NYU

TANDON
SCHOOL OF
ENGINEERING

NEW YORK UNIVERSITY

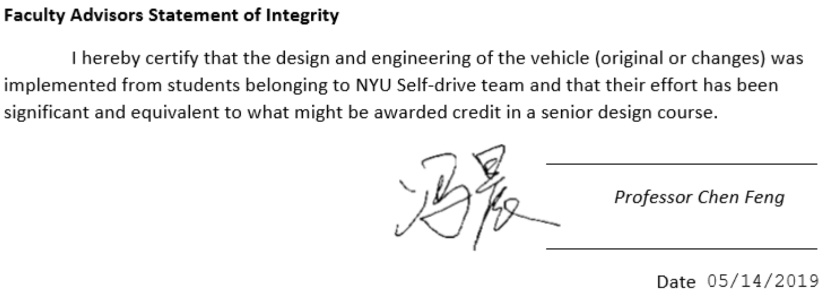
“Little Beetle”

Self-Drive Design Report

Team Leaders

Shivam Bhardwaj
shivam.bhardwaj@nyu.edu
(Team Captain)

Feng Chen
cfeng@nyu.edu
(Faculty Advisor)



Faculty Advisors Statement of Integrity

I hereby certify that the design and engineering of the vehicle (original or changes) was implemented from students belonging to NYU Self-drive team and that their effort has been significant and equivalent to what might be awarded credit in a senior design course.

冯晨

Professor Chen Feng

Date 05/14/2019

Table of Contents

Conduct of design process, team identification and team organization

1.1 Introduction

New York Universitys Vertically Integrated Project Team: NYU Self Drive, is made up of 26 engineers with knowledge in robotics, computer vision, management, machine learning, and deep learning. NYU Self Drive has transformed a 2010 low-speed golf cart into a fully functioning self-driving vehicle: Little Beetle. Our mission is to better educate our future engineers to facilitate Robotics and Artificial Intelligence research to build the worlds safest driver. Our team has experience in environments like RoboCup Competitions, Combat Robots, CyberTruck Challenges, e-Yantra and Research Expos. Our team of engineers are passionate about the autonomous vehicle industry and have contributed their skills and time to work on Little Beetle, our competition car. With teamwork, dedication and passion contributed from all members, NYU Self Drive has successfully built an autonomous vehicle that can take on different road conditions with the four state of the art algorithms we have programmed in the car. Since this is the second time our team will take part in the IGVC competitions, this year we decided to leverage our team effort by expanding the team size and improve certain aspects of the car, such as the control layer, where a new model predictive control (MPC) was implemented for controlling the steering angle and the velocity of Little Beetle.

1.2 Organization

The overall team is made of 26 undergraduate and graduate students and our Faculty Supervisor, Professor Chen Feng. Shivam Bhardwaj is the team captain who is responsible for overseeing the mechanical, electrical, computer science and managerial aspects of the team. He communicates with different subgroups, supports the Operations Manager, Kyra-Lee Harry, during weekly team meetings and provides guidance and support for the entire team. With diligent members who research and implement different algorithms in the car, the car has depth perception, speed estimation using stereo cameras, path planning algorithms, map generation and obstacle detection and avoidance. With Shivams insight, the team has been able to us four state of the algorithms in order to successfully challenge other industry autonomous vehicles. The six

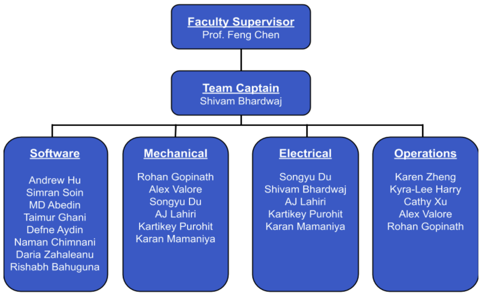


Figure 1.1: Team Structure

members of the mechanical team has built both a mini car for testing and the actual car for competition with weatherproofing technology, a drive-by-wire kit, and a mounting system. The electrical team, made up of five members, applied a power distribution system in the car that increases battery capacity, maximizes run time, and optimizes recharge rate. The electrical team also made sure the electronics suite was safely placed in the car. The mechanical team worked on mounting the sensors, designing 3D models, applying FEA stress, and vibrational analysis on the mechanical and electrical components. The software team consists of eight members. The software team focuses on navigation and localization of the car, path planning, obstacle detection, lane detection, and control. The mechatronics team used ROS to combine the computer vision, and the control code as well as reading data coming from the sensors. The five electrical team members made sure that power is distributed properly to all components of the vehicle system. This included DC motors, motor controllers, controller modules, emergency and safety elements. Furthermore, in collaboration with the mechanical team, circuit boards were designed and implemented for controlling the velocity and the steering angle. The members of the Operations team is made up of five students. Kyra-Lee Harry is the Operations manager where she is responsible for conducting weekly team meetings, approving purchase orders, managing the budget, giving status update reports to the supervisor and sponsors weekly and supporting overall Operations team. Each team member of the Operations team plays a critical role in supporting the technical teams by coordinating with each sub-team, providing updates on the progress of each member, designing and creating our website and work on our marketing strategy and overall image. Together, these four sub-teams and our supervisor make up the NYU SelfDrive Team.

1.3 Design assumptions and design process

Our team had a 5 step strategy in the process of transforming our LSV. This strategy is derived from the Verification and Validation in Scientific Computing.

Due to the modularity of the team organization into sub-teams this year, we focused more on improving the hardware design of the car and on adding more features to the functionality of the car through programming.

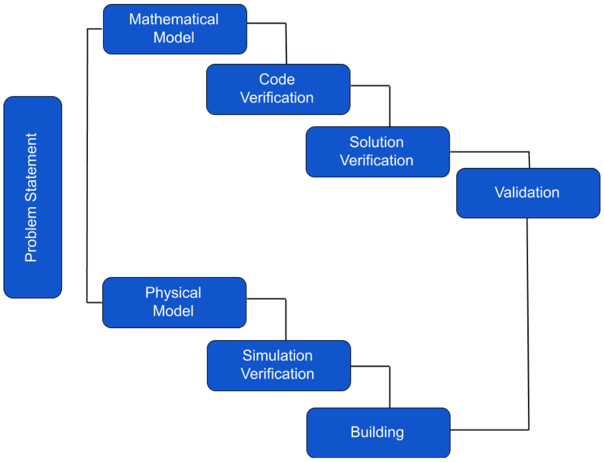


Figure 1.2: Strategy

Innovations in Vehicle Design

2.1 Innovative concept(s) adapted from other vehicles

The existing circuitry installed in the vehicle was redesigned and re-engineered to allow the team to control the car using controllers that use their own algorithms and effectively increase the response time. Individual power supply units, custom microcontrollers, microprocessors and chips have been tailored to the use of the Little Beetle. This was done in the most ergonomic way possible with the use of the existing car platforms while allowing mechanical leads to implement newer and more effective braking and steering technology.

2.2 Innovative technology applied to the vehicle

Two of the most innovative technologies that we used were the braking and the steering mechanism. In order to improve the braking mechanism and control the cars mechanical braking components entirely by the algorithms, an actuator was used to mechanically activate the existing braking mechanism, by allowing the manual use of the brake pedal.

Regarding the steering mechanism, a motor was used in combination with custom gears to control the steering angle values. This clever design has been modified in such a way that the car can still be mechanically controlled by a steering wheel.

Description of the mechanical design

3.1 Overview

A few modifications were made to the mechanical design of the car since last years competition, where NYU Tandon took part for the first time. The braking system was improved to be controlled both autonomously and manually. Part of the braking pedal is in touch with the tip of a linear actuator that was fixed horizontally with the possibility of being pushed manually. The front of the linear actuator is supported by an aluminum frame fixed under the braking pedal and the end is mounted slightly higher on an aluminum board. In that way, the braking pedal could be controlled through the linear actuator electronically without losing its manual function. In other words, the braking system can be used both manually by the driver and by the autonomous system. The steering system was also modified a rotating rod, that is connected to the center of the controlling gear and fixed perpendicularly to top surface of the gear, was the main addition to the system. The rod is suspended by a T-shaped aluminum frame that is fixed at the back of the windshield. When the rod is fixed perpendicularly, it rotates the controlling gear with maximum performance and minimum resistance.

3.2 Drive-by-wire kit

For the accelerator system we decided to disengage the accelerator pedal and the forward-reverse-neutral switches from the motor driver. Instead, we put an accelerator block between them. This block has a circuit of relayes driven by a microcontroller that supplies power to control the acceleration directly from the computer. For the steering system, we used a power steering kit. We attached it with an encoder that has a home positioning signal in order to actually program it as an absolute encoder. Our aim is to convert a DC motor into a precise, yet powerful servo motor. This is accomplished by having a proper feedback loop using a combo of Sabertooth and Kangaroo controllers. Since this motor has a double shaft, we decided to connect it in line with the steering shaft. This aspect gave us the leeway to have both autonomous and manual control of the steering system, while also providing a failsafe mechanism for the system.

3.3 Suspensions and Mounting systems

For the ZED Cameras on the Mini car, a platform was built into the chassis and reinforced with metal joints. This was then reconfigured into an exoskeleton using T-channels attached to the the chassis.

To mount the ZED cameras to our mini car, we cut holes both in the front hood and front trunk floor. This allowed us to connect L-shaped aluminum tubing perpendicularly to a square steel tube attached to the chassis. Next, we used L-brackets to reinforce the L-shaped aluminum tubing so it would twist or rotate. To build our platform for the cameras, a trapezoid shaped acrylic sheet was attached to the L-shaped aluminum tubing using L-brackets. To mount the computer monitor to the mini car, we started by removing the back plastic cover. Once removed, we attached 4 L-brackets to the back cover and the acrylic platform. Additionally, we used the stock monitor mount connected to a piece of L-shaped tubing and placed it in a groove between the windshield and hood.

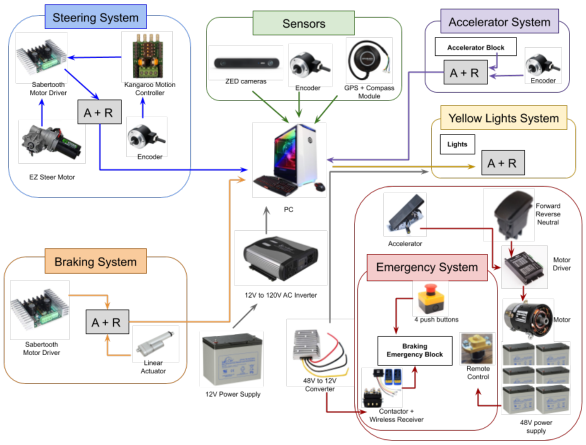
For the Little Beetle, we mounted the PC for easy adjustments to our code. In order to do this we placed the chassis in the trunk, where golf bags would typically go, to help with weight distribution. We then built a U-shaped brace using T-shaped aluminum frame and placed L-brackets at the ends. To mount the brace, we ran m5 bolts from the inside to allow for easy removal if needed.

3.4 Weatherproofing

In order to weatherproof both cars, all seams and wire needed to be protected and resistant to water and moisture. A seam sealer, commonly used in the automotive industry, was used for this end. We filled all cracks where plastic piece of the car met as well as holes where wires were being run. Afterwards, we made sure all wires were insulated and heat shrinked.

Description of electronic and power design

4.1 Overview



4.2 Power distribution system (capacity, max. run time, recharge rate, additional innovative concepts)

The majority of the car is powered by 48V battery consisted of six 8V T-875 deep cycle flooded batteries. The rated power of the car is 3.3 horsepower and 48V is provided such that the supplied current when the car is in full speed 19 mph is 50 Amps. The batteries will last 11.7h (702 mins) in total. The computer is powered with a 12V deep flooded cycle battery with an 12V to 120AC inverter and the battery could last around 6 hours.

4.3 Electronics suite description including CPU and sensors system integration/feedback concepts

A major goal of the electrical group was to design the systems in a way that the car could be easily switched from manual to autonomous mode. Figure indicates the power distribution and all the systems implemented, which include steering, braking, light, accelerator and emergency systems.

4.4 Safety devices and their integration into your system

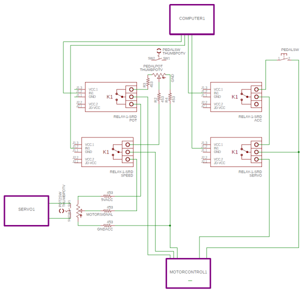


Figure 4.1: The Accelerator System

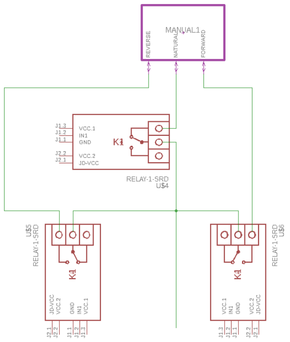


Figure 4.2: Forward/neutral/backward signal block

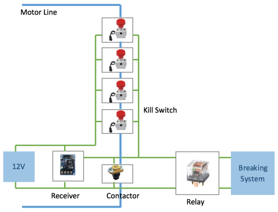


Figure 4.3: Emergency brake signal system

Description of software strategy and mapping techniques

5.1 Overview

The hierarchy of decision-making processes is usually: route planning, behavioral layer, motion planning, and control system. What is usually the input for behavioral layer we have expanded and called it vision perception layer instead. This layer is in charge of counting lanes, localizing the lane position of the car, object detection and recognition, and distance estimation. When the behavioral layer decides on the driving behavior to be performed in the current context, which could be follow lane, change-lane, turn-right, the selected behavior has to be translated into a path that is later tracked by feedback controller. We are using as path planner a combination of A* and a third order polinomio waypoint generator. To set the goal or target point, the farthest point from the center of lanes is chosen. A nonlinear mpc controller then minimizes the tracking error of the desired path.

5.2 Obstacle detection and avoidance

The goal of the project was to create an autonomous car, using as few resources as possible. This is why, instead of using LIDARs in conjunction with other various sensors, we opted to use cameras. We are using a ZED camera, produced by Stereolabs. This camera gives us a 110 field of view, capturing 1080p HD video at 30FPS (The Camera That Senses).

With each frame of the camera, a few different algorithms was implemented to detect obstacles. Originally we used YOLO, a method in which a single neural network can be used to return class predictions of the output unlike other methods that perform the operation in multiple steps. However, we found that a new method, CornerNet-Lite, yielded superior performance and accuracy, and switched halfway through the project (Deng Law, 2018). Since we already had a well-structured dataset, making this switch was not difficult and was a simple task of retraining our models.

However, we still needed to generate a depth map in order to know when and where to stop/turn/etc. Stereolabs provided an API in Python to wrap around the ZED camera software development kit. Using this, we were able to detect obstacles in the path. This depth map allowed us to know how far away obstacles were, in order to stop before hitting them or maneuver around them if necessary.

5.3 Software strategy and path planning

Decisions like deciding the best path around an obstacle or through an intersection are handled by our motion planner. It makes progress by setting short term goals in the interest of maintaining constant progress. For path planning, we found the RRT* algorithm to be a robust method of finding the optimal path towards our next goal.

5.4 Map generation

To generate the map, recognition as well as depth information from all the objects around is required. Crucial for our map was the identification of objects labeled as miscellaneous, which we have achieved. Based on the generated map, the behavioural layer indicates whether to stop, turn, or continue.

5.5 Goal selection and path generation

The algorithm used for goal selection and path generation is advanced lane detection. The purpose of the algorithm is to find the radius of curvature of the vehicle, as well as the offset of the vehicle from the lane using camera calibration, image pre-processing and lane detection. Most of the code used Python and OpenCV, library which is widely used for real-time computer vision algorithms. In the first stage, the ZED camera was calibrated on a set of object points on a chessboard. Calibration was needed in order to ensure the establish an accuracy standard for the output of the algorithm.

The next step involved image-preprocessing using distortion correction to raw images, color transformation and perspective correction. The point of perspective transformation was to visually switch from the drivers perspective to a perpendicular vision of the road lanes, known as birds-eye view. The warped image can be noticed in the image below.

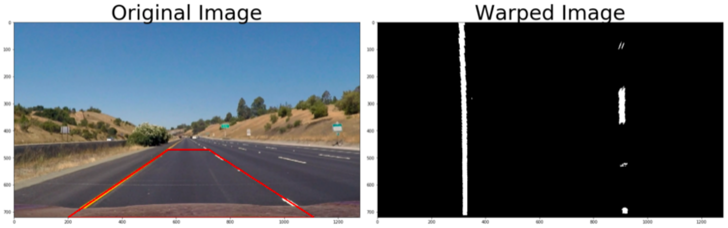


Figure 5.1: *Comparison between original, unprocessed image vs warped image (birds-eye view)*

The most important step, lane detection, was performed firstly, by identifying the lane pixels using the preprocessed images from the previous step and secondly, by running sliding window search in order to fit the pixels in a polynomial function.

The last step involved the computation of the radius of curvature using the best x and y-coordinate fits from the previous step. The output of advanced lane detection, the radius of curvature, was essential in our project to the path planning and steering algorithms.

Figure 5.2: *Lane detection algorithm output*

Description of Failure Modes, Failure Points and Resolutions

6.1 Vehicle failure modes (software, mapping, etc) and resolutions

Due to the low speed limit of the competition, the physical limitations of our car and the rapid pace of development, extensive effort into software failure modes was not deemed of critical importance. However, in the interest of future development and minimizing the risk of any incidents, we proposed a number of basic failure modes in order to ensure that our car operates smoothly.

Having several modes for the car and rules in place for switching between those modes allows us to maintain safe vehicle operation. In the OFF mode, the vehicles brakes are applied and no signals from the motion planning unit are accepted. The vehicle can be switched to the OFF mode from the electronic failure points mentioned in the next section. Next is the MANUAL mode, in which the human is in control of the car. The drive-by-wire system will be disengaged and the human will be in full control of the vehicle. The only way to get out of the OFF mode is to switch to the MANUAL mode, in order to avoid any unexpected autonomous driving.

Then there is the DRIVE mode, in which the vehicle will transition from the MANUAL mode to the DRIVE mode automatically after 10 seconds, 1. If the vehicle is not interrupted, e.g. by E-stops or other cancelling procedures, and 2. If the computer is properly receiving signals from all sensor inputs.

If any object is detected within 5 feet of the car, the brakes are automatically be applied. This is done with the stereo camera depth estimation.

If no signal is received from the motion planner (due to camera failure, connection error, optical obscuration, etc.), the car automatically enters OFF mode in which the brakes are applied and the car will not take any action.

6.2 Vehicle electronic, electric failure points (electronic, electrical, mechanical, structural, etc) and resolutions

Overall, we experienced minor power loss due to short battery life and minor camera malfunctions caused by bugs in the software programs. On the mechanical side, we identified that the slippage of encoders may pose a moderate risk. The main issues with the vehicles failure points are summarized in the table below. In order to identify and resolve most of them, visual inspection and eventual re-boot of the computer is needed.

Table 1. Failure Points Analysis

Failure Point	Type	Risk	Resolution
Loss of main power(48V battery system)	Electrical	LOW	Physical Inspection
Loss of auxiliary power(12V battery system)	Electrical	LOW	Use Kill switch and then Physical Inspection
Camera Malfunction	Software/ Electronics	LOW	Use Kill switch and then Physical Inspection/ code inspection
Slippage of Encoders	Mechanical	MODERATE	Visual Inspection/ Code inspection

6.3 All failure prevention strategy

Unforeseen circumstances and failure points can be difficult to prevent. However, all the unfortunate circumstances that may occur during the competition helped us develop a prevention strategy. Based on previous experience, we identified that the best prevention strategy is modularity, which ensures the independence of most components of the car. The failure of one subsystem does not propagate to another. Moreover, the power batteries for computer and other electrical components are separate so that no signal interference takes place. The hierarchy of decision-making processes usually involves route planning, behavioral layer, motion planning and control system. We have used this standard and modified the route planning by just deciding a preference of direction since the environment is unknown.

Also, all failures that may occur due to calibration will be resolved at this point. Our team members are prepared to deal with such situations by testing and tuning the car manually on site before the competition. Special attention will be given to batteries, which need to be fully charged in order to avoid power failure and to software.

6.4 Testing (mechanical, electronic, simulations, in lab, real world, etc.)

Table 2. Testing

System	**Type**	**Method**
Steering wheel mechanism	Electromechanical	Manual Calibration
Brake mechanism	Electromechanical	Manual Calibration
Lighting System	Electrical	Conductivity Check and code inspection
Computer	Electrical/ Software	Physical Inspection/ Code inspection
Encoders	Electrical	Manual Testing by rotating the wheels/ steering
MPC(Model Predictive Controller)	Software	In simulation by putting random obstacle and sticking to a pre-generated path

6.5 Vehicle safety design concepts

For the safety of the passengers, we have included seatbelts and DOT (Department of Transportation) approved laminated windshield on the car. For crash avoidance: headlamps, reflectors, mirrors and other lights and signals are incorporated. We have left the manual system of the car engaged at all times in case the autonomous features are not performing well. The vehicle's brakes, steering, and suspension systems are robust enough to handle swift turns and rough terrain. Since a lot of modifications are done on the vehicle, the team has made sure that all electrical systems are well insulated.

Simulations Employed

7.1 Simulations in Virtual Environment

The Citiscape dataset was primarily used in order to train the neural network in a realistic setting for image segmentation. However, in order to collect the synthetic dataset, a virtual world was designed in Unity. This synthetic dataset is called Katalinas dataset and includes obstacles characteristic of the setting, such as barrels and potholes. Katalinas dataset has been made open source so that anyone can expand the environment. The assigned color based labels in the second set of images are described in our website along with the code to transform ground truth colors into ground truth label values .

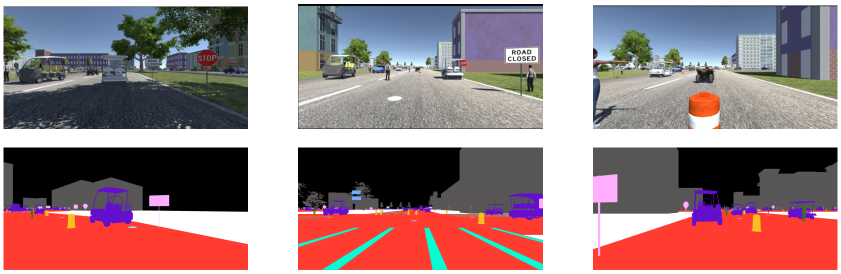


Figure 7.1: Snapshots of Katalinas dataset

7.2 Theoretical concepts in simulations

With respect to the theoretical model we used MATLAB to test our model predictive controller and shifted to Unity to test it on Udacity open source platform.

Testing and Results

8.1 Component testing, system and subsystem testing

The systems of the car can be categorized into the yellow lights system, emergency system, acceleration system, braking, steering, sensor and display system. The individual subsystems of the car including their components were tested separately on a software, electrical and mechanical level. Moreover, as full sized golf cart was not feasible to test drive in the city, a mini version of the car was designed. This mini car allowed for further testing of the performance of each of the subsystems present in the car.

Initial Performance Assessments

9.1 Up-to-date self-driving car state

The subsystems performed as expected. A GUI was used to run all the actuators of the car using keyboard input. The vehicle has not yet been tested in autonomous mode with all the subsystems running simultaneously.

Bibliography

[1] *The Camera That Senses Space and Motion.* Stereolabs, `www.stereolabs.com/zed/`.

[2] *CornerNet-Lite: Efficient Keypoint Based Object Detection* Hei Law, Yun Teng, Olga Russakovsky, Jia Deng
`arXiv:1904.08900`