Design Document

ARC - Autonomous RC

Tao Chen, Cierra Shawe, Daniel Stoyer

June 5, 2017

CONTENTS

1 Introduction 5

2 Original Requirements Document 5

3 Requirements Changes 6

4 Design Document 8

4.1 Original Design Document 8

4.2 Changes in Design 9

5 Tech Review 9

5.1 Original Tech Review Document 9

5.2 Changes in Tech 10

6 Weekly Blogs 10

6.1 Cierra's Blog Entries 10

6.1.1 Fall Week 4 10

6.1.2 Fall Week 5 10

6.1.3 Fall Week 6 10

6.1.4 Fall Week 7 10

6.1.5 Fall Week 8 10

6.1.6 Fall Week 9 11

6.1.7 Fall Week 10 11

6.1.8 Fall Week 11 11

6.1.9 Winter Week 1 11

6.1.10 Winter Week 2 11

6.1.11 Winter Week 3 11

6.1.12 Winter Week 4 11

6.1.13 Winter Week 5 12

6.1.14 Winter Week 6 12

6.1.15 Winter Week 7 . . . 12
6.1.16 Winter Week 8 . . . 12
6.1.17 Winter Week 9 . . . 12
6.1.18 Winter Week 10 . . . 12
6.1.19 Winter Week 11 . . . 12
6.1.20 Spring Week 1 . . . 13
6.1.21 Spring Week 2 . . . 13
6.1.22 Spring Week 3 . . . 13
6.1.23 Spring Week 4 . . . 13
6.1.24 Spring Week 5 . . . 13
6.1.25 Spring Week 6 . . . 13
6.1.26 Spring Week 7 . . . 13
6.1.27 Spring Week 8 . . . 14
6.2 Tao's Blog Entries . . . 14
6.2.1 Fall Week 4 . . . 14
6.2.2 Fall Week 5 . . . 14
6.2.3 Fall Week 6 . . . 14
6.2.4 Fall Week 7 . . . 14
6.2.5 Fall Week 8 . . . 15
6.2.6 Fall Week 9 . . . 15
6.2.7 Fall Week 10 . . . 15
6.2.8 Fall Week 11 . . . 16
6.2.9 Winter Week 1 . . . 16
6.2.10 Winter Week 2 . . . 16
6.2.11 Winter Week 3 . . . 16
6.2.12 Winter Week 4 . . . 17
6.2.13 Winter Week 5 . . . 17
6.2.14 Winter Week 6 . . . 17

6.2.15 Winter Week 7 . . . 18
6.2.16 Winter Week 8 . . . 18
6.2.17 Winter Week 9 . . . 18
6.2.18 Winter Week 10 . . . 18
6.2.19 Winter Week 11 . . . 18
6.2.20 Spring Week 1 . . . 19
6.2.21 Spring Week 2 . . . 19
6.2.22 Spring Week 3 . . . 20
6.2.23 Spring Week 4 . . . 20
6.2.24 Spring Week 5 . . . 20
6.2.25 Spring Week 6 . . . 21
6.2.26 Spring Week 7 . . . 21
6.2.27 Spring Week 8 . . . 21
6.3 Dan's Blog Entries . . . 23
6.3.1 Fall Week 4 . . . 23
6.3.2 Fall Week 5 . . . 23
6.3.3 Fall Week 6 . . . 23
6.3.4 Fall Week 7 . . . 24
6.3.5 Fall Week 8 . . . 25
6.3.6 Fall Week 9 . . . 25
6.3.7 Fall Week 10 . . . 25
6.3.8 Fall Week 11 . . . 26
6.3.9 Winter Week 1 . . . 26
6.3.10 Winter Week 2 . . . 26
6.3.11 Winter Week 3 . . . 27
6.3.12 Winter Week 4 . . . 27
6.3.13 Winter Week 5 . . . 28
6.3.14 Winter Week 6 . . . 30

6.3.15 Winter Week 7 . . . 30
6.3.16 Winter Week 8 . . . 31
6.3.17 Winter Week 9 . . . 32
6.3.18 Winter Week 10 . . . 33
6.3.19 Winter Week 11 . . . 33
6.3.20 Spring Week 1 . . . 33
6.3.21 Spring Week 2 . . . 34
6.3.22 Spring Week 3 . . . 34
6.3.23 Spring Week 4 . . . 34
6.3.24 Spring Week 5 . . . 35
6.3.25 Spring Week 6 . . . 36
6.3.26 Spring Week 7 . . . 36
6.3.27 Spring Week 8 . . . 37

7 Project Poster 38

8 Project Documentation 38

9 New Technology Learned 39

10 What We Learned 39

10.1 Cierra . . . 39
10.2 Tao . . . 39
10.3 Dan . . . 40

11 Appendix 1 42

12 Appendix 2 45

1 Introduction

Research into consumer/hobbyist, high performance RC vehicles was requested by Oregon State University via Mr. Kevin McGrath. This project was requested to determine if it is possible to apply high-speed performance during autonomous navigation and obstacle avoidance to a modified RC car at a cost less than four thousand dollars (USD). Autonomous RC (ARC) sought to push the boundaries of what is possible for autonomous RC vehicles. Our research shows that components are decreasing in cost and increasing in performance. The cost-barrier to autonomous research is decreasing dramatically. Our documentation and parts list provides would-be researchers a launching point to continue the work we started in ARC. Our client was the same person who requested the project, Mr. Kevin McGrath. Mr. McGrath is an instructor at Oregon State University. [Who are the members of our team?]The ARC team members are Tao Chen, Cierra Shawe, and Daniel Stoyer. [What were their roles?]Tao was our software and robotics expert, he worked extensively with our software package and got our car working in simulation, he was responsible for the areas of Motion Model, Path Planning, and Autonomous Algorithms (e.g. obstacle avoidance, parallel parking, etc.). Cierra was our electronics and hardware expert, she designed all the mounting hardware used to anchor the sensors to the RC car and did all the wiring/soldering, she was responsible for the ares of Vision Systems, Sensors, and Hardware. Dan was team leader, responsible for making sure the team was on track to hit milestones and Capstone deadlines on time. He was also responsible for overseeing the areas of Image Analysis, User Interfaces, and Radio Communication. [What was the role of the client? (i.e. supervision only, participate in development, etc.)]

2 Original Requirements Document

3 Requirements Changes

No major or minor changes were made to the requirements document. We found that our initial assessment was accurate throughout the project and our client agreed.

Fig. 1. ARC Fall Schedule Gantt Chart

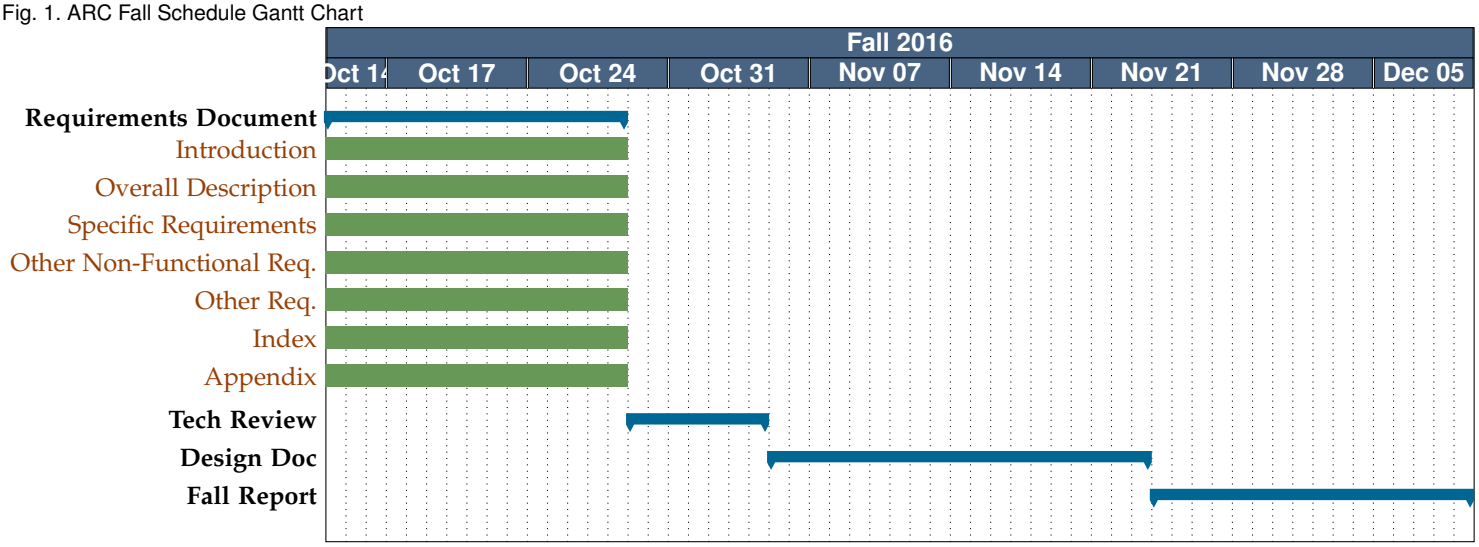


Fig. 2. ARC Winter Schedule Gantt Chart

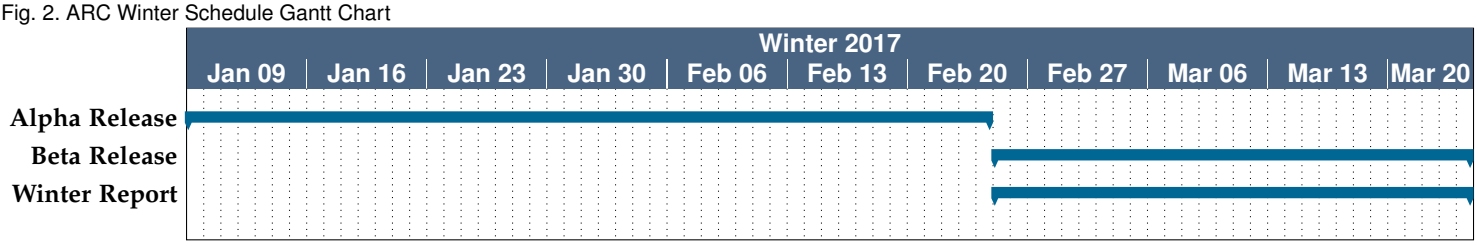
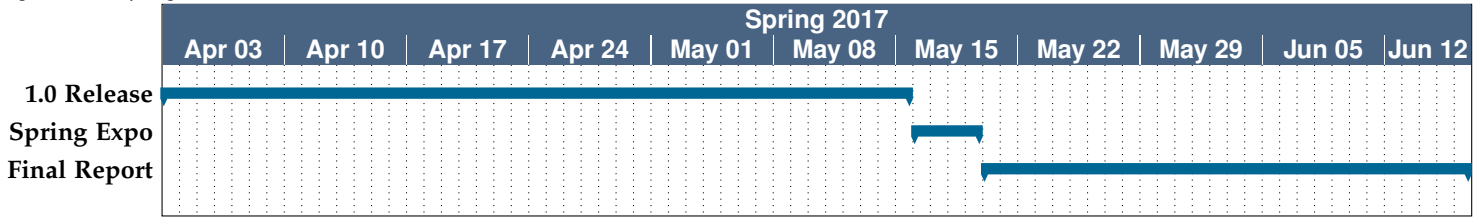


Fig. 3. ARC Spring Schedule Gantt Chart



4 DESIGN DOCUMENT

4.1 Original Design Document

4.2 Changes in Design

We had to make several changes to our design throughout the life of the project. We moved away from stereo vision for obstacle avoidance and went with lidar instead. This had a cascading effect on other parts of our design. Since we didn't use stereo vision, our image analysis design also changed. We didn't need to worry about communicating images over the network so our communication protocols became much more simple.

5 Tech Review

5.1 Original Tech Review Document

5.2 Changes in Tech

As our research and development process matured, we discovered different ways of doing things and changed direction on some of the technology we used for the project. We changed from stereo vision to lidar for depth-finding and obstacle detection. We changed from using the PXFMini autopilot with the Raspberry Pi 3 to using a PWM controller with an Arduino to control motors and servos. We want to use the Q Ground Control GUI to control the vehicle remotely, but had to scratch that when we abandoned the PXFMini. We switched to using RVIZ (visualization package for ROS) for vehicle control

6 Weekly Blogs

6.1 Cierra's Blog Entries

6.1.1 Fall Week 4

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.2 Fall Week 5

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.3 Fall Week 6

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.4 Fall Week 7

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.5 Fall Week 8

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.6 Fall Week 9

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.7 Fall Week 10

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.8 Fall Week 11

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.9 Winter Week 1

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.10 Winter Week 2

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.11 Winter Week 3

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.12 Winter Week 4

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.13 Winter Week 5

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.14 Winter Week 6

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.15 Winter Week 7

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.16 Winter Week 8

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.17 Winter Week 9

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.18 Winter Week 10

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.19 Winter Week 11

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.20 Spring Week 1

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.21 Spring Week 2

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.22 Spring Week 3

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.23 Spring Week 4

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.24 Spring Week 5

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.25 Spring Week 6

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.26 Spring Week 7

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.1.27 Spring Week 8

- **Worked On**
- **Problems Encountered**
- **Plans for next week**

6.2 Tao's Blog Entries

6.2.1 Fall Week 4

- **Worked On**
  - Worked on the problem statement performance measure section.
- **Problems Encountered**
  - None.
- **Plans for next week**
  - None.

6.2.2 Fall Week 5

- **Worked On**
  - Have not contributed so far.
- **Problems Encountered**
  - None.
- **Plans for next week**
  - None.

6.2.3 Fall Week 6

- **Worked On**
  - The SRS document.
- **New**
  - Have a better and clear view of the project.
  - An abstract block diagram that depicts the structure of the project.
- **Problems Encountered**
  - Still a lot of unclear/unspecified aspects of the project.
- **Plans for next week**
  - Individual writing.

6.2.4 Fall Week 7

- **Worked On**

  - SRS document.
  - Tasks breakdown list.
- **Problems Encountered**
  - Still a lot of uncertainties on what hardware/interface we are using.
- **Plans for next week**
  - Finish SRS document.
  - Finish tech review.

6.2.5 Fall Week 8

- **Worked On**
  - Worked on the SRS document and the Tech Review. Did research on efficiency of specific algorithms. Since our vehicle will eventually run in different environments, possibly not predefined, algorithms must be robust as well. Thought about the project as a whole. Realized that we could only have a more and more thorough understanding of the different aspects of this project.
- **Problems Encountered**
  - We still need more hardware. The SRS document hasn't been finished yet. The research we are doing currently may turn out to be in the wrong direction.
- **Plans for next week**
  - Tech review and the SRS document.

6.2.6 Fall Week 9

- **Worked On**
  - Tech review and the SRS document.
- **Problems Encountered**
  - Haven't started the design document yet.
- **Plans for next week**
  - Start the design doc ASAP. Get more hardware in hand.
  - We'd better have a clear view (graph/list) of we need to accomplish along with potential solutions.

6.2.7 Fall Week 10

- **Worked On**
  - None.
- **Problems Encountered**
  - None.
- **Plans for next week**
  - None.

6.2.8 Fall Week 11

- **Worked On**
  - None.
- **Problems Encountered**
  - None.
- **Plans for next week**
  - None.

6.2.9 Winter Week 1

- **Worked On**
  - Ros on the NUC is fully functional.
  - Tested the Lidar unit. It worked great.
  - Went through some tutorials on Ros.
- **Some thoughts on the project**
  - Start our development on the NUC and see if it is possible to migrate all functionalities to the Intel board at the end.
  - Start with the Lidar to implement obstacle avoidance.
  - The two cameras we were given provide two separate feeds. I think we need two video streams in one feed such that we only need to dedicate one port to it and it will provide a better interface to interact with the images.
  - Need to mount the two cameras on something rigid.

6.2.10 Winter Week 2

- **Worked On**
  - Ros tutorials.
- **Problems Encountered**
  - A lot of software only supports certain versions of Ros.
- **Plans for next week**
  - Car model on Gazebo.

6.2.11 Winter Week 3

- **Worked On**
  - More Ros tutorials. RC car model on Gazebo. Simple shape with a rectangular box as the chassis and four wheels.
- **Problems Encountered**
  - Ros Indigo only works with Gazebo 2.X, which doesn't have the model editor built-in. So we have to code our URDF file.
- **Plans for next week**

- Finish the RC car model over the weekend. Make it move in Gazebo. Add sensors to it.

6.2.12 Winter Week 4

- **Worked On**
  - Modified the AutoRally car model.
  - Added a lidar unit and a camera unit to the model.
  - Retrieved data from the two sensors and display results in real time in Gazebo successfully in simulation.
  - Migrated part of the AutoRally codes to our own package.
  - Tried to figure out the dependencies between nodes/packages within the AutoRally project so that we can reuse some of them.
  - Working on a documentation as well.
- **Problems Encountered**
  - AutoRally is a huge project and there is a lot of physics involved. It might take a little bit longer than expected to go through all the nodes/package.
  - The lidar unit worked flawlessly in Gazebo. However, the point cloud produced by the camera unit was rotated 90 degrees up. Worked on it for one day, trying to rotate it so that it actually reflects the reality. No luck.
- **Plans for next week**
  - Keep on working on migrating codes. Plan to finish it by the end of this week. And then we can add our own stuff to it.
  - Try to get the car moving in the simulation.

6.2.13 Winter Week 5

- **Worked On**
  - No entry.
- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

6.2.14 Winter Week 6

- **Worked On**
  - No entry.
- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

6.2.15 Winter Week 7

- **Worked On**
  - No entry.
- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

6.2.16 Winter Week 8

- **Worked On**
  - No entry.
- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

6.2.17 Winter Week 9

- **Worked On**
  - No entry.
- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

6.2.18 Winter Week 10

- **Worked On**
  - No entry.
- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

6.2.19 Winter Week 11

- **Worked On**
  - No entry.

- **Problems Encountered**
  - No entry.
- **Plans for next week**
  - No entry.

Note:

None indicates no blog posts were made for those weeks.

No entry means blogs and notes were made but did not get posted and will appear on later blog posts.

6.2.20 Spring Week 1

- **Worked On**
  - In the last two weeks of winter term I didn't do the whole lot mainly because of finals. However, I managed to get the car to follow paths. It turned the path planning algorithm didn't generate paths for the car to follow. Instead, it generates commands to drive to car. That made things very easy. All I did was translate the commands for the controller. So that's done. I have decided to use the teb_local_planner as our path planning package. I integrated it into the smart_driving package and set it to work on car-like robot. According to the ros wiki for the car-like setup, the configuration is only experimental. Just something to keep in mind. It may sometimes output unpredictable behaviors. Since the teb_local_planner also does obstacle avoidance, obstacle avoidance was handled.
- **Problems Encountered**
  - The steering commands oscillate a lot. The oscillation makes the movements of the car unnatural. We need to look deeper into the code to make sure the steering is smooth.
  - Obstacle avoidance requires a lot of computing power. It requires so much that it's undoable on this laptop. The algorithm does obstacle avoidance by computing the costmap each time it sees a new objects. (detected by laser scans) A costmap is basically a 2D grid that describes the environment and helps to calculate the most efficient path. On this computer, a new costmap takes on average 3 seconds to compute. When there's no objects, the algorithm works fine on this computer. Note that the costmap is only updated when objects are detected. Not sure if the NUC can handle it. Need to find out.
  - The software can be simplified. When I have been doing is just adding new nodes. I didn't consider if the nodes are actually necessary. It's because I didn't want to spend time on going over the AutoRally packages. For example, when I need to translate some data, I just make a node to do the task. In fact, I could have just remap the message name in the launch file. That will reduce a lot of overhead. It will possibly make the software run faster. I will make a table to figure out the data flow and delete all unnecessary nodes.
- **Plans for next week**
  - Continue working on the navigation.

6.2.21 Spring Week 2

- **Worked On**

  - We decided to move away from AutoRally a little bit because it was too complicated. We tried ROS stage, which was a 2D simulation platform. It's simpler than Gazebo. We quickly got the teb_local_planning working because we already had it working with AutoRally.
- **Problems Encountered**
  - I am not sure if stage can simulate gps and imu. It will only simulate laser scan right now. If we couldn't get the imu and gps simulated on stage, it is fine. We can do it directly on the car.
- **Plans for next week**
  - Get the imu and gps working.

6.2.22 Spring Week 3

- **Worked On**
  - Got the odometry in Stage working.
  - Worked on the poster.
- **Problems Encountered**
  - Stage doesn't simulate IMU.
- **Plans for next week**
  - Keep working on Stage and the poster.

6.2.23 Spring Week 4

- **Worked On**
  - Didn't do a whole lot because I was at a point where if I couldn't get the imu data I couldn't move forward in integrating the navigation stack. Dan and I were working on it. But we didn't make much progress. I had a lot homework to do. So I left it to Dan. Looks like he hasn't gotten it working yet.
- **Problems Encountered**
  - The IMU driver provided by ROS outputs imu data. However, the state estimator receives a constant pointer to the data. We need to find a way to transmit the data.
- **Plans for next week**
  - Keep working on the IMU. If that's done. Move on to the GPS.

6.2.24 Spring Week 5

- **Worked On**
  - Worked on the GPS. The data from the GPS was not reliable.
  - Looked for other options for state estimation, because we couldn't the way around the constant pointer.
  - Found one package that seemed promising.
- **Problems Encountered**
  - The AutoRally state estimator was a perfect package to use. But we didn't have the right hardware. The new package we found was intuitive but had a lot of parameters that needed to set up.

  - We are still not able to set it up.
- **Plans for next week**
  - Keep trying the new package.
  - Start preparing for Expo (demo).

6.2.25 Spring Week 6

- **Worked On**
  - Midterm progress report.
  - IMU and GPS.
  - Code Freeze. Documentation.
  - Simulation.
- **Problems Encountered**
  - Our software is very environment-dependent, meaning that it might not be able to run on another computer if the environment was not set up right.
  - GPS data was still not reliable. Can't just work with the IMU.
  - We might need wheel encoders.
- **Plans for next week**
  - Keep preparing for the expo.
  - Keep experimenting the new package.

6.2.26 Spring Week 7

- **Worked On**
  - Prepared the system for expo demo.
  - Expo pitch.
  - Expo.
- **Problems Encountered**
  - A lot of software was installed on the laptop.
  - Wasn't able to run the leddar during Expo, which was kind of embarrassing.
- **Plans for next week**
  - Relax.
  - Relax more.
  - Relax even more.
  - Relax a little bit more.
  - Then start working on the final write up.

6.2.27 Spring Week 8

- **If you were to redo the project from Fall term, what would you tell yourself?**

  - I would tell myself that my team would win the first prize at the end of the year.
  - I would tell myself to get as many sensors as possible.
  - I would tell myself this project was one of the coolest projects so do your best.
- **What's the biggest skill you've learned?**
  - Using ROS.
  - Communication.
- **What skills do you see yourself using in the future?**
  - Since I will be doing robotics, I think knowing how to use ROS will greatly benefit me.
  - I want to do research or development, which means I will most likely be working in a team. Good communication skill will also benefit me lot.
- **What did you like about the project, and what did you not?**
  - I like everything about this project.
  - I don't like the fact that we were given so many writing assignments.
- **What did you learn from your teammates?**
  - I think my understanding of sarcasm improves a little bit.
  - Most importantly, it's collaboration. I was working in a team with more than 15 people. I didn't really learn a lot from that team. A three person team is a good starting point. We can distribute the work load evenly and still have a complete picture of the project.
- **If you were the client for this project, would you be satisfied with the work done?**
  - I will be satisfied but a little disappointed because the car can't drive itself.
- **If your project were to be continued next year, what do you think needs to be working on?**
  - To make the project a little easier, I think we need to get another car first, or make one ourselves. We didn't have much information about the car we have right now. and that car is too much for our purposes. I want the wheels to be completely rigid or just with thin tires. I don't want the car to have less complicated suspensions and drive train.
  - With a new customized car, we can add wheel encoders as well as arrange the hardware in a more elegant way. With the rigid wheels and the new suspension and drive train, we can gain more accuracy to the location estimate.
  - If we have more time, we could possibly customize our own self driving package.
- **Speak a little about your expo experience.**
  - It was very hot. And I didn't get enough free soda.
  - We got a big table, which was nice.
  - We were at a good location too.
  - People seemed to enjoy our project and presentation.

6.3 Dan's Blog Entries

6.3.1 Fall Week 4

- **Worked On**
  - Created the problem statement tex template.
  - The problem statement proposed solution section.
- **Problems Encountered**
  - Not getting feedback on the revised problem statement as quickly as I would have liked.
- **Plans for next week**
  - Create LaTeX template for the SRS.
  - Research LaTeX Gantt charts.
  - Start on SRS.
  - Start on filling out a Gantt chart.

6.3.2 Fall Week 5

- **Worked On**
  - Created an SRS tex template.
  - Did not use IEEEtran.cls because it does not format the SRS properly.
  - I found a template online that has the proper formatting and modified it to follow the IEEE 830 documentation.
  - https://github.com/Eisenbarth/SRS-Tex
  - Created a new 'srs' git branch.
  - Has a new 'srs_template' folder with the template tex file, makefile, and supporting resources.
  - Worked on the SRS.
  - Filled out Software Interfaces and Communications interfaces.
  - Found templates for LaTeX Gantt charts.
- **Problems Encountered**
  - Trying to write a document that assumes we know what components we will need and will be using without really knowing what those things are.
- **Plans for next week**
  - Work on SRS final draft.
  - Work on / finish Gantt chart.
  - Start thinking about the tech review.

6.3.3 Fall Week 6

- **Worked On**
  - Corrected our SRS tex document to use IEEEtran.cls.
  - Corrected the heirarchy format to be numeric, instead of the default Roman numeral.
  - Gantt chart:

    * Created LaTeX document for rendering a Gantt chart:
    * Added files and folder for gantt chart to repo
    * Added comments explaining how to set up groups and tasks.
    * Filled out SRS group with subtasks.
    * Created system for groups to track subtask progress.
    * Fixed formatting.
    * Finished basic layout for Gantt chart
    * Integrated gantt_chart.tex into arc_srs_template.tex
- **Problems Encountered**
  - Getting the Gantt charts to render in LaTeX. It took around 9 hours, and it's not even close to scale-able.
- **Plans for next week**
  - Work on finishing SRS final draft.
  - Determine what my part of the tech review is.
  - Work on tech review.

6.3.4 Fall Week 7

- **Worked On**
  - Tech Review: Researched image analysis software, telemetry radios, and user interfaces for drones.
    * Image analysis software
    * Found DroneKit-Python, ArduPilot, and LibrePilot.
    * Telemetry radios
    * Found 3DR 915 MHz Transceiver, RFD900 Radio Modem, and OpenPilot OPLink Mini Ground and Air Station 433 MHz
    * User interfaces
    * Found QGroundControl, DroneKit-Android, and LibrePilot.
    * Worked on the write up for these components.
- **Problems Encountered**
  - The documentation for the drone software is rather vague when it comes to information on path-finding and image analysis capabilities.
  - DroneKit documentation was somewhat confusing. At times it seemed like it was its own project, but at other times it referenced ArduPilot, making me think that DroneKit is a Python API wrapper on top of ArduPilot.
- **Plans for next week**
  - Finish SRS document.
  - Finish Tech Review document.
  - Start:
    * Planning the design document.
    * Planning the progress report and presentation.
    * Planning the poster.

6.3.5 Fall Week 8

- **Worked On**
  - Completing tech review and additions to the srs document.
  - Researched hardware for telemetry radios.
  - Researched software for UI to be able to communicate with the vehicle.
  - Researched software for image analysis for environment mapping and depth finding based off of sensor data.
- **Problems Encountered**
  - Not too much different that other weeks: the document formats require the writer to have concrete examples and plans for where the project is going. Since this is a research project, and has not been done before, our work-flow really requires some experimentation before we can establish what is going to be used.
- **Plans for next week**
  - Work on creating a plan that will be used in the Design Document.
  - Create the design document.

6.3.6 Fall Week 9

- **Worked On**
  - Filled out specific requirements section of the SRS.
  - Wrote ”Radio Communications”, ”Image Analysis”, and ”User Interfaces” sections of the the tech review.
- **Problems Encountered**
  - Having been delayed on the previous documents, we are about 1 to 1.5 weeks behind schedule. We need to finish up the srs and tech review still. This makes starting on the design document difficult, if not impossible.
- **Plans for next week**
  - Write the design document.

6.3.7 Fall Week 10

- **Worked On**
  - Finishing the srs and design documents.
  - Wrote up how we will test our implementation of the ARC project. My part of the progress report is to synthesize my comments in these progress posts into a coherent subsection of the report. For the presentation, I provided summaries on power point slides, these included graphics and diagrams.
- **Problems Encountered**
  - Our implementation for many components of ARC is currently unknown. More time is required to know exactly what specific software will be used on our project. We need to dive into the APIs used and see how compatible they are and see if we can write our own limited API (meaning simple conversions). So far we have only really had time to look into what open source resources are available at a higher level. This makes writing a detailed design document (in terms of actual code implementation) difficult. So, the design document is being written

in terms of experimental plan. We have milestones that our project needs to meet. After succeeding in the first milestone, we continue to the next.

- **Plans for next week**
  - Write up my section of the progress report.
  - Record the progress report presentation.

6.3.8 Fall Week 11

- **Worked On**
  - Progress Report Created LaTeX template. Wrote the purpose and goals, weekly summaries for weeks 1-6.
- **Problems Encountered**
  - It's finals week, enough said.
- **Plans for next week**
  - It's Christmas break, but I plan to start looking into implementation, how the APIs will work.

6.3.9 Winter Week 1

- **Worked On**
  - Setting up the development environment for Linux and ROS.
    - We will be using the Robotics Operating System (ROS) for communication and control of our vehicle. Linux is the operating system of choice for the ROS platform.
    - I worked through orientation tutorials for ROS to become more familiar with the platform.
- **Problems Encountered**
  - No real problems encountered in week 1.
- **Plans for next week**
  - Get Linux installed on my laptop to dual boot, also get Linux running in VMware for further testing/development capabilities.

6.3.10 Winter Week 2

- **Worked On**
  - Installed Linux to dual boot on my laptop.
  - Got ROS running in Ubuntu 14.04 and 16.04.
  - Got GT Autorally working in 14.04 and 16.04 in both "native" Ubuntu and in a VM.
- **Problems Encountered**
  - My laptop died on Tuesday, so I ordered a new one and got it set up on Thursday/Friday.
- **Plans for next week**
  - Start digging into how Autorally uses ROS and Gazebo (a simulator) to see how we can use it in our project.
  - Look into direct computer-to-computer communication.

6.3.11 Winter Week 3

- **Worked On**
  - Continued with development environment setup.
  - More work in ROS.
  - Found software library for our IMU.
  - Started looking into computer-to-computer direct communication.
- **Problems Encountered**
  - The research I have done stated that new computer should be able to do direct ethernet communication without a crossover cable. I was not able to get the computers to communicate. I need to get a crossover cable to see if that is the issue, or if some other setting is incorrect.
  - We received a corrupt image for the PXFMini, so we will need a new one sent to us.
  - We were hoping we could find a ready-made computer model for our RC car for ROS or Gazebo, we have not found one as yet. This means that we might need to spend quite a bit more time building a model for simulation, if we want to go that route for development.
- **Plans for next week**
  - Get computers talking to each other directly over ethernet.
  - Get telemetry data from the RC to a remote computer.
  - Get command from remote computer to RC car.

6.3.12 Winter Week 4

- **Worked On**
  - Got computers talking with each other directly over ethernet.
  - In Ubuntu on both systems:
    Edit wired connection set ipv4 to manual set the IP address subnet mask, and gateway The networking service may need to be started: sudo service network-manager restart Got Raspberry Pi3 talking with computers over ethernet as well.
  - Worked on tutorials for ROS publishers and subscribers
  - Got ROS publisher on computer A talking with ROS subscriber on computer B.
  - This test is between my personal laptop and the Raspberry Pi 3.
    - It is important that /.bashrc have the following lines:
      For computer running ROS master (this is the Raspberry Pi, and note that IP addresses are the same): export export ROS_MASTER_URI=http://192.168.1.2:11311 export ROS_IP=192.168.1.2 For client computer (note ROS_IP is local IP address): export ROS_MASTER_URI=http://192.168.1.2:11311 export ROS_IP=192.168.1.3
  - Got the NUC talking to the Raspberry Pi 3 in ROS following a publisher/subscriber tutorial:
    http://docs.erlerobotics.com/robot_operating_system/ros/basic_concepts/examples/publisher_and_subscribers
  - Able to get internet working over ethernet. This will allow us to update the Raspberry Pi3, install needed ROS packages and other library dependencies.
- **Problems Encountered**

  - The keyboard layout on the Raspberry Pi 3 was set to UK by default.
  - Fixed the layout permanently with: sudo raspi-config
  - Other settings can be configured for the Pi 3 from the same config menu.
  - WiFi is not working on the Raspberry Pi 3, internet is not working, in general.
  - Able to get internet working over ethernet.
  - The PXFMini did not power on.

    Discovered that the PXFMini cape was connected to the Raspberry Pi 3 incorrectly.

    After connecting correctly, the Rpi3 boots, but the PXFMini LED status lights show a code that tells us that it could not launch. This could be normal, or caused by an error. The documentation says that a vehicle selection should appear when the Rpi3 boots up, but I do not see that.
  - Upgraded the Raspberry Pi 3 using sudo apt-get dist-upgrade and then the rpi would not boot any more.
    * It turns out that the upgrade somehow wiped out the required kernel7.img needed for the OS.
    * Copied a kernel7.img from another system image, along with other missing files, the rpi now boots, and WiFi is now working too!
  - Discovered that our power supply (USB power from the NUC) is under-powered for the Raspberry Pi 3, especially with the PXFMini attached. We will need to try using the DC power converter and the passthrough to the Rpi. We will need to get a voltage-meter to make sure our power is correct (need 5V, 2.5A).

- **Plans for next week**
  - Get proper power to the Raspberry Pi 3.
  - Get the PXFMini runnning properly.
  - Make servos and motors run via the ROS and the PXFMini.

6.3.13 Winter Week 5

- **Worked On**
  - Got the PXFmini launched and armed with instructions for installing binaries for APMrover2.
  - Got the Raspberry Pi 3 (RPi3) connected to the internet via ethernet using the original Erle OS image.
    * Steps for connecting to internet using ethernet:
      In /etc/network/interfaces, uncomment the eth0 section.
    * Steps for getting WiFi internet working with home router you will need to edit two files, interfaces and wpa_supplicant.conf:
      · Back up original /etc/network/interfaces file.
      · In /etc/network/interfaces, comment out the lines for wlan0 and add the following lines
      · auto wlan0
      · iface wlan0 inet dhcp
      · netmask 255.255.255.0
      · wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
      · Back up original /etc/wpa_supplicant/wpa_supplicant.conf
      · Edit /etc/wpa_supplicant/wpa_supplicant.conf:

- Comment out current network block
- Add new network=... block containing the following lines:
- ssid=”¡name-of-network-ssid¿”
- psk=”¡network-password¿”
- Reboot the Rpi3.

* I was able to publish to mavros topics and confirmed that the subscribers did get the information. So far it has not caused the actuators to fire.
* I was also able to confirm that the pxfmini is talking with the rpi, I can see the imu data being transmitted and changing when I move the unit around.

- **Problems Encountered**
  - Continued to struggle with getting internet working using WiFi on the RPi3 and the original Erle image. This is important to get the proper software installed.
  - I can’t compile any of the code that is given in the examples because dependencies are missing... I’ve gotten one script to actually run, but it doesn’t do anything for the pxf. and the documentation is so incomplete, it just assumes that ”everything” (whatever that is) is set up and ready to go to launch the given script. And in places where it tells you to do something, it doesn’t explain what is necessary for it to run, or even what the code is doing. When I try to compile a cpp file (ground_rover.cpp) with include¡ros/ros.h¿ the g++ compiler says it can’t find the file, which means that the ros library wasn’t installed into the system include directory. I can compile using an include option and point to the correct directory for the ros include libraries, but then it errors out on a different file not found in one of the included headers of ground_rover.cpp.
  - The instructions from erle-robotics in ”first steps” show a mavros node ’raspicam_node when running rosnode list, that node is not listed after going through their instructions. If I run rostopic list I get a similar output, but rostopics and rosnodes are two different things.
  - Getting WiFi working for the internet breaks launching rostopics. The wifi settings must be set back to original to restore rostopic settings. – Use erle-reset.
  - After talking with our client, Kevin McGrath, we have determined that something is probably not set up properly, or strangery of some kind is afoot. The rpi3 and Erle PXFmini should work with the erle-image ”out of the box”, but it does not on our rpi3, for an as-of-yet unknown reason. Kevin and I are going to try to figure out what is going on.
  - Well, I am now able to publish (via command line) to any of the mavros topic subscribers. I have only published to few of them, but I know how to do it now. I am also able to subscribe to topics to see what they are sending. The pxfmini is ”working”, in the sense that is communicating with the rpi3. I can see the IMU data, which changes when I move the pxfmini. I don’t know if the numbers or correct, or even usable, but at least they are numbers...
  - Unfortunately, nothing I have tried has gotten the actuator to move. When I arm the pxfmini, the actuator fires erratically until I send a message via mavros/rc/override, then the motor stops, but further messages do not cause the motor to do anything.

    At this point, I feel like I could spend the rest of the term figuring this out. Obviously that is not an option.

    I’m talking with Kevin about this problem. He’s also having a hard time getting the pxfmini to work via

software.

- Progress is going very slowly due to very poor documentation for the pxfmini and mavros (an api for RC communication).

- **Plans for next week**
  - It's week 6 and the midterm report and presentation is due.
  - Get our required One Note project environment setup and filled out.
  - Make revisions to the documentation for my areas of focus.
  - Make my portion of the video presentation.

6.3.14 Winter Week 6

- **Worked On**
  - Got the OneNote workbook up.
  - Added document and presentation PDFs to OneNote.
  - Added revision table templates to OneNote
  - Added "template" or empty revision tables to documents (Problem Statement, Tech Review, SRS, Design).
  - Made revisions to:
    * tech review
    * design doc
    * Looked at srs and updated gantt chart.
    * Created template for winter midterm report.

- **Problems Encountered**
  - None.

- **Plans for next week**
  - Install QGroundControl on my system and see if we can control the pxfmini.
  - Try to get "gtest" (a library necessary for one of the erle tutorials) installed.
  - make a determination if we want to continue with trying to use the pxfmini autopilot or switch to a simple controller.

6.3.15 Winter Week 7

- **Worked On**
  - Got the "Teleoperating" tutorial installed. The tutorials allows control of the RC vehicle using arrow keys on the keyboard. Kevin McGrath figured out how to install it:
    * mkdir -p /erle_ws/src
    * cd /erle_ws/src
    * git clone https://github.com/erlerobot/gazebo_cpp_examples

    * git clone https://github.com/ros-perception/vision_opencv
    * git clone https://github.com/ros-perception/image_common
    * cd ..
    * cd /usr/lib/arm-linux-gnueabihf/
    * sudo ln -s libboost_python-py34.so libboost_python3.so
    * sudo apt-get install libgtest-dev
    * catkin_make –pkg ros_erle_cpp_teleoperation_erle_rover
  - The package now installs, and the keyboard changes the input to the rostopic /mavros/rc/override, but the motor does not activate.
  - Able to set the parameter SYSID_MYGCS, this is needed to override RC in and enable computer control of the autopilot.
    * To set using $ rosrun mavros mavparam set SYSID_MYGCS 1, the autopilot must be launched and armed.
  - Able to use /mavros/rc/override to publish data to /mavros/rc/in. This is very important because it is necessary for controlling motors. The motors are still not moving, however.
  - Installed GCS software (ArduPilot) and connected telemetry radios. Now able to see attitude of the autopilot in a gimble. But nothing else is connecting, not able to control motors at all. Cannot disarm/arm the autopilot from GCS.

- **Problems Encountered**

  - Using the above ”Teleoperation” tutorial, I am able to change the channel values for steering and throttle to /mavros/rc/in. I can see these values change on the GCS via the telemetry radios. This means that data is being sent over mavros and the telemetry radios are communicating properly. So, the problem is somewhere between /mavros/rc/in and the PXFmini. At this point we have spent almost 4 weeks trying to get the PXFmini to work. This was supposed to work ”out of the box”. We will move on to a different autopilot.

- **Plans for next week**

  - Work out the kinks while installing AutoRally.

6.3.16 Winter Week 8

- **Worked On**

  - Went through uninstalling and reinstalling Autorally.
  - The autorally softare package has a few different parts to it and odd/obscure errors can occur during installation. My goal was to find errors that arise and document how to fix/correct them so that users will have a better installation experience and get the system up and running faster.
  - Found the following error:

```
CMake Error at autorally/autorally\_control/CMakeLists.txt:19 (find_package): By not pr

Could not find a package configuration file provided by "Eigen3" with any of the follow
```

```
Eigen3Config.cmake
eigen3-config.cmake

Add the installation prefix of "Eigen3" to CMAKE_PREFIX_PATH or set "Eigen3_DIR" to a d
```

  - This was resolved by changing the calling CMakeLists.txt file:
    Original: find_package(Eigen3 REQUIRED)
    Changed to:
    find_package(PkgConfig)
    pkg_search_module(Eigen3 REQUIRED eigen3)

  - I was able to reproduce the above error and verify the above fix worked consistently. I have not been able to test this on other systems.
  - Worked through the uninstall process and identified where files were install to, this mostly had to do with GTSAM installation. Autorally itself is fairly localized and doesn't install any libraries to /usr/..
  - I have a fairly solid grasp of the uninstall/reinstall process for autorally, which means that as we move forward with integrating our own software we will be able to provide a more smooth experience for other users through documentation, and perhaps automation of the installation process.
  - Still not sure if the Eigen3 error is something that will occur on a first-time install. I don't recall seeing the error before.

- **Problems Encountered**

  - It took a few days to figure out what all needed to be cleaned out of the system to be able to reinstall the software. Then it took another day or two to figure out how to resolve installation errors that we hadn't seen before.

- **Plans for next week**

  -
  - Get autorally installed with the new files Tao is working on.
  - Get Telemetry radios working without the pxfmini...

6.3.17 Winter Week 9

- **Worked On**

  - Integrating Tao's implementation in to the installation process.
  - Able to launch rviz using a custom configuration file. This allows us to load a specific configuration on different computers which will eliminate the need for tedious setup of the rviz environment.

- **Problems Encountered**

  - None this week.

- **Plans for next week**

  - Continue with building our installation process.

6.3.18 Winter Week 10

- **Worked On**

  - None

- **Problems Encountered**

  - My wife had a pregnancy-related emergency and our son was born via emergency c-section.

- **Plans for next week**

  - None.

6.3.19 Winter Week 11

- **Worked On**

  - None.

- **Problems Encountered**

  - I was in the NICU in Eugene.

- **Plans for next week**

  - None – Spring Break

6.3.20 Spring Week 1

- **Worked On**

  - Talked through the complexity of AutoRally with Tao and Cierra. We determined that AutoRally is too complex to try to integrate into our project in the time-frame we have left. We decided to drop AutoRally and focus solely on ROS for the navigation stack.

- **Problems Encountered**

  - The physical USB port on one of your telemetry radios broke. We cannot do radio communication until it is fixed.

- **Plans for next week**

  - Get the IMU, Lidar, Leddar, and GPS drivers installed on the NUC.

6.3.21 Spring Week 2

- **Worked On**
  - Installed the IMU and Lidar drivers on the NUC.
  - Got rostopics for the IMU and Lidar publishing data and able to visualize the data in RViz.
  - Worked with Tao on figuring out how to get another simulation (Stage) working with our hardware packages.
- **Problems Encountered**
  - The Leddar unit is not being recognized in Windows or Linux via USB. It was working earlier, but isn't now. We will try the RS485 interface to see if the Leddar unit is bad, or just the USB interface.
- **Plans for next week**
  - Integrate IMU and GPS.
  - Get simulation stable using GPS to smooth out the IMU calculations.
  - Refine hardware mounting on the car

6.3.22 Spring Week 3

- **Worked On**
  - Edited the outcomes and findings for the poster.
  - Continued to try to get IMU and GPS data converted into something the AutoRally state estimator will accept.
- **Problems Encountered**
  - Still having no luck converting the data we need.
- **Plans for next week**
  - Help Tao with the simulations and Cierra with the hardware mounting.

6.3.23 Spring Week 4

- **Worked On**
  - Attempting to convert our ROS-published imu data to something that imuGPSestimator can use. imuGPSestimator is from GT AutoRally, we need it to integrate our GPS and IMU data to determine the position of the car in the world.
- **Problems Encountered**
  - We need the imu and gps data to accurately track the position of our car in the world. Integrating that data is not trivial. This is absolutely required to do any autonomous navigation in the real world. It turns out that this very problem is what was really helpful in having an autopilot. The autopilot does all this integration for you. Without the autopilot we have to come up with an integration and interpretation solution on our own.

  - We do not have time to develop our own software to integrate GPS and IMU data to accurately track the position of our car in the world.
    * We are trying to use the estimator from GT AutoRally to integrate GPS and IMU data. However, imuGPSestimator subscribes to a different data type (IMUConstPtr) than the data type published by by our imu (IMU data type).
    * We are trying to resolve this difference by converting IMU to IMUConstPtr. This has proven very challenging. There is no documentation on what IMUConstPtr actually is. The definition shows us that it's super type is shared_ptr from the Boost library.
    * shared_ptr seems to be a generic type used to track pointer usage and garbage collect automatically when the pointer is no longer used. It does not seem to be relevant to our need to convert from IMU to IMUConstPtr.
  - So far, we have not been able to convert the data.

- **Plans for next week**
  - Do as much as we can to get our software running the car.
  - Get the poster more-or-less finalized and approved by our client.
  - Get all the code we have been working on/with uploaded to our Github repo.
  - Submit our poster for print before May 1.

6.3.24 Spring Week 5

- **Worked On**
  - Continued researching how to convert our IMU data into the constant pointer data structure that AutoRally wants.
  - Reviewed the poster put together by Cierra and Tao while I've been caring for my wife and newborn son.
  - Worked with Tao on finding different simulations to possibly use during Expo.

- **Problems Encountered**
  - Adjusting to a newborn baby is rough!
  - Getting accurate state estimation for the vehicle is proving problematic.
  - We found out that we misunderstood poster submission requirements. We did not submit the printing job to Printing Media Services within the allotted time-frame for Engineering Expo posters.

- **Plans for next week**
  - Continue to hack away at getting sensor data integrated and experiment with our navigation stack in simulation.
  - Work on our mid-term submission requirements.
    * Written progress report
    * Progress report video

    * Expo pitch
    * Final project write up rough draft.
  - Practice project presentation for the advisory board.

6.3.25 Spring Week 6

- **Worked On**
  - Worked with Tao testing a simulation package called Stage. It is very light-weight and, used with RVIZ, it allows use to control and environment and display our car much more simply than Gazebo allows. We will likely use this combination, Stage with RVIZ, to present at Expo.
  - Wrote my section of the mid-term progress report.
  - Tao, Cierra, and I recorded the mid-term progress video.
  - Wrote up my section of the write-up rough draft, which was very similar to the progress report.
  - We worked on our Expo pitch and project presentation. I am doing the intro, Tao is talking about what we did, and Cierra is wrapping things up in the conclusion.
- **Problems Encountered**
  - We are still struggling with getting the data from the navigation stack to be usable on actual hardware. The main problem is that the simulation is using "perfect" data for the sensors, in real life the sensor give somewhat inaccurate data. This can be countered by adding redundant sensors to increase noise, which smooths out the data, and/or add an odometry encoder to the car to track distance traveled which, added to the data from the sensors, will give the software a more accurate estimation of the state of the car.
- **Plans for next week**
  - Last minute work on expo presentation.
  - Build stand for the car.
  - Give project presentation to the advisory board.
  - Do Expo.

6.3.26 Spring Week 7

- **Worked On**
  - Built stand for the car.
    * Designed base and legs to raise and support the vehicle off the ground.
    * Painted the stand with a gloss finish coat.
  - Gave project presentation to the advisory board.
  - Presented our project at Expo.
- **Problems Encountered**
  - No major problems encountered this week.

- **Plans for next week**
  - Continue to try to implement functionality for hardware.
  - Build out some of the interfaces to allow for future projects to implement the hardware.
  - Continue working on the instructions/documentation so that future projects will be able to get to the point we are at very quickly and go from there.
  - Work on the Project Write Up, add more detail and refine the findings and conclusion sections.

6.3.27 Spring Week 8

- *If you were to redo the project from Fall term, what would you tell yourself?*
  - Identify critical points of failure. Things that could possibly set us back significantly if they were to fail.
  - Test those points early, if possible.
  - Be more proactive about getting help for problems. Don't struggle with a problem for more than a few days before getting help.
- *What's the biggest skill you've learned?*
  - I learned about the Robotics Operating System (ROS), a general purpose, open source platform for robotics communication and operation. ROS provides a wide range of libraries for communication and control of many kinds of robots, from mechanical arms to rovers, like the car we worked on during Capstone. This project was complicated enough and covered a long enough period of time that I was able to research ROS in depth. I went through beginner tutorials to understand the basics of how ROS works and then applied some of the concepts from the tutorials to our specific needs in our project. I was able to go through a fairly complete learning cycle to become comfortable with how ROS works. In addition, I had no prior experience with autonomous vehicles, this project gave me exposure to the world of autonomy and gave me a greater appreciation and understanding of what is involved for vehicles to operate autonomously.
- *What skills do you see yourself using in the future?*
  - Computer Science is about solving problems (or trying to...), this project required us to answer the question, "Is high performance autonomous navigation possible on inexpensive hardware?" We had to identify the problem we were trying to solve, research possible techniques/approaches for solving the problem, test our solutions, make adjustments and try again, communicate with a client and conform to requirements, work as a team for a "long" period of time, then arrive at a conclusion and present our findings and conclusion.
  - These experiences help develop/refine a wide range of skills, both technical and relational, that apply directly to the CS field and life in general.
- *What did you like about the project, and what did you not?*
  - I liked my client and my team. Cierra and Tao were great teammates and always positive, eager to work on the project. Kevin was a great client, giving us pointers and keeping expectations within reason.
  - I liked that we had several terms to work on the project, this allowed us to do something fairly significant and

complex, versus only having one term as would be the case with a normal course.
  - I liked the project itself, learning about autonomous vehicles and the current state of the hobbyist-level in this field was very interesting and it was fun to try to get our car working.
  - I was not a fan of having to do so many "Progress Report" videos. They were quite time-consuming and a little frustrating. Each progress report took around a week to put together, that's about 4 weeks taken from Fall and Winter terms. I realize that accountability needs to be maintained, perhaps the written report would suffice? It seemed like our weekly meetings with our TA were like a running version of the video.
  - I did not really like the "Wired" article. I get that the point was to practice our expo pitch and interact with others, but grading us on the formatting and style of writing was a bit much.

- *What did you learn from your teammates?*
  - I learned from them how to have fun with the process. I think at the beginning of the project my demeanor was too serious. Their enthusiasm and general positivism helped me lighten up and have more fun (I don't know if they would agree, haha). I learned more about electronics from Cierra and more about software simulations from Tao.

- *If you were the client for this project, would you be satisfied with the work done?*
  - I think I would be satisfied. Our objective was to determine if it's possible to make a high performance autonomous RC car with inexpensive hardware and open source software. We put more than sufficient work to come a informed conclusion on the matter.

- *If your project were to be continued next year, what do you think needs to be working on?*
  - A team would need to take our hardware and software platform and work on creating a state estimator with the sensors and data we are using. The state estimator is need to keep the car in control during autonomous operation.

- *Speak a little about your expo experience.*
  - Expo was a fun experience for me, overall. We gave a presentation of our project to the industry advisory board and then talked with attendees during the event. It was fun explaining what we did and why we did it. I talked to quite a few kids and it was really fun seeing them interested in the project and answering their questions.

7 Project Poster

8 Project Documentation

- How does the project work?
  - What is its structure?
  - What is its Theory of Operation?
  - Block and flow diagrams are good here.

- How does one install your software, if any?
- How does one run your software?
- Are there any special hardware, OS, or runtime requirements to run your software?
- Any user guides, API documentation, etc?

9 New Technology Learned

- *What web sites were helpful?*
  - http://wiki.ros.org
  - https://github.com/AutoRally/autorally
- *What, if any, reference books really helped?*
  We did not use any reference books.
- *Were there any people on campus that were really helpful?*
  Our client was Kevin, who was also our instructor. So it was very helpful. He provided us with all the hardware and gave us instructions on how to use them.

10 What We Learned

- We experienced how a research project was approximately carried out.
- We got to work in a diverse team, which helped us improve our communication skills.
- When working in a small team, the relationship between teammates became more intimate, which make us become more considerate people.
- ROS was a big part of this project, and we only had little to none experience working with. We learned a lot on how to use ROS. What we were able to accomplish was incredible.
- MORE TO ADD.

10.1 Cierra

- What technical information did you learn?
- What non-technical information did you learn?
- What have you learned about project work?
- What have you learned about project management?
- What have you learned about working in teams?
- If you could do it all over, what would you do differently?

10.2 Tao

- What technical information did you learn?
  - A basic structure of a project.

  - Using ROS.
  - Version control.

- What non-technical information did you learn?

  - Communicating with teammates.
  - Explaining concepts and reporting to client.
  - Recording progress.
  - For me personally, this project made me like to think about problems with the big picture while focusing on the individual components.

- What have you learned about project work?

  - Coding could be easy if the project was planed out nicely.
  - To make a good plan, one must view the project from different prospectives, and listen to others' opinions.
  - Keeping track of progress is extremely important.
  - Stick with the plan, and yet quickly switch other solutions when the original does not work out.

- What have you learned about project management?

  - Lists are useful for keeping track of hardware.
  - Version control is important but be careful when using it.

- What have you learned about working in teams?

  - Good teammates are key. It depends on luck.
  - Always try to create a positive atmosphere even things aren't working so well.
  - Helping teammates out is not sacrificing. You gain more than you lose.

- If you could do it all over, what would you do differently?

  - I'd probably prioritize things differently and look for other options when something didn't work out. For example, we could have worked on the software and hardware simultaneously, which I think was our original plan. But then we hit a road block that totally stalled our work on hardware from progressing. Other than that, I don't think I would do much different.

10.3 Dan

- What technical information did you learn?

  - How to develop for robots:

    The Robotics Operating System (ROS) is an open source general-purpose software platform for developers working with all manner of robots. ROS allows the developer to use its extensive libraries to implement custom code for controlling robots and provides communications and navigation libraries that are very useful for autonomous vehicle applications. We wrote some programs in C and python using Ros libraries to communicate with the vehicle.

  - How to set up simulators:

    We used both Gazebo and Stage are simulation environments that we used on this project. I learned how to set up and run both.
  - RC cars and Autonomy

    I had never worked with RC cars, or tried any sort of robotics, so diving into the field of autonomous vehicles really stretched me. There is *so much* that goes into research and development of autonomous vehicles. The computer has to either know the exact state of the vehicle via sensor data, or be able to estimate the vehicle's state closely enough to operate.
  - CMake

    CMake is a platform for build and testing software. It is open source. ROS uses CMake as part of its package installation platform, called Catkin.
  - I learned more about bash and Linux, creating a script to launch AutoRally, spawning multiple terminals and processes.
  - I learned about lidar and point-clouds. I had never seen lidar in action before. I learned that there are different types of lidar. We used IR and some sort of LED technology.

- What non-technical information did you learn?

  - I learned that "autonomous" has many different meanings when related to vehicles. My assumption was that it meant that cars could navigate pretty much anywhere on their own. I found out that a car can be considered "autonomous" if it performs *any* operation on its own. For instance, Georgia Tech's AutoRally project claims their car is "autonomous" and the car does go around a track without a user controlling the car directly. However, the car does guide itself, it follows predetermined way-points and performs no obstacle avoidance what-so-ever. The barriers around the track were in place to merely keep the car from going completely off the rails should something go wrong. The car did self-correct steering around the turns and kept the car under control will power-sliding around corners. So, the car was autonomous, just not in the way that I had though.

- What have you learned about project work?

  - Plan for setbacks and do not be disappointed when they happen.
  - It's really hard to work on a complicated project part time while taking other classes.
  - Ten hours of trial and error will save an hour of good design planning.

- What have you learned about project management?

  - It would be very helpful to use project management software.

    We did *not* use any sort of project management software for this project, by "project management software" I mean packages such as MS Project, or Redmine. The weak attempt at project tracking using manually implemented Gantt charts with no guidance on what to use or where to go for it was appalling. But I digress...
  - Milestones need to be clearly identified and consistently communicated throughout the life of the project.

- What have you learned about working in teams?

- Good communication is key to keeping a team operating smoothly.
- When everyone pitches in and they are enthusiastic about the project, the work is quite fun.

- If you could do it all over, what would you do differently?
  - Identify critical points of failure. Things that could possibly set us back significantly if they were to fail. Test those points early, if possible.
  - Be more proactive about getting help for problems. Don't struggle with a problem for more than a few days before getting help.
  - Start with a rover that has an autopilot integrated into it, possibly even already has some sensors on it. Our team had *no* experience going into this project. Trying to start from scratch was probably a bit too much, if the goal was to implement high-speed performance. It would have been more realistic to attain if we had started with a vehicle that had some sort of autonomous capability already and then we "upgrade" it to perform at a high rate of speed.
  - I would use project management software.

11 APPENDIX 1

Essential Code Listings. You don't have to include absolutely everything, but if someone wants to understand your project, there should be enough here to learn from. If you worked within a larger project, something like a patch file might be a good way to go.

```
/*********************************************
* @file lidarDetection.cpp
* @author Tao Chen <chentao@oregonstate.edu>
* @date Feburary 21, 2017
* @copyright 2017 Oregon State University
* @brief ROS node to generate new waypoints based on lidar scan
*
* @details When an obstacle is detected, it alters the original
* set of waypoints to go around the obstacle. This node subscribes to
* the laser_scan message and the current_list_of_waypoints message and
* publish the new_waypoints message.
*********************************************/

#include "lidarDetection.h"

namespace autorally_smartdriving{
    LidarDetection::LidarDetection(){
        l_lidarSub = l_nh.subscribe("/autorally_platform/laser_scan", 10, &LidarDetection::gatherLidarData, this);
        l_lidarPub = l_nh.advertise<sensor_msgs::LaserScan>("scan", 10);
```

```
20
21          samples = 2000;
22          laser_frequency = 10;
23          scan.ranges.resize(samples);
24          scan. intensities .resize(samples);
25      }
26
27      LidarDetection::~LidarDetection(){}
28
29      void LidarDetection::gatherLidarData(sensor_msgs::LaserScan data){
30          int  counter = 0;
31
32          ros :: Time scan_time = ros::Time::now();
33
34          scan.header.stamp = scan_time;
35          scan.header.frame_id = "lidar";
36          scan.angle_min = data.angle_min;
37          scan.angle_max = data.angle_max;
38          scan.angle_increment = data.angle_max / samples;
39          scan.time_increment = (1 / laser_frequency) / samples;
40          scan.range_min = data.range_min;
41          scan.range_max = data.range_max;
42
43          for(counter = 0;  counter < samples; counter++){
44              scan.ranges[counter] = data.ranges[counter];
45              scan. intensities [counter] = data. intensities [counter];
46          }
47
48          // publish scan to scan topic
49          l_lidarPub.publish(scan);
50      }
51  };
52
53  int  main(int argc,  char** argv){
54      ros :: init (argc,  argv,  "LidarDetection");
55      autorally_smartdriving::LidarDetection lidarDetection;
56      ros :: spin() ;
57  }
```

Code Example 1. Example Custom ROS Node

```
<launch>

  <!-- ************** Global Parameters *************** -->
  <param name="/use_sim_time" value="true"/>

  <!-- ************** Stage Simulator *************** -->
  <node pkg="stage_ros" type="stageros" name="stageros" args="$(find stage_launch)/stage/empty.world">
    <!-- <remap from="base_scan" to="scan"/> -->
    <remap from="base_scan" to="base_scan_0"/>
  </node>

  <node pkg="move_base" type="move_base" respawn="false" name="move_base" output="screen">
    <rosparam file="$(find smart_driving)/config/costmap_common_params.yaml" command="load"
        ns="global_costmap" />
    <rosparam file="$(find smart_driving)/config/costmap_common_params.yaml" command="load"
        ns="local_costmap" />
    <rosparam file="$(find smart_driving)/config/local_costmap_params.yaml" command="load" />
    <rosparam file="$(find smart_driving)/config/global_costmap_params.yaml" command="load" />
    <rosparam file="$(find smart_driving)/config/base_local_planner_params.yaml" command="load" />

    <param name="base_global_planner" value="global_planner/GlobalPlanner" />
    <param name="planner_frequency" value="1.0" />
    <param name="planner_patience" value="5.0" />

    <param name="base_local_planner" value="teb_local_planner/TebLocalPlannerROS" />
    <param name="controller_frequency" value="5.0" />
    <param name="controller_patience" value="15.0" />

    <param name="clearing_rotation_allowed" value="false" />
  </node>

  <node name="map_server" pkg="map_server" type="map_server" args="$(find
      stage_launch)/maps/empty_world.yaml" output="screen">
    <param name="frame_id" value="/map" />
  </node>

  <!--<node pkg="amcl" type="amcl" name="amcl" output="screen">
    <rosparam file="$(find stage_launch)/cfg/amcl_params.yaml" command="load"/>
    <param name="initial_pose_x" value="0" />
```

```
    <param name="initial_pose_y" value="0" />
    <param name="initial_pose_z" value="0" />
  </ode>-->

  <!-- <include file="$(find leddar)/launch/leddar.launch">
    <arg name="serial" value="AJ04071" />
    <arg name="frame" value="base_laser_link_0" />
    <arg name="fov" value="45" />
    <arg name="range" value="50" />
  </include> -->

  <include file="$(find stage_launch)/robot_localization.launch"/>

  <node name="odom" pkg="autorally_smartdriving" type="odom" />
  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find stage_launch)/stage/rviz_navigation.rviz"/>
</launch>
```

Code Example 2. Example Custom Launch File for Stage

12 APPENDIX 2

Anything else you want to include. Photos, etc.