Dokalman

University of Cincinnati

24th Annual Intelligent Ground Vehicle Competition

Joseph Knight ✦ Tony Iacobelli ✦ Joe Hirschfield ✦ Douglas Flick ✦ Lucas Boswell Matt Kraemer ✦ Matt Thomas ✦ Griffin Ramsey ✦ Evan Baumann

Contact Information for Team Members provided in Team Organization

Submitted 5/15/2017

CERTIFICATION:
I certify that the engineering design in the vehicle Dokalman (original and changes) by the current student team identified in this Design Report has been significant and equivalent to what might be awarded credit in a senior design course.

____

Doctor Janet Dong, Advisor janet.dong@uc.edu

Introduction

The University of Cincinnati Robotics Team has a proud tradition of competing in the Intelligent Ground Vehicle Competition. The first iteration of Dokalman was entered in the in the 22nd Intelligent Ground Vehicle Competition. In the four years since its original design, we have completely re-designed the robot and are now on our third iteration of this robot. The following list of improvements have been since the last IGVC competition.

- There were various improvements to the physical arrangement of the robot:
  - Many electrical components were moved to a dedicated enclosure to provide better protection, cable management, and cooling.
  - A second battery was added over the front axle.
  - The wheelbase was widened and the axle was better calibrated to the frame.
  - The wheels were replaced to provide for better handling on a variety of terrains.
- The electrical system has been improved based on last year's lessons learned:
  - The electrical system was changed from 12 to 24 base voltage.
  - A new wireless E-Stop system was designed to provide better reliability and longer range.
  - Expanded functionality has been added to the safety light, also making it a status indicator.
  - The Power Distribution Units were integrated with the software stack, enabling automation of startup, shutdown, and blown fuse procedures.
- The modular software system has more interoperability:
  - Nearly all devices on Dokalman communicate over the TCP/IP network. Some networking components were replaced to provide better network resilience, reliability and lower power consumption.
  - Motor controller drivers were custom written providing a additional features and a massive increase in stability.
  - A switch from the two high resolution cameras to three cameras gives Dokalman a wider viewing angle for obstacle avoidance.
  - A new lidar unit was acquired. This unit provides a much higher resolution over the old unit in addition for being appropriate for outdoor use.
  - CPU intensive algorithms (such as warp perspective) were optimized for maximum performance.
  - The entire code base was reviewed and reformatted to reduce the total lines of code, ensure code quality, improve human readability, and future compatibility.
- 3D printing was once again heavily leveraged to create custom mounting solutions:

- A custom lidar mount with integrated cooling and power conversion from 24 Volt DC to 12 Volt DC was printed.
- Custom visors were designed and printed for each of the cameras.
- The wireless e-stop case was designed and printed for maximum space efficiency and protection.

The function of these changes is further explained in the following sections describing the control and hardware system design for Dokalman.

Team Organization

This year, the team was significantly smaller compared to previous years, allowing for team members to take on large responsibilities for the completion of the robot. Team members were able to carry out full life cycle development as the robot progressed from prototype to end product. Coordination between roles was perpetual as each constituent layer of the design was worked on simultaneously.

Role	Name	Major	Email*	Grad Year
Captain/Software Lead	Joseph Knight	Computer Engineering	knightjp	2018
Lead Engineer	Evan Baumann	Mechanical Engineering	baumanea	2020
Vision Lead	Douglas Flick	Computer Science	flickdm	2018
Navigation Lead	Lucas Boswell	Electrical Engineering	boswellj	2018
Network/Documentation	Tony Iacobelli	Information Technology	iacobeaj	2019
Navigation/Drivers	Joe Hirschfield	Computer Science	hirschjb	2020
Navigation	Jordan Jacob	Computer Engineering	jacobjr	2020
Software	Griffin Ramsey	Mechanical Engineering	ramseygd	2019
Mechanical Engineer	Sandro Leon	Computer Engineering	leonso	2018

*Note: Emails are suffixed username@mail.uc.edu

Design Process

The design process of this year's Dokalman was carried out simultaneously by the vision, software, navigation and hardware teams. Teams worked closely together in order to coordinate interconnected components and the boundaries between design layers.

Innovative Concepts and Technologies

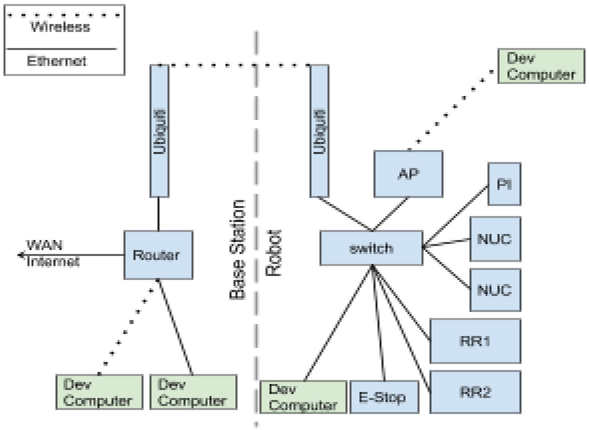
Some of the more innovative concepts that we included into this year's design are as follows:

Electronics enclosure

Our electronics enclosure allows for our electronics "package" to be transplanted easily from one robot to another. This allows us to essentially plug and play the entire brains of our robot into another machine. As our mechanical frame is designed and improved upon the electronics can evolve separately. This was inspired by the payload that is used at competition and is a similar size and shape.

Network System Design

A large goal for the system design of Dokalman was to convert legacy serial devices over to ethernet equipped devices. We have found TCP/IP to be easier to troubleshoot and more reliable. The robots network was designed to support these devices as well as debugging. Devices on the robot communicate with each other via gigabit ethernet, via a layer 2 switch. This network, when not running in competition mode, is bridged to our base station network for developer computers via a Ubiquiti long distance network link. This allows developers to connect to a stable network without having to follow around right behind the robot. In practice, this allows developers to be located in shade and with available power.

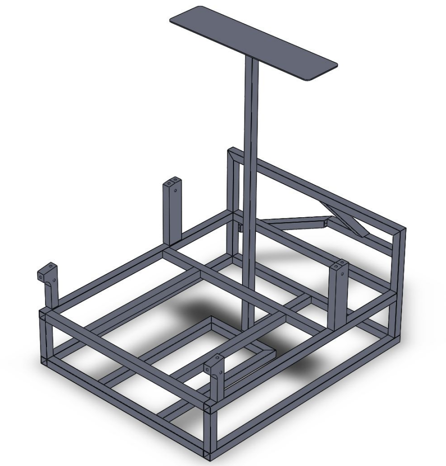


This network diagram depicts the location and connectivity of devices on Dokalman.

The robot network is entirely self supporting, however adding the base station allows for developers to connect from distance, as well as providing local DHCP, NTP, and DNS services. Where possible, the base station can also provide internet access for robot and developer machines.

Mechanical Design

Frame Design



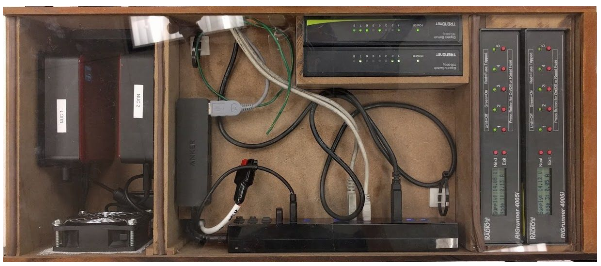
Building upon the success of last year's CAD modeling efforts, this year's hardware team further refined the frame and shell with the goal of achieving greater stability and weatherproofing. The front of the welded 41-30 alloy steel tubing frame was redesigned to accommodate two marine batteries. By accommodating both batteries over the center of the front axis, more grip was achieved by the wheels. By extending the front axis by approximately 8 inches this allows the wheels to apply more force to turn the robot. Not only were we able to move the center of gravity to a more desirable location but also correct a previous axle alignment issue.

Weatherproofing

Dokalman can handle most weather it will encounter. There are multiple tight fitting panels used to keep the majority of water out of the electronics. The electronics are also protected by our new custom project box. Parts outside of the panels are kept in enclosures or are water resistant.

Electronics Enclosure

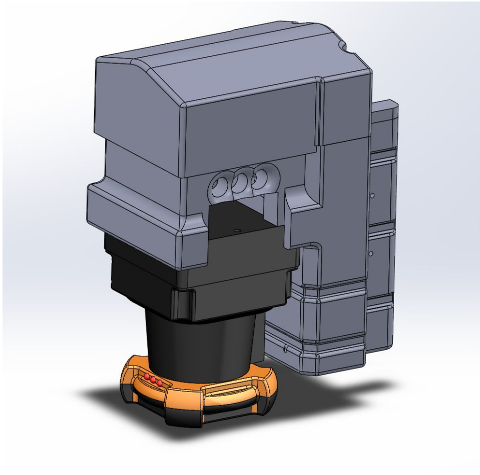
A project box was constructed to centralize and protect all core electronic components. Power distribution, power conversion, network switches, computing resources, and USB hubs are located within the box. Special attention was given to the thermal, cable, and power needs of each component within the box. The optimized layout reduced the total footage of cable on the robot as well as increased serviceability by making components easily removeable



as all components are pressure-fitted. The two-layer design enables distinct cooling zones to enable thermal separation of the hottest components. The ventilation system also employs filters to ensure that dust, pollen, and dirt do not coat the electronic components.

3D Printing

For many of the components which were not made of steel tubing, 3D printing technology was utilized. Among the pieces that were modeled with Solidworks, Sliced with Simplify3D, and printed using a Genuine Prusa i3 Mk2 and RepRap Prusa i3 printer were the cases for electrical components and improved mounting brackets. The ability to custom-design these 3D printed components added a layer of protection and specialization which would have been difficult to achieve otherwise. Furthermore, we lowered cost by using 3d Printing as PLA plastic filament is significantly less expensive than the raw materials that would have been needed to mill the same parts. The model pictured left is the custom lidar mount with a representation of the lidar unit as well.



Suspension

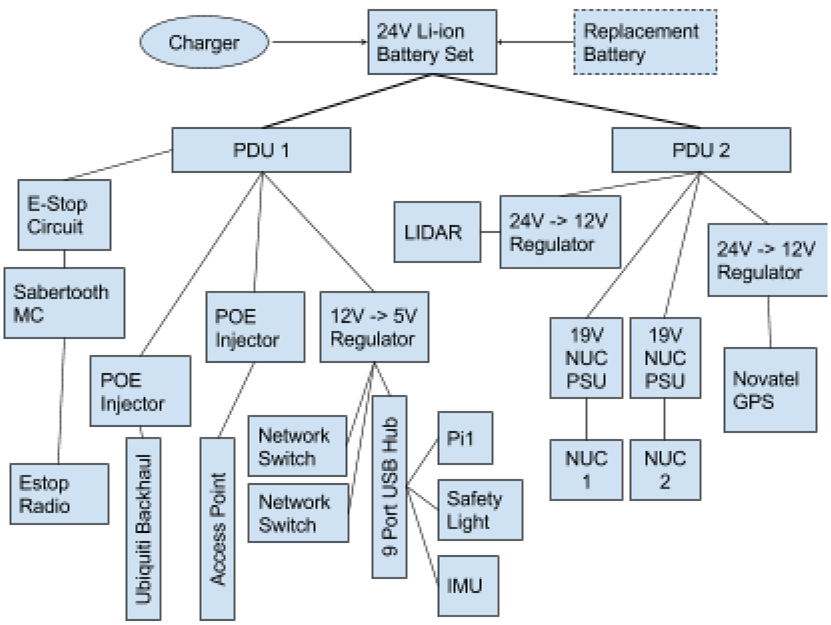
Due to the lower speeds and low grade of the terrain the robot was designed for this was not a huge factor. While we are actively thinking about how to improve the robot with suspension being high on the list, this was not a priority due to time constraints and finances.

Electronic and Power Design

The electrical design for Dokalman was driven with several goals in mind, many based on difficulties in previous designs. The central batteries connect to the 2 Power Distribution Units (PDU) which then connect to all the powered systems on the robot. These PDUs have the ability to enable and disable all the individual components on the robot, as well as read their live current load. Incorporated in the PDUs are slow trip electronic fuses, which cut the current if a configurable current limit is passed. The biggest advantage of the electronic fuses is that they can be easily reset, which allows us to set very close tolerances to protect the devices.

From a hardware perspective, we have mounted all the electronics to a single board that allows us to service the frame by removing the single board and not disconnecting all the components. This board, made of acrylic, also serves to electrically isolate the devices from each other.

One important aspect of the robot's design was strictly working with DC voltage. In past experience using inverters and A/C adapter leads to wild voltage zero points and dangerous behavior when connecting non-referenced sources together, such as computers and sensors. In Dokalman, all voltages are relative to battery negative, and there is no alternating current to cause the floating grounds. This achieved both by careful part selection and the use of DC to DC step up and step down converters, for 18V and 5V devices, respectively



The electrical diagram to the left describes the hierarchy of powered devices in the robot. Each line off the PDU can be individually measured and disabled.

An important electrical improvement allows the main battery to be swapped out by connecting a new battery in parallel and removing the drained one. This allows the team to hot-swap batteries for longer runtime, or plug the charger directly into the robot and run off external power

With the improvements made to power distribution, the robot can operate in competition mode for 3-4 hours of continuous use. However, in lower functionality mode, Dokalman can last significantly lower. The largest power demand for the robot are the NUC computer cluster and the motion controller. If the robot is running at idle, without moving, it will last for 20 hours running the master controller and networking gear.

Wireless E-stop System Design

Dokalman's estop system was upgraded this year to be simpler and better integrated into the robot's electrical system. When the wireless e-stop button is pressed, a dedicated e-stop AVR disables the robot's motion controller by disabling the power channel on our power distribution board. This prevents powered motion, and the motors bring the robot to a sudden complete stop. The advantage of this system is the fact that the control system can detect the disabled motion controller and stop the software from commanding motion as well as displaying an e-stop warning on the safety light.

Software Design

Dokalman is built using the ROS, or Robot Operating System, framework. ROS specifies a method of data exchange between independent software codebases working in a network to control the robot's position. ROS fits within the team's development style because it leads to concurrent development, test isolation and well compartmentalized code.

Navigation System Design

The navigation system is a multi-level software system designed to learn and store the robot's location, as well as the locations of all obstacles around it. By recording the entire environment into a virtual map, smarter pathfinding decisions can be made.

The lowest level of localization focuses on compiling wheel encoder and accelerometer data to generate accurate odometry for the robot. The high percentage of the robot's weight on the drive wheels reduces wheel slippage. Most importantly, the wheel encoder data is passed through an extended kalman filter with the output of a 3D accelerometer to detect slippage during acceleration and braking. To correct for drift in the robot's overall localization, the LIDAR and GPS are used for position correction built with fixed world objects. All combined, the odometry allows the robot to continuously evaluate its position relative to start.

GPS System Design

To localize the robot on earth, Dokalman is equipped with Novatel Propak V3 GPS and amplified antenna. Via a serial interface on NUC1, the NMEA output from the GPS parsed at a 2 hz rate within the robot software. This GPS location is used to find the location of the robot on earth, in UTM coordinates to the nearest origin point. This transform is inversely broadcast as a descendant of the map frame of the robot, to prevent reparenting of the fixed map frame. With this transform, any other UTM coordinate can be translated to a relative transform from the robot. Using this system, the wayfinding software simply attaches UTM translated goals in this frame for the robot to navigate.

Map Generation

With an accurate position of where dokalman is in the world, the map can be used to record the position of obstacles in the course. A costmap approach is used to track the positions of obstacles such as lines and barrels. The LIDAR detection layer takes in laser sweeps of the surrounding area. A separate layer takes in the visible lines from all three camera systems mounted on the mast, allowing the cameras to work together to build a complete map. This costmap is persistent during the run, and is continuously built and refined as the sensors discover new areas of the world.

Pathing Planning

To plot a route through the costmap, the robot takes a 2 step approach. The global planner, which takes a low resolution view of the entire known map, uses a modified A* route planning approach to set intermediate goals from the robot's present location to its final goal. These intermediate goals are delegated to a local planner to achieve, which takes a higher resolution look both at the environment in the immediate vicinity of Dokalman, and at the shape of Dokalman itself. It is this layer that handles the back out logic described below in failure modes.

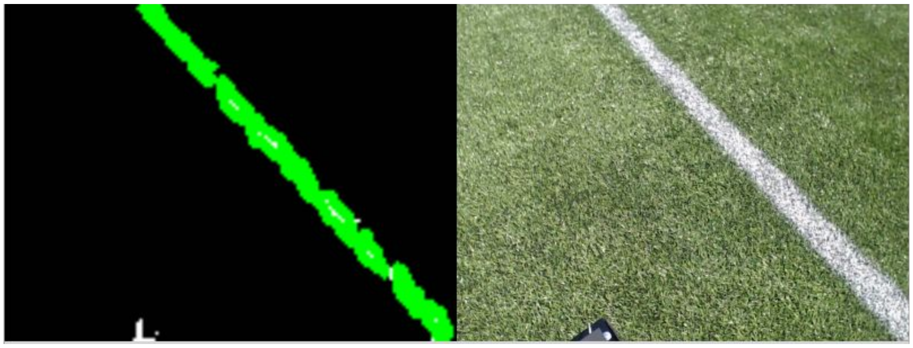
Goal Selection

To set the global goals, dokalman is programmed to find GPS waypoints in the order to match the course. However, we added additional limitations to its route planning strategy analogous to horse blinds. To the route planning algorithm from driving straight at the exit of the maze at the beginning of the run, the goal setting limits the turning radius for the global plan to force dokalman to drive forward and make the first corner. This radius also prevents backtracking or turnarounds in the maze, without preventing error recovery behavior in the local plan.

Obstacle Detection and Avoidance

Through the use of our two costmaps, various sensor data can build a virtual world that describes the real world with associated "costs." This allows the robot to see objects and make decisions about them such as the likelihood of running into an object. We can essentially compute a bitmap where each pixel has a threshold for the how close we can get to an obstacle without hitting it.

Vision System Design



One of the largest problems we encountered in the previous year was some of our algorithms employed were extremely computationally intensive. Major changes in the realm of our vision systems have been to refactor code to make it simpler and easier to understand, and finally to look back at the intensive algorithms and optimize them.

One example of this has been in the transformation of our points from 2D pixels to 3D real world coordinates. Previously we were taking the entire image and transforming it and then finding the points we were interested in. Now we find the points we are interested in and only warp those. In testing we saw that CPU utilization went from 85% to 15% across 4 Cores on an Intel i3.

Another issue we faced at previous competitions was that the algorithms that were employed could not properly handle dynamic lighting conditions. With lane detection being fundamental to the competition, this was a considerable complication. This has been a particular focus of ours for this year's competition.

After refactoring our vision stack we went from three independent nodes to two independent nodes that deploy the OpenCV Api and the ROS framework . The two nodes are the driver node and the vision node. The driver node communicates directly to the cameras and manages their settings such as frame rate and brightness. The vision node is responsible for lane detection but has also been designed in such a way to allow additional components to be added effortlessly. The vision node makes use of a Lane Detection class which utilizes convolution matrices to filter the lanes from the image. The node begins by resizing an image so that the lanes are of a known size. After which we apply a whiteness filter so that each pixels whiteness is calculated. After this we apply our convolution matrix to a 2D filter. Finally we transform our pixels to real world coordinates and pass those to our navigation stack.

Failure Handling

Software Failure

Failures in the software mapping algorithms are most often manifested as ghost obstacles detected and tracked by the planners but do not actually exist in the real world. This is caused by the occasional bad read from the LIDAR, or from line shaped items detected by the vision system. We correct for this by allowing the sensors to clear out obstacles previously recorded if they are no longer present when returning this position. This fits well with the navigation's fallback strategy for impossible navigation tasks.

When the route planner can't find a route through a local costmap, it defaults to a backout and spin behavior. In our experience, unplannable maps often contain non-existent obstacles. Reversing and spinning slightly gives the cameras and lidar a new perspective on the environment so that the map can be cleared out and a proper route planned.

Electrical Failure

Electrical faults are most often caused by current overdraw from the motor controller. While the system is rated to a very high current load, we have limited it with a software fuse to a specific maximum current to protect the batteries and drive train. If the robot is up against a immobile obstacle and attempts to move forward, it will trip the circuit and be estopped until the circuit is re-enabled.

Vehicle Safety Design

In additional to preventing significant current in a motor stall configuration, care was taken during the design phase to enclose chains and sprockets inside and underneath the shell of the robot to prevent dangerous pinch points and hazards. To work on the drivetrain, we can remove the mast, batteries and electronics package and flip the entire chassis over to access the equipment underneath.

Simulation

Simulation was done by creating a virtual machine on a proxmox server and looping sensor data over various interfaces. The benefit to using proxmox is that it is open sourced software with no vendor lock-in and allows for compartmentalization. Using Proxmox we could clone a base image and produce copies all of which were identical to our competition machine. Each machine could easily be managed from a simple web interface and allowed users to have access to a linux machine. From this programmers of all levels and knowledge could get exposed to linux and even if they managed to damage the underlying system we could easily roll-back the machine to a stable state.

Much of our sensor data had been recorded during a prior run and looped on the system. Custom udev rules were written to loop the sensor data over the same names as the devices the competition robot would have. For example Video Feed was looped and users could subscribe to a constantly running feed of what competition would look like. Complete with Ohio weather that changes every 5 minutes (Normal Real time Behavior) so that the environment is dynamic.

The reason we chose this method was due to time constraints, finances, and greater control over our simulation setup. This does not require a machine with a graphics card nor does it limit our developers from having to worry about destroying the machine they are working on due to a lack of experience with linux. Developers are encouraged to learn more about linux through trial and error.

Performance Testing

All the critical systems on Dokalman were tested in isolation before added to the robot. A list of systems below had specific test strategies.

Gearbox and chain drive system - Bench tested with 8020 frame and grades. Design was limited by the traction of the surface under the wheels. Electrical measurements fell in expected range for the components used. 24V was selected for superior torque for a given current.

LIDAR system - tested on independent wheelchair robot in various lighting and distance configurations.

Vision system - Several different webcam models were tested, and the Logitech C920 was selected for its onboard encoding. The vision pipeline was designed to input and output video at several points to allow sections to be tested individually.

Electronics Enclosure - prototyped out of disposable materials and heat tested with electronics operating under maximum load to verify adequate cooling outdoors in the heat.

Performance Assessment

Our testing shows successful route planning and obstacle avoidance for barrels and lines. Dokalman can find GPS waypoints accurately to within 1 meter for the center of the robot, functionally hitting the target waypoints. A lack of ideal proving ground has limited our ability to test fencing and potholes. Our self navigation averages a 1 m/s speed when making straight lines or slight turns. The route planning loop occurs at a 20Hz rate, so robots reaction time is limited by acceleration and deceleration limits rather than planning frequency. Obstacles can be added as soon as they are detected by the sensors. For the LIDAR, this can range as far as 30 meters for a reflective object with a 270 degree viewing angle. The vision system can see 160 degrees side to side for about 6 feet out from the robot. The robot has been tested at full weight on a 35 percent concrete slope without slippage or stall.

Appendix A: Bill of Materials

Part	Manufacturer	Model No	Quantity	Unit Price	Total
Frame	Alro	1” steel square tube	360 ft	$360.00	$720.00
Batteries	U.S. Battery	US 36DCXC	1	$135.00	$135.00
Motors	AmpFlow	F30-400 w/ Gearbox	2	$200.00	$400.00
Motor Driver	Sabertooth	TE-091-260	1	$190.00	$190.00
Processing Computers	Intel	DC3217BY	2	$1,000.00	$2,000.00
Master Computer	Raspberry Pi	2B	1	$35.00	$35.00

Cameras	Logitech	C920	2	$69.99	$139.98
Wireless Estop	3Built	RES12VU	1	$74.99	$74.99
Motion controller	Kangaroo	X2	1	$23.99	$23.99
GPS	Novatel	ProPak V3	1	$14,995.00	$14,995.00
DC Transformer	RioRand	RRDCCI12245V5A25W	2	$15.00	$30.00
Wheel Encoders	Sparkfun	COM-10932	2	$40.00	$80.00
Controller	Sony	Dualshock 3	1	$44.99	$44.99
Body Panels	Plaskolite	MC-100	2	$108.99	$217.98
Misc 3D Printed Parts	n/a	n/a	4	$15	$60.00
				Total	$21,508

Acknowledgements

We would like to thank our advisors Professor Janet Dong and Professor Paul Talaga, as well as the University of Cincinnati, without all of whom none of this would be possible.