Dave

Franklin W. College of Engineering

2011 Intelligent Ground Vehicle Competition Entry

Gray Thomas, Matthew Alvarado, Lucas Hill,

Jaime McCandless, James Regulinski, Varun Mani,

Chris Marra, Tomas Lamar, Nicholas Hobbs

Faculty Advisor: Professor David Barrett

Required Faculty Statement: I certify that the engineering design of the new vehicle, Dave, described in this report, has been significant and equivalent to what might be awarded credit in a senior design course.

David Barrett

Professor David Barrett

The team also received extensive funding and resources from Olin College of Engineering integral in the development of Dave.

Part	Estimated Cost to Team ($)	Estimated Cost ($)
u-blox Odometry Enabled GPS	$0	$600
Autec Wireless Emergency Stop	$0	$700
VectorNav Vn-100 IMU	$0	$450
Dell Latitude D630	$0	$2000
HD2.0 Heavy Duty Optical Encoder	$100	$100
Aluminum Sheet Metal	$400	$400
HW-CC121006CHQR NEMA Boxes (3)	$300	$300
Hokuyo Ethernet LIDAR	$0	$4500
USB Cameras (2)	$40	$40
Vicor VI-213-IW Voltage Regulators	$40	$40
Blue Sea Systems Fuse Boxes (2)	$24	$24
PCB Boards (2)	$120	$120
Linksys WRT54GL Wireless Router	$50	$50
Pittman GM14904S016-R1Motor with Encoder	$0	$350
Pittman GM9236S027-R1	$0	$260
CommFront RS422 -> TTL Converter	$30	$30
NI cRIO-9104	$3000	$3000
NI 9403 C Series 32-Ch, 5 V/TTL Bidirectional Digital I/O Module (2)	$700	$700
NI 9401 8 Ch, 5 V/TTL High-Speed Bidirectional Digital I/O Module	$250	$250
NI 9263 4-Channel, 100 kS/s, 16-bit, ±10 V, Analog Output Module	$350	$350
NI 9205 32-Ch ±200 mV to ±10 V, 16-Bit, 250 kS/s Analog Input Module	$0	$800
NI 9870 4-Port, RS232 Serial Interface Module for CompactRIO	$600	$600
	$6004	$15664

Table 1: OGRE Expenses for the Semeseter

3 Design Process

The team started this year by reviewing notes from the 2010 IGVC competition. Reflections on the faults of the previous year's robot, Athena, lead the team to focus on vision for the first semester and on electrical system design and environmental mapping for the second. The team also invested itself in reading research papers in the first half of the semester. The reflection on the readings led to several high level goals for the year:

1. This year's vehicle platform should expand on the original software architecture introduced by Athena with a more logical organization and with more potential for higher-level behavior.
2. The vehicle should be made more accessible and logically organized electrically to allow better in-field debugging
3. The vehicle should be joystick-controllable within the competition software so that it can drive without the aid of the team.
4. The vehicle should store and maintain a map to enhance its path planning and obstacle avoidance behaviors.
5. The vehicle's cognition speed should be increased to allow operating velocity to exceed two miles per hour.
6. The vehicle should be durable and able to function in all weather conditions.

Throughout our design process we attempted to make our robot accessible in terms of development future generations of students. This goal directly influenced many of or decisions and led us to choose and implement the best solutions regardless of time or cost. This dedication to ease of use, transparency, and simplicity will serve to benefit our future competition teams.

Dave strives to be a robust, modular platform that pushes the boundary of our ability to design intelligent systems. In the following sections we will describe the implementation of these design principles.

4 Mechanical

4.1 Major Mechanical Innovations

The primary goals of this semester's mechanical redesign were to rebuild our electronics boxes so they are robustly mounted, clearly compartmentalized, and waterproof. To achieve this, we replaced the old Tupperware electronics boxes used on Athena with

with an integrated 500 CPR encoder. This motor is also reduced by an internal 65.5:1 gearbox. The actuator is connected to the LIDAR by a four bar crank rocker linkage. The linkage is configured such that continuous rotation of the motor rocks the LIDAR, protecting the delicate sensor from software malfunctions.

5 Electrical Systems

The overall cabling and electrical design of the robot is divided into three main categories: power distribution, motor controllers, and cRIO. Each of these three components was physically separated into boxes in the port, starboard and aft of the robot respectively. This demarcation aids in trouble-shooting problems and adds modularity to the electrical system such that each set of components is relatively independent of the others.

5.1 Power Distribution

Actuators generally have much higher current requirements than computing or sensing systems, meaning that rapid changes in actuator motion can cause unwanted transients in the power lines. Dave's power distribution system addresses these issues while remaining organized and expandable by regulating the power for the sensors and the computing hardware. A simplified wiring diagram is provided in Figure 2, illustrating the modularity of our design.

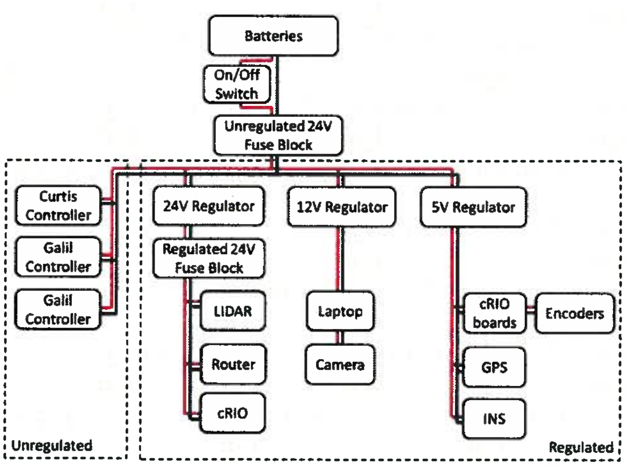

Figure 2: Simplified Electrical System

Power is provided to the robot by two 12V sealed deep-cycle lead-acid batteries. They are wired in series to each other in order to provide 24V power to the entire robot. Master power to the robot is controlled by a marine power switch. There are fuses between the battery charger and each of the batteries, as well as a terminal fuse on the positive side of the 24V series connection.

This primary power line runs through the marine switch into a 12 terminal ATC style fuse block which contains all of our unregulated power, coming from the batteries. The fuse box outputs power to each of the three motor controllers and to 24V, 12V, and 5V regulators. A 24V Vicor voltage converter regulates the power into a cleaner 24V and outputs it to another fuse block, used to distribute 24V power to the more sensitive electronics - the router, LIDAR, cRIO, and other sensors. The 12V and 5V regulators are both Castle Creations CC-BEC Pros, which are programmable switching regulators. These are used for onboard laptop power and additional sensor power, respectively.

Dave uses a Dell DC Adaptor to provide the onboard laptop with power and avoid requiring an AC inverter.

5.2 Motor Control

Dave's three motors are powered by two Galil PWM servo amplifiers and one Curtis 1228 motor controller. They are all controlled by analog inputs from the cRIO, and are wired up from the mother boards with cable checks.

5.2.1 Emergency Stop

The main electrically-based safety feature is the emergency stop (ESTOP). There are three ways to trigger an ESTOP: by the physical button on the robot, through the wireless remote, and in software. The ESTOP circuit is designed such that all three methods would result in all three motors being inhibited or braked, with the further goal that any failure of the circuit would have the same result.

encoder on the steering column runs at 500 counters per revolution. The encoder on the position end of the LIDAR tilt mechanism updates at 2048 counts per revolution, while the one on the LIDAR tilt motor also operates at 500 counts per revolution. The drive motor encoder with 500 counts per revolution is yet to be attached since the failure of the previous encoder.

5.3.4 Main Camera

The Firewire camera is a Sony DFW-SX910 mounted at the top of the robot mast. The camera's native resolution is 1280x1024, and it requires a powered firewire port to operate. Because most laptops and Cardbus Firewire cards do not provide any power, we have connected the camera through a powered PCMCIA Firewire card that runs off of Dave's 12V power.

5.3.5 Side Cameras

Dave will have two additional USB cameras mounted near its front wheel which will reduce the size of Dave's blind spots and allow the main camera to see further ahead without sacrificing the ability to avoid lines close to the front wheel.

5.3.6 GPS

The GPS sensor intended for Dave is a Navcom SF-2050G, which has less than 10cm horizontal accuracy via differential correction. Because of long lead-times for repair, the team switched to a UBlox EVU-6T. It communicates over serial to the cRIO's built in port, and has accuracy down to approximately 2m.

5.3.7 LIDAR

The LIDAR currently on the robot is a pre-production Hokuyo UXM-30LX-E. The robot originally - and may again in the future - used a SICK LMS-291-S05, but at the time of writing it is out for repair. The Hokuyo communicates with the robot over TCP using the SCIP 2.0 protocol. It is connected to Dave's local network using a manually set IP address and port number. The LIDAR has a 190° field of view, with 0.25 ° resolution, and a 20 Hz update rate.

5.3.8 Inertial Navigation

Dave's inertial navigation system is a development board for the VectorNav VN-100. It continuously outputs both attitude (orientation in 3D-space) and heading (digital compass) over serial and is read on the FPGA serial port.

5.3.9 Hall Effect Sensor

Dave uses a Hall Effect sensor to center its steering mechanism. We chose this system over limit switches for the higher reliability over physical limit switches.

5.4 Computer Hardware

5.4.1 Wireless Router

Because cRIO communicates with the LIDAR and laptop over a network, Dave includes a Linksys WRT54GL wireless router configured to host an 802.11g network with WPA2 Personal encryption.

The router assigns static IP addresses to known hosts, and these lists are hardcoded into the software. Each team member's laptop gets a separate static IP reservation for both its wireless and wired connections.

We relocated the wireless antenna to the top of our mast in order to extend the broadcast range, and boost the router's output power to the maximum within FCC limits for ideal field operation.

5.4.2 Computation

This year Dave was upgraded with a brand-new National Instruments cRIO based control system in conjunction with a complete re-wiring of all of its subsystems. The cRIO is a NI 9014, designed for rugged performance. It has a 400 mhz PowerPC processor, 128 MB of RAM, and 2 GB of persistent storage. It is attached to an NI 9104 reconfigurable chasis that has eight module bays and a 3M gate Virtex-II FPGA.

Dave's computation is spread across three different processors. The architecture includes an FPGA, a real time processor, and a personal computer – the standard student laptop used by our school. The FPGA is a 3 million gate Xilinx XC2V3000, with 96 multipliers and it runs nominally at 40 MHz, but is capable of higher speeds. The RT processor is a 400MHz Power PC processor with 128 MB of memory and 2 GB of storage. The laptop has two Intel Core II protected from the discontinuities of the actual encoder ticks. This raises the abstraction barrier for the midbrain and frees up a huge number of cycles which would be misspent polling digital inputs.

6.2.2 Motor control

Dave's hindbrain also calculates a PID control voltage using reconfigurable gains at 100Hz. The hindbrain takes direct care of the Emergency stop and the green indicator light.

6.2.3 Flow Control

Each region of Dave's brain has software consisting of many parallel processes controlled by a flow control process. Dave's flow control determines when it should turn off and on and when it should become autonomous. The flow control also handles the reset command, the switching of computers, and sometimes what overall behavioral mode Dave is in. The behavior created by the flow control on the hindbrain prevents it from moving any motor until it has explicitly been given the begin signal by the midbrain or by the operator. At this point it switches to the running mode.

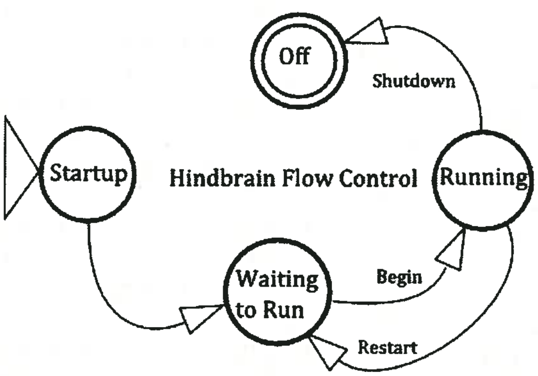

Figure 6: Flow Control

6.2.4 Low Neurons

Dave's brain has four neuron bundles between the hindbrain and the midbrain. They are the efferent and afferent how, where, and what bundles. The act bundle carries the motor control set points from the midbrain to the hindbrain, and also includes flow control information which allows the robot to be turned on and off from the top down. The what bundle holds information about what is around the robot. If the Sick LMS 291 used on Athena were not in Germany being repaired, it would carry processed LIDAR data. This architecture was built in part to be used by future generations of Olin students as an example or template codebase, and in inclusion of placeholder neuron bundles highlights the parallels between the architecture on the different brain levels. The where bundle holds information about Dave's location—it answers the "where am I?" question. Finally the how bundle communicates Dave's state information, including debugging indicators and cable checks. These are an invaluable debugging tool.

6.3 Midbrain

Dave's Midbrain handles the low level obstacle sensing and avoidance which keeps it safe while the forebrain comes up with slower but better thought out plans.

6.3.1 Act arbiter

The midbrain act loop arbitrates efferent neuron bundles which describe which directions it is safe for Dave to travel and how fast it can go in each direction. The bundles contain an array of speed values with indices corresponding to angles within the semi-circle directly in front of the robot. A speed value of one for any of those angles means that whatever behavior generated the neuron bundle does not object to driving in that direction at the maximum speed allowed by other processes.

For example, the bundle from the obstacle avoidance behavior might contain a value of one for every angle—the case for no obstacles, or contain values of one for all angles except a few congruent angle sectors where the value is much lower—the case where there is one obstacle.

For each angle, the arbiter multiplies the values from each of the input bundles to produce the final allowed speed in each angle. The robot will select the angle where it can go the fastest and set the steering angle proportional to this maximum speed angle. It sets the drive speed set point based on the value of the final speed by multiplying the maximum allowed motor voltage by that number. The LIDAR set point will also be set based on this maximum speed so that the LIDAR can be lowered when Dave is moving slowly

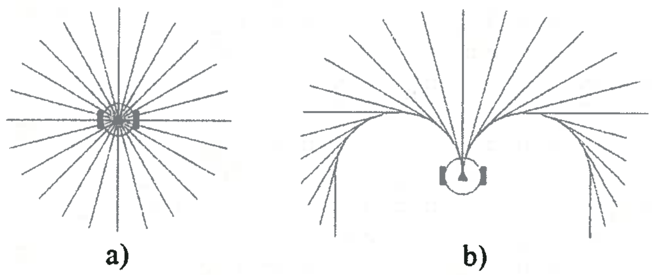

Figure 8 a) represents the original approximation of Dave's movement not accounting for his non-holonomic vehicle dynamics b) represents how vehicle dynamics are modeled with the non-holonomic path reduction algorithm. (1)

The curvature correlates to a radius r_c which can be used to draw a circle whose circumference represents the sharpest turn Dave can make. When combined with exploded obstacle data additional drivable paths can be eliminated depending on an obstacles proximity to these circles. If a circle surrounding an exploded obstacle intersects a maximum turn circle all paths which correspond to driving around the obstacle on the outside are deemed impassable.

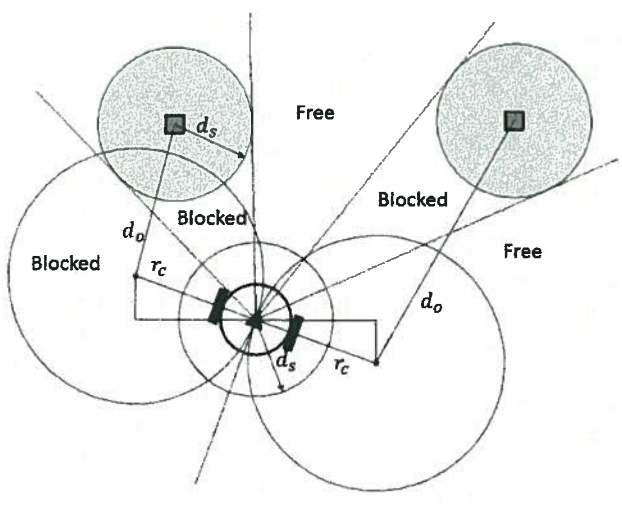

Figure 9 shows how paths are eliminated based on obstacle position combined with vehicle dynamics approximations (1).

Two determine if the two circles intersect the distance from the center of the exploded obstacle to the center of the maximum turning radius circle d_o must be less than the sum of both circles radii. This approximation results in an even more realistic approximation of Dave's drivable paths and allows obstacle avoidance to be performed using an accurate vehicle dynamics model.

6.3.4 GPS Parsing – Where Am I?

Dave receives and processes GPS data through a serial connection using the NMEA protocol. The data is received by the cRIO. There the raw data is parsed and converted into latitude and longitude positions which are then transmitted through a where neuron to other processes on the cRIO and PC.

6.3.5 High Neurons

The information between midbrain and forebrain is handled by a client server pair for each of the four neuron bundles. Transmistting over UDP allows for very fast communication at the cost of lossy transmission. This works well for the majority of the information passed between the two, since old data is not useful on the whole. A notable execption is the mapping algorithm which needs a separate TCP server client pair to ensure data transmission.

6.4 Forebrain

Dave's forebrain runs the slower, lower priority algorithms which make its motion more efficient and wiser than the simple reactive behaviors on the midbrain.

6.4.1 Waypoint Navigation

Dave employs a algorithm to navigate from waypoint to waypoint. Iterating through a list of way points, Dave takes the current waypoint and finds the Cartesian vector between its current location and the goal. If the magnitude is greater than the threshold the angle then the output efferent neuron is a normally distributed function centered on that target angle. If the distance between Dave and the waypoint is less than the waypoint completion threshold Dave moves on to the next waypoint.

6.4.2 GPS to Cartesian conversion

To run the waypoint navigation algorithm Dave transforms the waypoints from decimal degrees to meters. This is done relative to a global GPS reference point defined close to the testing grounds. This allows the transform to approximate the globe as flat. All

6.5 Mapping

Mapping has been added as an additional functionality to Dave for this year's competition. Mapping was broken up into two areas: global and local. Each map is represented as a 2-D occupancy grid whose values indicate the certainty that an object is in that location. The global map is maintained on the forebrain. It keeps a representation of the world surrounding Dave in memory in terms of discrete sections knows as sectors. Each sector is a 6 foot square with a resolution of 4 inches. A 3x3 local subsection of the larger global map is maintained on the midbrain. The local map's values are constantly updated using the most recent LIDAR data. Transmission of map data from the forebrain to the midbrain is done using a TCP connection between the two. This ensures a lossless transfer of sectors.

To account for Dave's movement both maps on the local and global level are "scrolled" according to Dave's position within them. When Dave's position in the map enters a specified update region new sectors will replace the oldest and least relevant ones. In the forebrain this involves reading new sectors from disk and writing old sectors to disk as binary files. On the local map this involves receiving new sector data from the forebrain and sending old sector data back for storage. The old sector data from the local map replaces the corresponding sector data on the global level.

To allow for use over multiple runs, each sector is defined by its distance from a reference position close to Dave's starting location. This allows us to approximate the Cartesian position of a sector from the global origin directly without including the curvature of the Earth. The Cartesian positions of the sectors are used to define the sectors when reading from and writing to disc as well as in the internal local and global map definitions. The global latitude position will be fixed throughout multiple runs to ensure that relative sector definitions are consistent between trials.

6.5.1 Local Map Update

Updating of Dave's local map was based off of the historgramic in-motion mapping method proposed by Borenstein and Koren (2). The method involves manipulating the cells of the local map based upon the certainty an object is located at that point. Once an obstacle is detected its location in the environment is converted into a location on the local map. Those cell values corresponding to that map location are then incremented by a specified value. To prevent accumulation of noise in the map all the cells in the straight line from that obstacle to Dave are decremented with another specified value.

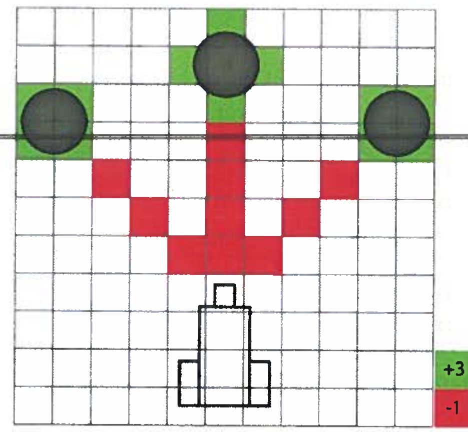

Figure 11 Representation of the incrementing and decrementing method used to populate the local map.

As suggested by Borenstein and Koren our implementation uses values of +3 and -1 to respectively increment and decrement the local map. This method creates a topographic representation of the world where hills represent high obstacle certainty and valleys represent drivable paths. In addition to incrementing and decrementing the map based on object positions, cells which Dave travels through will be reinitialized to their default value. Since Dave is able to drive through the cells there is a near 100% chance they are free and as a result should receive a value accordingly.

When Dave moves enough to require a new sector of the map to update the local representation it will update the local map with the appropriate sectors from the global map. The sectors in the local map which are scheduled to be replaced will then be sent up to the global map and replaced according to their globally defined positions.

Metric	Brian (2009)	Athena (2010)	Dave (2011)
Line Avoidance Reaction Time	0.1 s	0.067 s	0.1 s
Obstacle Avoidance Reaction Time	0.1 s	0.014 s	0.05 s
Localization Update Rate	2 Hz	50 Hz	50 Hz
Local Map Size	NA	NA	114 Kb
Global Map Size	10 Kb	NA	1.5 Mb
LIDAR Update Rate	10 Hz	70 Hz	20 Hz
Obstacle Detection Range	3 m	7 m	9 m
Heading Update Rate	2 Hz	50 Hz	50 Hz
GPS Watch Circle	3 m	3 m	3 m
Top Speed	4.5 mph	4.5 mph	9.5 mph
Average Operating Speed	0.5 mph	3 mph	3 mph
Ramp Climbing Ability	15% grade	15% grade	15% grade
Running Battery Life	1.8 h	1.8 h	1.8 h
Standby Battery Life	10 h	14 h	14 h

Table 2: Key Performance Metrics

Dave is based on the Chimp—platform by Doran Electric Vehicles. This personal transportation vehicle (PTV) is designed to carry a 300 pound rider at 11 mph on grassy or paved surfaces. The additional electric and mechanical systems on Dave total less than 100 pounds. Therefore, we expect performance well within the base vehicle's specifications. In our testing, we have never been limited by the overall power and speed of the base vehicle.

Each individual behavior was tested in a controlled environment over a range of basic test cases. Further integration will take place in the weeks to come on the Olin robotics test track at which point we will get a more complete performance measure.

8 Conclusion

Dave represents a drastic improvement over the team's 2010 entry. Dave's carefully engineered mechanical and electrical systems add durability while providing a modular base for future development. Dave's novel use of a biologically based architecture simplified and clarified the prior's year's multilevel architecture implementation. The implementation of a new obstacle avoidance algorithm incorporating Dave's vehicle dynamics has increased the readability of its avoidance behaviors. The addition of a mapping has increased the reliability of obstacle avoidance on a local level and allowed for the development of higher level path planning algorithms. Similar improvements in image filtering have resulted in more reliable and robust line detection. The result is a vehicle whose reliability, modularity, and functionality exceeded all previous years. However, the ultimate proof of these methods will be her performance at competition. The Olin College Ground Robotics Engineering group proudly presents the college's 2011 IGVC entry: Dave.

9 References

1. *VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots.* **Ulrich, Iwan and Borenstein, Johann.** Leuven, Belgium : IEEE, 1998.

2. *Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance.* **Borenstein, J and Koren, Y.** 1991, IEEE Journal of Robotics and Automaton, pp. 535-539.