

Engineering Design Team

Ali Attarwala	BIOE	Sophomore	Dean Khan	BIOE	Junior	Joseph Rodriguez	CS	Freshman
James Bittler	ME	Junior	Edwin Kim	ME	Freshman	Rayna Saito	BIOE	Senior
Matthew Byrne	CS	Sophomore	Amit Kotadia	EE	Freshman	Jaspreet Sethi	CS	Freshman
Daniel Egerton	EE	Senior	Anup Kotadia	EE	Senior	Avner Shanan	CS	Junior
Marios Fanourakis	EE	Sophomore	Karthik Kotha	CS	Junior	Nick Shaskevich	CS/ChE	Junior
Dalia Garcia	CE	Senior	Nick Kutella	EE	Senior	Travis Strutzenberg	ME	Sophomore
Garrett Graham	ME	Junior	Nick Novak	EE/ME	Senior	Wai Yin Szeto	ME	Freshman
Martin Hamper	ME	Junior	Sanket Patel	ME	Junior	Curtis Warren	CS	Freshman
Adela Isovich	CHEM	Sophomore	Steve Piasecki	ME	Sophomore	Keith Weber	EE	Sophomore
Archit Joshipura	CE	Junior	Rafal Radulski	CS	Senior	Jonathan Wrightsell	EE	Senior

The engineering design in this vehicle by the current student team has been significant and equivalent to what might be awarded in a senior design course.

Faculty Advisor Certification Signature: *Robert Cottrill*

1 Contents

1	Contents	1
1	Introduction	3
2	Design Process	3
2.1	Organization	3
2.2	Implementation	3
2.3	Testing	4
3	Mechanical Design	4
3.1	Mechanical Innovation	5
3.2	Chassis	5
3.3	Vehicle Interior and Compartments	5
3.4	Vehicle Exterior	5
3.5	Enclosure Ventilation	6
3.6	Drive Train	6
3.7	Paint	6
4	Electrical Design	6
4.1	Electrical Innovations	6
4.2	E-Stop	7
4.3	Headlights	7
4.4	Power Generation	8
4.5	Motion Control	8
4.6	Embedded Drive Train Control System (EDCS)	8
4.7	Computers and Computer Mounting	9
4.8	Signal Multiplexer	9
4.9	Sonar Sensors	9
5	Software design	10
5.1	Software Innovation	10
5.2	System Integration	11

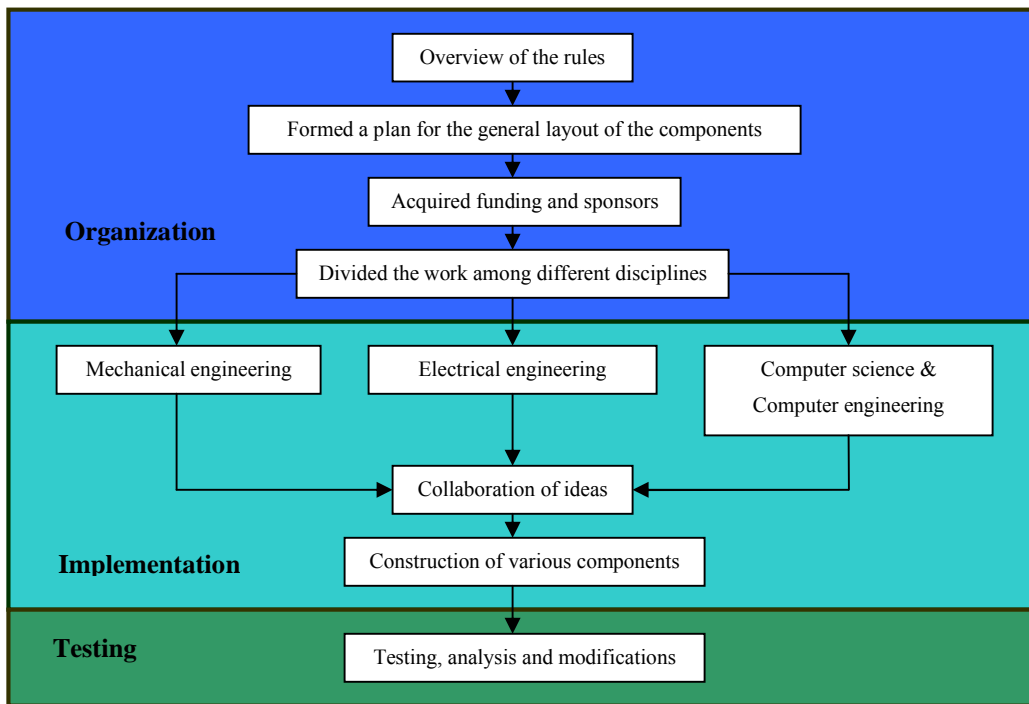
5.3	Drive Control.....	11
5.4	Stereo Vision Disparity and Recovering the Z value	11
5.5	Object Detection.....	12
5.6	Bounds Detection	13
5.7	Multiple Sonar Scanner Explanation.....	13
5.8	Goal Planning: Map Based.....	14
5.9	Goal Planning: Image Based	14
6	Vehicle Cost.....	15

1 Introduction

Enyo is a fully autonomous robot built by the Engineering Design Team at the University of Illinois at Chicago. An autonomous robot is a complex system of software, electrical, and mechanical engineering. To build Enyo, many engineering disciplines collaborated on this robot. This report details the design process, predicted performance, mechanical, electrical, and software components of Enyo.

2 Design Process

This diagram details the design path followed to construct Enyo.



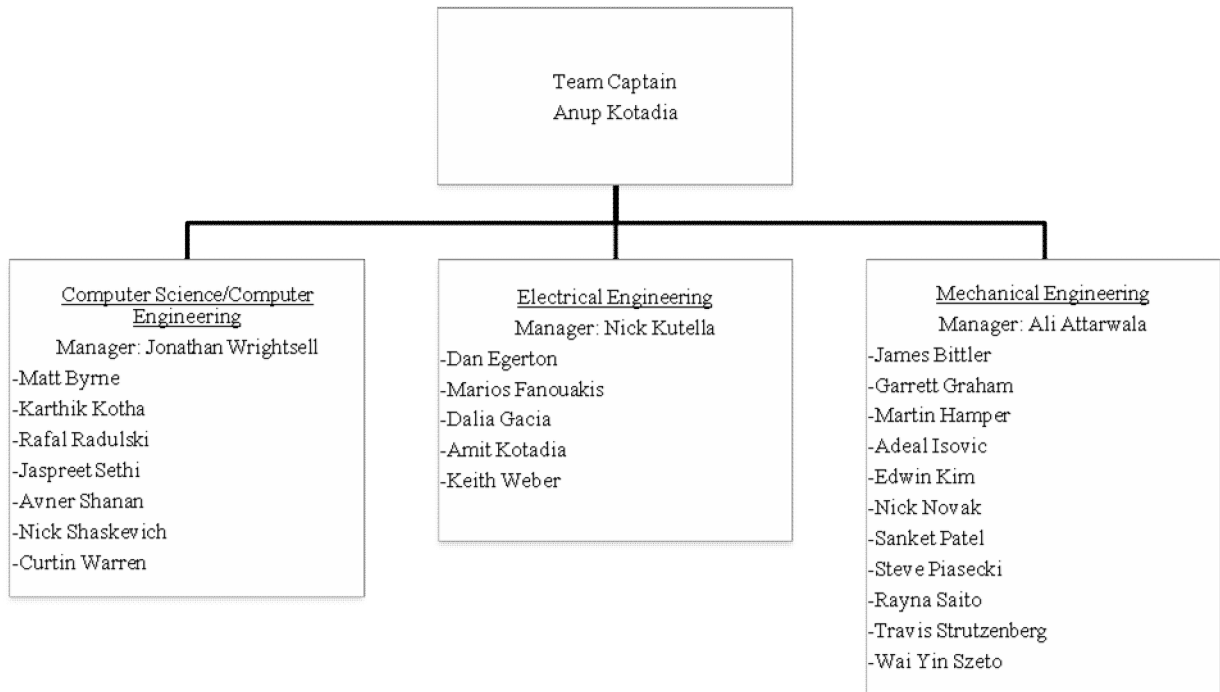
2.1 Organization

As a team, the rules of IGVC were reviewed and discussed. A timeline of approximately 9 months was given to complete this task. The task consisted of everything needed to build an autonomous robot, including all of the main hardware and software components. The preliminary budget for the robot was chosen to be \$8000. The team then gathered money and donations from within the university and corporate sponsors.

2.2 Implementation

The figure below clearly outlines how the work was managed. The team was divided into three main groups. These groups were a mechanical engineering group, an electrical engineering group, and a software group. Frequent

meetings of all group members were held to share ideas and progress with other engineering disciplines. These meetings helped to assure that Enyo would work with no major design flaws.



During idea collaboration and construction, particular emphasis was put on the safety, durability, and reliability of Enyo. Safety was always an issue, to make sure there were no fires from faulty electrical wiring and so that Enyo could be shut down quickly if acting erratically. Durability was another key concern. A long life span was a priority to provide a working platform for the generations of students to come.

2.3 Testing

Testing and analysis of the new design is the longest and most important. The goal was to assemble a working robot no later than a month before the competition to allow for testing. Priority was given to some systems so that particular components could be tested before this deadline. Currently Enyo is in this stage of the design process.

3 Mechanical Design

The main goals for designing the vehicle were simplicity, safety, durability, and reliability while being able to achieve its objective of enclosing and protecting all of the systems necessary for autonomous navigation. Simplicity in design allows for undemanding manufacturing processes, reduction in costs, and minimization of potential failures. Accessibility is crucial for trouble free repair and maintenance. Durability is necessary to represent a real world application. Lastly, reliability of all components was worked into the design.

3.1 Mechanical Innovation

The design of Enyo was meant to be an improvement in efficiency and manufacturability over the previous design robot, Achilles. Increased accessibility was a major concern for the new chassis as well as more ground clearance. The frame is made up of two parts, a top and bottom frame. The top frame is constructed of T-slot aluminum, which allows quick design and assembly of new a top chassis if needed. The bottom of the frame is 8" from the ground to clear small obstacles. This height also allows for clearing a hill with a maximum of a 30-degree incline.

3.2 Chassis

The vehicle frame is constructed out of alloy 4130 military grade 1" by 1" steel tubing and 1" by 1" 80/20 T-slots. The steel tubing was chosen due to our experience with the material and its relative ease of use. The T-slots allow for a flexible design that can be easily adjusted should the need of a modified design arise. The T-slots also allow parts to be attached without the need of drilling holes. The vehicle's frame is a demonstration of a design that can be very rugged, durable, and strong, while also keeping it lightweight.

3.3 Vehicle Interior and Compartments

With all of the considerations for what was going inside, it was important to make sure that there was adequate space for accessibility. The majority of the components that comprise the vehicle are contained inside the frame and it is important to be able to access them conveniently. To accomplish this, the frame is partitioned into a lower and upper compartment. The lower compartment was designed specifically to house the batteries, motors, motor controllers, and the drive train. All of the heaviest components are nearest to the ground to lower the center of gravity. The upper compartment houses the computing equipment and generator. These two compartments are easily separable for access to the vehicle interior.

3.4 Vehicle Exterior

With all of this open space, there comes the problem of exposure to the elements. An important objective is to allow the vehicle to be able to operate while it is raining. With that in mind, the vehicle needs to be water tight, but at same time be easily accessible. The frame is covered in sheets of polycarbonate and aluminum. Sheets of aluminum, $\frac{1}{8}$ " thick, are used to cover the bottom of the vehicle. It provides enough material to sound protection while allowing the vehicle to maintain its lightweight performance. The polycarbonate is painted to prevent direct exposure to the sun's rays causing unnecessary heating. Similar to the aluminum, it serves as protection without adding much weight. The covers are attached directly to the frame using weld nuts and machine screws, directly mounting them to the frame. There are rubber gaskets in between the polycarbonate and the frame to ensure water resistance.

3.5 Enclosure Ventilation

One of the major concerns is keeping the vehicle cool. Heat must be exhausted, but the vehicle must also be kept watertight. Cool air convection by fans is the method used to cool the vehicle. Ten fans are used to cool the vehicle. The fan inlet and outlets are covered by screens, to prevent water from entering.

3.6 Drive Train

The vehicle features a sprocket and chain drive train. The chain driven system allows freedom in placement of the motors and wheels. On each side of the lower compartment is the drive train for that side. Each wheel has its own motor enabling the wheels to turn in opposite directions. This tank drive provides a much needed zero turning radius for ease of obstacle avoidance. Choosing the tank drive therefore not only makes the navigation of the robot less computationally intensive but also reduces the mechanical complexity of the system as opposed to using steering.

3.7 Paint

Enyo's side panels are made of polycarbonate and thus we used spray paint designed for plastic. The design is a camouflage design based on military desert camouflage. We have a three-color design, using a light tan as the base and olive green and a darker brown for the swatches. The swatches are all smooth and horizontal, as to better hide in a desert environment.

4 Electrical Design

Enyo's wiring is concise and simple. The electrical components were located at specific locale for simplification and efficient use of space. All of the circuit boards and sensors are modular so if a malfunction occurs, easy isolation and repair is possible.

The batteries are positioned in the bottom of the robot for a center of gravity close to the ground. It also minimizes average distance to all powered components. All power cables and signal wires are run through wiring duct to free up Enyo from cluttered wires. The circuit boards for interfacing with the computers are placed near their respective wheel encoders in order to reduce wiring. All high current systems are protected by circuit breakers. The circuit components are secured with Velcro to enable easy manipulation of circuit location and accessibility.

4.1 Electrical Innovations

The main electrical innovations in Enyo are based on reliability and safety. All systems are designed to be safe by failing gracefully. If any system fails, the robot immediately stops moving. In addition, all electrical systems have been made more reliable by employing watchdog timers. This ensures that if a system stops working, the system restarts itself automatically. This allows the robot to recover from error conditions without any human intervention. In addition, all circuitry is composed on custom circuit boards that are manufactured in the team's workshop. This ensures that the circuits are as small and robust as possible while cutting costs.

4.2 E-Stop

The emergency stop system (E-stop) is a system designed to stop the robot quickly in emergencies. The user may activate the E-stop in two ways, either by a wireless remote control or by pressing an onboard switch. The E-stop is composed of two modules: a handheld transmitter unit that wirelessly transmits a "STOP" signal to the robot and a receiver unit onboard the robot that receives the signal from the transmitter and is responsible for cutting power to the motors. The receiver unit also monitors the onboard switch to determine whether to activate the E-stop. The transmitter unit has a power switch and a red pushbutton switch that when pressed, activates the E-stop. The behavior of the E-stop is that if either the onboard switch or the wireless transmitter provides a STOP command, the robot will stop moving. The robot will resume movement only when both sources provide a GO signal. The heart of both modules is an Atmel ATmega8 microcontroller running at 16 MHz. In addition, a Radiotronix RK-Wi.232FHSS-250-FCC-R wireless module is present on both E-stop modules to allow wireless communication. The wireless module uses spread spectrum technology to ensure data encryption and prevent interference or jamming from external sources. In addition, the wireless module has a range of 5 miles in open-air line of sight, which is substantially greater than the range of 50 feet required by the official rules. The E-stop transmitter operates by continuously sending a signal to the receiver module through the wireless module. The signal will be a stop signal if the red button is pressed or a go signal otherwise. The receiver module gets the signal and based on the received signal and the state of the onboard switch, the receiver will decide on whether to cut power to the motors and thus stop the robot. The E-stop cuts the power by deactivating a relay that is in series with the motors. The relay is active in normal operation. Furthermore, the E-stop can recognize a lost connection between the transmitter and receiver and stops the robot as a safety precaution until the connection is restored. The E-stop is able to stop the robot in less than a half a second of activation. All cables and wires that are part of the E-stop employ strain relief to increase the mechanical reliability of the system. The receiver module accepts a power supply of 12 volts and the transmitter uses a 9V battery as its power source. Both modules employ a watchdog timer as a safeguard to recover gracefully from equipment failure within 15 milliseconds. The transmitter module has indicator lights for power and indicating whether the transmitter is issuing a stop command. The receiver module has indicator lights on the main robot control panel showing the status of the onboard switch, the transmitter signal, or lack thereof, and whether the E-stop is currently stopping the robot. These lights allow quick inspection to verify correct operation of the E-stop system.

4.3 Headlights

The robot has a headlight system that serves the purpose of maintaining an adequate amount of visibility for the robot to function. The purpose of the headlight unit is to switch on automatically when the environment lighting becomes too low for the robot to assess its path properly. Two Northern Industrial Halogen 12 Volt Spotlights, each rated at 55 Watts, provide the necessary lighting. A custom circuit configured specifically for operating the headlights controls the headlight unit.

4.4 Power Generation

Enyo has two sources of power. The circuitry and motors are powered by two 12 Volt 26 Amp Hour sealed lead acid batteries arranged in parallel configuration. Lead acid batteries were chosen due to their high power to weight ratio and relatively low cost. The computers require AC power and so a 2000 W gasoline generator is used to power the computers. The decision to use a generator was made because the computers consume a large amount of power and a prohibitively large amount of batteries would be needed to power the computers for an extended period.

4.5 Motion Control

The vehicle uses two 3 HP 12 volt DC motors which are controlled by PWM motor controllers. The motors are connected to the motor controllers using 10 gauge wires. The motor controllers that are being used are the Victor 883's, which have a rating of 24 volts at 60 amps of continuous current. The motor controllers receive a PWM signal from the Embedded Drive Train Control System. The speed will be controlled by adjusting the duty cycle on the PWM, which will in turn adjust the speed of the motors proportionally.

4.6 Embedded Drive Train Control System (EDCS)

The EDCS is a system designed to interface the software controls system to the motors that drive the robot. The system consists of a module for the left side of the robot and a module for the right side of the robot. Each module is responsible for measuring the wheel speed and controlling the power level of the motor for its side. An Atmel ATmega8 microcontroller operating at 16 MHz is the heart of each module. Each module communicates with the software controls system via RS-232 serial communication. Each module measures the wheel speed of its respective side with a US Digital HD25 shaft encoder. The encoder produces two pulse trains with frequencies linearly dependent on wheel speed. The pulse trains are 90° out of phase and the leading of one pulse train with regard to the other indicates the direction of wheel motion. The encoder sends the two pulse trains to the microcontroller and the microcontroller uses the pulse trains to determine the wheel speed and direction of rotation. The microcontroller sends this data to the software controls system through the serial communication interface in the form of an integer at a rate of 20 times a second. Positive values of the integer indicate forward motion, negative values of the integer indicate reverse motion, and the magnitude of the integer indicates the magnitude of the wheel speed. Furthermore, each module continuously listens for commands to change the amount of power supplied to the motor under the module's control. The power level commands take the form of an integer sent to the module by the software controls system through the serial communication interface. Positive values of the integer indicate that power should be applied to move the motor forward, negative values of the integer indicate that power should be applied to move the motor backwards and the magnitude of the integer indicates the amount of power to deliver to the motor. The microcontroller converts this data to a pulse width modulation (PWM) signal and sends it to the motor controller. The motor controller uses the PWM signal to provide the correct amount of power to the motor. All cables and wires that are part of the EDCS employ strain relief to increase the mechanical reliability of the system. The EDCS has a female 9-pin serial port to connect the EDCS to a serial cable. A standard serial cable and an USB to serial adapter make the connection between the EDCS and the computer running the software controls system. A Maxim

MAX232 IC interfaces the serial communication voltage levels with the voltage levels for the microcontroller. Each EDCS module accepts a power supply of 12 volts and employs a watchdog timer as a safeguard to recover gracefully from equipment failure within 15 milliseconds. Each module contains a power indicator LED and LED bar arrays to allow rapid inspection of the module to verify proper operation. The LED bar arrays display motor power levels and wheel speeds as bar graphs to indicate the magnitude of their values in real time.

4.7 Computers and Computer Mounting

The robot contains two computers to allow parallel execution of the software and increase performance. One computer will run the vision system and the other will run the hardware control system as explained in the software section. Both computers use the ASUS P5E LGA 775 Intel X38 ATX Intel motherboard. This motherboard was chosen for its ability to handle components that produce more heat when overclocked. Each computer uses the Intel Core 2 Quad Q8200 CPU. Intel's quad core processors give the best performance because four cores allow four processes to run concurrently on each computer. Our software architecture is designed such that each component runs as its own program. These programs run simultaneously and require their own thread of execution.

Each computer has an 800-watt power supply unit to provide electrical power and meet the GPU and CPU power requirements. The power source for both computers is a Honda EU2000iA gasoline generator. A gasoline generator is used because batteries cannot supply sufficient power for long durations.

Both computers have 8GB of RAM to handle large amounts of data efficiently. Each computer has an 80GB laptop hard drive. Laptop hard drives are used because they are engineered to handle vibrations. Each computer uses the Ubuntu Linux 64-Bit operating system that enables the use of 8GB of RAM.

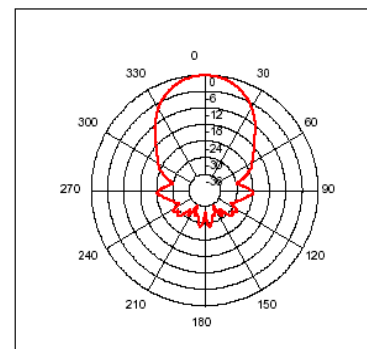
The computer that runs the vision system has an ATI Radeon HD 4870 X2 GPU. This card uses two ATI Radeon HD 4870 GPU's combined allowing faster processes for the vision system. The other computer has a NVIDIA Ge-Force 9800GTX+ to serve as a backup GPU.

4.8 Signal Multiplexer

This system allows the robot to be controller either autonomously or by radio control. This system has three modes. One mode allows the robot to be exclusively autonomously controlled. The second mode allows the robot to be exclusively controlled by remote. The final mode allows the robot to switch between autonomous and radio control by the switch of a button on the radio control. This allows quick restoration of control by a human if the robot malfunctions. In addition, the system recognizes the absence of a radio signal and stops the robot as a safety feature.

4.9 Sonar Sensors

Connected to the AVR board are four ultrasonic sonar sensors that can identify objects between 3 centimeters and 6 meters. The figure on the right shows the beam pattern of the range finder. This system is used primarily to detect objects at close range. This system also contains a circuit that



senses any loss of power, preemptively warning of impending failure, and shuts down the motors to prevent the vehicle from running haywire.

5 Software design

Robotic software is often times complicated and computationally intensive. For both of these reasons, a divide and conquer design philosophy in which each individual problem is solved in a separate module was implemented for the robot. To speed development of a rather complicated series of components, existing programs and frameworks were adapted to suit the needs of the robot. The algorithms used to solve each part of the problem were researched thoroughly and planned to ensure seamless integration and accomplishment all of the required tasks.

5.1 Software Innovation

The most significant innovations to Enyo come in the software portion. Multiple software components and techniques give Enyo an edge over other robots.

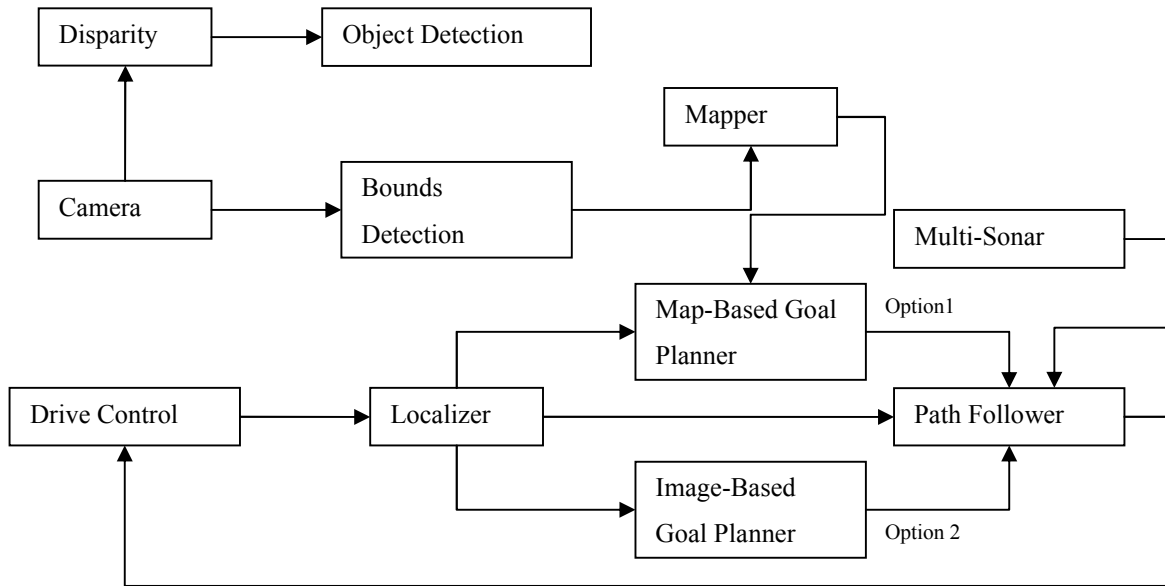
Stereo vision enables Enyo to see further and get a better understanding of what obstacles lie ahead. The types of obstacles Enyo can see are drop offs, switch backs, and past obstacles.

The graphics processor creates the disparity map. This technique provides an implementation 4 times faster than the next competitor does. Having the quick disparity map allows for better reaction times.

Look-ahead path planning is used with a probabilistic understanding of location of Enyo and obstacles. Enyo does not just obviously follow the lines and avoid obstacles along the way. It creates a map of its surroundings and intelligently chooses the best path to take.

An innovative framework to avoid duplication of effort called Orca is used. Orca was written by universities in Berkeley, CA and Sydney Australia specifically for the DARPA Urban Challenge. Enyo uses some of their framework and components that implement well known and tested algorithms such as VFH+, Wavefront Propagation, and Skeleton path planning.

5.2 System Integration



5.3 Drive Control

The software component Drive Control uses a proportional integral derivative (PID) control algorithm for precise control based on a target velocity. This algorithm takes, as input, a target velocity ["setpoint"], and the current velocity of the wheel ["process-variable"]. The output is then the sum of the PID transfer functions multiplied by the gain. Gain values are chosen for a fast rise time, minimal overshoot, and minimal steady-state error. This component interfaces with the EDCS to complete the interface between software and hardware.

5.4 Stereo Vision Disparity and Recovering the Z value

In the field of machine vision, it is a common problem to be able to take what a camera sees and translate it to world coordinates. A single camera standing still does not provide enough information to recover the full world coordinates of what it is seeing. Stereo vision was chosen for its ability to recover world coordinates without movement. It was also chosen for its ability to be used in conjunction with line detection to map lines in 3-D. This additional information could then later be used to preplan paths far in advance.

In stereo vision, two parallel cameras are aligned such that they lie on the same horizontal plane. The physical distance between the centers of the two cameras is used in the transform. Each image is treated as a two dimensional matrix of three values: red, green, and blue. The upper leftmost corner is designated the index (0, 0). The bottom right most corner is the largest coordinate value in the image. A point in the world will then be mapped in to the two images differently. There will be a horizontal distance in pixels that the point in the world is shown in the left and right images. This distance is known as disparity. By calculating the distance for each point in the image, a disparity map can be formed. This is a 2-D matrix, or image, that contains integer values with the disparity at that particular pixel. It can be displayed as a grayscale image as shown above. White color pixels are close and gray color pixels are farther away.

The cameras themselves do not perfectly conform to the perspective transform by default. The perspective



transform assumes a pinhole camera is being used and thus no lens distortion. Lens distortion, or barrel distortion as it is sometimes known, causes straight edges in the world to appear curved. Correcting for the distortion to conform to the needed pinhole assumption is called rectification of the images. A commercial stereo camera was

used that came pre-calibrated with easy to use C function calls to rectify the images. Once the distortion is gone correlation between pixels in the left and right images (shown above) must be done to determine which pixel in the left image matches a pixel in the right image. Usually this is done by taking the absolute difference of intensity of each pixel along a row in one image with respect to a single pixel in the other. This creates a 3-D matrix of absolute difference values from which a minimum value may be determined. The minimum value is determined for each x, y coordinate in the matrix, the z index which contains the minimum value at the point is the disparity. The two images will never necessarily contain the same exact value at a matching mapped point due to noise and variation in lighting. There might be other pixels in the same row that contain the same value as the current pixel. Either way an incorrect disparity choice for a given pixel is easily made. To correct for this error several steps are taken.

First, a weight window is applied at each pixel to determine the difference. So instead of just taking a single difference value at a single pixel, several neighboring pixel differences are weighted in on and summed together to form a new difference value for a given point. This software uses a Gaussian mask of 5x5 to determine the difference value for a given point.

Second, a 5x5 median filter is applied to the disparity map image. This eliminates stray values, what is known as salt and pepper noise. This creates a very smooth grayscale image that can then be used to determine the z values for each point.

The number of calculations to develop a disparity map adds up very quick with even relatively small images of 320 by 240 pixels in dimension and checking for 32 disparity values. The graphics card is therefore used to speed up these operations dramatically. By using a modern day graphics card to do the image processing, in this case the Stereo Vision component achieves about 30 frames per second, or 30 Hz. A similar implementation done on the systems normal x86 processor from the commercial vendor who supplied the camera was only able to achieve around 8 frames per second, or 8 Hz.

5.5 Object Detection

With the output of the disparity image as well as line detection results, our code processes these inputs to determine the presence of obstacles. While the disparity images relays depth & the presence of objects, the line detection relays what the boundaries are. We iterate over the disparity image in a rectangular fashion and take the average of the pixel values to determine a normalized "depth" for that region, as the grayscale image directly depicts the "z" value. The lighter the average of this rectangular cell, the closer a certain object is; a threshold value

indicates whether the object is close enough to acknowledge or disregard it. After taking all cells that exceed this threshold, object detection places these cells on an occupancy-grid map, where the average color of each cell depicts how far away that "object" is located.

5.6 Bounds Detection

Bounds detection is achieved by exploiting the small range of colors that constitute boundaries. For each pixel in the image received from the camera, a value is calculated which represents, roughly, the likelihood that said pixel is part of a boundary. This value can be derived from a simple formula, with the red, green, and blue components of the pixel given as inputs. In essence, the algorithm calculates how close the pixel is to gray, and discards pixels that have too much color or are too dark. The result of this process is a grayscale image that represents the boundaries of the course from the camera's position. The picture shows an example of the bounds detection input, intermediate results, and output.

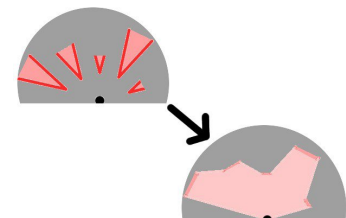


The benefits of this algorithm versus other approaches are twofold. First, due to its simplicity, this algorithm is extremely easy to understand. In addition, because of this simplicity, it also has very acceptable performance characteristics. Secondly, this algorithm is extremely consistent in terms of behavior: unlike other, more complicated approaches, there are very few potential corner cases for this algorithm. This observation will make debugging much easier, and ensures that the line detection component remains defect-free

Of course, this algorithm does have disadvantages. Largely, these problems revolve around things that cannot be controlled in software: the peculiarities of the course (especially regarding color), the quality of the camera output, the weather on the day of the contest, etc. Unfortunately, it is difficult to assess whether these problems will crop up in practice or not. Due to the lack of any hard data on these issues, this algorithm was chosen because of its simplicity. Should it be necessary, substituting a different algorithm is simple.

5.7 Multiple Sonar Scanner Explanation

Once the physical setup and parameters of each scanner are known, software algorithms process the readings from each scanner to construct a 180-degree view of how much clear-space is in front of the vehicle. Since each scanner has a limited beam-width, either there may be overlapping between two scanners' readings or there may be gaps depending on the beam-widths and spacing. For a certain angle "A" within the 180 degree view ahead, we have chosen to average



the two readings and report this average as the maximum distance of clear-space when two or more scanners overlap

a certain region. We have also chosen linear regression methods to fill in the gaps from surrounding scanners' readings where no scanner has absolute coverage. The firing pattern of the array can also be manipulated on demand – this may be necessary if interference becomes an issue. These estimates provide adequate information to goal planning algorithms in order to achieve an acceptable level of close-range obstacle avoidance. The figure to the left shows how five discrete scanner readings are interpreted as an area of “free-space.”

5.8 Goal Planning: Map Based

The final step is Object Mapping. This component creates an overhead view of the 3-D map just created by the Position Interpolation. A bird’s eye view of the course ahead, provides a base for the following goal-planning component.

The goal-planning component determines the general flow of the course and defines a sequence of points from the robot to the goal. The input is an occupancy grid map with only lines and a separate image with only objects are preferred. An occupancy grid with relative robot position is required. The output of this component will be a sequence of points as defined by a configuration file. The file will list the number of points that are required to be outputted by the component.

If given the boundaries of the course through the line detection component, a general direction (slope of the lines) can be found. A goal is set at an arbitrary distance along the imaginary line that represents the desired direction to move in. This goal is placed sufficiently far enough from the robot so that hopefully the goal is beyond the reliable set of data currently available to the robot. This means this component does not have to deal with specific goal-planning strategies such as choosing one path from many or determining that a goal is out of bounds. With a goal set, the robot then determines a set of points that will minimize the chance of the robot touching any obstacles. The robot will then divide these points into the number given to it by the configuration file and send it to the component LocalNav that will direct the movement of the robot to each of the points. Since the goal is outside of the reliable set of data, the goal planner component will update the set of waypoints frequently to ensure an accurate path. If an obstacle is detected between refreshes, the updated path will be able to include the newly detected object. This component will also allow the robot to take into account where it has been in recent history in order to estimate the next goal appropriately. Thus going backwards only occurs in an extreme situation. Since the goal is sufficiently far away, the robot should not be required to go back on its self unless the robot receives insufficient data. This also allows room for error since the goal planner component continually refreshes. Unfortunately, the success of this component is dependent entirely on the effectiveness of the other components, since they produce the data used by this component.

5.9 Goal Planning: Image Based

Image Goal Decider attempts to choose the best next destination for the robot based on the input image. The image is analyzed based on colors. The algorithm detects pixels and areas of image containing white lines and obstacles. Grass color is considered safe ground. Based solely on this information, the component gives a suggestion on where the robot should go.

6 Vehicle Cost

2008-2009 Intelligent Ground Vehicle Competition (IGVC) Budget Report Summary

Manufacturer	Part	Unit Price	Quantity	Total Price	Price Paid
Power System					
Power-Sonic	12V 26 Ah Lead Acid Battery	\$38.80	2	\$77.60	\$77.60
IFI Robotics	Victor 883 Motorcontroller	\$150.00	2	\$300.00	\$300.00
Cooper Bussman	40 Amp Auto Reset Circuit Breaker	\$2.12	2	\$4.24	\$4.24
Durakool	100 Amp Relay	\$18.12	1	\$18.12	\$18.12
Honda	EU2000i Generator	\$1000.00	1	\$1000.00	\$1000.00
N/A	Wire (all sizes)	\$40.00	1	\$40.00	\$40.00
Electronics					
Radiotrix	RK-Wi.232FHSS-250-FCC-R Radio	\$250.00	1	\$250.00	\$0.00
Atmel	ATmega8 Microcontroller	\$3.44	6	\$20.64	\$20.64
US Digital	HD25 Optical Encoder	\$250.00	2	\$500.00	\$0.00
Spektrum	DX7 Radio Controller	\$300.00	1	\$300.00	\$300.00
Hemisphere GPS	A100 GPS Smart Antenna	\$1500.00	1	\$1500.00	\$0.00
Point Grey Research	Bumblebee 2 Stereo Camera	\$2000.00	1	\$2000.00	\$2000.00
N/A	Miscellaneous Components	\$100.00	1	\$100.00	\$100.00
Computers and Thermal Management					
Various	Desktop Computer	\$1000.00	2	\$2000.00	\$2000.00
Scythe	DFS123812H-3000 Cooling Fan	\$7.95	10	\$79.50	\$79.50
Link Depot	DVI Cable	\$8.99	2	\$17.98	\$17.98
IO Gear	USB Hub	\$7.99	2	\$15.98	\$15.98
Drive Train					
Palmer Industries	Model 200 Motor	\$533.00	2	\$1066.00	\$1066.00
McMaster-Carr	Sprockets, Gears, Shafts	\$300.00	1	\$300.00	\$300.00
McMaster-Carr	Roller Chain	\$80.00	1	\$80.00	\$80.00
Northern Tool	Wheels	\$42.99	4	\$171.96	\$171.96
Chassis					
McMaster-Carr	Steel Tubing	\$246.84	1	\$246.84	\$246.84
McMaster-Carr	Aluminum Sheets	\$300.00	1	\$300.00	\$300.00
McMaster-Carr	Polycarbonate Sheets	\$150.00	1	\$150.00	\$150.00
80/20 Inc.	T-Slot Aluminum Tubing	\$350.00	1	\$350.00	\$350.00
Total Cost		Retail Cost	\$10,888.86	Discounted Cost	\$8,638.86