



# THE UNIVERSITY OF WESTERN AUSTRALIA

School of Electrical, Electronic and Computer Engineering

FINAL YEAR RESEARCH PROJECT THESIS  
**Autonomous Driving with Dynamic Path Planning**

Written by:

**Wai Lam William Lai (21696421)**

Academic Supervisor:

**Professor Thomas Bräunl**

Submitted: 22/10/2018

Word Count: 6153



## Acknowledgments

I would like to thank my project supervisor Professor Thomas Bräunl, for his vision, leadership and ongoing advisory throughout this practical research project, especially for the provision of the opportunity to tackle the challenging and interesting project.

I would like to express my appreciation to the REV SAE team core members, as well as the supporting PhD students Mr. Thomas Drage and Mr. Ki Li Lim for their professionalism, collaboration on the teamwork, assistance, and knowledge sharing during the research project to achieve the challenging goals. Special thanks to the team leader Chao Zhang who had contributed a lot of time in conjunction with his technical skills/knowledge to aid the progression of this project.

Finally, I would like to thank my wife Monica and my parent for their ongoing support and encouragement throughout the course of the year.

## Abstract

Path planning is one of the crucial elements for autonomous driving. A Hermite spline interpolation algorithm has been used to ensure smooth manoeuvre in between series of waypoints is discussed in this paper. Followed by proposing a customised algorithm based on graphic-search path planning to achieve the real-time dynamic path planning that avoids static obstacles. Coordinated with sensing devices including odometry from wheel encoders and a single layer LiDAR scanner to detect static obstacles in front. Cone following can be achieved without the predefined mapping situation. Cones within detectable distance will be evaluated against the limited steering range, the optimal path which is collision-free on the track is constituted from choosing the largest driving range free of obstacles amongst some possible path candidates. The simulation and experimental results of the algorithms demonstrated the potential of practical application for fully autonomous driving.

Although Simultaneous Localisation And Mapping (SLAM) technique can aid to generate global map for path planning purpose. However, it is not within the design project scope. Skid effects would be an option to consider for maximise the driving speed of the vehicle. However, this mechanical dynamic subject is not within the scope of this paper and hence not being considered.

## List of Acronyms

AI	Artificial Intelligence
DTEC	Driving Training and Education Centre
EV	Electric Vehicle
FSD	Formula Student Driverless
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
LTS	Long Term Support
MATLAB	Matrix Laboratory
PhD	Doctor of Philosophy
PID	Proportional, Integral, Derivative
REV	Renewable Energy Vehicles
RAC	Royal Automobile Club
ROS	Robot Operating System
SAE	Society of Automotive Engineers
SLAM	Simultaneous Localisation And Mapping
UWA	University of Western Australia
YAML	YAML Ain't Markup Language

## Table of Contents

Acknowledgments.....	II
Abstract.....	III
List of Acronyms.....	IV
1. Introduction .....	1
1.1 Motivation .....	1
1.2 Background .....	1
1.3 Problem Statement .....	2
1.4 Project Goals .....	3
2.0 Literature Review.....	4
2.1 Localisation .....	4
2.2 Driving control .....	4
2.3 Path planning.....	5
3.0 Design Process .....	6
3.1 Physical limitations .....	6
3.2 Design requirements.....	7
3.3 Design tools.....	9
4.0 Final Design .....	11
4.1 Waypoints driving .....	12
4.2 Cone Driving .....	14
5.0 Results and Discussion .....	18
5.1 Results of waypoint driving .....	18
5.2 Evaluations of waypoint driving .....	19
5.3 Results of cone driving from simulation .....	20
5.4 Results of cone driving from the field.....	20
5.5 Evaluation of cone driving from the field .....	21
6.0 Topics for Further Investigation .....	22
6.1 Road edge detection with SLAM .....	22
6.2 Skid effect .....	22
6.3 Involvement with AI model .....	22
6.4 Reverse motion.....	22
7.0 Conclusion .....	23
Bibliography .....	24
Appendix A.....	26

# 1. Introduction

This chapter entails the motivation for this work, the background of the project, states the problems at hand, and describes the predefined project goals.

## 1.1 Motivation

From the statistical report of [1], there were about 1,200 fatalities due to road crashes in Australia during the annual record from August 2017 to 2018. From the global perspective, there were around 1.25 million fatal road crashes [2]. Study from [3] indicates over 90% of the vehicle accidents on road are constituted by human driver error, with reasons including fatigue, drunk driving, speeding and inadvertent drivers on traffic conditions.

Autonomous driving can potentially reduce the road accidents that are caused by human driver intervention as well as increasing safety from aggressive driving behaviour, mitigate the risk of fatigue/improve the comfort level from driving commute route, and provides an extra transportation option for people who unable or not suitable to drive on the road. Thus, fuel efficiency and transit time saving can be achieved via optimised driving pattern.

Renowned commercial enterprises including General Motors, BMW, Toyota and Tesla have envisioned the opportunities for future automobile businesses, hence their recent involvement with invested funds and resources into research and development activities on autonomous driving industry [4].

Therefore, it is believed the content within this paper would contribute to one of the most important aspect for the contemporary Engineering research topics.

## 1.2 Background

To promote sustainable energy for the future, the Renewable Energy Vehicle (REV) project which is led by Professor Thomas Bräunl has initially converted the Formula-SAE car into a pure Electric Vehicle (EV) since 2010. Over the years of development by the final year Master students, PhD students and student managers, the REV project has subsequently progressed the autonomous driving capability by enabling the vehicle with drive-by-wire technology to electronically controls of steering, brakes, and accelerator. In addition, the SAE vehicle is equipped with array of sensory devices including Inertial Measurement Unit (IMU), Light Detection and Ranging (LiDAR), digital camera, hall sensors on all wheels and GPS to sense the environment. The SAE vehicle by then is capable to follow the waypoints that have been previously captured and stored inside the pre-recorded map [5].

At August 2017, Germany located on the opposite side of the Earth compared to Australia has held a Formula Student Driverless (FSD) competition to encourage Engineering students get hands on experience at autonomous vehicles [6]. This act is another factor that stimulated the REV project to undergo further autonomous driving research and development efforts.

At year 2018, the REV SAE team have divided the autonomous driving project into 6 sub-projects; Video processing, LiDAR processing, Sensing and reliability, Localisation and mapping, High level software and Driving control and path planning. The author will be focused mainly on Driving control and path planning by utilising SAE vehicle as a nonholonomic car-like robot.

### 1.3 Problem Statement

This paper will address two scenarios of autonomous driving. The first case would be waypoints driving with consideration of safety, smoothness and accuracy. Although it has been examined in 2013 as mentioned in [5], however, the SAE vehicle has undergone various modifications on hardware as well as software. Especially the project this year have introduced and utilised a new software platform called Robot Operating System (ROS) to coordinate the interfacing between arrays of sensors and software modules (nodes). It is necessary to reimplement the waypoints driving to ensure the SAE vehicle can manoeuvre from the starting position to the goal position successfully via a path that is joined by a set of waypoints.

The second case would be cones following without a predefined map beforehand, using the FSD 2018 event TrackDrive setting (Fig.1) as the guideline [7]. It is expected to enable the continuity of SAE vehicle driving autonomously through the designed cones track in a closed loop environment.

- ■ Yellow/Blue Cone
- ▲ ▲ Small/Big Orange Cone
- ◁ Red TK Marking & TK Equipment  
(Shape undefined)

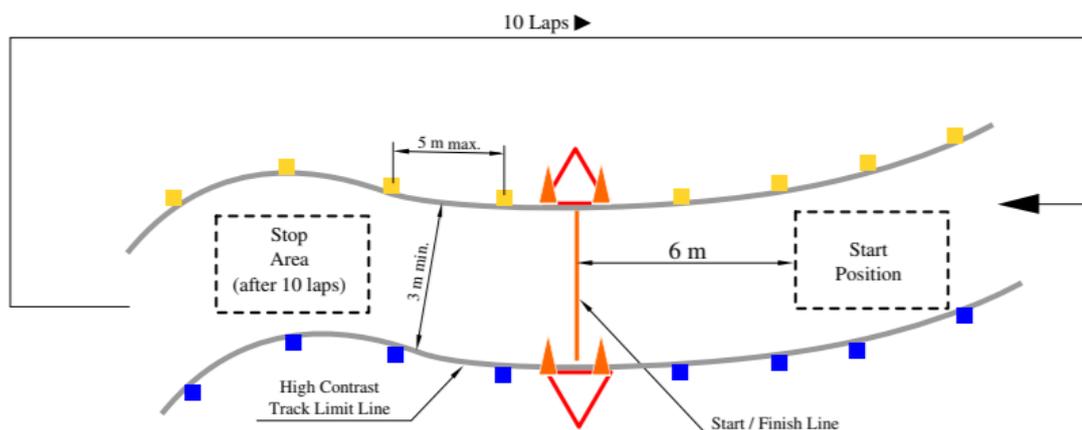


Figure 1: TrackDrive requirement of FSD2018 [7]

This would involve cones placement along the entire track constituted of straight and curvature sections. Cones placement are 5 meters maximum distance apart from the adjacent cones on either side of the track and minimum of 3 meters for the track width, with denser population at any corners. To replicate the similar track settings, the project team has constructed the testing path with the same sized orange traffic cones (Fig. 2a & Fig. 2b).



*Figure 2a: Straight section of the cone track*



*Figure 2b: Corner section of the cone track*

## 1.4 Project Goals

The primary goal for this project is to research, design, test, and implement local path planning algorithms that utilise the equipped sensing hardware, thus enable the SAE vehicle to drive autonomously in the scenarios of the abovementioned section 1.3 safely and reliably. The secondary goal for this work is to provide a starting point for the future students who wish to participate into the REV project and expand the features that has been developed along the project. And lastly, to provide positive impact to the Engineering research area and contribute to the acceleration of autonomous driving industry in the near future that also benefits the community as stated in the motivation at section 1.1.

## 2.0 Literature Review

### 2.1 Localisation

For an autonomous vehicle to drive within an unknown environment, it is critical to know its position and orientation relative to the start position or the global coordinates for moving around the environment without losing the direction and ultimately the control [8].

One applicable method called “dead reckoning” can achieve short-term localisation without the needs of global sensors. By knowing the starting position and orientation, and the information extracted from the wheel encoders while driving around, the current position of the robot can be updated. However, there is a drawback from the accumulated accuracy issue for long-term driving from the sensors or wheel slippage. An on-board compass can be applied as a heading sensor to overcome this issue and provide absolute orientation [8].

Matching between the local robot coordinates and global world coordinates can be done via an x-y plane transformation with a translation and a rotation [8]. The global coordinates can be defined by a general formula obtained from [8] is presented below:

$$[o_x, o_y] = Trans(r_x, r_y) \cdot Rot(\varphi) \cdot [o_{x'}, o_{y'}] \quad (1)$$

Where  $o_x$  and  $o_y$  = global coordinates,

Trans = translational movement,

$r_x$  and  $r_y$  = global position,

$\cdot$  = dot product computation,

Rot = rotational movement,

$\varphi$  = global orientation, and

$o_{x'}$  and  $o_{y'}$  = local coordinates

### 2.2 Driving control

The SAE vehicle control system comprised of two parts: 1. A “low-level system” to manage the function of motor controllers, brake servomotor, and steering controller. 2. A “high-level system” that handles the sensor information computation, human driver interfacing, PID control parameters calculation, and send commands to the low-level system [5].

Drivable map is represented by the waypoints in Cartesian coordinates that were converted from a latitude/longitude pair inputs such as Google Maps. The formulas below show the relationship [5]:

$$x = R_E \frac{2\pi}{360} (\varphi - \varphi_D) \quad (2)$$

$$y = R_E \frac{2\pi}{360} \cos|\varphi| (\lambda - \lambda_D) \quad (3)$$

Where  $R_E$  denotes as the Earth's radius: 6378 km [9];  $\varphi$  and  $\varphi_D$  denotes as the latitude observed and latitude from Datum;  $\lambda$  and  $\lambda_D$  denotes as the longitude observed and the longitude from Datum respectively.

By implementation of the drivable map, a safety boundary can also be set to control the vehicle allowable driving area and stop promptly at the event of losing control during autonomous testing [5].

Applying waypoint with practically large in size is recommended as to increase the probability for the vehicle to reach the point successfully. To attain a desired finish heading and optimum trajectory, thus cater for the obstacle avoidance and smooth driving experience over the sharp edge turning. A decent steering algorithm to dynamically compute the short distance path ahead is required, with cubic splines calculation through four waypoints including the current location shall be integrated into the design [5].

### 2.3 Path planning

To avoid the SAE vehicle driven over any invalid path, a general path planning algorithm is consisted of four parts. They are namely as base frame generation, potential maneuvers projection, costs function computation and selection of the optimum maneuvers [10].

A set of waypoints are sent to the SAE vehicle and the algorithm will figure out an ideal path as the base frame based on a parametric cubic spline calculation through the waypoints. Once vehicle start moving, a set of path candidates would be determined and projected based on the current speed, turning angle, orientation and differences to the goal. Each potential maneuver would be calculated with a cost function that considered different cost factors including deviation from the base frame, smoothness for the continual driving, and path clearance to obstacles. The weighted cost would be summated, and the preferred path is chosen based on the lowest cost, and command will be sent to the drive control algorithm to follow the route accordingly [10].

Extracted from [10], the Fig. 3 beneath illustrate the simulated process of choosing the optimum path in green, potential maneuvers in blue, black as the base frame and obstacles in red on the x-y plane.

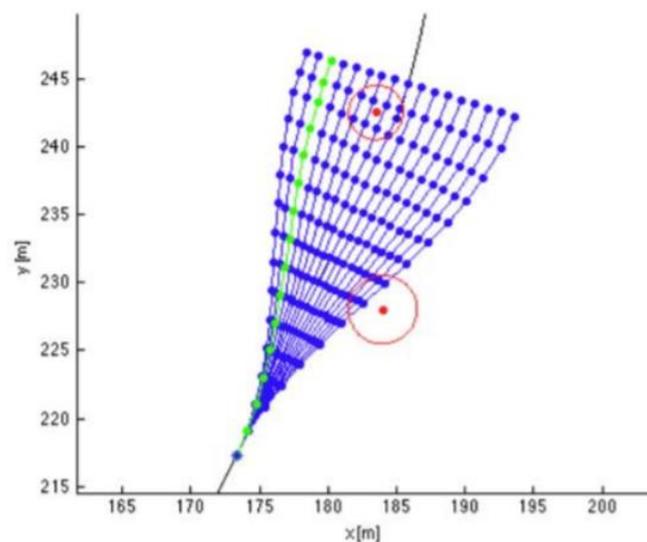


Figure 3: Simulated computation of path selection [10]

### 3.0 Design Process

Before the actual implementation of path planning, constraints and other design requirements shall be considered. Design constraints can be derived from the physical limitations of the vehicle including the possible driving direction, maximum steering angle, and the size of the vehicle. This will aid to design a viable driving path with avoidance of collision to obstacles and prevent the potential risk of a trapped situation when driving through a confined track environment.

#### 3.1 Physical limitations

The SAE vehicle is a nonholonomic system that the velocities and other derivatives of the vehicle position are constrained. This prohibits the driving in arbitrary direction such as transverse motion.

Measurements have been made to find out the physical dimensions of the SAE vehicle.

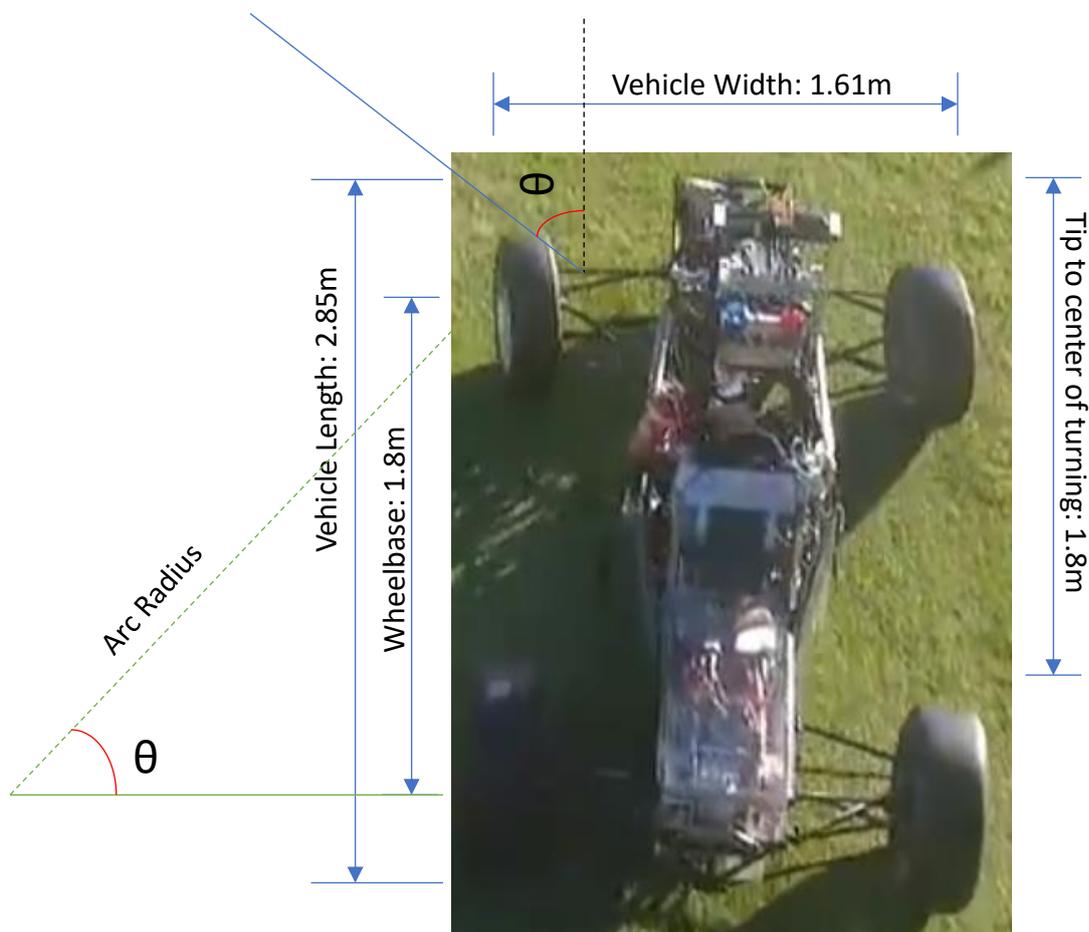


Figure 4: Dimensions of the SAE vehicle

Depicted from the Fig. 4,  $\theta$  is denoted as the maximum steering angle that the SAE vehicle can physically turn. It is measured at the ground level to be approximately to  $\pm 25^\circ$ , which dictates one of the design constraints with maximum size of steering interval to be  $50^\circ$ .

It is assumed that the SAE vehicle is not 100% Ackermann steering with the existing mechanical design, built and modified from the previous year's students. Therefore, the minimum arc radius can be determined from the simplified model shown in Fig. 4 based on the maximum steering angle and wheelbase of the vehicle:

$$Arc\ Radius_{min} = \frac{Wheelbase}{\sin(|Steering\ Angle_{max}|)} \quad (4)$$

From the measured wheelbase distance and  $Steering\ Angle_{max}$ ,

the resultant  $Arc\ Radius_{min} = \frac{1.8m}{\sin(25)} \approx 4.26m$ . With this restriction in place and adding 10% safety factor, the track design should consider the sharpest arc radius (i.e. perform a 180° turn) of any turning point to be  $4.26*1.1 \approx 4.7m$ .

The SAE vehicle is not capable to reverse during the autonomous driving mode. Therefore, this constraint will contribute to the additional testing time on the field in the event of failure to the path planning and/or the drive control, and the vehicle might be driven to an undesired position includes reaching a dead end. Thus, manually repositioning of the vehicle is required before further testings can be continued.

### 3.2 Design requirements

There were 5 design requirements identified for this design, they are safety, controllability, comfortability, obstacle avoidance, and adaptability.

#### **Requirement 1 – Safety of manoeuvre of SAE vehicle during autonomous mode**

The SAE vehicle is a full-sized car compared to most of the lab robot, it is more than 250 kg in weight and capable of driving up to 80 km/h. Due to the safety reason, the maximum output from the throttle has been limited to 40 km/hr at the low-level design. The testing of path planning and drive control algorithms shall be conducted on the simulator before the actual field testing, this can lower the chance of malfunction program execution in autonomous driving mode.

Before commencing on road test drive of the SAE vehicle, the risk assessment forms shall be completed and signed by the safety authority, safety checklist shall be evaluated to ensure all the safety features including onboard and the remote emergency stop buttons are function as intended at the beginning of the autonomous mode driving.

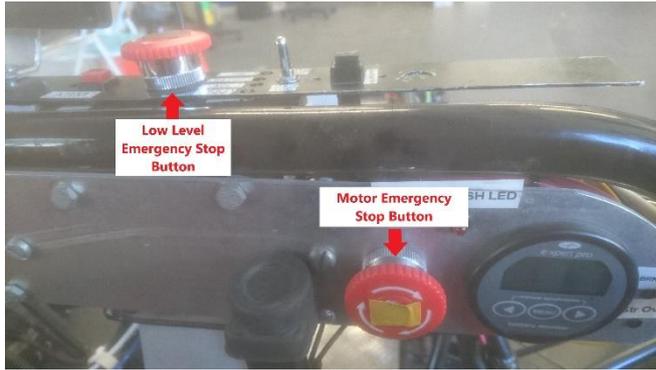


Figure 5a: Onboard Low-Level Emergency Stop Button (left) for steering, throttle and brake control, and Motor Emergency Stop Button (right) for Motor control



Figure 5b: Remote Emergency Stop Button for brake and motor control

Hardware or software failure might occur during the field testing process. To reduce the potential risk of crashing accidents that might cause injury to the general public and damage building structures, the chosen testing venue shall be isolated to the public access if possible. If the test field is publicly accessible, safety boundary shall be developed with highly visible cones to signify others that the test area as a dangerous terrain and notify them if they attempt to cross the boundary.

## Requirement 2 – Controllability of speed and steering direction

The configuration space is defined as  $(x, y, \theta, s, \phi)$  on a 2-dimensional plane, where  $X$  and  $Y$  are denoted as the Cartesian coordinates on the plane,  $\theta$  is the current orientation of the SAE vehicle,  $s$  is the speed and  $\phi$  is the angle difference between the current position  $(X_0, Y_0)$  and the next target point  $(X_1, Y_1)$ . To increase the controllability of motion of the vehicle, the entire path can be broken down into combinations of straight lines and curvature sections.

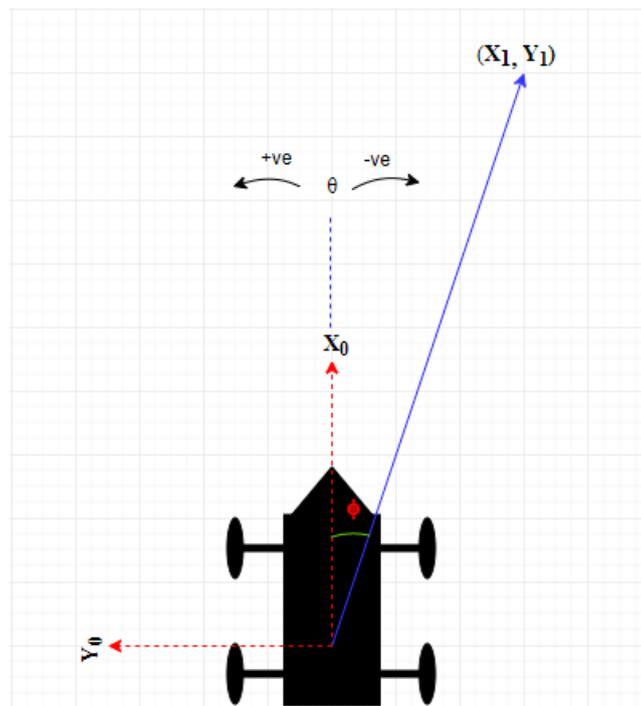


Figure 6: Configuration space with dash line (red) extends of vehicle's coordinates

### **Requirement 3 – Comfortability during motion**

Although the objective of this project is to achieve autonomous driving, a driver is still required to control the SAE vehicle in the during test drive to minimise safety implication to the surrounding. Therefore, the driving pattern should avoid potential oscillation during the movement from current to next position, this can be achieved with ensuring the correct orientation heading to the next waypoint.

### **Requirement 4 – Obstacles avoidance**

This is a crucial requirement which also in line with requirement 1, as for human driving behaviour at normal situation, the vehicle should be controlled to avoid collision with any objects nearby while travelling from one location to another.

### **Requirement 5 – Adaptability**

Adaptive path adjustment for the cones following case, the design shall be able to implement on any track settings with minimal updates of configuration parameters, especially in the case where predefined mapping is unavailable.

## **3.3 Design tools**

The designed functions for path planning and motion control to achieve autonomous driving of the SAE vehicle are programmed with C++ language. Those functions are integrated as control modules into an open source robot framework called Robot Operating System (ROS) [11], which is the main design tool for this project.

### ROS

The current ROS version applied to this project is Kinetic Kame and running on Ubuntu 16.04 LTS platform, which is a Linux distribution based on Debian and free of charge operating system [12].

ROS provides services including hardware abstraction, low-level device control, implementation of commonly-used tools, message-passing between nodes, and package management [11]. To design a particular function as a node, via the subscribing (listen) and publishing (talk) of message type on the input and output topics, the message-passing scheme enables intercommunications between nodes [13]. System integration is therefore simplified since different nodes created by individual project team members can apply the information from the same topic to perform various tasks. It also supports data capture, logging of runtime errors and data replay.

ROS also offers a visualisation toolkit called RViz [14], simulation of the designed nodes can be visualised on the screen by selecting the corresponding topics and their associated message types. The performance of the simulation can be checked against with the information prompted on the traditional command prompt window.

### Draw.io

It is a free online diagram software that has been applied to create, store, and modification of flowchart and other customised diagrams to support the design process [15].

### Peek

It is a freeware that is used to create the screencasts by recording the simulations on RViz in MP4 format video [16], then playback the video with the specific framerate to aid the evaluation of the results.

### MATLAB

It is an auxiliary tool being applied in this project [17]. It can be used to perform matrix manipulation to solve simultaneous equations and visualise the outcome on the screen by using their built-in plotting function.

### Test fields with cones

During the entire project, there were three test fields (UWA sports sciences oval, grass patch area between UWA's James oval, and RAC DTEC close to Perth international airport) being utilised to build cones track with different arrangements, they served as the important platforms for verifying the designed planners.

## 4.0 Final Design

Based on the design process and two scenarios of autonomous driving described in previous sections, there were two path planners being designed and implemented in this project. The first one is waypoint driving, which is a global path planner that handles longer distance by generation of a complete path that composed of a set of waypoints to be followed. The second one is cone driving, which is a local path planner that handles shorter distance by observing portion of the path, with more precision in drive control even in continuous form and taking into account of any unexpected obstacles that may interfere the path with obstacle avoidance.

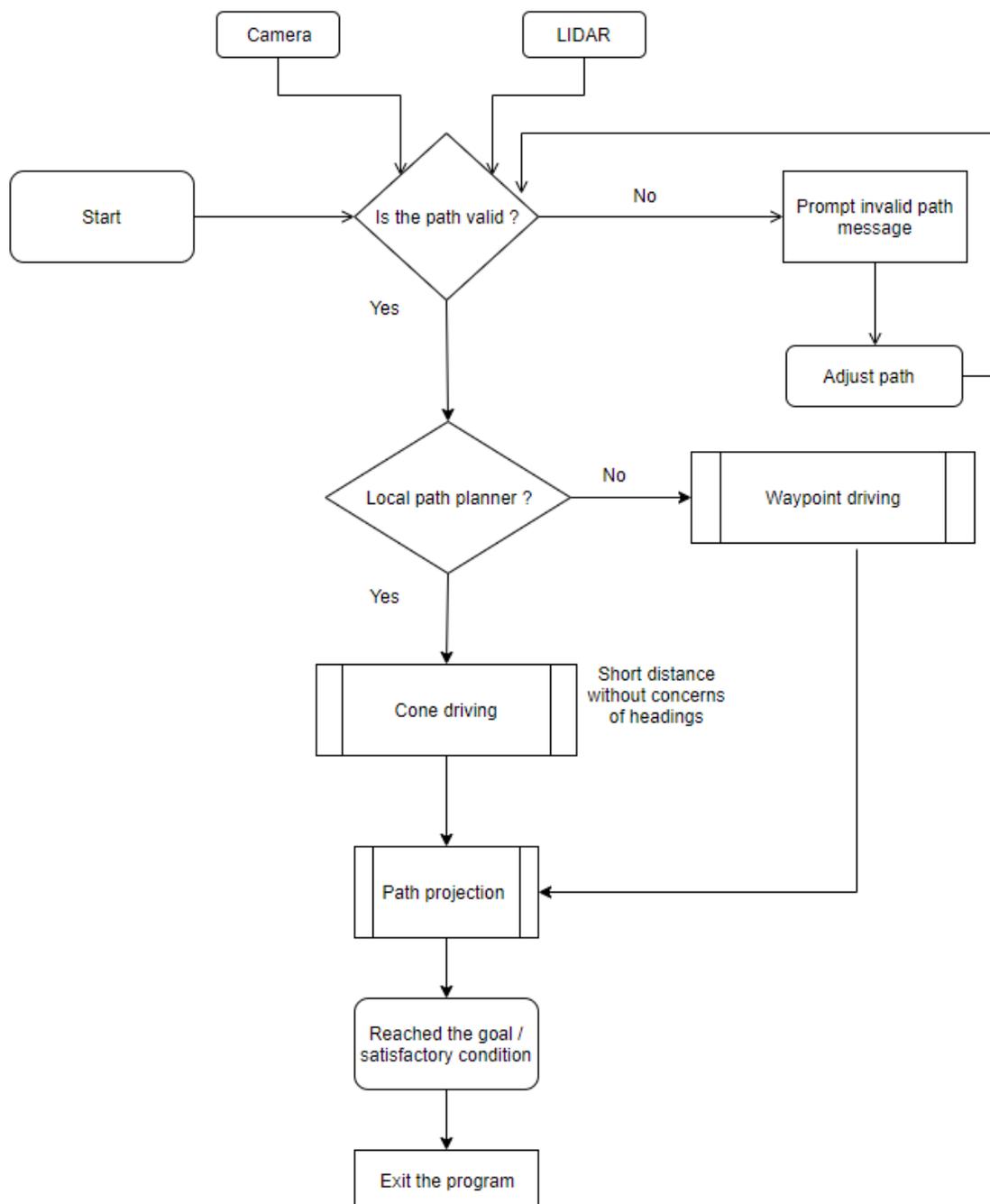


Figure 7: Logic of the final design

## 4.1 Waypoints driving

The underlying idea behind waypoint driving is to drive a set of predefined points in between the starting position  $P_1$  with initial setting ( $x = 0, y = 0, \text{orientation } \theta = 0$ ) to the destination position  $P_2$ , which can be obtained through the differences in the GPS coordinates. These waypoints can either be stored in Cartesian coordinates in an array or in this design case, they are selected based on the driver's preference by selecting position points on RViz, which are then confirmed on the console. The SAE vehicle will start driving once the path is established from the waypoints.

To fulfil the requirement 1 listed in section 3.2, the design of waypoint driving has incorporated a stopping logic in the event of lost track from the current position to the target position (exceed the threshold limit) with respect to their absolute distance and angle difference between them.

For requirement 2, the resultant steering angle  $\phi$  is computed based on the difference between  $\phi$  (angle difference of current position to target point obtained with atan2 function) and  $\theta$  (the vehicle's current orientation extracted from wheel odometry). Steering direction would be left, right or stay straight in current direction if  $\phi$  is positive, negative and zero respectively. Based on the absolute distance between the current position and the next target point, the speed control can be adjusted accordingly to enable faster driving on the straight sections while slowing down at the sharper curvatures.

For requirement 3, all waypoints shall be driven continuously and smoothly with the consideration of the correct heading to the subsequent point, a spline approach has been implemented within this design to generate the desired path.

The Hermite spline interpolation technique [18] has been implemented on this design to generate the desired path. It consists of four vectors:

1. Current position ( $P_1$ ),
2. Target position ( $P_2$ ),
3. Tangent of departure from current position ( $T_1$ ),
4. Tangent of approach to target position ( $T_2$ ).

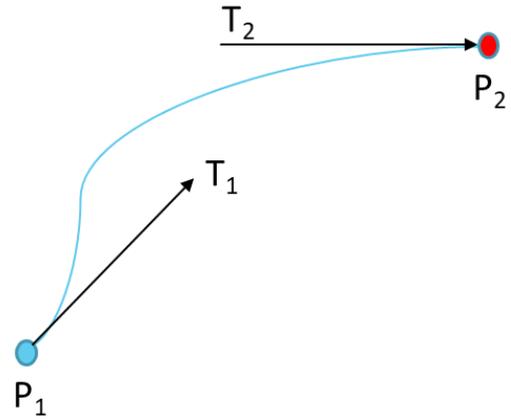


Figure 8: Illustration of four vectors

and four Hermite basis functions  $H_n(u)$ :

$$H_1(u) = 2u^3 - 3u^2 + 1 \quad (5a)$$

$$H_2(u) = -2u^3 + 3u^2 \quad (5b)$$

$$H_3(u) = u^3 - 2u^2 + u \quad (5c)$$

$$H_4(u) = u^3 - u^2 \quad (5d)$$

Where  $0 \leq u \leq 1$  is the interpolation parameter derived from the intervals of the absolute distance between the current position and the target position.

The construction of the resultant path  $f$  is then evaluated from the product between the vectors and the Hermite basis functions.

$$\text{Resultant Path } f(x, y, \varphi) = H_1P_1 + H_2P_2 + H_3T_1 + H_4T_2 \quad (6)$$

To ensure the continuity and smoothness of the resultant path, the Hermite spline in this design has considered the previous waypoint position  $(x_{i-1}, y_{i-1})$ , current waypoint position  $(x_i, y_i)$ , and next waypoint position  $(x_{i+1}, y_{i+1})$ . The pointer of the waypoint will scan through the rest of the path and update its position.

Once the waypoints are chosen, two static paths are shown on RViz (see Fig. 9). The first path (green) consist of straight lines that interlink all the points with the arrow at the end showing the destination heading. The second path (purple) is the desired path which consists of a smoothed curvature that passes through all the waypoints.

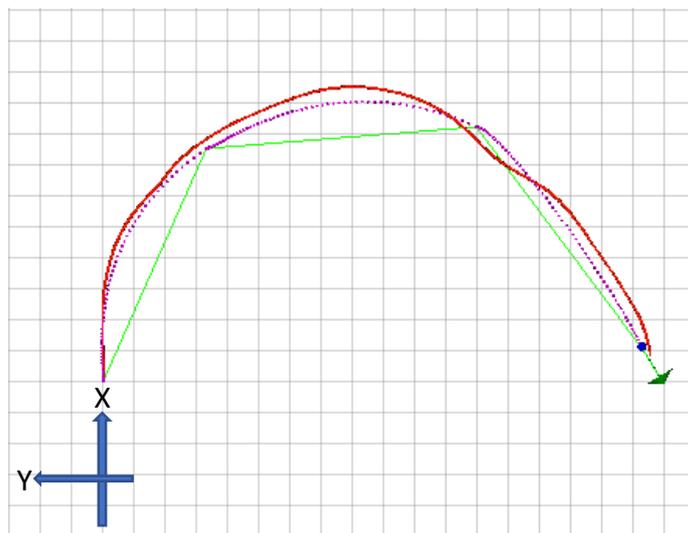


Figure 9: Calculated waypoints as output on RViz with inputs from equations (5) and (6)

The actual simulated driving pattern (red) is a dynamic path, which is determined based on the distance measurement between the current position of the vehicle against the desired path with a predefined tolerance range, and physically limited  $Steering\ Angle_{max}$  ranged between  $\pm 25^\circ$ . The desired path is constituted by a finite number of points generated from the Hermite spline interpolation. Pythagoras theorem is applied to find out the relative distance from the vehicle's current position to the next desired path point, and simple geometry to find out the angle differences from the current vehicle's pose to the next desired path's pose. To avoid oversteering or understeering, parallelism is computed from the slope differences for both the driving path and the desired path. With this logic in place, the vehicle is either turning left ( $\varphi > 0^\circ$ ), right ( $\varphi < 0^\circ$ ) or driving straight  $\varphi$  ( $\varphi = 0^\circ$ ) either at a constant speed or slowed speed for a sharper turn in order to reach the goal point.

## 4.2 Cone Driving

The cone driving planner applies object detection by LiDAR (with maximum scanning range up to 20 m) [19], to search for cones that is within the predefined coverage area and determines their size and their relative location to the SAE vehicle. It also accepts cone locations from either the camera vision after image processing or on the pre-recorded map. This planner enables real time processing for the vehicle to continuously compute a valid path from the small section of the track (constructed with cones), that is similar to the graph-search method [20]. Then search and follows the optimal path safely without collision with the cones until no more valid track is detected.

The overview of the cone driving procedure is detailed in Algorithm 1 [13].

---

**Algorithm 1** Cone driving

---

**procedure** CONEDRIVE(cones in range)

    init steering range to [-25,25]

**for** all cones in range **do**

        evaluate collision range with cone

        exclude the collision range from steering range

**end for**

**if** steering range is empty **then**

        stop

**else if** all steering range  $\leq$  threshold **then**

        select largest steering range

**else if** all steering range  $>$  threshold **then**

        select steering angle with minimum change in current direction

**end if**

    drive toward centre of steering range

**end procedure**

---

It first defines a detection range, which is typically set to about 4 meters scanning radius in the form of a semi-circle in front of the SAE vehicle (as the LiDAR is mounted at the front edge of the car's structure). Next, initialise the steering interval from  $-25^\circ$  to  $25^\circ$ . Iterate through all the cones detected within the detection range, and evaluate against the collision range, which is based on the clearance radius of the vehicle, size (horizontal spans) of the cones and the wheelbase distance that dictates the curvature center of the vehicle and the suitable steering angle. After that a set of steering intervals is updated with exclusion from the collision range. Evaluate the steering interval set and if is emptied, stop the vehicle as there is no feasible path to drive. If all steering intervals are smaller or equal to size of the

threshold interval, then choose the largest one as the desire interval. Otherwise if there are some steering intervals large enough (exceed some thresholds), then selects the desire interval that makes least amount of change to the current steering. After the evaluation, applies the mean value of the desire steering interval as the optimal angle, and starts/continue driving based on that angle.

To fulfil the requirement 1 listed in section 3.2, the design of cone driving has incorporated a stopping logic in the event of detected obstacle interferes the path, for instance, human walking through the track. The default processing range is set to 4 m, it is the scanning radius measured from the LiDAR to detect any objects in front of the vehicle. The planner is designed to update the processing range proportionally based on the current driving speed, this is to accommodate for a longer stopping distance at faster speed. In the event of any obstacle is detected within the path, the vehicle is set to stop from brake engagement and stay an idle period of 10 seconds, recheck the state of the path then reinstate autonomous driving once the path is cleared.

For requirement 2, the resultant steering angle  $\phi$  is the mean of the desire steering interval. This interval is computed based on the evaluation of the steering boundary that avoids the collision with cones. The initial steering range is computed based on the arc radius for the vehicle to turn with consideration of the room for obstacle avoidance.

$$Arc\ Radius = \frac{x_1}{\sin(2 * \text{atan2}(x_1, y_1))} - Shift \quad (7)$$

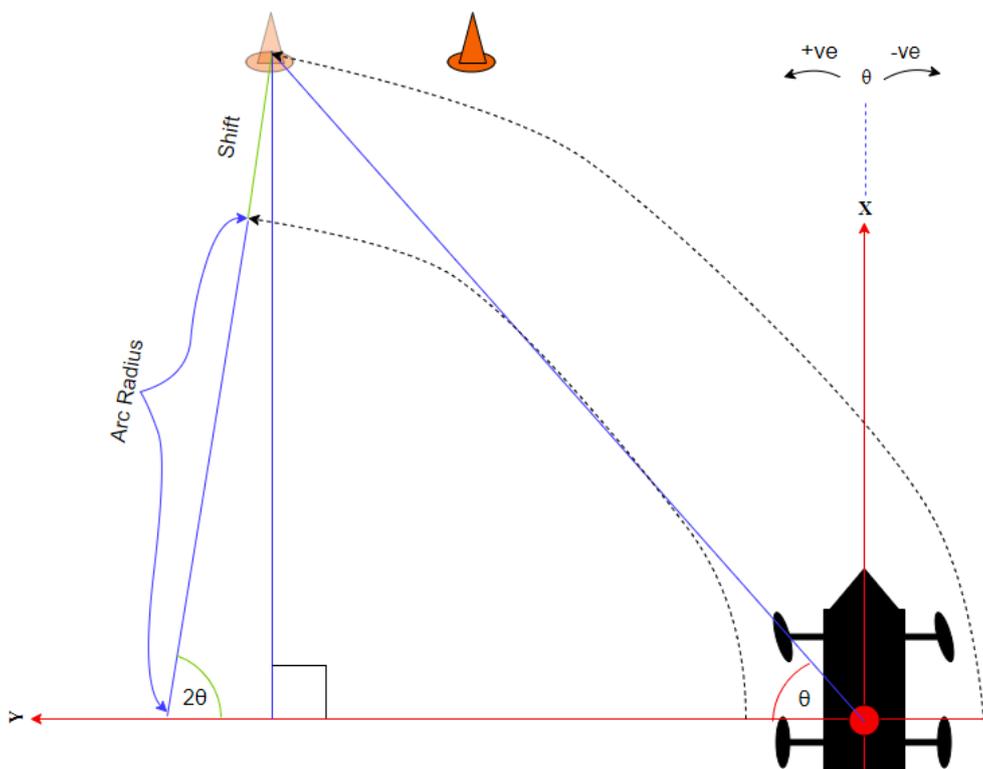


Figure 10: Applied the shifted center of the (semi-transparent) cone to obtain the Arc Radius

Where Shift denotes the shifting factor that consists of clearance radius measured from the base link (turning center) of the vehicle and half size of the detected cone;  $x_1$  and  $y_1$  denotes the forward and the lateral distance, respectively between the shifted center of the detected cone to the base link of the vehicle.

Thereafter the possible steering angle  $\theta'$  that forms part of the initial steering range can be determined by:

$$\theta' = \text{asin}\left(\frac{\text{Wheelbase}}{\text{Arc Radius}}\right) \quad (8)$$

Then the cone driving planner will iterate through all the detected cones and evaluates the possible steering ranges in order to obtain the valid ones. This is achieved by excluding the driving ranges that have exceeded the maximum steering angle the vehicle can physically turn ( $\pm 25^\circ$ ) as well as the one that will collide with the cones. The valid steering ranges are further processed by comparing their size of interval to search for the largest one and set it as the desired steering interval. Otherwise, if multiple of them are deemed to be large but in the same size, select the one that requires the least amount of steering with respect to the vehicle current orientation and set it as the desired steering interval.

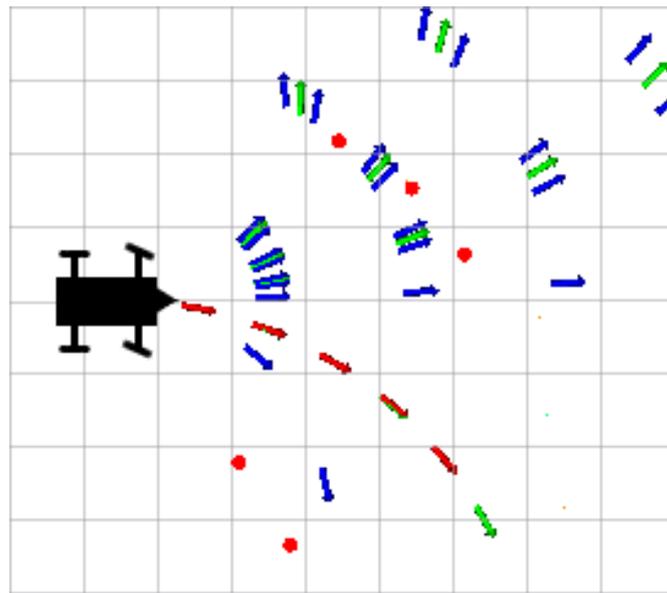


Figure 11: Curvature driving on the desired path (red) shown on RViz

Depicted from the Fig. 11, groups of steering intervals (blue), with their mean value (green), and the desired path (red) is projected based on the resultant steering angle  $\varphi$ , which is evaluated from the mean of the desired steering interval (the largest steering interval amongst the group). The vehicle in this case will follow the desired path by turning right.

Speed control has been applied based on the resultant steering angle  $\varphi$ . The desired speed typically sets to 6 m/s. When  $\varphi$  is in between the interval of  $[-5^\circ, 5^\circ]$ , 100% of the desired speed is being applied to drive the vehicle. Then slows down slightly with 80% of the desired speed if  $\varphi$  is in between  $[-20^\circ, -5^\circ]$  or  $[5^\circ, 20^\circ]$ , and set 60% of the desired speed when  $\varphi$  is in between  $[-25^\circ, -20^\circ]$  or  $[20^\circ, 25^\circ]$ .

For requirement 4, the vehicle will only proceed to drive when a valid path is available. The obstacle avoidance mechanism will set a clearance distance with the configurable clearance radius measured from the base link of the vehicle and half of the width of the detected obstacle (the cone in this design). Then applied this clearance distance as the safety boundary to avoid the collision of the cones in front.

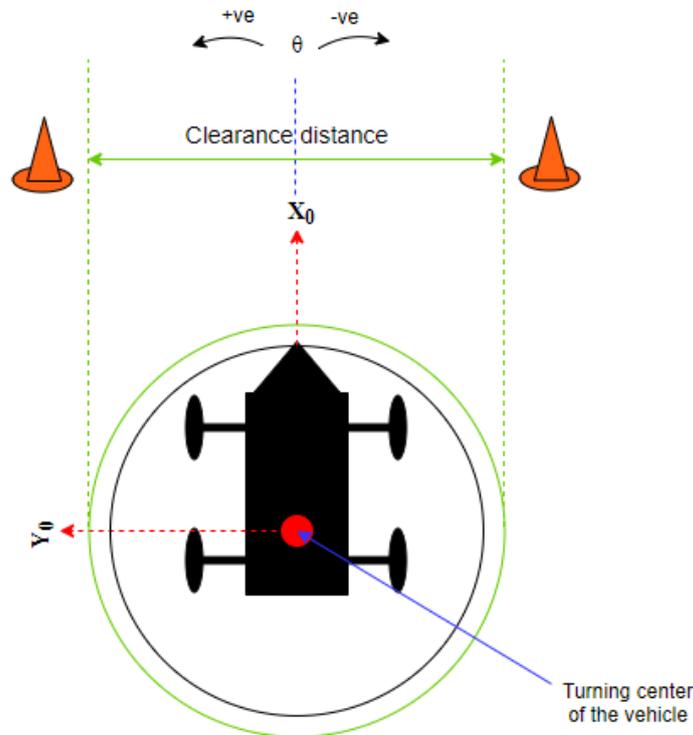


Figure 12: Clearance distance with respect to the cones and the vehicle

Depicted from Fig. 12, the two orange cones in front of the vehicle setup a potential drivable path. The inner circle (black) is projected from the clearance radius from the base link, the length of this radius is typically set to 1.4 m, and the outer circle (green) is generated base on the size of the inner circle with additional radius distance from the half of the width of the cone. The diameter of the outer circle is the clearance distance, and a feasible path is determined when this distance is smaller than the measured gap between the cones.

For requirement 5, the cone driving planner is capable to handle different track settings. Along this year, various closed loop cone tracks have been built by the project team members on the field for testing purpose. Since the formation of those tracks are not the same, the cone driving planner is required to be adaptive.

The adaptability is achieved by adjusting the three crucial parameters on the yaml extension configuration file [21]. The configurable parameters including the desired speed as the reference driving speed, the processing range for the object detection coverage, and clearance radius to evaluate collision free boundary. Instead of recompilation of the cone driving planner, rerun of the planner is sufficient to test on different settings such as faster driving speed to determine the planner's limit.

## 5.0 Results and Discussion

### 5.1 Results of waypoint driving

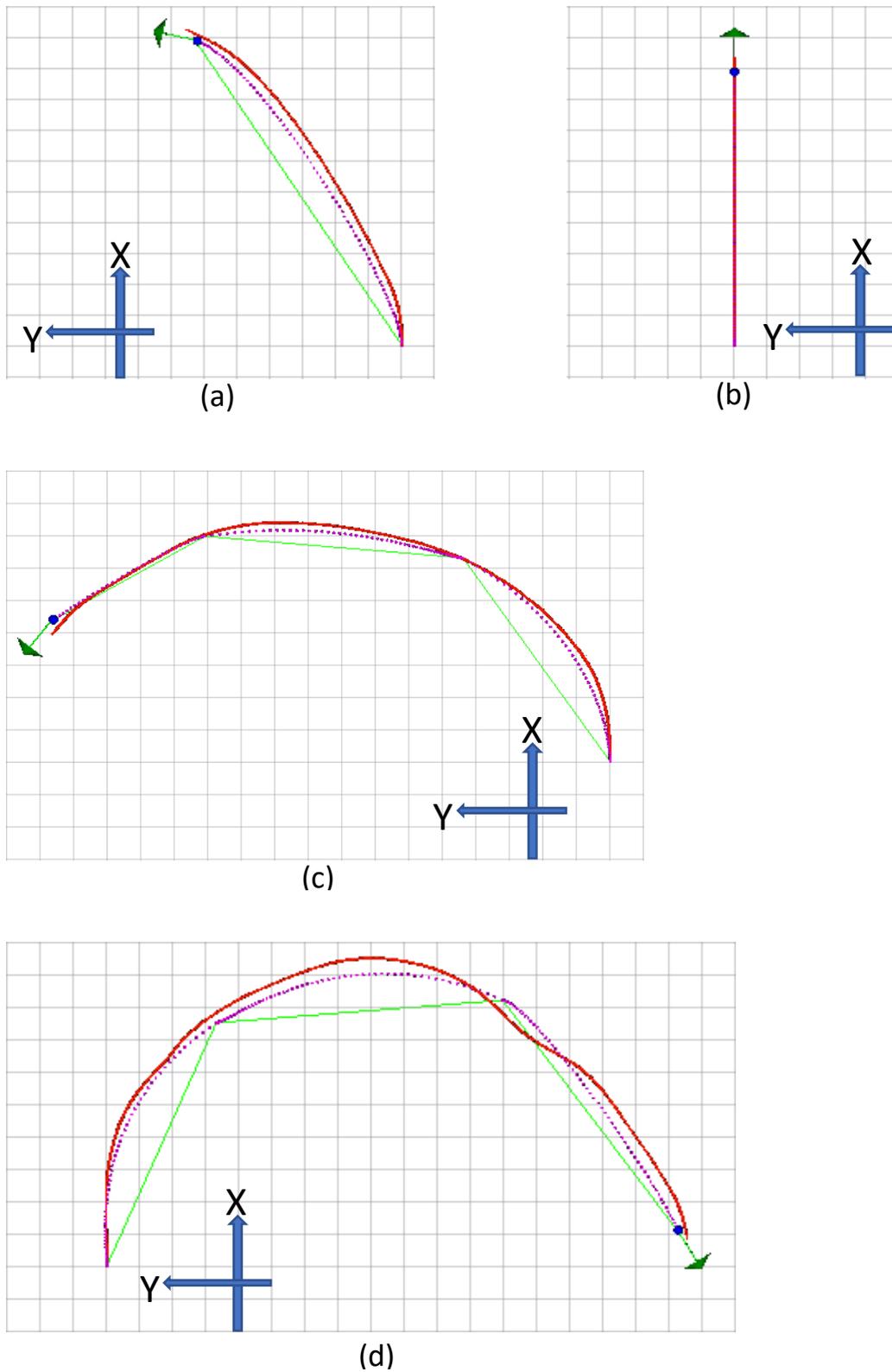


Figure 13: Generated driving path  $M_i$  (red), its ground truth  $D_i$  (purple) and the linear displacements of waypoints (green). [Grid size: 1 m  $\times$  1 m]

## 5.2 Evaluations of waypoint driving

The performance of waypoint driving is evaluated based on measuring its path planning accuracy through the calculation of waypoints across several simulated driving scenarios as illustrated in Fig. 13. This accuracy is quantified by the car's projection error  $\epsilon_p$  (the maximum deviation between the driving path and the ground truth), and its root mean square error  $\epsilon_{rms}$ .

$$\epsilon_{rms} = \sqrt{\frac{\sum_{i=1}^n (D_i - M_i)^2}{n}} \quad (9)$$

Where  $n$  denotes the total number of records;  $D_i$  and  $M_i$  denotes the distance of the ground truth and the driving path, respectively at record  $i$ .

The measurements based on the simulated driving scenarios are tabulated in Table 1.

Table 1: Error and distance measurements from the generated paths

<b>Fig. 13</b>	$\epsilon_p$ (m)	$M_i$ (m)	$D_i$ (m)
(a)	0.313	12.438	11.875
(b)	0.000	8.875	8.875
(c)	0.250	21.375	21.125
(d)	0.500	27.375	26.438

Using the values of  $D_i$  and  $M_i$  in Table 1,  $\epsilon_{rms}$  is computed to be 0.561 m, when compared with the average design track width of approximately 3.5 m, the accuracy can be obtained by:

$$Accuracy (\%) = \left( 1 - \frac{\epsilon_{rms}}{Track\ Width_{Avg}} \right) * 100 \quad (10)$$

Which is evaluated to be 84% accurate. This accuracy indicates that  $D$  is relatively close to  $M$  across the total  $i$  records. Additionally, as the increase in track complexity (such as through the addition of sharper turns and more segments) contributes to a greater increase of  $(D_i - M_i)$  as compared to the increase in  $\epsilon_p$ . As expected,  $\epsilon_p$  is non-existent when a perfectly straight path is generated as shown in Fig. 13(b).

### 5.3 Results of cone driving from simulation

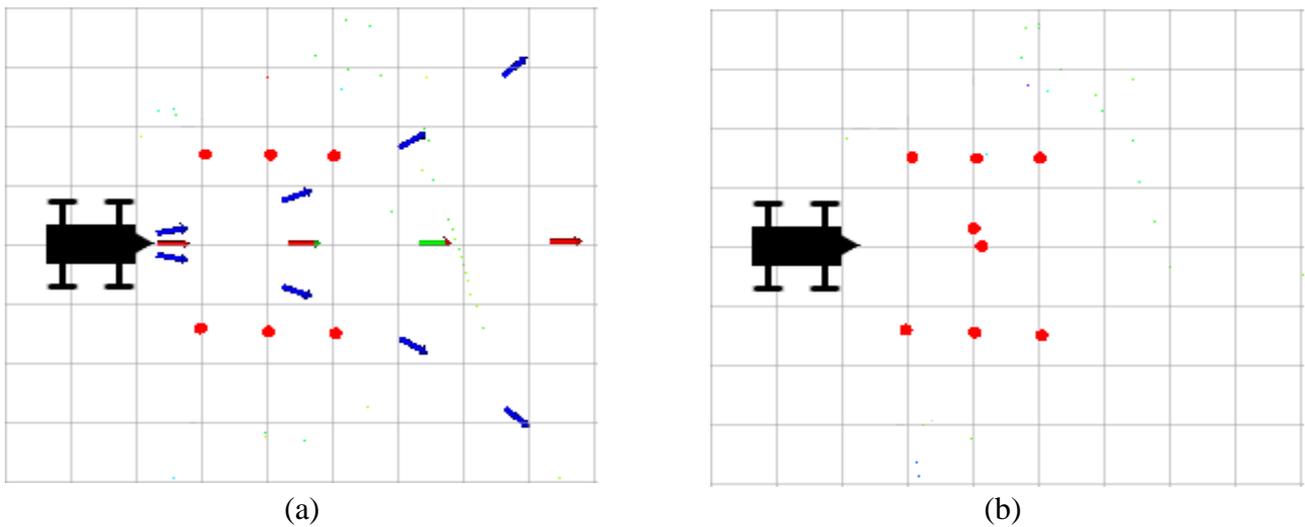


Figure 14: Cone driving simulation outputs on Rviz (a) without obstacle and (b) with obstacles

Two driving conditions have been pre-recorded and being played back to display on Rviz (see Fig. 14). Cone driving planner have been tested on the recorded scene to verify the stopping logic. Depicted from Fig. 14a, the red dots representing the detected cones track within the 3.5 m predefined detection range, unviable paths are blue, driveable path is green which is taken from the mean range of the unviable paths, and the final selected path is red to proceed the driving action. As illustrated in Fig. 14b, when the obstacles have interfered the previously drivable path, the cone planner evaluated that no valid path is exist in front and stop process has been activated.

### 5.4 Results of cone driving from the field

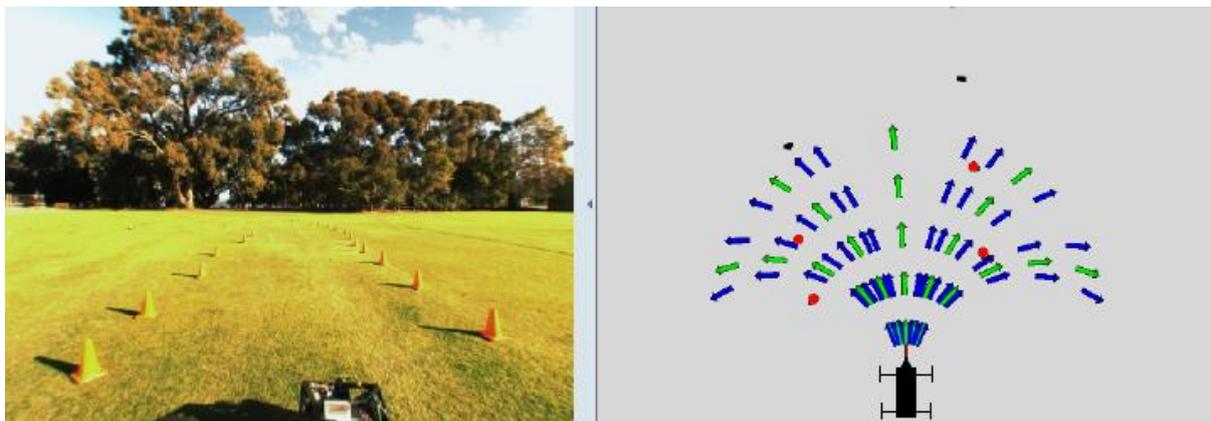


Figure 15: Visual display of cone driving in the test field (left) and the corresponding visualisation in RViz [13]

## 5.5 Evaluation of cone driving from the field

Three different field test records have been utilised to verify the accuracy of the cone driving planner, the projected path that is directing the SAE vehicle to drive outside of the cone track will be deemed as the undesired path (error). The evaluation will be based on the ratio between number of frames with error and number of total frames.

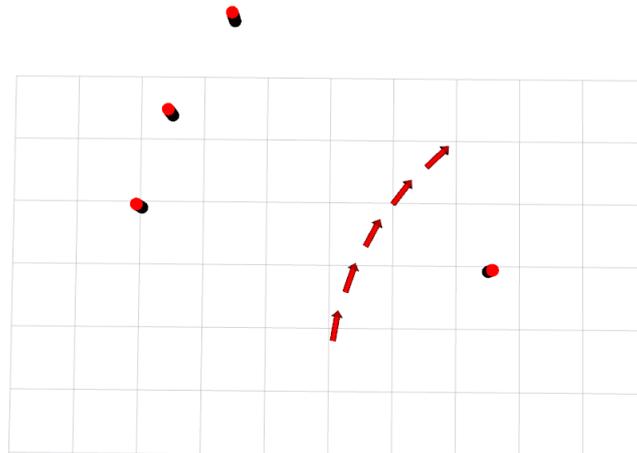


Figure 16: Playback frame with desired path

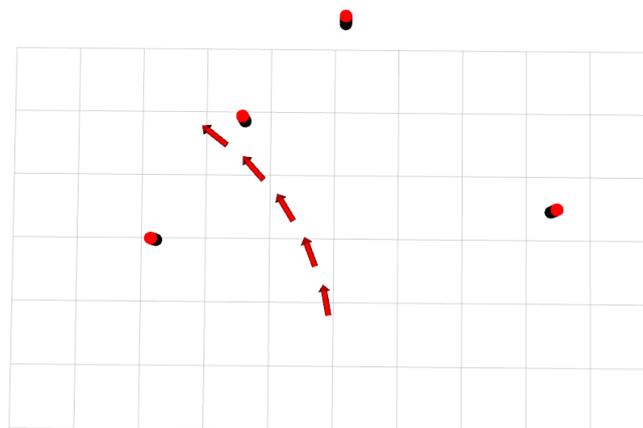


Figure 17: Playback frame with undesired path

Table 2: Percentage error based on number of occurrences of undesired path

Record	Frames with error	Total number of frames	% Error
1	7	84	7.14
2	3	62	4.84
3	1	24	4.17

The overall percentage error is less than 8%, which is consider acceptable. However, at the scene of those field tests, there were no collision with the cones track. It is perhaps the SAE vehicle has scanned more cones in front during manoeuvring. Hence, the cone planner adapts the path adjustment in time to fulfil the obstacle avoidance requirement.

## 6.0 Topics for Further Investigation

Further study of certain areas can enhance the current available features on the SAE, they are road edge detection with SLAM, skid effect consideration, AI modelling inclusion, and reverse motion implementation.

### 6.1 Road edge detection with SLAM

Road edge detection planner, similar concept to the UWA initiative event in August 2018 that introduce a driverless shuttle bus to manoeuvre the campus on a predefined route [22]. The major difference is enabling SAE vehicle with visual SLAM [23], by using the combination of LiDAR scanner and camera to observe the unknown environment, simultaneously update the vehicle position from the odometry and previously stored landmarks of the generated map. Then drive autonomously through the UWA internal roads, rather than traversing the allocated stops and follows the track from a pre-programmed map. The challenge is safety of manoeuvring around the campus within an environment that is consist of static and dynamic obstacles. This involves sensors coordination to detect the surrounding, path planning to dictate the valid path of traverse, driving control to regulate the speed of the SAE vehicle, and facilitate obstacles avoidance to prevent damage any of the UWA properties or other traffic, and potential injury to living objects due to the cause of collision.

### 6.2 Skid effect

The frictional forces between the wheels and the ground with the related skid effect can be considered as the design inputs to establish a robust velocity profile. Once being implemented, it is anticipated to improve the run time performance of the vehicle.

### 6.3 Involvement with AI model

Various Artificial Intelligence (AI) methods can be adopted into this project to gain knowledge of the driving path and applies prediction logic to enhance the obstacles avoidance feature, it would be especially beneficial to the driving on road scenario when the SAE vehicle is sharing a limited movement space with other traffic. Post processing with the recorded image data can be trained to classify object type more accurately. Thus, adaptive learning and more versatile path planning with optimised drive control can be accomplished to potentially achieve the level 5 (fully automated) autonomous driving standard [24].

### 6.4 Reverse motion

Current design has not included the reversing motion at the autonomous driving level. It would be useful to have this extra feature in the situation when the SAE vehicle is jammed in the dead end during its manoeuvre. This would require a rear viewed camera version to observe the environment during reversing.

## 7.0 Conclusion

The Hermite spline interpolation has served the path smoothing purpose of waypoint driving, as well as with the correct heading considered. However, the drive control to follow the smoothed path requires more works including optimisation to increase the performance and reduce the root mean square error. Other spline approaches, for instance, the natural spine could be a smoother interpolation method and easier for the drive control to trace the curvature section, even though natural spine might increase the computational complexity and time as a matter of solving the linear system.

On the other hand, the cone following planner have proved the autonomous driving on the close loop cones track without collision of the cones is achievable, the concerns of correct heading to enable the smoothness of the manoeuvre is not necessary as the algorithm is favoured to choose the optimal steering angle in the least amount of turning from the vehicle current orientation. Once all the crucial parameters have been calibrated and manually driving through the track to verify the feasibility of the valid path, then SAE vehicle is able to autonomously, safely and continuously manoeuvring the track.

## Bibliography

- [1] T. and R. E. Bureau of Infrastructure, "Road Deaths Australia—Monthly Bulletins," 14-Aug-2018. [Online]. Available: [https://bitre.gov.au/publications/ongoing/road\\_deaths\\_australia\\_monthly\\_bulletins.aspx](https://bitre.gov.au/publications/ongoing/road_deaths_australia_monthly_bulletins.aspx). [Accessed: 23-Sep-2018].
- [2] ASIRT, "Road Safety Facts," *Association for Safe International Road Travel*, 2018. .
- [3] K. L. Young and P. M. Salmon, "Examining the relationship between driver distraction and driving errors: A discussion of theory, studies and methods," *Safety Science*, vol. 50, no. 2, pp. 165–174, Feb. 2012.
- [4] Daielle Muoio, "RANKED: The 18 companies most likely to get self-driving cars on the road first | Business Insider," 24-Sep-2017. [Online]. Available: <https://www.businessinsider.com.au/the-companies-most-likely-to-get-driverless-cars-on-the-road-first-2017-4?r=US&IR=T#/#18-baidu-1>. [Accessed: 25-Sep-2018].
- [5] T. H. Drage, "Development of a Navigation Control System for an Autonomous Formula SAE-Electric Race Car," p. 86, 2013.
- [6] L. Laursen, "Students Race Driverless Cars in Germany in Formula Student Competition," *IEEE Spectrum: Technology, Engineering, and Science News*, 16-Aug-2017. [Online]. Available: <https://spectrum.ieee.org/cars-that-think/transportation/self-driving/students-race-driverless-cars-in-germany-in-formula-student-competition>. [Accessed: 25-Sep-2018].
- [7] FS Germany, "FSG2018\_Compensation\_Handbook." 2018.
- [8] Thomas Bräunl, "Localization and Navigation," in *Embedded Robotics*, Springer, 2006, pp. 197–205.
- [9] Tim Sharp, "How Big Is Earth? - Radius, Diameter and Circumference Explained." [Online]. Available: <https://www.space.com/17638-how-big-is-earth.html>. [Accessed: 17-Oct-2018].
- [10] Thomas Churack, "Improved Road-Edge Detection and Path-Planning for an Autonomous SAE Electric Race Car," UWA, Perth, 2013.
- [11] Open Source Robotics Foundation, "ROS/Introduction - ROS Wiki." [Online]. Available: <http://wiki.ros.org/ROS/Introduction>. [Accessed: 16-Oct-2018].
- [12] Canonical Ltd., "About the Ubuntu project | Ubuntu." [Online]. Available: <https://www.ubuntu.com/about>. [Accessed: 16-Oct-2018].
- [13] Kai Li Lim, Thomas Drage, Chao Zhang, Craig Brogle, William W. L. Lai, and Timothy Kelliher, Manuchekhr Adina-Zada, and Thomas Bräunl, "Evolution of a Reliable and Extensible High-Level Control System for an Autonomous Car," [Under Review].
- [14] H. R. Kam, S.-H. Lee, T. Park, and C.-H. Kim, "RViz: a toolkit for real domain data visualization," *Telecommun Syst*, vol. 60, no. 2, pp. 337–345, Oct. 2015.
- [15] T. Groß, "draw.io - Diagrams For Everyone, Everywhere," *draw.io*. .
- [16] P. Wolfer, *Simple animated GIF screen recorder with an easy to use interface: phw/peek*. 2018.
- [17] The MathWorks, Inc., "2-D and 3-D Plots - MATLAB & Simulink - MathWorks Australia." [Online]. Available: [https://au.mathworks.com/help/matlab/learn\\_matlab/plots.html?s\\_tid=srchtitle](https://au.mathworks.com/help/matlab/learn_matlab/plots.html?s_tid=srchtitle). [Accessed: 16-Oct-2018].

- [18] Thomas Bräunl, ""Evolution of Walking Gaits," in *Embedded Robotics*, Berlin: Springer, 2006, pp. 345–346.
- [19] SICK AG, "LMS111-10100 | Detection and ranging solutions | SICK." [Online]. Available: <https://www.sick.com/au/en/detection-and-ranging-solutions/2d-lidar-sensors/lms1xx/lms111-10100/p/p109842>. [Accessed: 10-Oct-2018].
- [20] Brian Paden, Michal Cáp, Sze Zheng Yong, , Dmitry Yershov, Emilio Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles," *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, vol. 1, no. 1, pp. 33–55, 2016.
- [21] Oren Ben-Kiki, Clark Evans, Ingy döt Net, "YAML Ain't Markup Language (YAML™) Version 1.2." [Online]. Available: <http://yaml.org/spec/1.2/spec.html>. [Accessed: 15-Oct-2018].
- [22] Jon Bassett, "Driverless electric bus takes to the road at UWA," *Community News Group*, 09-Aug-2018.
- [23] Søren Riisgaard, Morten Rufus Blas, ""SLAM for Dummies (A Tutorial Approach to Simultaneous Localization and Mapping)." [Online]. Available: [https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam\\_blas\\_repo.pdf](https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-412j-cognitive-robotics-spring-2005/projects/1aslam_blas_repo.pdf). [Accessed: 17-Oct-2018].
- [24] Hope Reese, "Updated: Autonomous driving levels 0 to 5: Understanding the differences," *TechRepublic*. [Online]. Available: <https://www.techrepublic.com/article/autonomous-driving-levels-0-to-5-understanding-the-differences/>. [Accessed: 08-Apr-2018].

## Appendix A

Check list for safety tripping version 1.2

Created on 21/08/2018

Created by: WL

Which part in the system	Function	Condition under test drive
HB battery level check	If anything is abnormal to the heartbeat warning/error will be print to screen.	It works, stay the same as 3.5 V before and after the field test
LED on top dash – HBT [10Hz binary square wave sampled at 50Hz]	Blinks on and off at normal state but stays on after a trip due to lost heartbeat.	It works at the tripping condition due to lost heartbeat, by pressed down e-stop button on the base station
Dead man’s switch on dash	Allows simulation of the heartbeat by a person in the car. <b>1 Hz interval minimum</b>	It seems to response when pressing the dead man switch on dash, however it requires to press in a very fast speed for it to work, suggestion to improve: 1. Switch replacement with easier pressing button 2. Apply push then let-go check on the software setting instead of frequently button press. However, it is not recommended to focus on response with the dead man switch in the emergency event, as the <b>observer</b> should apply the <b>e-stop button</b> on remote base station to safely stop the vehicle. And <b>driver</b> should apply <b>brake, e-stop on dash</b> and <b>observe</b> the road condition
Low-level controller WDT [ER2]	Reset the controller in the event of main loop freeze - puts the controller to trip state	When Autonomous mode driving is activated but ROS is not started, it will not publish any topics, then the brake will engage and trips eventually

Which part in the system	Function	Condition under test drive
Steering motor current [ER4] + LED on front dash – STR Over	Indicator of excess steering force if someone steers against the motor) will result in steering motor disconnection that causes a trip of the safety supervisor, by wiring the relay contacts in series with the safety supervisors e-stop feedback.	Not a safety practice to test this feature. In the emergency event, it is recommended to engage brake first. As excess steering might bring the vehicle to completely lost control position
Safety supervisor with DES condition	Indicate a trip came from the dash traction motor e-stop being pushed in or from the low level safety box.	It works when pressed down the e-stop next to the steering wheel to stop the SAE
High level controller	At the tripped state, it will re-trip if the high level controller hasn't also reset and is still sending trip commands	Required to restart ROS SAE
LED on top dash - ACL	Throttle voltage error. Either no power on traction motor controllers, or there was a positive throttle signal after enabling the motor controllers (which could have caused the car to unexpectedly take off).	Tested in the lab and works. SAE cannot be ARMED while accelerator pedal is pressed
LED on top dash - ESTOP	lights on all trips. If HBT and ACL don't also light then it either came from the dash e-stop, low level safety box or was commanded by the high level controller.	It has happened, possible from low level safety box. Suspect abnormal brake engagement due to the over sensitivity of brake hall sensor initiate the trip
LED on top dash – Safety mode	This <b>blue LED</b> should be always switched <b>on</b>	When it's off, then it is not depending on HBT anymore (i.e. heartbeat remote e-stop will not trip the SAE)
LED on front dash – BRK Over	Indicate that SAE is tripped due to abnormal brake action	Activated, suspect abnormal brake engagement due to the over sensitivity of brake hall sensor initiate the trip
No new command in 300 ms [ER5]	SAE will trip as Jetson is not responding	It works as intended
Steering sensor out of bounds [ER6]	Extreme steering condition that will initiate the trip.	It might damage the steering motor if happened. Have not happened

Which part in the system	Function	Condition under test drive
High level controller	Reset speed to zero when no new speed command received in 100ms. This is in case demo program crashes.	Tripped because of not receiving command due to program is crashed. Set speed to zero but kept the same steering
Serial port connection	If the connection to safety serial port is failed, the program will exit.	Happened, due to the USB connection lost detected from the 4 ports USB hub. Required to remove USB connection/reconnect from the Jetson's end

Other Notes:

1. USB port on Jetson (due to power overload by the USB hub) can be the cause of tripping and requires reboot of Jetson to fix the issue. Possible solution is powering the USB hub from the 5 V source instead via the round power plug.
2. If speed set above 3 m/s will trip SAE as of the 17/08/2018 field test.