Oakland University Proudly Presents:

Replicant

IGVC 2013

Student Members:

Mike Truitt – Senior, Electrical/Computer

Kevin Hallenbeck – Senior, Electrical

Steve Grzebyk – Masters, Electrical

Mike Norman – Senior, Electrical

Gregory Hickman – Masters, Mechanical

Brian Neumeyer – Freshman, Mechanical

Britni Reese – Junior, Electrical

Chris Parks – Junior, Electrical

Suzanne Neal – Junior, Engineering Chemistry

Parker Bidigare – Freshman, Computer

Kevin Benfield – Freshman, Electrical

Nick Pletz – Sophmore, Electrical

Savvas Koupparis – Sophmore, Mechanical

Ziyad Al Obaidi – Sophmore, Computer

Oscar Sanchez – Sophmore, Computer

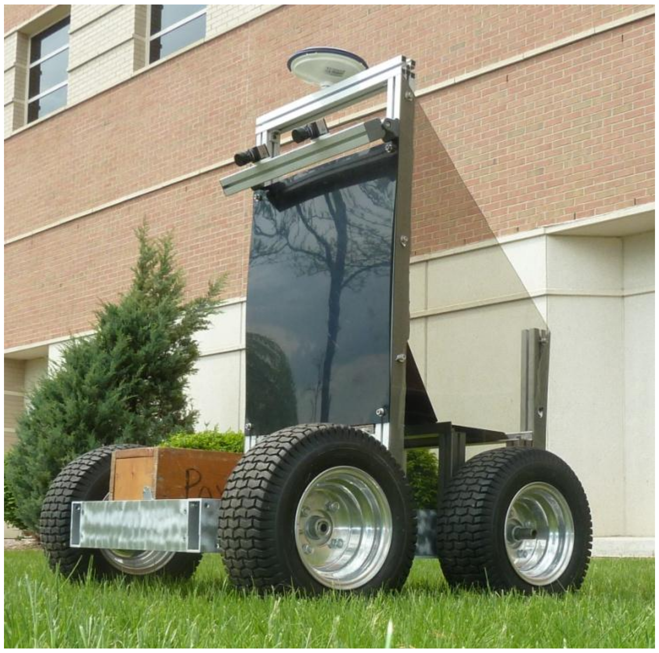
David Guoin – Freshman, Mechanical

Micho Radovnikovich – PhD, Systems

Lincoln Lorenz – PhD, Systems

Sami Oweis – Ph.D, Systems

Faculty Advisor: Dr. Ka C. Cheok



Fixxated Motion

molex one company › a world of innovation

Oakland UNIVERSITY

BATTERY GIANT

DATASPEED Inc. MAKING MOBILE EASIER™

I certify that the engineering design present in this vehicle is significant and equivalent to work that would satisfy the requirements of a senior design or graduate project course.

Signed,

________________________________, **Dr. Ka C. Cheok,** Faculty Advisor

1 Introduction

Oakland University is proud to enter Replicant into the 21st annual Intelligent Ground Vehicle Competition! Replicant is a rugged four-wheel drive platform, employing differential drive steering **(Section** 3**)**. Custom electronics, including an H-bridge **(Section 4.1)** and an embedded microcontroller board **(Section 4.3)**, were designed to meet the specific requirements of the IGVC vehicle. All software systems, including stereo vision processing **(Section** 7**)** and map-based path planning **(Section** 8**)**, were simulated and integrated in the powerful Robot Operating System (ROS) environment **(Section** 6**)**.

1.1 Team Organization

The Oakland University team consists of 18 members, with a composition of 3 graduate students and 15 undergraduate students. Figure 1 shows the organization of the team and how the responsibility is distributed. Each major sub-system of Replicant has its own captain responsible for taking the lead role in its development. The captains were in charge of managing the rest of the group and guiding them towards completion of their respective components. It is estimated that about 1200 person-hours were invested in the development of Replicant.

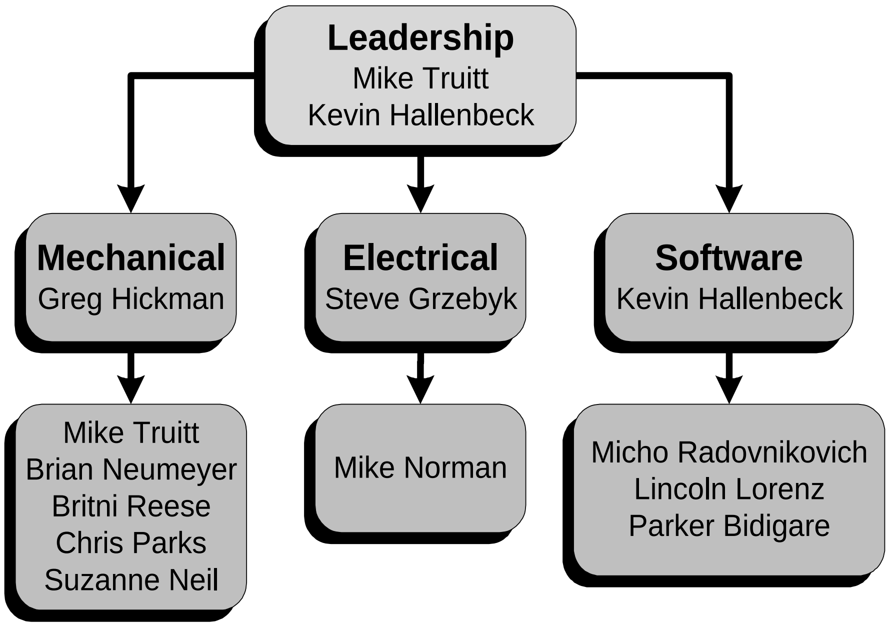


Figure 1 – Organizational chart of the team.

1.2 Design Process

A classic 'V-Model' design process was followed to develop Replicant, shown in Figure 2. After defining the requirements of Replicant, a design was formed using CAD, and a detailed simulation environment was formed to develop the navigation system. After implementing the design and integrating the various components, a rigorous test cycle began, where consistent failure points were identified and rectified through minor adjustments or larger design modifications.

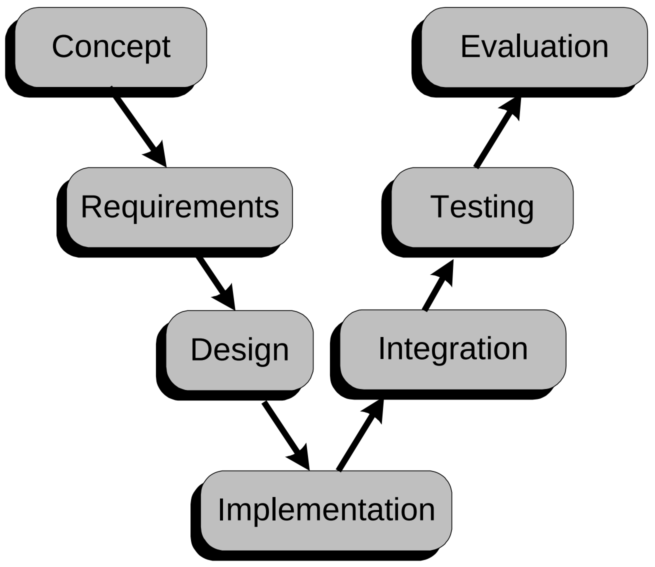


Figure 2 - V model design process.

2 Innovations

Below is a list of the main innovative aspects of Replicant's design. They are summarized here, and discussed in more detail in their respective sections.

- ❖ Custom H-Bridge design and fabrication **(Section 4.1)**
- ❖ Custom embedded microcontroller board **(Section 4.3)**
- ❖ Automatic camera calibration **(Section 7.2)**
- ❖ Multi-rate Kalman pose estimation **(Section 8.1)**
- ❖ Effective interpretation of stereo vision point cloud data **(Section 7.3)**
- ❖ 3D map-based path planning **(Section 8.3)**
- ❖ Ability to create and load reusable maps of the environment **(Section 8.2)**

3 Mechanical Design

3.1 Chassis

Replicant's chassis was constructed without using any pre-fabricated components. The chassis was created using mainly tubular aluminum, selected for its low weight and high strength to weight ratio. The batteries were placed as low as feasible in frame, and positioned in relation to the payload such that the robot's center of gravity is in the midpoint of the four wheels and 6 inches above the ground. In the middle of the chassis exists a large upright frame. This frame provides a raised camera platform, and an electronics mount that allows the robot's electrical components to be easily monitored and serviced.

Figure 3 shows a CAD design of the frame. The axles are securely mounted with two bearings that limit the radial load, as well as possible lateral deformation or twisting. The axles were machined with a flange on one end to mount against the motor face, isolating the motor shaft from excessive load, and distributing it along the motor face. This method of load distribution avoids large amounts of additional friction due to bearing rub.

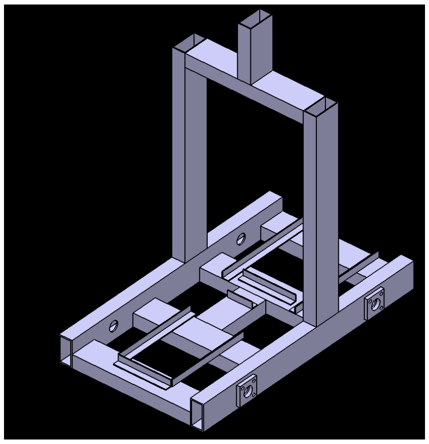


Figure 3 - CAD model of Replicant's frame.

3.2 Drive Train

At the beginning of the design phase, based on past IGVC experience, it was mandated that Replicant must be able to perform a zero-point turn. This capability greatly simplifies the path planning and control. The simplest and most widely used drive method that enables zero-point turns is differential steer. Additionally, differential steer simplifies the mechanical aspects of the drive train, since it operates solely on wheel rotation, and does not require any additional moving parts.

Each of Replicant’s four wheels is driven by a brushed DC motor with an integrated gearbox. The output shafts of these motors attach directly to the wheels. These simple motor mounts eliminate the need for heavy and bulky transmissions, and enable consistent locomotion while traversing irregular terrain. Additionally, the low center of gravity and six-inch ground clearance allow the robot to easily traverse inclined surfaces, with very small risk of rollover.

4 Electronic Components

4.1 H-Bridges

Replicant’s H-bridges are completely custom-designed PCBs. Based on past experience with other H-bridges such as IFI’s Victor series, it was desired to use an H-bridge that is more flexible, robust, and capable of chopping the motor power at a much higher frequency. A conventional single-channel PWM signal controls the speed and direction of the H-bridge output.

Key features of the H-bridges are:

- On-board fuses
- Automatic fan control
- Reverse battery protection
- Over-current protection
- Over-temperature protection
- Serviceable components

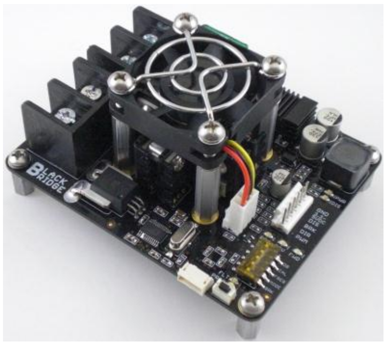


Figure 4: Custom H-bridge

4.2 Sensors

Replicant is equipped with an array of sensors that allow it to detect obstacles, compute its location, heading and speed, and be operated in a safe and reliable manner. The sensor array consists of:

- NovaTel FlexG2-Star GPS receiver
  - 8 Hz, less than 1 meter accuracy
- 2 uEye UI-2220SE color USB 2.0 Cameras
  - 768x576 resolution, ½’’ CCD sensor, 50 Hz
- Hokuyo UBG-04LX LIDAR sensor
  - 4 meter range, 240 degree field of view, 35 Hz
- InvenSense ITG-3200 tri-axis gyro
- Honeywell HMC5843 tri-axis magnetometer
- US Digital E3 Wheel 2500 CPR encoders
- DX6i wireless R/C aircraft joystick
  - Embedded controller-based manual control and wireless E-stop

4.3 Computing Hardware

Embedded Controller Board

The embedded controller board is a custom PCB that was designed specifically for Replicant. The motivation to do so resulted from using off-the-shelf FPGA and microcontroller development boards in the past which were not specifically designed for the application. The embedded control board is designed to satisfy all hardware needs, and provide the flexibility of a commercially available development board without the extra bulk of unused functionality.

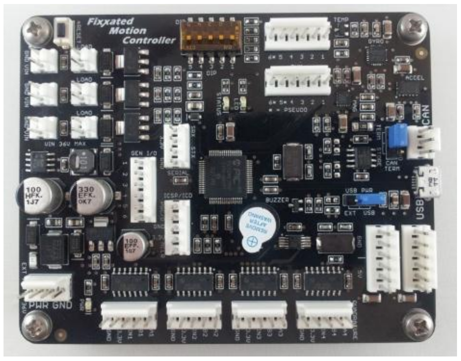


Figure 5 - Custom embedded controller board.

Key features of the embedded control board are:

- ❖ 32-bit Microchip PIC microprocessor
- ❖ Dedicated hardware quadrature counters to efficiently perform high-resolution encoder data processing
- ❖ Integrated accelerometer, gyro, and magnetometer for robust robot pose estimation
- ❖ High speed USB communication
- ❖ Battery voltage monitoring
- ❖ Power switches and distribution
- ❖ General I/Os, input capture, PWM

Laptop

All high-level processing is performed on a Lenovo Thinkpad W530 laptop. The processor is a quad core, 3.4 GHz Intel i7, and has 16 GB of RAM. To make the laptop robust to the vibration encountered on a ground vehicle, a solid state drive is used instead of a conventional hard disk drive. The operating system is Ubuntu 12.04, and runs the "Fuerte" distribution of ROS.

4.4 Power Distribution

Figure 6 shows a block diagram of Replicant's power distribution system. The power for the robot comes from two 12V lead acid batteries, wired in series to make a 24V system. The entire electrical system is routed through a main circuit breaker for protection. The robot has the ability to power up the embedded control board and sensors separate from the H-bridges for testing and safety purposes. The H-bridges are driven through a custom solid state relay PCB. The relay board is fully isolated and prevents improper ground return paths from occurring. The batteries can either be charged on-board, or quickly replaced by another set to achieve continuous runtime.

4.5 Safety Considerations

Many precautions were taken into account when designing Replicant's emergency stop system. In addition to two conventional turn-to-release E-stop switches, a DX6i joystick is used for disabling the motor output wirelessly. The DX6i has a range of several hundred feet. To protect against a variety of failure conditions, the drive control system automatically turns off the motors if it fails to receive commands from the computer or joystick after 200ms.

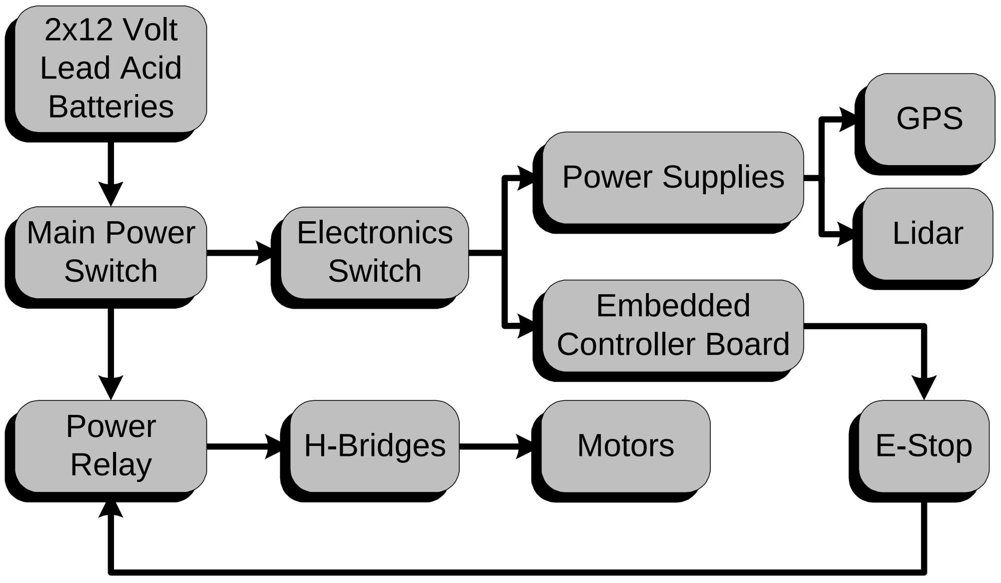


Figure 6 - Replicant's power distribution system.

5 Embedded Controller

5.1 Data Gathering

The microcontroller runs a Real-Time Operating System (RTOS) to manage its tasks. Data from the inertial sensors is gathered at a rate of 200Hz and streamed to the laptop using custom USB drivers.

5.2 Drive Control

Closed loop velocity control is critical to accurately follow a plan generated by higher level software. Closed loop velocity PI controllers were implemented on the embedded microcontroller for each wheel. The control loop is the highest priority task in the operating system of the microcontroller, and runs at 100 Hz. The velocity feedback is reported to higher level software for localization. Velocity commands come from higher level software and the DX6i wireless joystick.

6 ROS Software Platform

Replicant's software systems are implemented on the Robot Operating System (ROS) platform. ROS is an open-source development environment that runs in Ubuntu Linux. There is a multitude of built-in software packages that implement common robotic functionality. Firstly, there are many drivers for common sensors like LIDAR, cameras and GPS units. There are also general-purpose mapping and path planning software modules that allow for much faster implementation of sophisticated navigation algorithms.

6.1 Efficient Node Communication

A ROS system consists of a network of individual software modules called "nodes". Each node is developed in either C++ or Python, and runs independently of other nodes. The nodes are all controlled by the ROS core. Inter-node communication is made seamless by a behind-the-scenes message transport layer. A node can simply "subscribe" to a message that another node is "publishing" through a very simple class-based interface in C++. This allows for the development of easily modular and re-useable code, and shortens implementation time of new code.

6.2 Debugging Capabilities

One of the most powerful features of ROS is the debugging capability. Any message passing between two nodes can be recorded in a "bag" file. Bag files timestamp every message so that during playback, the message is recreated as if it were being produced in real time. This way, software can be written, tested and initially verified without having to set up and run the robot.

Another convenient debugging feature is the *reconfigure_gui*. This is an ROS node that allows users to change program parameters on the fly using graphical slider bars. This tool is invaluable, since most robotic vehicle controllers require precise adjustment of several parameters, and being able to change them while the program is running is very beneficial.

6.3 Simulation Using Gazebo

Gazebo is an open source simulation environment with a convenient interface to ROS. To rigorously test Replicant's artificial intelligence, simulated IGVC courses were constructed. These courses contain models of commonly encountered objects: grass, lines, barrels, and sawhorses. The configurations are designed to emulate the real IGVC course as accurately as possible. The simulation environment proved invaluable to the development process, since, unlike recorded data, the simulation responds to robot decisions and generates appropriate simulated sensor data.

7 Computer Vision

7.1 Stereo Vision

Most LIDAR sensors can only detect objects on one plane. At past competitions, this limitation caused problems, especially in the case of the sawhorse-style obstacles, seen in the foreground of the image in Figure 8. The horizontal bar of the obstacle would not be in the scan plane, thereby going undetected, and the vehicle would frequently try to fit between the two legs of the obstacle.

Replicant addresses this severe sensor limitation by using stereo vision. Applying open-source functions for stereo image matching, the images from the stereo camera pair are processed to generate a 3D cloud of points corresponding to everything in the frame.

The two stereo cameras are mounted eight inches apart near the top of the robot. The image capture is hardware synced to make sure the left and right images are always matched, even at high speeds. Some obstacles of uniform color do not have enough texture to find confident matches between images. Because of this, data near

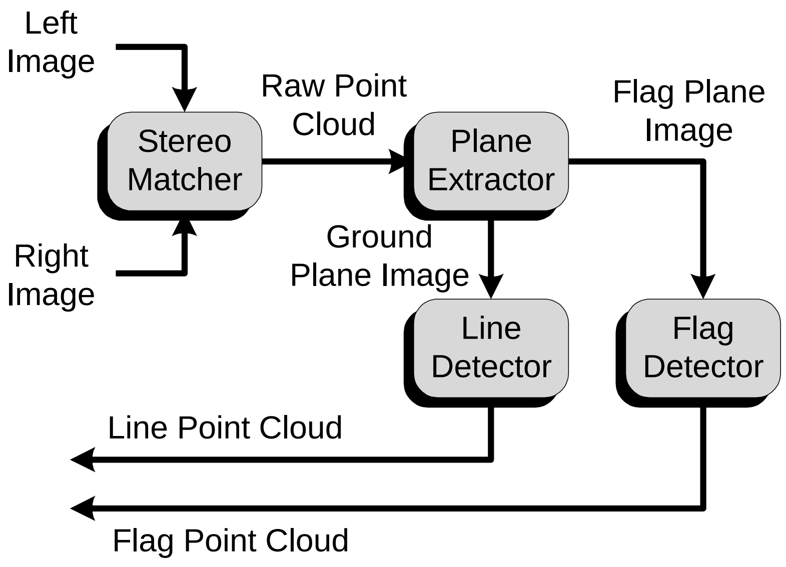


Figure 7 - Block diagram of the vision system modules.

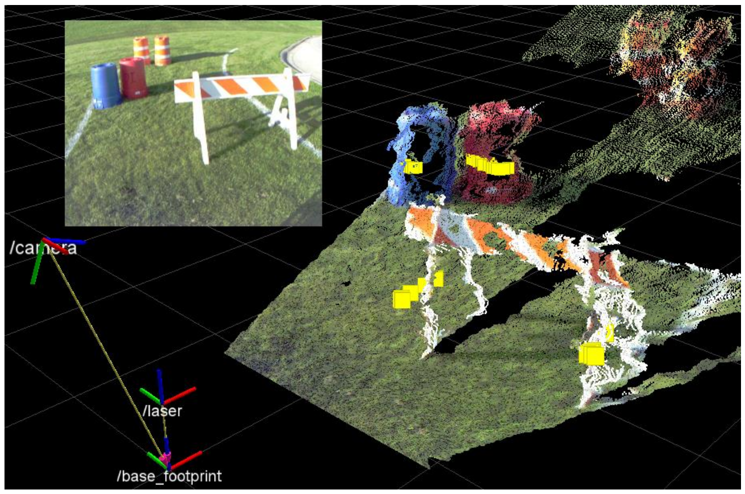


Figure 8 – Example stereo point cloud and LIDAR scan

the center of the uniform obstacles tends to be absent, while edges, grass, and everything else is reliably detected and mapped to a 3D point.

Figure 8 shows an example image and the corresponding point cloud. Each point in the cloud is marked with the color of the image pixel is corresponds to. On top of the point cloud is LIDAR data, indicated by the yellow squares. This example shows how much information the LIDAR misses in certain situations, and exemplifies the capabilities of a carefully implemented stereo vision system.

7.2 Automatic Camera Transform Calibration

To calibrate the transform from the cameras to the ground, an automatic algorithm was developed using a checkerboard. Assuming the checkerboard is flat on the ground, the 48 vertices in an 8x6 grid are optimally fitted to a plane equation to detect the camera position and orientation. This transform is used to place stereo and LIDAR data on the map from different coordinate frames. Figure 9 shows an example of how the transform calibration is performed on a real image. The calibration algorithm has proven to be very reliable and yields very accurate point clouds.

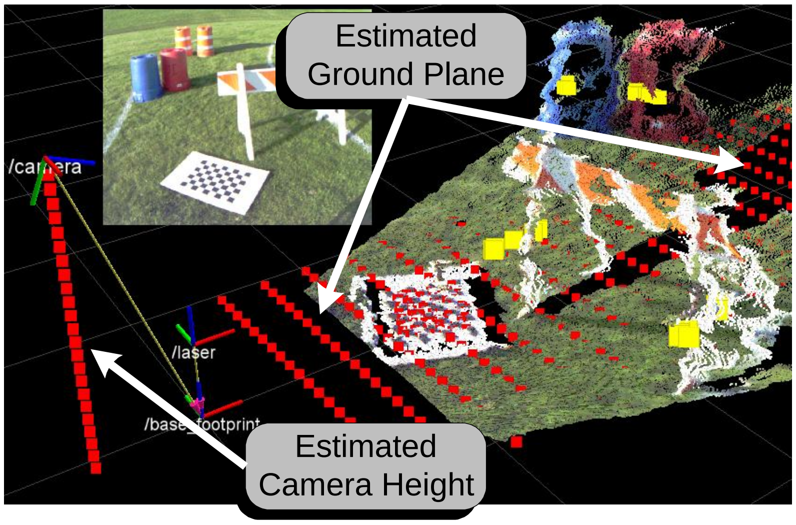
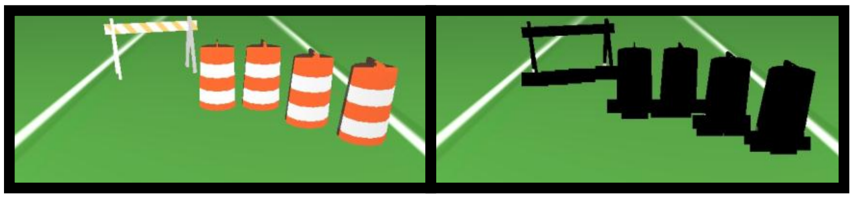


Figure 9 - Example of the automatic transform calibration procedure.

7.3 Plane Extraction

The plane extractor is responsible for analyzing the raw point cloud output from the stereo matching algorithm. The goal is to generate images that contain only pixels within a certain height window, and black out the rest. Specifically, the two planes of interest are the ground plane, where the lines will be detected, and the flag plane, which is used for the flag detection algorithm. The height window to detect points on the ground plane is adjusted according to experimental results, and the height window for the flags is set according to the expected height of the flags on the course.

Additionally, the height information of obstacles is used to eliminate obstacle pixels from the generated ground plane image in order to make the line detection algorithm more robust. Figure 10 shows examples of ground plane image generation, where pixels corresponding to points above the ground are blacked out, as well as regions of the ground pixels that correspond to where objects meet the ground. Notice how the resulting image primarily contains just grass and line pixels.


b) Simulated Images

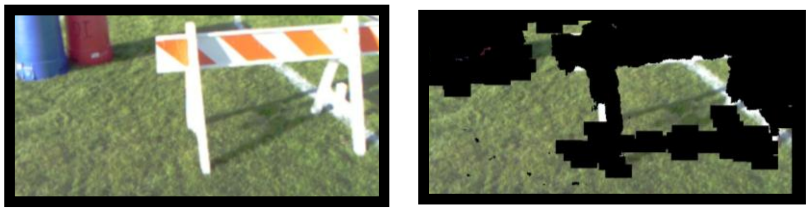
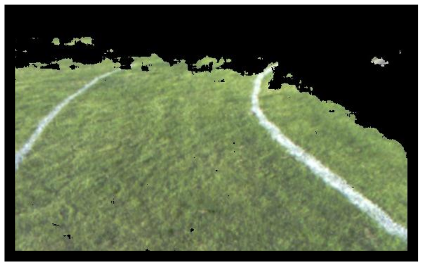
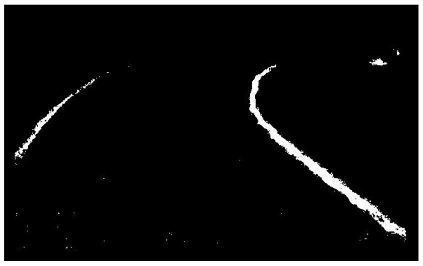

a) Real Images

Figure 11 - Example ground plane images in simulated and real scenarios.


a) Ground Plane Image


b) Threshold Image

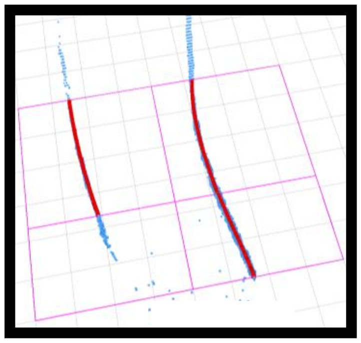

c) Quadrant Line Fitting

Figure 10 - Example line detection and curve fitting.

7.4 Line Detection

After separating out the ground plane points and generating a ground plane image as in Figure 10, the line detector operates on this image to extract the lines. It is assumed that most of the pixels either contain grass or lines, so a simple, yet effective way to segment the lines from the grass is an adaptive median filter that is robust to lighting changes.

The resulting segmented binary image is then used to fit a 3rd order polynomial curve in 4 quadrants in front of the vehicle that combine to accurately represent the true nature of the lines. This polynomial fit is computed by a standard recursive linear least squares algorithm. However, in order to make the algorithm robust to noise, the weight of each detected line pixel is discounted by a likeliness factor that is dependent on its distance from the current curve fit. For stability reasons, the parameter estimates follow simple first-order dynamics to rule out rapid changes that are impossible.

Figure 11 shows an example of the line detection system. In Figure 11.c, the light blue points are the 3D locations of the segmented line pixels in Figure 11.b. The red points are the curve fits in each quadrant.

7.5 Flag Detection

The flag plane image from the plane extractor is input to the flag detector. In this image, red and blue flags are separated by thresholding hue. To direct the robot towards the correct path, artificial lines are drawn on the map. Red flags draw to the right, and blue flags draw to the left. This blocks invalid paths and funnels the robot into the correct path. Figure 12 shows a simulated flag scenario illustrating this approach, where artificial lines are shown in white.

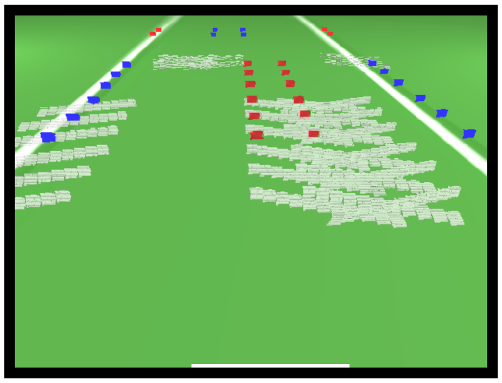


Figure 12 - Example flag negotiation logic.

8 Navigation System

8.1 Kalman Filter

The Kalman Filter fuses data from many sensors to accurately estimate position and orientation. Each sensor updates at a different rate, and the filter updates with the fastest sensor, 200Hz. This results in accurate dead-reckoning between slow GPS updates. To avoid the discontinuity of traditional Euler angles, the orientation is represented using a quaternion. In the two dimensional case, the yaw angle can be represented by a 2D vector. Table 1 shows information about the sensors being fused together, and which state variable each is measuring.

Table 1: Kalman Filter Sensors

Sensor	Rate	Measurement(s)
Wheel Encoders	100 Hz	Linear and Angular Velocity
Gyro	200 Hz	Angular Velocity
Magnetometer	160 Hz	Orientation
GPS	8 Hz	Position

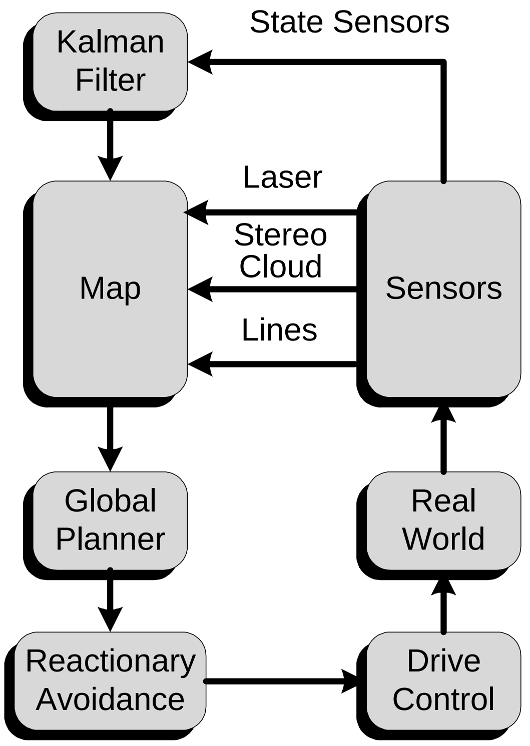


Figure 13 - Diagram of the path planning modules.

8.2 Mapping

The Kalman Filter was found to be accurate enough to build a map without Simultaneous Location and Mapping (SLAM). SLAM requires a cluttered environment to match incremental data, but obstacles on the IGVC course are relatively sparse.

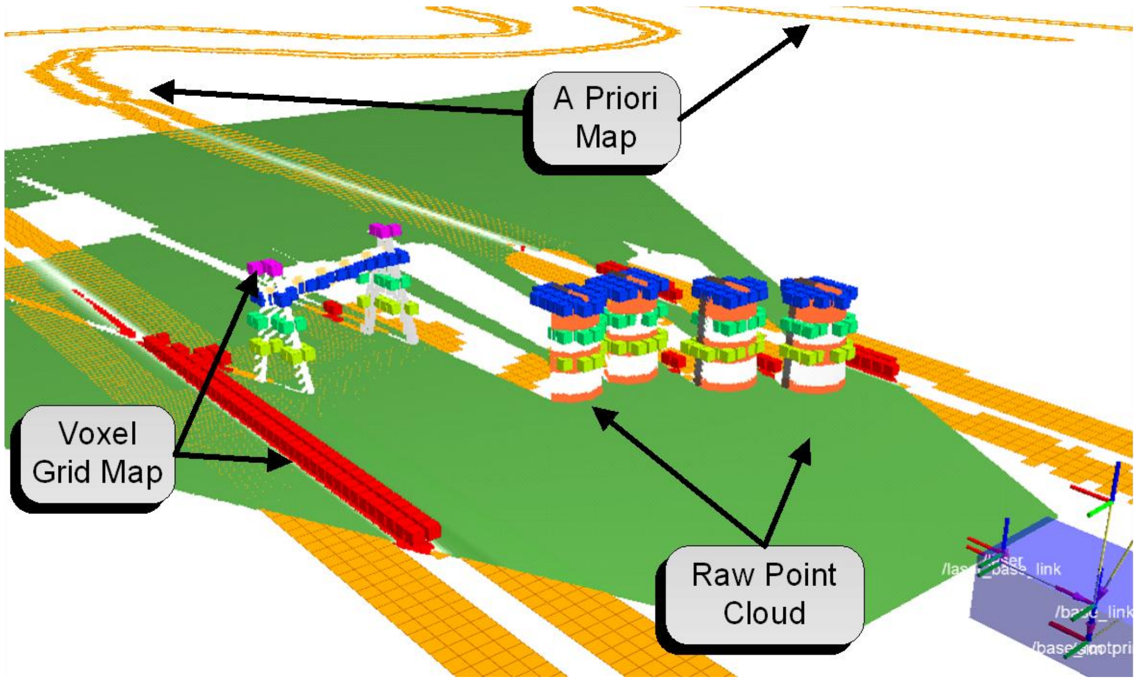


Figure 14 - Example of the mapping procedure.

Replicant's mapping algorithm places time-stamped information on the map using the Kalman estimated position and orientation. The map is represented in 3D as 5 planar layers. Object information from three sources is placed on the map: 3D stereo data, LIDAR data, and detected lines. Line data is on the ground plane, LIDAR data is parallel to the ground plane and elevated, and 3D stereo data is present in all heights.

Each sensor can mark and clear space on the map. Clearing is done by tracing a ray through 3D space from the sensor source to the obstacle, and clearing every cell in that path. Two instances of the mapping algorithm, global and local, run in parallel. The local map is a 15 meter square with 5 cm resolution. The global map is a 100 meter square with 10 cm resolution. The global map can be initialized a priori with a map generated from a previous run.

8.3 Global Path Planning

The global planner uses the global map to select the path to a goal point with minimum cost. The occupied squares on the map are inflated using the robot's width and an exponential cost function. The global planner is not allowed to plan a path passing through any inflated square.

Furthermore, the cost function allows the path planner to choose the optimal path, even in both cluttered and open environments. The global planner implements Dijkstra's algorithm, and is an open-source package built in to ROS.

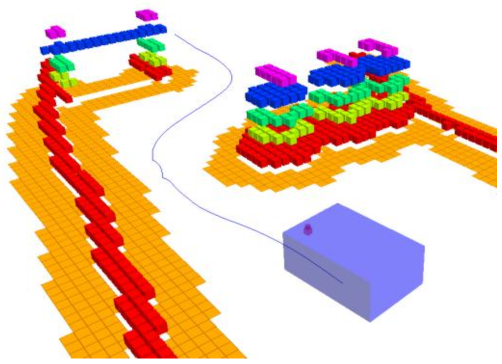


Figure 15 - Example of global plan computed from 3D map.

Figure 15 shows an example of a global path output from the planner. The differently colored cubes represent the different layers of the 3D voxel grid map. The orange squares represent the inflated 2D projection of the 3D voxels onto the ground plane.

8.4 Reactionary Avoidance

Reactionary avoidance uses the local map to avoid collisions. This is the last stage in the path planning, and is responsible for overriding commands that would cause a collision. Future collisions are detected by simulating trajectories along evenly spaced turning radii. The best trajectory is the one closest to the requested command that doesn't result in a collision. A safety factor is also applied to avoid driving unnecessarily close to objects. The speed of the trajectory is scaled inversely by the angular velocity to prevent quick turns which could smear data on the map.

9 Performance Analysis

9.1 Maximum Speed

Replicant's motors spin at 157 RPM at nominal load, so combined with 15 inch diameter wheels, the resulting maximum speed is 10.3 mph. This estimate correlates with the observed performance.

9.2 Ramp Climbing Ability

At nominal load, the drive motors provide 101 in-lbs of torque. Assuming a realistic vehicle weight of 175 lbs, this corresponds to a max slope of 18 degrees. However, experiments have shown that Replicant can handle much steeper slopes, up to about 30 degrees, although not at the nominal load of the motors.

9.3 Reaction Time

The ROS system running on the laptop gathers position data at 200 Hz, LIDAR data at 35 Hz, and images from the stereo camera pair at 20 Hz. The artificial intelligence systems were designed to be able to handle this frequency easily, thereby allowing the robot to make new decisions at the slowest sensor sampling rate 20 Hz = 50 ms.

9.4 Battery Life

The AGM batteries on Replicant provide a total of 35 AH. The sensors and embedded controller board consume a total of approximately 1 amp. Experiments have shown that the drive motors consume a total of 25 amps maximum in a grass environment typically encountered at IGVC. Based on these observations, total battery life is approximately 1.5 hours, which is about the amount of time achieved in experiments.

9.5 Obstacle Detection Range

The Hokuyo LIDAR has a range of about 4 meters, but has shown to provide very low-noise distance measurements. The stereo cameras are oriented to see 5 meters away from the vehicle, but experiments show that the 3D point cloud measurements are most reliable within 4 meters.

9.6 GPS Accuracy

Under normal conditions, the Novatel FlexPackG2-Star GPS receiver is accurate to within 1 meter, which is enough positional accuracy to reach the small waypoints on the Auto-Nav Challenge course. However, the Kalman filter algorithm **(Section 8.1)** fuses the GPS readings with the rest of the sensors to eliminate some of the noise and to provide faster position updates based on dead reckoning.

10 Vehicle Equipment Cost

A breakdown of the cost of the components on Replicant is shown in Table 2.

Table 2: Cost Breakdown of Vehicle

Item	Cost	Cost to Team
FlexG2-Star GPS Unit	$1,000	$1,000
Lenovo Laptop w/Mount	$1,526	$1,526
Hokuyo LIDAR	$3,500	$3,500
uEye Cameras	$1,834	$1,834
Camera Lenses	$300	$300
Batteries	$320	Donated
Motors	$724	$724
Wheels	$420	$420
Frame Material	$665	$665
H-Bridges	$600	$600
Wheel Encoders	$480	$480
Misc.	$120	Donated
Total:	**$11,489**	**$11,049**

11 Conclusion

Replicant has proven to be very rugged, efficient and reliable, performing well while driving on any kind of terrain. The new artificial intelligence design shows promising results, and the Oakland University team has great confidence going into this year's competition.

Acknowledgements

The Oakland Robotics Association would like to thank several people and organizations. The team first of all thanks their advisor, Dr. Ka C. Cheok, for his invaluable advice and guidance. Thanks also to the School of Engineering and Computer Science (SECS) and its dean, Dr. Louay Chamra, for generous funding and lab space to work in. Immense thanks to Peter Taylor, machinist in the SECS for his help and guidance in the fabrication of Replicant. Finally, thanks to our external sponsors Molex, Battery Giant, Dataspeed Inc. and Fixxated Motion LLC, whose donations of certain components of the vehicle were critical to its development.