for real time debugging was created that displays the exact information Q is using to navigate, such as current GPS location, current heading, target heading, SICK array, compass reading, a waypoint checklist, battery voltage, sensor initialization, and the control loop state machine state.

5.4 Intelligence Algorithms: VFH+

A modified form of the Vector Field Histogram (VFH) algorithm was used for obstacle avoidance. Using a SICK LIDAR sensor, a compass, and camera data, a polar histogram representation of the obstacles is created and paths are planned accordingly. The algorithm utilizes a cost function which considers the target direction, wheel orientation and previous direction. When there is more than one opening between obstacles, this cost function is used to select the best candidate direction. The opening with the lowest cost is chosen. The result is a heading which represents the best path. Several improvements from previous years were instantiated. The scan, which covers 180 degrees, was split into 180 separate sectors, rather than 30, for path consideration for better resolution. Detected obstacles were widened to account for the robot radius and turning ability. Finally, the SICK data has been split into three distance thresholds, with the nearest threshold (at one meter) receiving priority in motor functions. The further two thresholds, at two and three meters, are considered in path planning. They influence the cost calculation in the same manner as the first threshold, with robot direction and target heading taken into account, but have less weight than the threshold at one meter.



Figure 8: VFH Loop

Special VFH States

Several VFH states have been added to Q outside the “normal” state described above. These states are triggered by specific signatures detected by Q’s sensors. Examples of this would be when the ramp is detected by the SICK and Ping sensors, or when slalom gates are found from the cameras. When the ramp is detected, Q will give lowest cost to the path of the steepest gradient, so that lateral motion on the ramp is limited. When slalom flags are detected, a high cost will be placed on paths outside of the slalom gates.

Smart Path History

Q now uses a smart path history to calculate a target heading to help navigate and avoid turning around. Q uses GPS co-ordinates taken at time intervals to calculate and store past headings. A weighted average of this history gives a general heading that roughly follows the direction of the course. This assists Q in determining the general direction of the on track. This is illustrated in Figure 9. However, using a simple averaged path history fails while

traversing a complex chicane. While traversing a chicane, Q may be required to turn at a direction that is radically different than the general direction of the course. Using heading readings from inside the chicane to calculate an averaged path history may result in a target direction that leads Q into a dead end. Such a scenario is identified in Figure 10.

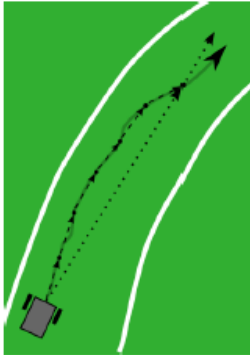


Figure 9. Usage of path history to determine general direction of the course

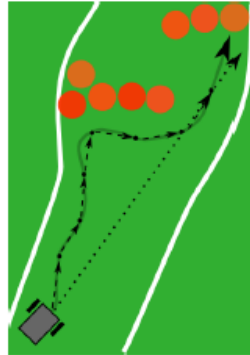


Figure 10. General path taken by Q with simple path history. Here, Q is driven to a corner.

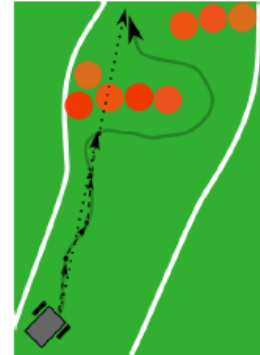


Figure 11. General path taken by Q with environment aware path history. Here, Q successfully traverses the chicane

(Note: In the above diagrams, the solid grey lines denotes the path taken by Q, the black dash lines represent heading calculated between time intervals and the black dotted lines represent the averaged heading)

To overcome the challenge of traversing a chicane, Q uses an environment-aware path history. Q uses the concept of an obstacle field to select an alternate path. Here, an obstacle field is defined as an environment where there is a large density of obstacles such as in a chicane. When Q is not in an obstacle field it continuously logs path history to calculate a new target heading. But when Q encounters an obstacle field, data collection for the path history is temporarily suspended and the target heading calculated before entering the obstacle field is held constant. This helps Q maintain a sense of direction even while traversing a chicane. The Figure 11 illustrates how, this algorithm helps Q traverse a chicane.

5.5 Waypoint Navigation

The autonomous and navigation algorithms were merged. This was accomplished by switching from the autonomous algorithm to the navigation algorithm after the first GPS waypoint is reached. When the robot is in the navigation portion of the obstacle field, Q creates an X-Y coordinate system with its current location as the origin and plots each waypoint with coordinates based on the distance of the waypoint from Q's original location. Q then moves toward the closest waypoint by assigning a very low cost to the heading of the target waypoint in the VFH cost calculation. When an obstacle is encountered, Q plots the obstacle on the X-Y graph and routes a path around it. When the waypoint is reached, Q approaches the next closest waypoint with obstacles that have been plotted considered in shortest distance calculations. If the closest waypoint changes when

Q is navigating to a waypoint (i.e. going around a wall and increasing distance from the original waypoint) then Q will pursue the closer waypoint having plotted obstacles along the way. The GPS readings are accurate to within one meter.

5.6 Image Processing

The new camera has been positioned for maximal viewing range with bottom of the image closest to the robot. The obtained images go through blue plane extraction since blue is the primary color in the RGB spectrum that is prominent in the painted grass but not in the surrounding green grass or the barrels. Next entropy thresholding algorithm is used to obtain binary image. Particle filter is then applied to remove random noise that is not part of the line. The image is then dilated to make sure the line is connected. Then elongation filter is used to test the elongation factor of each object found. Since white reflected blob normally have a comparatively small elongation factor, this filter remove all but the line in the binary image. The resulted image is then calibrated using a Labview program which has real-world coordinate for each pixel location. This image processing loop is shown in figure 12.

The throughput of the vision system has limited the reaction time of the robot. The algorithm has undergone pipelining. The various subtasks in the algorithm now run parallel to each other thereby reducing the execution time and increasing throughput. The vision code has also been ported to the FPGA of the EVS, thus decreasing processing time.

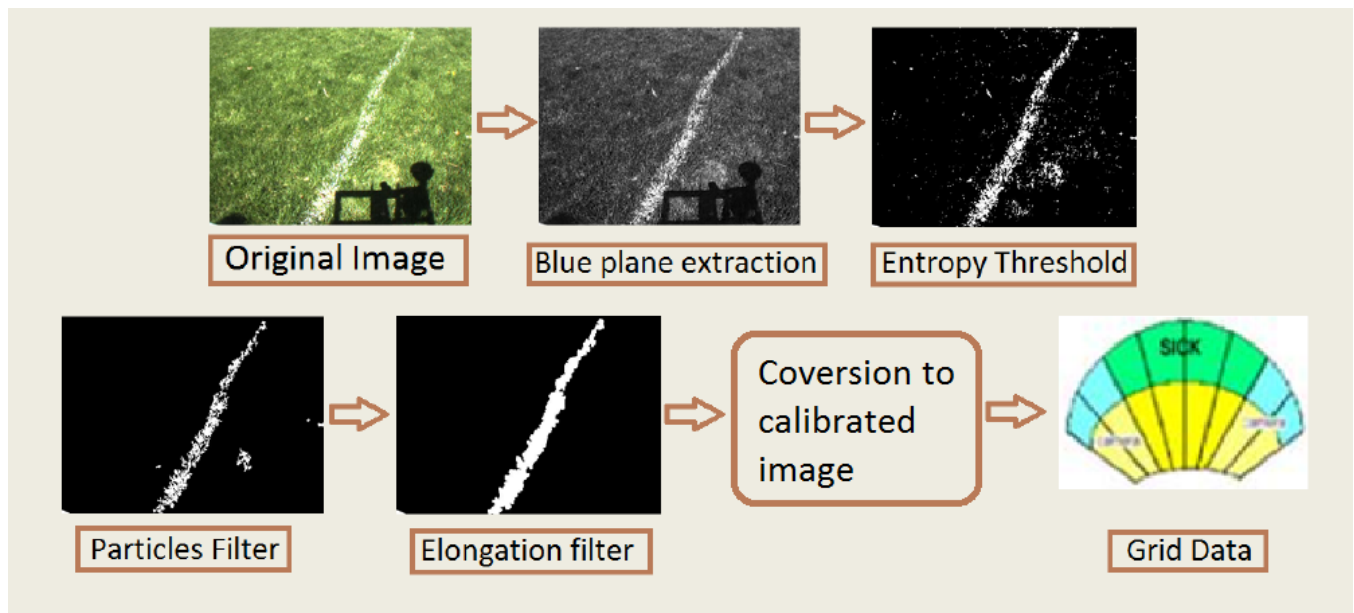


Figure 12: Image processing loop

To detect different colors, this process is repeated three times on each image by parallel loops: one loop finds green, one finds red, and the main loop finds white. When either red or green colors are detected that match the

colors of the slalom flags, a new VFH loop is initiated that steers Q through the gates. Due to its parallel nature, this extra processing does not affect throughput.

In addition, new software was written to check the brightness of each input image by calculating the mean and standard deviation of the grayscale level on the blue plane. The result is then compared to a lookup table. If the image is determine to be too bright or too dark, the gain of the camera is decreased or increased respectively. Otherwise, no change is made to the gain.

5.7 Motor Control

The motor control uses shaft encoders and feedback loop to maintain desired motor speed. A simplified kinematic model has been used to convert the speed and heading generated by the VFH algorithm to motor controller opcodes. Further, the motor control system is now a closed-loop one. Feedback from the optical encoders relays velocity and relative position information back to Q from the shafts. Q can now ensure that it is traveling the desired speed using PID control. This feedback is also essential in the JAUS competition.

6. JAUS

The Joint Architecture for Unmanned Systems (JAUS) implementation on Q is intended to provide an interface for the robot to communicate with its subcomponents, convey information about its state to the user and for the IGVC challenge to the COP. Although the main JAUS specification document has several messages and protocols that need to be implemented for complete JAUS compliance, Q must support a core set of messages for the IGVC 2013 Interoperability Challenge. Q does not have many processors and sensors to interface with the Compact RIO and the sensors used are fairly easily accessible. Building JAUS components on the Compact RIO as well as the Vision System was deemed unnecessary as it would have had substantial overhead in communication. Q contains a single JAUS component that handles all of the services required for each of the six tasks in the Interoperability Challenge. Q's internal communication structure is used to pass data between sensors and navigation algorithms. A JAUS thread communicates with each of these internal components whenever specific data is required by an outside source. The wireless router is used as the gateway between Q and other JAUS subsystems.

7. SAFETY

Safety has been given the utmost priority in the design of Q both for the electrical system and the mechanical system. Wires of gauge 10 were used to connect power sources to the motors and filter, and 16 gauge wires were used elsewhere. A circuit breaker was used for the entire electrical system. Slow-blow fuses were instantiated into the connections to each component from the power board to ensure that individual electronic devices did not receive too much power. A single phase dual stage power line filter inserted to prevent transient current from the motors.

Three main motor safety measures have been implemented. They include the motor control board FETs, the physical emergency-stop, and the wireless emergency-stop. The motor control board FETs are now controlled by the program to disconnect power from the motor whenever the system is in an idle state. The physical and wireless e-stop features also work by immediately cutting power from the motor controllers as soon as one of them is triggered.

7.1 Wireless Emergency Stop

The wireless Emergency Stop is controlled by a Seco-Larm SK-919TD1S-UP transmitter that uses 315 MHz RF transmission with a one-channel receiver. The range was extended to 150 feet by installing antennae on Q and on the transmitter.

8. VEHICLE COST

Component List	Price (US\$)	Cost Incurred (US\$)
Basler Camera + Wide Angle lens	900	900
Crescent A100 DGPS	2000	0
Caster Assembly	150	150
Chassis	650	0
Encoders	100	100
Honeywell Compass (HMR-3300)	750	0
Motors	1000	0
NI cRIO-9022 Real-Time Controller	3199	3199
NI cRIO-9133 Reconfigurable Chassis	1999	1999
NI EVS-1464RT Embedded Vision System	4499	4049
SICK LMS291 Laser Range Finder	4000	4000
Power Supply Board	150	0
Encoded Receiver/Transmission Set	77	77
Remote Control	180	0
Roboteq AX3500 Motor Controller	400	300
Wiring/Electrical	50	50
Parallax Ping Ultrasonic Sensor	30	30
Total	20134	14854

9. CONCLUDING REMARKS

The changes made to Q hopefully will lead to improved autonomous performance. Our goal has been to expand the range of situations that Q can handle when navigating so that each run will more likely be successful. We look forward to participating at the 2013 IGVC.

10. SPONSORS

Enterprise Rent-A-Car

Hemisphere GPS

Honeywell International Inc.

National Instruments

PerMobil Corporation

Travelers Insurance

Trinity College

11. REFERENCES

[1] "USA - Products - Support - Product Support - Permobil." Power Wheelchairs - Permobil. Web. 14 May 2010. <<http://www.permobil.com/USA/Products/Support/Manuals/drivers/>>.

[2] Wright, Adam, Orko Momin, Young Ho Shin, Rahul Shakya, Kumud Nepal, and David Ahlgren. "Application of a Distributed Systems Architecture for Increased Speed in Image Processing on an Autonomous Ground Vehicle (Proceedings Paper)." *Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques. Proc. of IS&T/SPIE Electronic Imaging*, San Jose. Vol. 7539. San Jose: SPIE, 2010. Print.

[3] Ulrich, Iwan, and Johann Borenstein. "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots." *Proceedings: 1998 IEEE International Conference on Robotics and Automation*, May 16-20, 1998, Katholieke Universiteit Leuven, Leuven, Belgium. *IEEE International Conference On Robotics and Automation*, Leuven, Belgium. Piscataway, NJ: Robotics and Automation Society, 1998. 1572-577. Print.