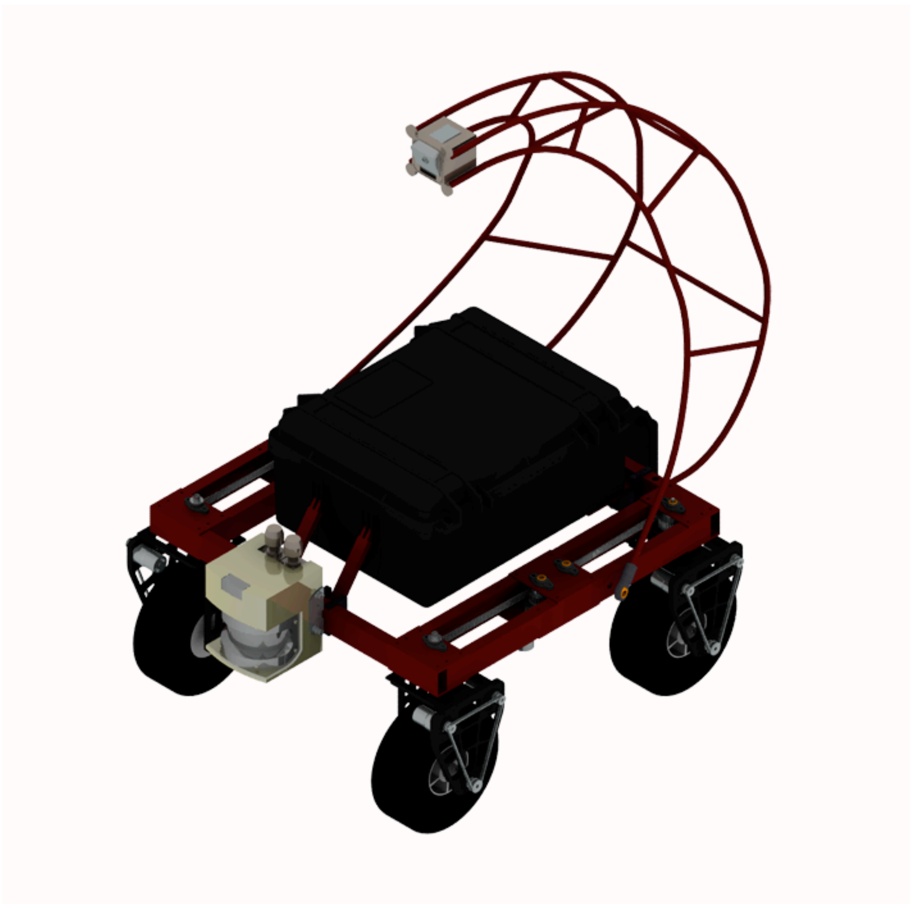
Cornell University
Cornell MineSweeper
Vehicle Design Report



Hamzah Sikander, Franklin Geeng, Steven Gilson, Tanya Gupta,

Michael Hsu, Frank Lee, Andres Mack, and Félix Pageau

Required Faculty Advisor Statement

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

____

Professor Ephrahim Garcia

Department of Mechanical and Aerospace Engineering, Cornell University

Cornell MineSweeper's Gladiator – Minesweeping is no game

Félix Pageau, Justin Charles Yee, Sarah Leung, Philip Stathis, Abe Cantwell, Tanya Gupta

Cornell MineSweeper project 2007-2008

Cornell University

Abstract

The Cornell Minesweeper project proudly presents Gladiator, the Intelligent Minesweeping Ground Vehicle that will be Cornell University's first entry at the Intelligent Ground Vehicle Competition. This modular robot featuring a Zero-point turn capability has been built from scratch by 33 Cornell's undergraduate students.

Keywords- unmanned ground vehicle, intelligent vehicle, mine detection, IGVC

1. Introduction

Cornell MineSweeper (CMS) project was founded in 2006 to develop an autonomous vehicle that can accurately detect land mines and facilitate their clearance. To validate the design of the autonomous robotics platform on which the mine detection sensors will be mounted, we present the Gladiator vehicle platform that will be Cornell University's first entry at the International Ground Vehicle Competition.

After developing a few test platforms, 31 undergraduates were divided into five subgroups (Frame, Control, Computing, Power and Business) that agreed to work together building a first full-blown ground robot. Two first-year engineering graduate students also joined the team; one advises on systems engineering while the other advises on management topics.

The Gladiator vehicle, featuring an innovative zero-point turn capability has been mostly built in-house by Cornell engineering undergraduates.

2. System design

After few component integration difficulties, the Cornell MineSweeper team opted to use a Systems Engineering process to handle the complexity of multi-disciplinary engineering tasks.

2.1 The system process

The International Council on Systems Engineering (INCOSE)'s systems engineering process[1] has been tailored to create the Gladiator's development process.

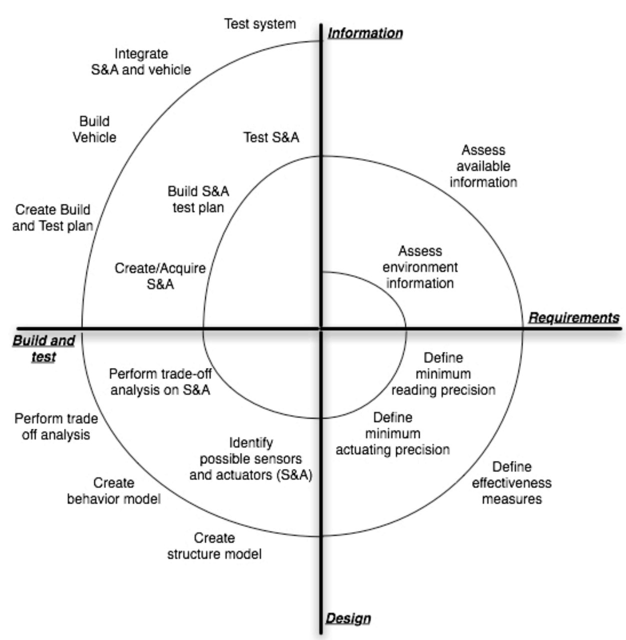


Figure 1 Intelligent Vehicle SE process

Figure 1 The Intelligent Vehicle SE process shows the two-iteration development process used to develop Gladiator. The first iteration focuses on the interaction between the robot and its environment through its sensors and actuators; while the second addresses the vehicle design elements. The Intelligent Vehicle SE process uses the Dive-And-Surface approach to subdivide the vehicle into subsystems and then do a bottom-up integration to bring everything together.

2.2 Objectives for the 2008 vehicle

For this year, in order to efficiently begin the project and ensure that all subgroups and team members are on the same page, the team established a series of eight objectives for the Gladiator platform:

1. Modular system
2. Size inferior to 1.2 m X 1.2 m X 2 m
3. Less than 35 kg
4. Four-wheel drive
5. Zero-point turn
6. Maximum speed of 8 km/h (5 mph)
7. Electronic case shall be IP54
8. Decision rate at 10 Hz

2.3 System overview

Using the Intelligent Vehicle SE process, the intelligent ground vehicle system was broken down into subsystems and sensors/actuators assemblies. Figure 2 below shows the vehicle design and a close-up view of its sensors.

Table I Sensor overview

Sensors	
Lane detection	Unibrain Fire-i camera
Obstacle detection	Sick LMS291 LIDAR
Odometry	US Digital E4P encoder
Orientation	US Digital MAE4 encoder
Position	Microbotics MIDG-II INS

Table I above provides the system with information about its environment. A Kontron 986LCD/m Core2Duo 2.2 GHz embedded computer is doing the data acquisition, processing and fusion for camera, the LIDAR and the INS while an AT Mega 2560 micro-controller uses the encoders to control the vehicle based on directions determined by the path planning on the computer. The micro-controller then manages the actuators from Table II .

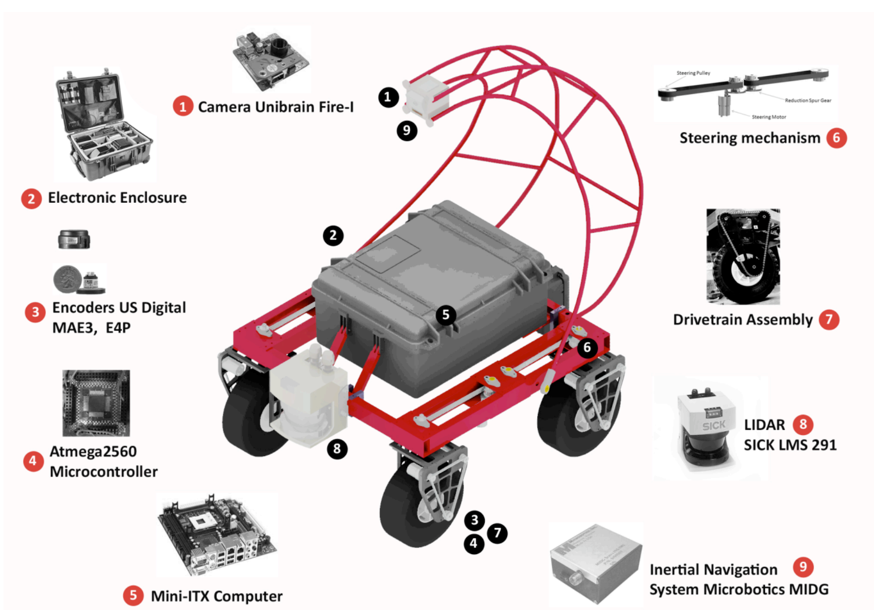


Figure 2 Gladiator’s sensors and actuators exploded view

Table II Actuator overview table

Actuators	
Propulsion	4WD chain driven 25cm wheels
Prop. motor	Maxon A-Max32 & 86:1 gearhead
Prop. info	4 independent wheels
Steering	2x independent steering pods
Steering motor	Maxon A-Max32 & 246:1 gearhead

2.4 Innovations and predicted values

The goal for the Cornell Minesweeper project at the 2008 IGVC is to present a working Intelligent Ground Vehicle that can pass the qualification, enter the Navigation challenge and the Autonomy challenge, and to place in the top 50^{th} percentile of the Design (category B) challenge.

The vehicle in its whole is a complete innovation since it is a brand new design from Cornell University and has been built from scratch by undergraduate students. More specifically, the Zero-point turn feature and the All-Wheel-Drive with an electronic differential are innovations that set the Gladiator project apart.

The following list summarize the principal predicted performance metrics for Gladiator whose calculation are explained in the respective subsystem section:

- Steady state speed on level ground: 2.24 m/s
- Speed up a 15° incline with an initial 2.24 m/s speed: 0.126 m/s
- Battery Life: 1.24 hours
- Maximum integrated obstacle detection range: 3 m
- Sand traps are surmounted by adjusting each individual wheel velocity until a calculated theoretical speed from the drive encoders matches the actual velocity provided by the INS. This minimizes wheel spin so traction can be maintained.
- The drive motors are capable to propel the robot over 0.05 m (2 in) potholes.

3. Mechanical design

Gladiator's mechanical design seeks to allow high maneuverability while traversing minefields by employing an all-wheel drive four-wheel-steering mechanism. To be cost effective when deployed in large numbers, Gladiator was constructed from inexpensive yet robust materials and components. Table III Mechanical datasheet table below summarizes the principal features of Gladiator's frame.

Table III Mechanical datasheet table

Mechanical Datasheet	
Material	Al-6061 & 6063
Length	1.18 m
Width	0.82 m
Height	1.20 m
Weight	81.7 kg
Ground clearance	0.0762 m
Payload capacity	20 kg
Driving wheels	4
Wheel diameter	0.254 m
Drive gear reduction	102.5:1
Steering gear reduction	23.61:1
Maximum speed	2.24 m/s
Speed on 15º ramp	0.126 m/s

3.1 Methods of design

Gladiator was conceptualized and then designed in detail with SolidWorks 2007. Using this computer-aided design (CAD) program, each component and assembly was examined and confirmed as properly constrained and non-interfering. Furthermore, finite element analysis (FEA) was done using COSMOSWorks to ensure that all structural components would maintain a factor of safety of at least two under the worst case dynamics loads.

Some of these components, such as spur gears and pulley belts, are off-the-shelf parts, while the structural components of the frame, such as the drive trains, steering pods, and crossbeams, are machined or CAM-fabricated. In addition, there are a number of rapidprototyped parts, such as the motor containers and encoder mounts.

3.2 Vehicle structure

Gladiator's chassis consists of two "steering pods" linked by two crossbeams, all connected to form a solid rectangular frame with parallel steering pods and parallel crossbeams. These steering pods and crossbeams are constructed primarily of 6063-T5 aluminum rectangular tubing and flat stock. Additionally, the payload and battery containers are mounted to the crossbeams, and the tail suspending a range of sensors is attached to the left and right steering pods and the rear crossbeam of the chassis.

Figure 3 CAD rendering of Gladiator's chassis and tail

3.3 Drive Train

Gladiator is equipped with four drive trains arranged in a rectangular configuration. One Maxon A-Max 32 motor with a Maxon GP32C gearhead provides a stall torque of 7 Nm and 20 W of power for each drive train.

The required torque was calculated taking into the consideration the ability to climb a ramp. When driving at the competition speed on an incline of 15º, the power required for each drive train is 17.68 W. Furthermore, with the design of the in-wheel drive train and the axis of the steering shaft aligned with the wheel's center ensures that no moment of force is produced when steering.

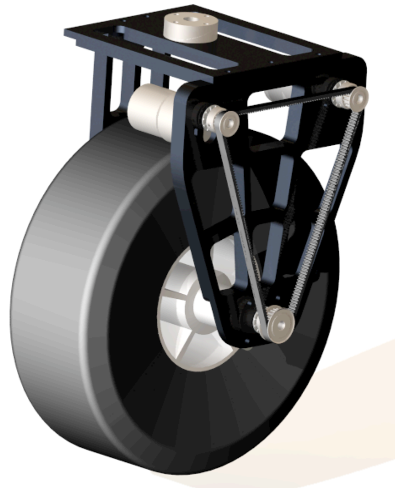


Figure 4 CAD rendering of drive train

The *CAD rendering of drive train* above shows the drive train assembled with sprockets and chains in order to reduce backlash, misalignment, and variations in belt tension.

Gladiator uses Radio Flyer all-terrain 0.25 m (10 in) wheels. The pneumatic wheels are treaded for outdoor use and provide damping in order to absorb vibrations.

3.4 Steering

Gladiator utilizes an innovative steering system, which allows for the vehicle to perform zero-point turns. The steering system consists of two steering pods moving the front and rear wheels at equal magnitudes, but in opposite directions. Orienting the wheels diagonal to each other parallel at 45º angles, allows the vehicle to make a zero-point turn. Additionally, this design allows the vehicle to go forward, rotate its wheels with a very small radius of turning, and then move in a direction 90º from the direction of its initial velocity.

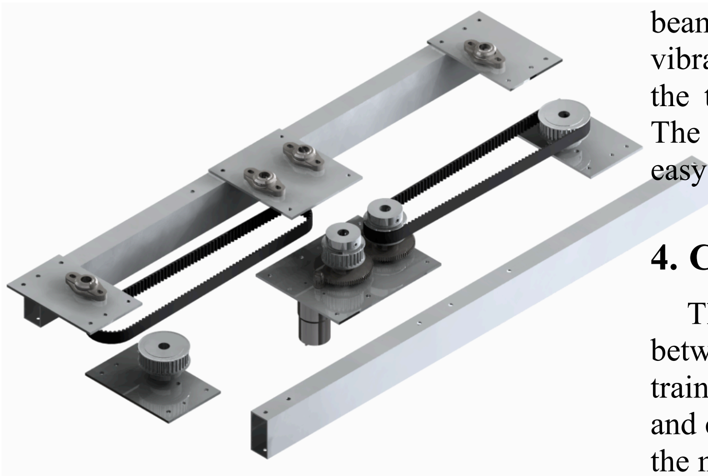


Figure 5 Exploded view of steering pod.

A DC motor is connected to the two spur gears in the center of the steering pod and provides torque for steering transmitted through the drive timing pulley, the timing belt, and the wheel timing pulley.

3.5 Enclosures and Sensor Mounts

Gladiator's payload enclosure is a hinged Pelican 1560NF case, with interior dimensions, 0.51 m x 0.39 m x 0.22 m, and rating of IP-65, which is protected against low pressure jets of liquid from all directions and completely against dust. To address the electronic heat dissipation issue, IP54 rated fans and filters are installed on the case. The Lithium-Polymer (LiPo) battery is housed in a separate smaller Pelican case attached to the rear of the vehicle for easy battery access. To isolate the LiPo battery in a separate compartment is part of the contingency plan to minimize damages in case of battery misuse that could trigger a battery fire.

In order for the camera to have the largest field of view possible, it needs to be mounted at least 1.2 m (4 ft) off the ground. To meet this requirement and retain the aesthetics of Gladiator, a tail structure with four curved beams merging at over the center of rotation of the robot was designed as a camera mount. The tail is constructed from 6061-T6 aluminum, and features cross members between each curved beams to provide increased rigidity against vibration. The camera is attached at the tip of the tail on a three-degrees of freedom mount. The curved design of the tail also allows for the easy access to the electronics.

4. Control system

The motor controller acts as the interface between the navigation computer and the drive train. By using pulse width modified (PWM) and continuous feedback loops to adjust outputs, the motor controller achieves accurate control of individual wheel velocities and the steering angles of each steering pod. Table IV below summarizes the characteristics of the control system. As seen in Figure 6 the system is built around a single Atmega2560 microcontroller. The Control board receives the desired heading and velocity over RS232 serial connection from the navigation computer as well as the current rotation speed and angle data from the encoders.

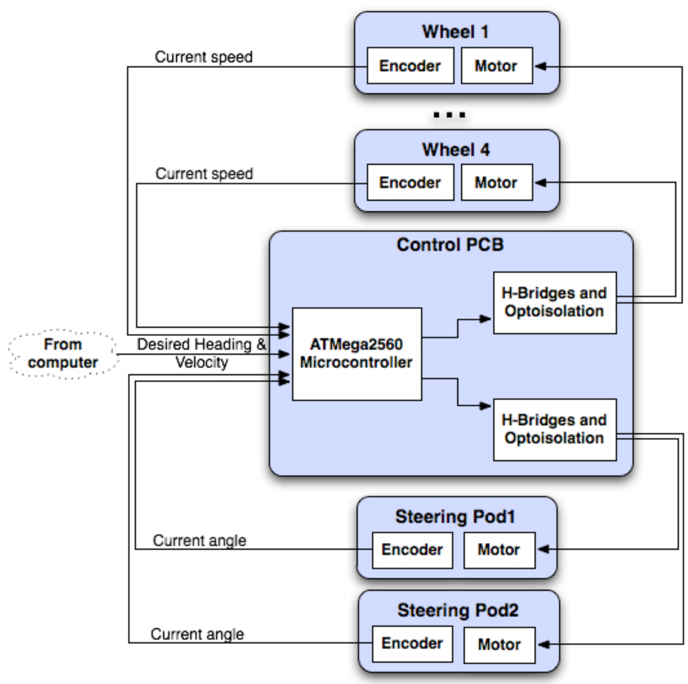


Figure 6 Control information flow diagram

The microcontroller outputs six PWM square wave 0-5V volt signals (2 for steering, 4 for propulsion) to the H-Bridges. These six signal channels are isolated using optoisolators to protect the microcontroller from back-EMF and noise. Gladiator's motors then receive a PWM voltage from the H-Bridges to set their speed.

Table IV Control datasheet

Control Datasheet	
Voltage	24 VDC
Max stall current	6A/channel, 6 chan
Microcontroller	AT Mega 2560
H-Bridges	Sabertooth dual 10A
Control frequency	250 Hz

4.1 Feedback

Active feedback for both velocity and steering angle accurately is used in this closed-loop control system. Gladiator uses analog optical encoder to determine the current velocity. The microcontroller then makes a fixed correction using the delta between the desired velocity and speed received from the computer and the current information obtained from the encoders. Though is system slightly underdamped, the 250 Hz control frequency quickly eliminate the overshoots. This process is illustrated in the state diagram in Figure 7 below. A similar approach is used in the control

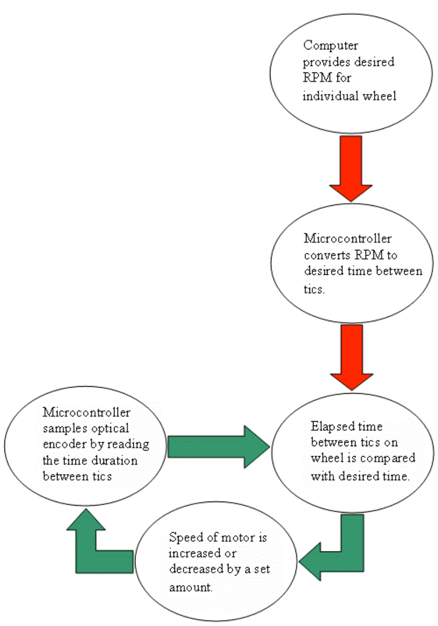


Figure 7 Control state diagram

of the steering angle except that steering pods are using magnetic encoders. If the current angle is not within five degrees of the desired value, the system responds by briefly powering the motor in the direction of the correction. This cycle is repeated producing a net correction to within tolerance. By using a closed-loop control system, involuntary angle changes caused by the terrain are actively compensated for.

5. Electronics

The power system is designed to provide a clean source of power, without fluctuations to the motor controllers and the processor. Those units require different voltages so we implemented a number of circuits that would help the process. Figure 9 below show block diagram of the electronics in Gladiator while Figure 9 below show the distribution of the 376 W of the power budget.

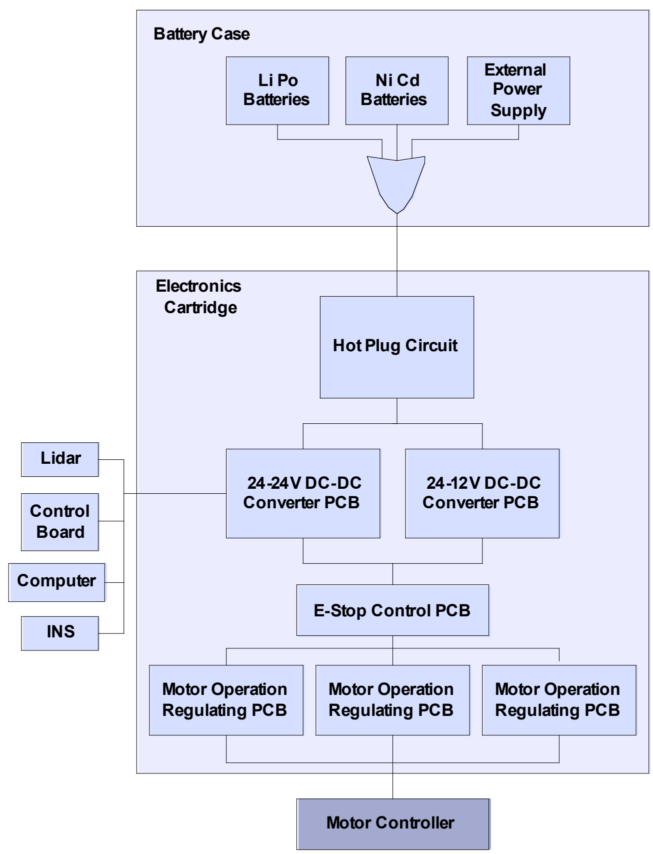


Figure 8 Power System Design Chart

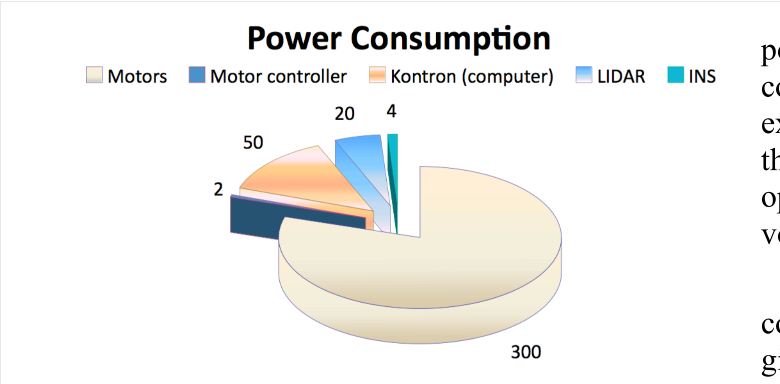


Figure 9 Power budget graph

5.1 Design process

The design process started by gathering all the power requirements of the components of the robot and trying to pick an appropriate source of power. The DC-DC converter modules were placed to handle fluctuations from the current and voltage coming from all power sources. Li-Po Batteries were chosen as the most efficient means of power and the Hot Plug Circuit was implemented to handle changes in power without affecting the component. The emergency stop controllers along with the Motor Operation Regulating PCBs were put in place as an effective way to stop all current supplied to the Motor Controller instantly when the manual and the remote E-Stop button were triggered. The free software provided by Advanced Circuits "PCB Artist" was used to create all the PCBs.

5.2 Components

Cornell Gladiator runs on 24 VDC Lithium Polymer (LiPo) batteries providing 12 Ah. Working with LiPo batteries are sensitive and requires special care. To address all possible safety concerns with the batteries, the team designed a LiPo battery-handling document that has been distributed amongst team members and whose knowledge is assessed before using the batteries.

The Hot Plug Circuit (HPC) offers multiple power input in the battery compartment. When connecting a secondary power source such as an external power supply or a replacement battery, the HPC ensures allows uninterruptible operation by providing steady current and voltage while isolating the two power sources.

The DC-DC Converter circuit essentially consists of a fuse and a DC/DC converter to give out a regulated 24 VDC and 12 VDC channel. Each channel can provide up to 4 amperes. The motor power input bypasses the DC-DC converter.

There are 3 motor operation circuit boards that are essentially controlled by relays protecting from overcharge.

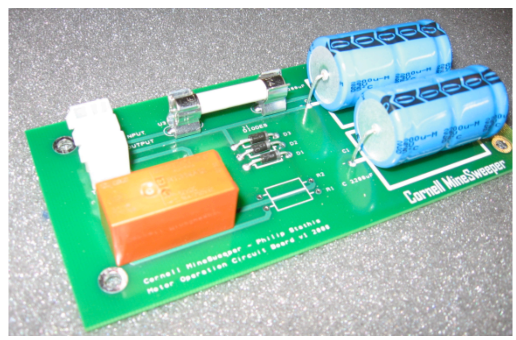


Figure 10 Motor Operation Regulating circuit

5.3 Emergency stop

The E-Stop circuit sends outputs to all the relays based on either the signal from the wireless e-stop or the pushing of the button for the mechanical e-stop.

This PCB handles mechanical and wireless emergency stop. For the mechanical E-stop, we have a push to stop a red 0.05 m (2 in.) in diameter button that is located 1.06 m (3 ft.) from ground. The wireless emergency stop is effective for at 22.86m (75 feet) and it has the same effect as the mechanical emergency stop, essentially cutting the power supply to the motors to stop the vehicle.

6. Software

6.1 Software architecture

Gladiator's software layered architecture[2] is designed with the intent of maximizing modularity, adaptability and concurrency. The software is implemented in Matlab, C#, C++ and Java. The CS team being composed of a fairly small number of members with radically different development experience, this technology "pot-of-luck" was the only way to ensure a rapid and efficient code development. The Figure 11 below summarizes the software architecture of Gladiator.

The Sensor layer manages the data acquisition from the sensors using two specialized daemon applications that let clients interacts with the sensors over TCP/IP. The camera frame grabber is utilizing Matlab API to do acquisition of frame from the camera and prepare them from processing. This layer acts as a proxy by minimizing the coupling between sensors and the navigation code.

The Control loop layer is responsible for data fusion and navigation. The navigation loop is asynchronous and runs at 10 Hz in order to produce heading and velocity information for the control system in a deterministic fashion.

Finally, the third logical layer, which is not part of the software but rather the control system, is responsible for vehicle control and propulsion.

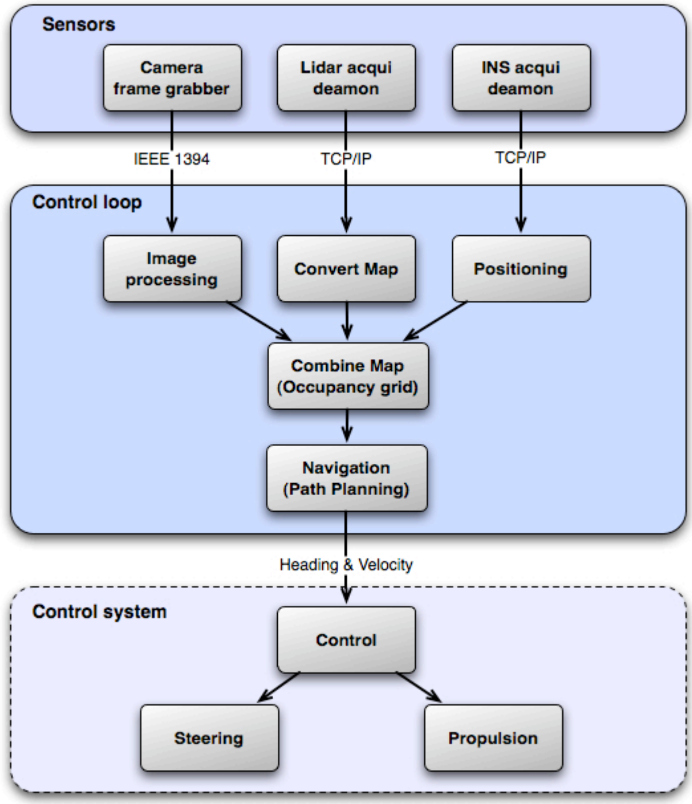


Figure 11 Software architecture

6.2 Control loop

The first modules from the Control loop are responsible for processing the information from the sensors to format it and filter noise.

6.2.1 Image processing

The Image Processing module acquires a new image frame using Matlab's firewire data acquisition to extract the lines position relative to the robot. The images are processed using the following algorithm:

Image processing steps:

1. The images are sampled in 320x240.
2. It is converted to grayscale.
3. Crop the image under the line of sight.
4. Canny edge detection.
5. Convert to binary map.
6. Geometric transformation to top view.
7. Convert to radial binary map.

The output is a radial binary map of the area in front of the robot.

6.2.2 Convert map

The role of the Convert Module is to transform the range information coming from the Lidar to radial map. It acquires the Lidar range values from the Lidar daemon via a TCP socket and then transform it to a radial map before outputting the newly created map.

6.2.3 Positioning

The Positioning module connects to the INS daemon to receive the global position of the vehicle (based on a combination of GPS and accelerometer data) and the orientation (based on a combination of compass and gyroscope data) and format it in both Latitude & Longitude format and X, Y, Z relative to the start point of the race.

6.3 Combine map

The first step in Combine map is to transform the binary radial based map from the camera to a radial range map. The radial based map is split into sectors. The distance to the closest object in a sector is determined using ray tracing. The Figure 12 shows an example of the process of ray

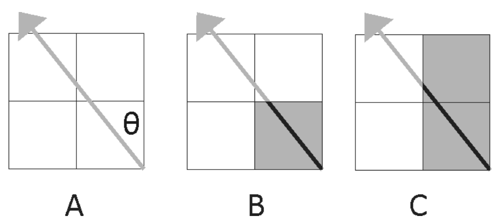


Figure 12 Determine the maximum travelable distance

tracing to determine the maximum travelable distance in a sector. To make the conversion more efficient, for every sector from $\theta - \Delta$ to $\theta + \Delta$, several angles inside that sector are picked randomly to determine the maximum travelable distance.

Once the radial range map has been generated using the maximum travelable distance, it is superimposes to the radial map from the Convert map module, so that it can be used in the navigation algorithm.

6.4 Navigation Algorithm

The navigation algorithm uses the local radial map computed by the combine map as its input. The navigation algorithm was designed with the local nature of the map in mind.

The navigation algorithm sweeps across the radial map starting from the preferred direction to find a direction that the Gladiator can move unobstructed. A direction is declared unobstructed if it is greater than a given threshold. The preferred direction is the front of the robot for the Autonomous challenge, and direction towards the closest goal for the Navigation challenge. If the preferred direction is not clear, the algorithm examines the sectors adjacent to the preferred direction. It continues to sweep across the map until it can find a direction with unobstructed movement closest to the preferred direction. If no such direction is found, the direction with the maximum possible unobstructed movement is outputted.

6.5 Determining Goal Sequence

This operation is only performed for the Navigation challenge. For the Navigation challenge, the robot needs to determine the sequence of goals that it needs to visit to finish the challenge as quickly as possible. To compute this sequence, the robot runs a Nearest Neighbor algorithm on a set of goals at the start, while ignoring any obstacles on the field. While this algorithm gives a good approximation for the Traveling Salesman problem, there may be cases where the robot encounters an unvisited goal on its way to another goal. To minimize the total distance traveled by the robot, the robot should visit this goal first.

In each iteration of the Control loop, the robot checks if it is getting closer to an unvisited goal on the way to another goal. If this is the case and if this goal is closer than some threshold distance, the robot switches its goal to the unvisited goal. Once it has visited this goal, it resumes to the previous goal.

Below is pseudo-code for the navigation:

```
directionLeft = preferred direction
directionRight = preferred direction
freeSectors = minimum number of sectors
  that gives enough clearance for the
  robot to navigate through
found = false
while (direction still in range)
if (directionLeft is free)
```

```
found = true
output directionLeft
if (directionRight is free)
found = true
output directionRight
directionLeft = directionLeft -
   freeSectors
directonRight = directionRight +
   freeSectors
if (not found)
Output the direction with the maximum
clearance
```

With the correct choice of the threshold value, the algorithm will force the vehicle to move along the line for the Autonomous challenge, as shown in the Figure 13

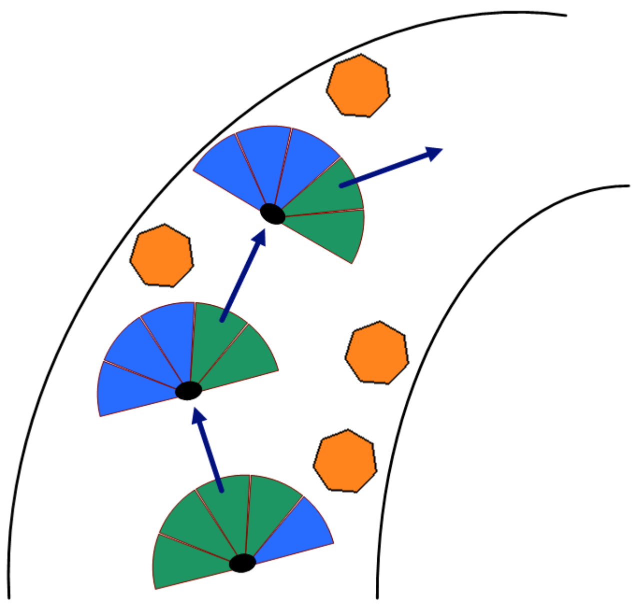


Figure 13 Navigation example diagram

7. Structure & History

Cornell MineSweeper is currently in its second year as a team and under the leadership of ECE undergrad Hamzah Sikander '09. The team's previous leader, Vikas Reddy '08, founded CMS along with Professor Ephrahim Garcia and stepped down this year to assume a non-leadership position in the controls division. In addition to controls, there are six other subdivisions that compose CMS—systems, management, business, computing, frame, and power and detection. Professor William Philpot also lends his aid to CMS in an advising role. A total of 33 enthusiastic and hard-working students are a part of CMS and have put in over 2,000 hours into the project.

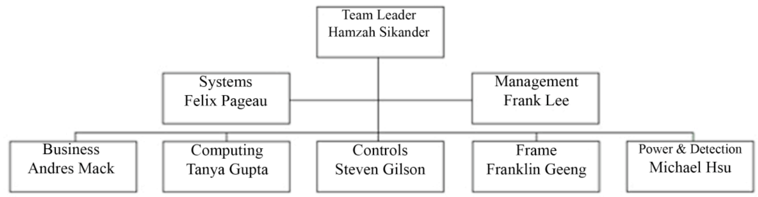


Figure 14 Team structure

Table V Team budget and vehicle cost

Equipment	Value	Cost
Amphenol connectors	$9,622.98	$0.00
Atmega2560 MCU	$125.90	$125.90
Drive trains	$1,020.00	$0.00
Frame and tail	$450.00	$450.00
LiPo batteries	$841.85	$841.85
Microbotics MIDG-II	$7,060.00	$0.00
Mini-ITX computer	$744.98	$744.98
Pelican 1560	$158.29	$158.29
Power circuits and cabling	$500.00	$500.00
Sick LMS 291	$5,137.50	$5,137.50
Steering pods	$400.00	$400.00
Unibrain Fire-I	$168.30	$168.30
US Digital MAE3 and E4P	$71.70	$71.70
Total:	**$26,301.50**	**$8,598.52**

8. Conclusion

Gladiator is a fully autonomous robot designed, constructed, and tested by Cornell University students. Gladiator's modular design allows the robot to be deployed and function across a whole spectrum of terrain types. The robot's advanced zero-point turn ability further bolsters Gladiator's mobility on the ground, allowing Gladiator to turn in place.

Safety stands as another key feature for Gladiator. For one, Gladiator is completely watertight so that its circuitry is guarded in the event of liquid-based hazards. The most sensitive component of Gladiator, its Li-PO battery, is given extra protection in the form of a secure, isolated case in the vehicle's rear. Lastly, Gladiator possesses reliable overcharge protection to prevent issues with overheating.

Cornell MineSweeper may be a rookie at the IGVC, but we have strongly devoted ourselves in preparing our innovative, humanitarian entry. We believe we will perform superbly in this year's contest and continue to better our entries in subsequent years.

9. Acknowledgement and thanks

The 2007-2008 Cornell Minesweeper would like to acknowledge the support and devotion of its team members, sponsors, our supervising professors Ephrahim Garcia and William Philpot, the CU Experiential Learning Laboratory staff, the Systems Engineering program, Sibley College of Mechanical and Aerospace Engineering, the School of Electrical and Computer Engineering and the Department of Computer Science.

We would like to particularly acknowledge the support of our sponsors:

- Cornell University
- Advanced Circuits
- Amphenol Aerospace inc.
- Gateworks Corporation
- Gladiator Technologies inc.
- Igus inc.
- Kionix inc
- Legend Technologies Pvt. Ltd.
- MaxBotic inc
- Microbotics inc.
- Sunstone Circuits

Finally, we would like to thanks all our families, friends, Prof. Andy Ruina, CUAIR, CUAUV, CU Baja Racing, Cornell Racing, and CU Snake arm for their support and help.

10. Team rooster

- Abe Cantwell
- Andres Mack
- Brian Cheng
- Cameron Salzberger
- Daniel Paz
- Daniel Wong
- Evan Levine
- Félix Pageau
- Frank Lee
- Frankling Geeng
- Gary Soedarsono
- Gil Lee
- Greg Meess
- Hamzah Sikander
- Harsh Chamria
- Hung Dang
- Jaydev Mahadevan
- Jawwad Asghar
- Joseph Stein
- Justin Charles Yee
- Karim Ibrahim Hamdoun
- Makoto Bentz
- Michael Hsu
- Naveen Dasa
- Philip Stathis
- Sarah Leung
- Saran Baskaran
- Siddharth Gauba
- Steve Gilson
- Steven Liu
- Tanya Gupta
- Tomasz Rutkowski
- Vaishal Patel
- Vikas Reddy
- Yong Sheng Kho

11. References

1 INCOSE: 'SYSTEMS ENGINEERING HANDBOOK', in Editor (Ed.)^(Eds.): 'Book SYSTEMS ENGINEERING HANDBOOK' (2000, edn.), pp. 383

2 Len Bass, P.C., Rick Kazman.: 'Software Architecture in Practice, Second Edition' (Addison-Wesley, 2006. 2006)

Annex I

Figure 15 Picture on the test field at Cornell on May 1st 2008