IMPROVED ROAD-EDGE DETECTION AND PATH-PLANNING FOR AN AUTONOMOUS SAE ELECTRIC RACE CAR

Thomas Churack

20751503

Supervisor: Professor Dr. Thomas Braunl

Submitted: 26th October 2015

Word Count: 6395

University of Western Australia, School of Electrical and Electronic Engineering

Table of Contents

Abstract

This paper describes the development and implementation of several improvements made to the UWA REV Autonomous SAE Electric Race Car, specifically with respect to its road-edge detection capabilities and path-planning intelligence.

The road-edge detection capability of the vehicle has been extended to allow the vehicle to make use of all the layers of incoming LiDAR scan data provided by the IBEO laser scanner and this data is also now adjusted in real time, so as to compensate for any variations in vehicular attitude.

One application for road-edge detection - the ability to map traversed roadways - is explored to test the efficacy of these improvements in practice.

The vehicle's road-edge detection capability sees further additional applications through the exploration and implementation of its integration in to the vehicle's path-planning subroutines, so as to facilitate the vehicle being able to autonomously remain on road while driving, rather than being manually instructed to follow curves through dense way-pointing. A lane-keeping algorithm is also proposed to further extend the vehicle's path-planning ability with a view towards making the vehicle practical to operate in an urban environment.

Results are demonstrated through simulations showing the successful integration of real-time road-edge avoidance, thus achieving the goal of the project.

Acknowledgements

The author would like to thank the following people for their assistance throughout the course of the project: Prof. Thomas Bräunl for the opportunity to undertake this project; Chris Blignaut, Marcus Pham, Thomas Drage, Ruvan Muthu-Krishna, Martin French and Sonia Miranda for their assistance and contributions to the SAE car and my project; Dana, Mum, Dad, Nan and Papa for their support, motivation and encouragement and Casey, Peter and Gurjun for their advice, assistance and suggestions.

Publications

Components of this project have been included in the following published manuscript:

T. Drage, T.Churack, T. Braunl (2015), "*LIDAR Road Edge Detection by Heuristic Evaluation of Many Linear Regressions*", presented at the IEEE 18th International Conference on Intelligent Transportation Systems and to be published by IEEE as part of the conference proceedings.

Nomenclature

REV	Renewable Energy Vehicle Project
SAE	Society of Automotive Engineers
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LiDAR	Light Density and Ranging
UWA	University of Western Australia
PID	Proportional-Integral-Derivative controller
GPU	Graphics Processing Unit
SLAM	Simultaneous Localisation and Mapping

Introduction and Background

Introduction

Figure 1: The Autonomous SAE Race Car as of 2010 [1], 2013 [2] and the end of 2015 (L-R)

The Autonomous SAE Race Car discussed in this paper was originally designed and built by UWA Motorsports as their inaugural 2001 entry in to the Formula SAE student design and performance competition organized by the Society of Automotive Engineers [3]. In 2010 the vehicle was modified to "to a battery-electric vehicle using a dual rear motor drive with an electronic differential" [3] by students as part of UWA's Renewable Energy Vehicle Project (REV) so as to meet the requirements of the then new Formula SAE-Electric vehicle class [3].

In 2013 the vehicle was chosen to become to the new test bed for autonomous technologies research by the REV Project [4] as a successor to the BMW X5 that had previously been converted by students to brake-by-wire and steer-by-wire [5,6]. The modifications made to the Autonomous SAE Race Car (the vehicle) in 2013 enabled: electronic control of the vehicles throttle, servomotor actuation of hydraulic brakes and motor-controlled steering [2], making the vehicle fully drive-by-wire and further building upon the research undertaken on the BWM X5. Additional research by students completed in parallel over 2013 produced basic autonomous driving capabilities for the vehicle [2].

While this addition of autonomous capability took the vehicle outside the scope of any Formula SAE competition, it allows the REV Project many potential research possibilities within the scope of autonomous driving and mapping, as well as allowing final year students within the REV project to explore the novel research area of autonomous high-performance race vehicles.

This paper describes several improvements made to the vehicle's road-edge sensing and mapping capabilities as well as the integration of this in to the vehicle's path-planning routines so as to facilitate autonomous on-road driving.

Motivation

The use of the Autonomous SAE Race Car as a research platform is in part motivated by the recently intense level of interest shown, and research and development done by, both traditional vehicle manufacturers such as Mercedes-Benz [7], Nissan [8, 9], Renault [8, 9] and Volvo [9] as well as modern major technology companies like Google [10], Apple [10] and Tesla (who has already put semi-autonomous vehicles in the hands of consumers) [11]. This suggests that research in this area has immense commercial potential and could result in the production of new technologies.

In particular, the fact that the Autonomous SAE Race Car was originally designed and used for high performance and competitive driving – areas that have had relatively little exposure to autonomy – provides a relatively unique context from which to undertake research. Traditionally, high-performance and competitive automobile research has resulted in many technological advances that have benefited vehicles of all applications, and as such, one could expect that the same could result from high-performance autonomous vehicle research.

The extension of the Autonomous SAE Race Car to be able to correctly identify and keep on roads would improve the autonomy of the vehicle allowing it to stay on-road (as would be required in a racing situation) without the need for the manual inputting of dense waypoints along any curved sections of road. In order for the vehicle to ever operate in an urban environment it would similarly be a requirement that the vehicle never deviates from the road.

Literature Review

An autonomous car is an automated vehicle that is capable of sensing and reacting to its environment and then navigating within it, without the need for human intervention. Research into the automation of cars goes back as far as the 1920s [12] where a car controlled remotely via radio was demonstrated by a car manufacturer. However it was not until the late 1980s that serious interest in the concept of a fully autonomous vehicle being developed was shown, with several universities and car manufacturers beginning research projects in the area, such as the Prometheus Project and the Autonomous Land Vehicle Project [13, 14, 15].

Since the mid-2000s several competitions and challenges have been hosted by organisations such as DARPA and VisLab, resulting in further exposure and encouraging research towards fully autonomous vehicles [16, 17]

The technologies developed in the pursuit of autonomous vehicles have resulted in benefits in various industries, with Rio Tinto using a fleet of autonomous haulage vehicles and autonomous drills on its mine sites [18, 19] and investigating the use of autonomous trains [20] to maximise safety and efficiency. Autonomous vehicles have also seen use as public transportation [21, 22].

Cars are becoming more and more autonomous over time, with technologies such as: adaptive cruise control [23]; automatic parallel parking [24]; blind spot monitoring [25] and intervention systems [26]; lane keeping [27] and automatic braking [28] coming to consumer vehicles over the last two decades. This increasing ability for vehicles to monitor and react to their environments has resulted in cars that are physically capable of fully autonomous driving being sold to consumers with only legislation and in some cases software limiting the use of full autonomy [29]. It is expected that autonomous vehicles could account for up to 75% of cars on the road by 2040 [30].

Prior research has been conducted on road and road edge detection through optical systems [31, 32, 33], radar [34, 35, 36] as well as through the use of Light Density and Ranging (LiDAR) sensors such as in the winning entry in the 2007 DARPA Urban Challenge [37] and in research at German [38] and Singaporean [39] universities. The methodology described in [36] utilises a feature-extraction algorithm based upon location of local maxima and minima in the LiDAR data as well as the variance of segments between these extrema. Other algorithms such as [39, 40, 41] rely on the presence of curbs and seek to identify and track curbs as features in the LiDAR data. The approach described in [38] is similar to the original algorithm employed here and attempts to seek appropriate linear fits to the road surface. In both of the latter two cases a Kalman filter is used to track the position of road edges temporally.

It has been shown that the accuracy of LiDAR based mapping can be improved by adjusting incoming LiDAR data for the sensor's pitch and roll, as measured by an on-board Inertial Measurement Unit (IMU) [42, 43, 44]. This strategy has been proven to be viable even for unmanned aerial vehicles [44], which require compensation for a much greater degree of pitch, roll and yaw than an autonomous land vehicle is likely to. By analysing LiDAR data consisting of multiple scan layers intersecting the roadway, an estimation of road shape and road curvature can be calculated [44, 45].

The problem of path-finding can be described as: "Given a start state, a goal state, a representation of the robot and a representation of the world, find a collision-free path that connects the start with the goal satisfying the system constraints" [46]. In mobile robotics a proven method to obtain the requisite "representation of the world" is via the use of LiDAR data to generate a virtual map in real-time both as the sole sensor [47] and in conjunction with data from additional sensors [48,49]. Similar LiDAR based map building approaches have been shown to be suitable for outdoor terrain [50, 51] and have been scaled up to larger autonomous vehicles with great success [52, 53]. These generated maps vary from simple two dimensional maps consisting of traversable regions, obstacles and unexplored regions [53, 54, 55], (suitable for simple path-planning algorithms) to more complex three dimensional maps that include the elevation of points from which more complex cost maps are generated [52, 56, 57].

System Overview

The Autonomous SAE Race Car consists of three main components: the vehicle's Control Software – currently being run on a Raspberry Pi 2; the Low Level Controller and the Safety Supervisor. The relationship between these components is depicted in Figure 2. The Low Level Controller is responsible for the actuation of physical vehicle components based upon instruction from the Control Software; such as the motor attached to the steering column or the servomotor attached to the hydraulic brakes, it also generates an artificial throttle signal when the vehicle is autonomously accelerating. The Safety Supervisor is designed to ensure the safety of the occupant and any bystanders should a fault occur with the Vehicle's Control software. This project consists of modification made only to the Control Software, so it is explained in greater detail below.

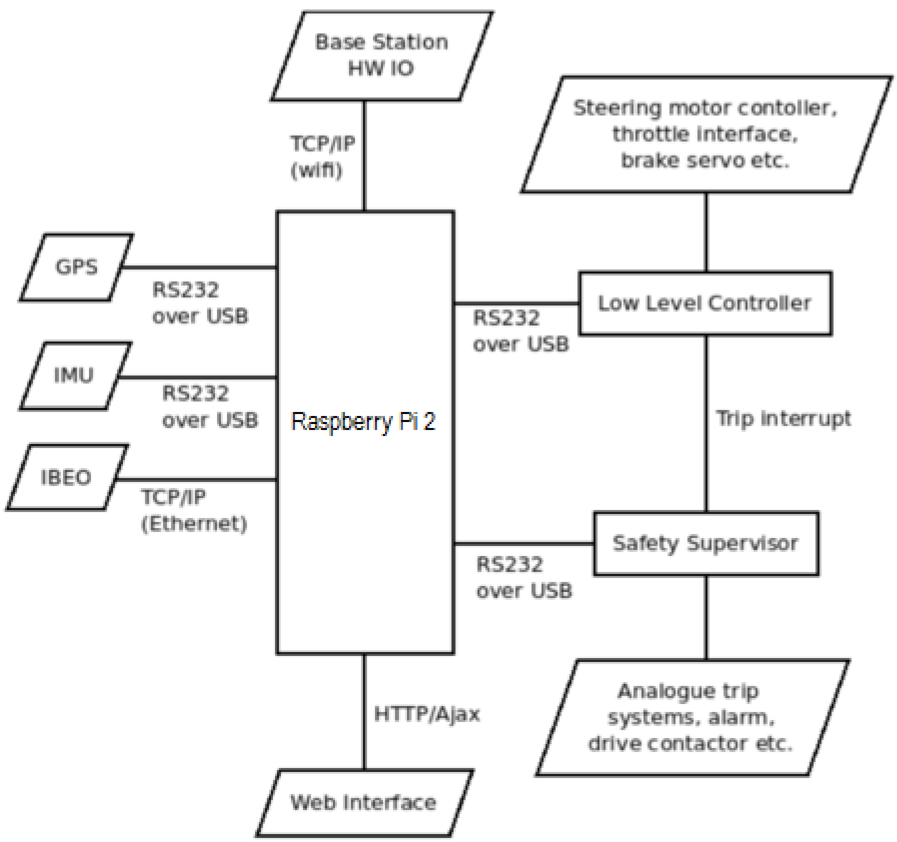

Figure 2: System Overview [2]

The Control Software reads in data from all of the on-board sensors, such as the GPS, Inertial Measurement Unit, and the IBEO LiDAR scanner; it processes this information so as to map and sense its environment as well as determine its location and orientation within this environment. The Control Software is also responsible for communication with the human operator(s) through the 'base station' – an external safety monitoring station, and a Web Interface - consisting of a series of dynamic web pages that are displayed through a web browser either on the screen mounted on the vehicle's dash or via any other web capable device connected to the vehicle's local network. The Control Software receives instructions from an operator through the Web Interface, these instructions consist of desired location(s) for it to drive to in the form of GPS waypoints. The Control Software then calculates various control parameters in order to drive the vehicle to these location(s) and based on these parameters, it instructs the low level system so as to accomplish tasks. This project focuses on the mapping and sensing of the environment and the subsequent planning of a path from the vehicle's current location to the desired location(s).

The Mapping and Sensing is performed through the analysis of a variety of incoming sensor data which is interpreted by software made up of a combination of C++ classes and device specific libraries. The resulting interpreted view of the environment and the vehicle's place within it are then passed in to the path-planning subroutines, which consists of calls to a static library that has been generated from MATLAB code. This path-planning library then creates intermediate waypoints to map out the path it has planned to approach the next desired location and passes these in to the vehicle control component.

The final component of the Control Software is the Vehicle Control, this component receives a set of immediate waypoints from the Path-Planning Component. This component will then sequentially drive the vehicle to each of these waypoints until it reaches the final one or receives a new set of waypoints. A waypoint is determine to be 'reached' if and when the vehicle determines that its current location is within a distance less than a defined tolerance of the waypoint. Upon reaching a waypoint the heading to the next waypoint is calculated and set as the desired bearing, this is then converted in to a low-level steering command by using a PID loop and this is sent through to the Low-Level Controller to turn the steering motor as necessary to steer the vehicle towards the next waypoint.

Road-Edge Detection

Background

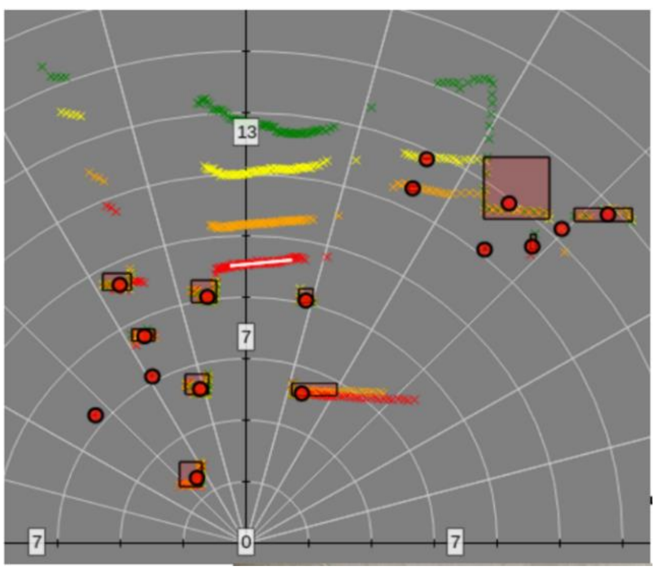

Figure 3: IBEO LiDAR Sensor Data, objects depicted by circular object centres with bounding boxes, scan layers colour coded and road segments depicted in white. Distances in metres.

A project undertaken by a previous final year student included the design and development of a road-edge detection algorithm for use on the Autonomous SAE car [2]. This algorithm operated by heuristically evaluating incoming LiDAR "scan data" - sets of polar angle/radial distance pairs, based on a set of criteria and assumptions about road characteristics. The algorithm also incorporated a Kalman Filter [58] so as to allow for time-averaged estimations of the road-edges' positions. The algorithm was shown to be able to determine road edges on both curbed and non-curbed roads in real-time while under dynamic conditions [2].

However, there were some limitations in the efficacy of this algorithm as of the completion of this student's project. The implementation of the algorithm was limited to the analysis of a single layer of incoming scan data (the layer closest to the vehicle). This meant the vehicle was only capable of detecting road-edges immediately in front of the vehicle, too late for the vehicle to react in any way. This incoming data was also not adjusted for the vehicle's attitude – its pitch and roll, this meant that the accuracy of the algorithm was reduced when the vehicle was driven over rough terrain or at high speeds. The detected road-edges were also discarded by the vehicle's control software as there was no integration of road-edge avoidance in the vehicle's path-planning subroutines and as such, the detected road-edges were unused.

Multi-Layer Algorithm

Before detection of road-edges can be integrated in to the vehicle's path-planning algorithm, the implementation of the algorithm on the vehicle must be modified to allow for the analysis of multiple layers of incoming LiDAR scan data. This is because LiDAR scan data is received in 'slices', and as such, a single, individual layer of scan data contains a significant number of points of data horizontally (left-right relative to the scanner), but it is only a single data point 'thick' in the forwards-backwards direction. By extending the algorithm to multiple layers of scan information the vehicle is able to detect deviations in the road from much farther away, while the vehicle still has plenty of time to react and modify its planned paths accordingly, without sacrificing the ability to monitor road-edges closer to the vehicle were it is more critical. This also allows for an accurate estimation of road shape and the calculation of road curvature to be generated [44, 45].

The implementation of the algorithm across multiple layers is to be implemented in its simplest form. Firstly, the incoming scan data from the IBEO LiDAR is filtered by scan layer, the data is then adjusted for the vehicle's attitude (as described below) and then finally, the original road-edge detection algorithm is run over each layer individually, returning one pair of road-edges per scan layer. The original algorithm itself will remain unmodified within the scope of this project.

Attitude Correction and Mapping

As the vehicle drives over a non-flat surface, its pitch and roll angles will constantly vary, this introduces noise into the Kalman filter, which would make it more difficult to track road edges over time. This error increases as the distance of scan points from the vehicle increases (i.e. for further scan layers) as slight changes in angle result in large changes in displacement at long distances, and this would also limit the usefulness of mapping detected road edges as without an accurate idea of the vehicle's orientation in global coordinates, only the local position of detected road-edges relative to the car can be determined. For this reason, the attitude correction for the vehicle's pitch and roll are performed prior to the scan information being passed into the road-finding algorithm, as this minimises what would otherwise be a significant source of noise to the Kalman Filter, and this has been proven to improve the accuracy of LiDAR generated maps [42, 43, 44].

Firstly, the incoming scan data must be converted from polar coordinates (relative to the IBEO LiDAR scanner) into three dimensional Cartesian coordinates. This is done through simple trigonometry.

The distance of each scan point from the scanner and its horizontal angle are the values returned by the LiDAR, and the vertical angle of the scan (degrees below the plane parallel to the car and its IMU) is pre-determined for each scan layer, by summing the angle below the horizontal for the LiDAR scanner itself (4.0°, for this project) and the angle of each scan layer relative to the scanner (2.9°. 1.58°, 0.65° and -0.29°, for this project). The Z-Coordinate (height in metres) of a point can then be calculated by simple trigonometry as the product of the absolute distance and the inverse sine of the vertical angle. The planar distance (absolute distance in the XY-plane, parallel to the car and the IMU's planes) is similarly the product of the absolute distance by the inverse cosine of the vertical angle. From the planar distance, the X (metres to the right) and Y-Coordinates (metres in front) can be calculated once again by basic trigonometry as the products of planar distance and the sine and cosine of the horizontal angle respectively.

In order to convert the local XYZ-Coordinate system to a global coordinate system, the scan points must be adjusted for rotation of the car. The IMU attached to the car measures both its pitch (forward/backward) and roll (tilting left/right) relative to the horizontal plane (the plane perpendicular to gravity). In order to correct for these rotations an origin must be chosen, as the LiDAR, GPS and IMU are all located in a line perpendicular to the vehicle, the point where this line intersects with the plane made by the bottom of the four wheels of the car was chosen. As this prevents the pitch, roll and global positioning data losing accuracy, is a simple transform from the LiDAR-as-origin coordinate system and once corrected for pitch and roll; gives coordinates relative to the ground's surface rather than the scanner, i.e. Z-Coordinate becomes the height above ground rather than above scanner. This transform is enacted by subtracting the height of the LiDAR (1.225m above the ground) from each Z-Coordinate.

To then correct for the detected pitch and roll of the vehicle (with respect to the new origin), standard 3D-rotation matrices are used.

For pitch correction: $\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{pmatrix}$, and for roll: $\begin{pmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{pmatrix}$,

where θ is the angle below the horizontal (pitch) and φ is the angle of the left of the vehicle above the horizontal and of the right of the vehicle below the horizontal. As the vehicle drives over a non-flat surface, its pitch and roll angles will constantly change, this introduces noise into the Kalman filter and can make it more difficult to track road edges over time, this error increases as the distance of scan points from the vehicle increases (i.e. for further scan layers). For this reason, the correction for the vehicle's pitch and roll are performed prior to the scan information being passed into the road-finding algorithm, as this minimises what would otherwise be a significant source of noise to the Kalman Filter.

The road-finding algorithm thus returns a pair of X coordinates (distances left/right relative to the vehicle) and a slope and intercept corresponding to a line of best fit through the scan points between these two X coordinates, which are corrected for pitch and roll but still local coordinates. In order to determine the location of road edges and scan points globally, these points must be rotated to account for the vehicle's heading at the time of the scan and the vehicle's current distance from the global origin.

The vehicle's heading at any time is calculated from a fusion of IMU and GPS data, from this data, the angle by which the points must be rotated in the XY-plane can be calculated. In order to rotate the lines of best fit that correspond to detect road segments, the end Y-coordinates of the line end-points are calculated. This is done by substituting the X-coordinates into the line, which can be displayed in the form of a standard linear function as:

$y = slope \times x + intercept$, the end-points of the line are then represented by X,Y pairs, which can be rotated about the vehicle origin by the transformation matrix: $\begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$, where ψ is 360° minus the vehicle's heading.

The detected road edges are now located by XY-coordinates independent of the car's rotation; however, they are still determined by their distance from the vehicle. In order to convert them to a global coordinate system a global origin must be chosen. The vehicle is pre-programmed with a default GPS coordinate (the Datum) representing the global origin, but this can be changed to the vehicle's current GPS location. The vehicle's GPS coordinates at any time can then be converted in to an approximation of distance North and East from the Datum which becomes the X and Y coordinates of the vehicle. Detected road edges can then be converted from XY- coordinates relative to the vehicle to global coordinates relative to the Datum by summing the converted X and Y coordinates of the vehicle with the relative coordinates of the detected road edges. Road segments can then be determined and plotted simply by fitting a line between each pair of XY coordinates.

On-Road Path-Planning

Background

The vehicle's path-planning algorithm as of the start of this project was implemented by a previous student as part of their Final Year Project. The algorithm operates as four distinct sub-tasks: the generation of a base frame, the generation of potential manoeuvres, the calculation of manoeuvre costs and the selection of manoeuvres [4], these steps are explained below.

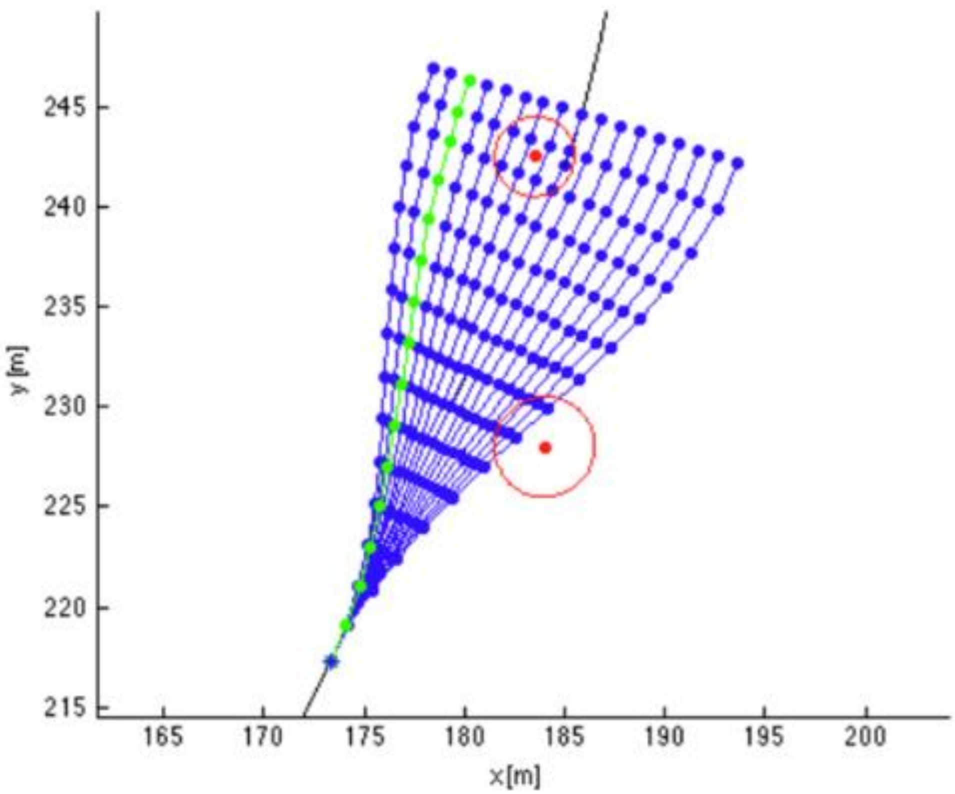

Figure 4: Simulated selection of path so as to avoid obstacles (in red). Chosen path in green. [4]

Prior to commencing a drive, the vehicle is provided a series of GPS waypoints; the algorithm then generates a 'base frame' (or ideal path) by calculating a parametric cubic spline through these waypoints. At each instance of path-planning, the vehicle's position along and relative to this 'base frame' can be determined. Then, a set of path candidates are generated from the vehicle model's information and the vehicle's current heading. Each of these paths are then given a cost calculated from a combination of cost functions representing: the path's distance from the base frame (to ensure the path does not deviate excessively from the ideal path), the consistency of this path with the previous path in terms of heading (to prevent 'jerky' driving), and the safety of the path (to prevent collisions or near misses). The path with the lowest cost is then selected by this algorithm and passed in to the vehicle's Control subroutines [4].

This Advanced Path-Planning algorithm allows the vehicle to dynamically avoid static obstacles by re-planning paths in real time as obstacles are detected [4].

Road-Edge Avoidance

One limitation of the previously described path-planning algorithm is the lack of any consideration of the need of a vehicle to stay on-road. With the improvements made to the road-edge detection of the vehicle – namely the increased range, accuracy and number of detected road edges – it became feasible to integrate the incoming road-edge data into the Advanced Path-Planning algorithm so as to allow the vehicle to plan paths that keep it safely on the road.

In order to implement this, an approach similar to that of the already implemented obstacle avoidance [4] was used. Firstly, as road-edges are detected they are appended to FIFO data structures(s) (in this particular implementation two Double-Ended Queues – chosen so as to prevent memory fragmentation errors, one for X-Coordinates and one for Y-Coordinates), in order to prevent excessive time and memory consumption these data structure(s) have a maximum size (tuneable in the source code), if the structure is filled and a new pair of road edges is detected, the oldest pair of road edges is discarded to make room. As long as the maximum size of these data structure(s) is large enough to hold the road edges from immediately in front of the vehicle, (i.e. to ensure the discarded road-edges are only discarded after the vehicle has driven past them and no longer has any immediate need for them) the discarding of road-edges would not impact on the ability of the vehicle to plan paths that stay on-road. All road-edges are still logged to allow for maps to be retroactively generated.

Then at each path-planning instance the current collection of road-edges along with the current collection of detected obstacles is added to a new data structure ordered as such: X-Coordinate, Y-Coordinate and then either the Safety Radius (the distance by which obstacles should be avoided) or the Road Edge Avoidance Radius, depending on whether the coordinates are for an obstacle or a road-edge.

This data structure is then passed into the Advanced Path-Planning subroutines, there road-edges are treated similarly to objects, that is, once potential paths are generated: those paths that pass within a Road-Edge Avoidance Radius of a road-edge are assigned a high safety cost - just like the paths that pass within an object Safety Radius of an object. Paths that are adjacent to paths that pass within a Road-Edge Avoidance Radius of a road-edge are also assigned a (lower) safety cost through the use of a Discrete Gaussian Convolution over a collision matrix [4, 59]. This should result in potential paths that cross (or come too close to crossing) the road's edge being discarded, and only paths that remain safely on-road being selected. This will allow the density of waypoints required for an on-road drive to be greatly reduced.

In fact with road-edge avoidance enabled, the vehicle would only require waypoints at intersections and would otherwise follow the road, rather than require any curved path to be manually mapped out densely with waypoints as was the case previously.

Lane-Keeping

One limitation in the new road-edge avoidance algorithm described above is that the vehicle will choose paths without regard for lanes, that is, it will drive on the right or left hand side of the road whichever it decides is closer to the 'base frame' (ideal path to the next waypoint). While this is not an issue within the scope of this project – as our vehicle is intended for race conditions (i.e. single lane, one-way driving) – a potential solution exists by way of the calculation of an additional cost, implemented into the total cost function for candidate path selection in the Advance Path-Planning subroutines, is proposed.

The cost-determination algorithm runs as follows: firstly, before starting the autonomous driving a variable is set to determine whether the vehicle should stay to the left or to the right of the road. Then at each path planning instance each of the generated candidate paths are checked for collisions with each road segment (where a road segment is defined by the line between a pair of road-edge coordinates). For each collision, the point of collision is compared to the 'ideal' collision point – which is the point that is a particular distance (Road-edge Offset, tuneable in the source code) along the road segment from the road-edge on the side to which the vehicle should be keeping (as depicted in Figure 5 on the next page).

Then the average of both the maximum distance from an ideal point from among all of its road segment intersections and the mean distance from ideal points from each intersection give the cost for each potential path. This cost should favour paths that closely follow the designated side of the road. The averaging of the maximum deviation and the mean deviation is to prevent the choosing of paths that deviate wildly for only one or a few segment intersections, as such a path might otherwise have a reasonably low mean deviation.

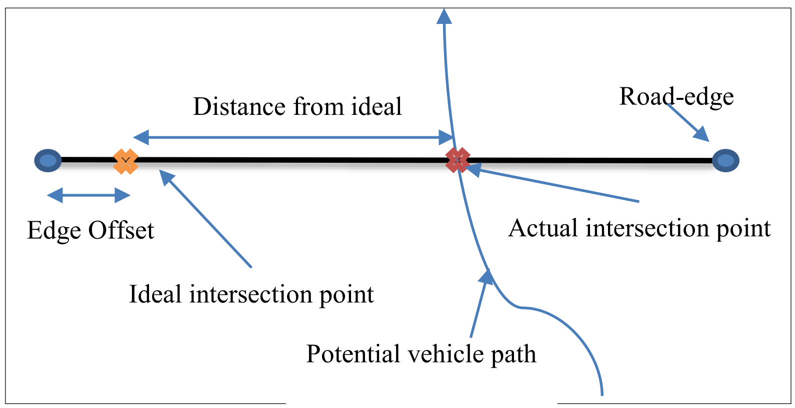

Figure 5: Lane-Keeping Cost Calculation Diagram

Results

Originally it was intended that each of the improvements to the vehicle were to be tested by simulation before being rigorously tested on-vehicle. However, due to significant issues with the vehicle due to a variety of factors - from hardware faults, sensor failures, damage to vital components due to other groups borrowing parts of the vehicle, and software bugs - the ability to test any of the improvements on vehicle was extremely limited. As such, the results presented below are mostly derived from simulations.

Multi-Layer Road-Edge Detection

The road-edge detection was extended across all four scan layers, an example is depicted below in Figure 6. The implementation used allows for road-edges to be detected along any arbitrary number of incoming scan layers by a simple modification to a variable in a header file, changing a variable to the desired number of scan layers.

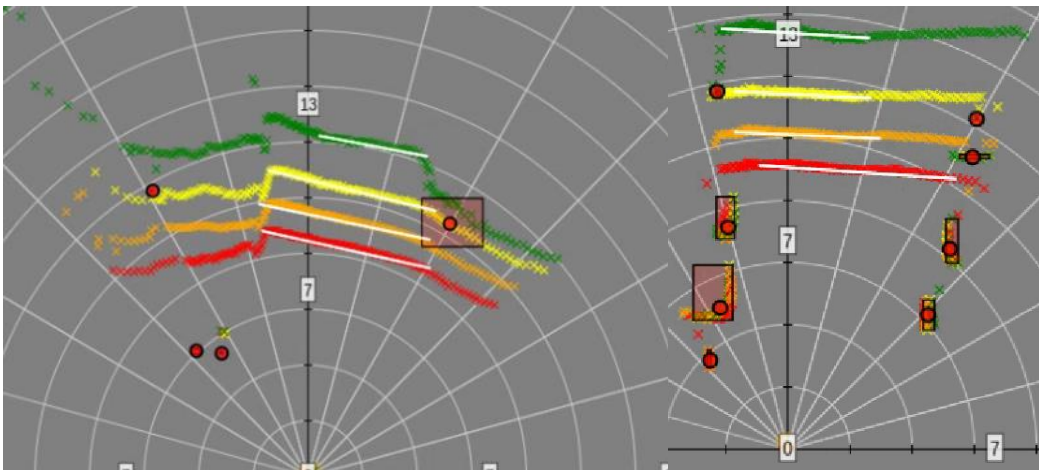

Figure 6: IBEO LiDAR Scan Data, showing road segments (in white) detected across all four scan layers. Distances in metres. Each layer of scan data is colour coded.

Attitude Correction and Mapping

A drive at very low-speeds (for safety reasons) was performed from the Automotive Laboratory at the Electrical and Electronic Engineering building to the oval adjacent to the Business School building and back again, and data was collected. The mapped road segments for this drive are displayed overlaid on a satellite image in Figure 7, it is evident from the overlay that the mapping is reasonably accurate with only the occasional misplaced road segment off to the sides – most likely due to the heavy volume of pedestrians walking alongside and in front of the vehicle.

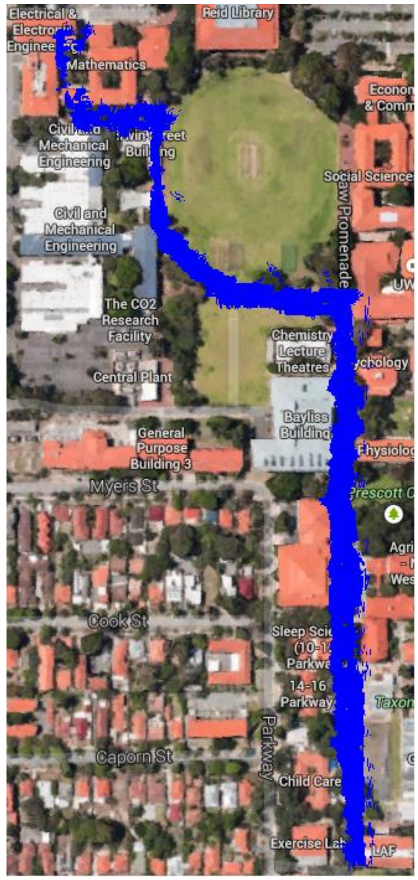

Figure 7: Detected, Adjusted and Mapped Road Segments overlaid on Google Maps Image

The incoming raw scan data was logged, allowing for a comparison between the accuracy of the road-edge detection with and without vehicle attitude adjustment. While the failure rate (the number of scan layers for which no road segments were detected) was very low in both cases, when the data was adjusted for the vehicle's attitude there was a 1.26% reduction in the failure rate. This is still a sizeable improvement considering that the surfaces driven over were paved, flat surfaces - that is, ones for which the vehicle's attitude would not vary significantly. It is therefore reasonable to assume that the accuracy of the attitude adjusted algorithm would further outperform the standard algorithm on rougher terrain, which unfortunately - due to time restrictions, the vehicle was not able to be tested over.

Road-Edge Avoidance

The road-edge avoidance algorithm was tested through several simulations run with the actual code in MATLAB.

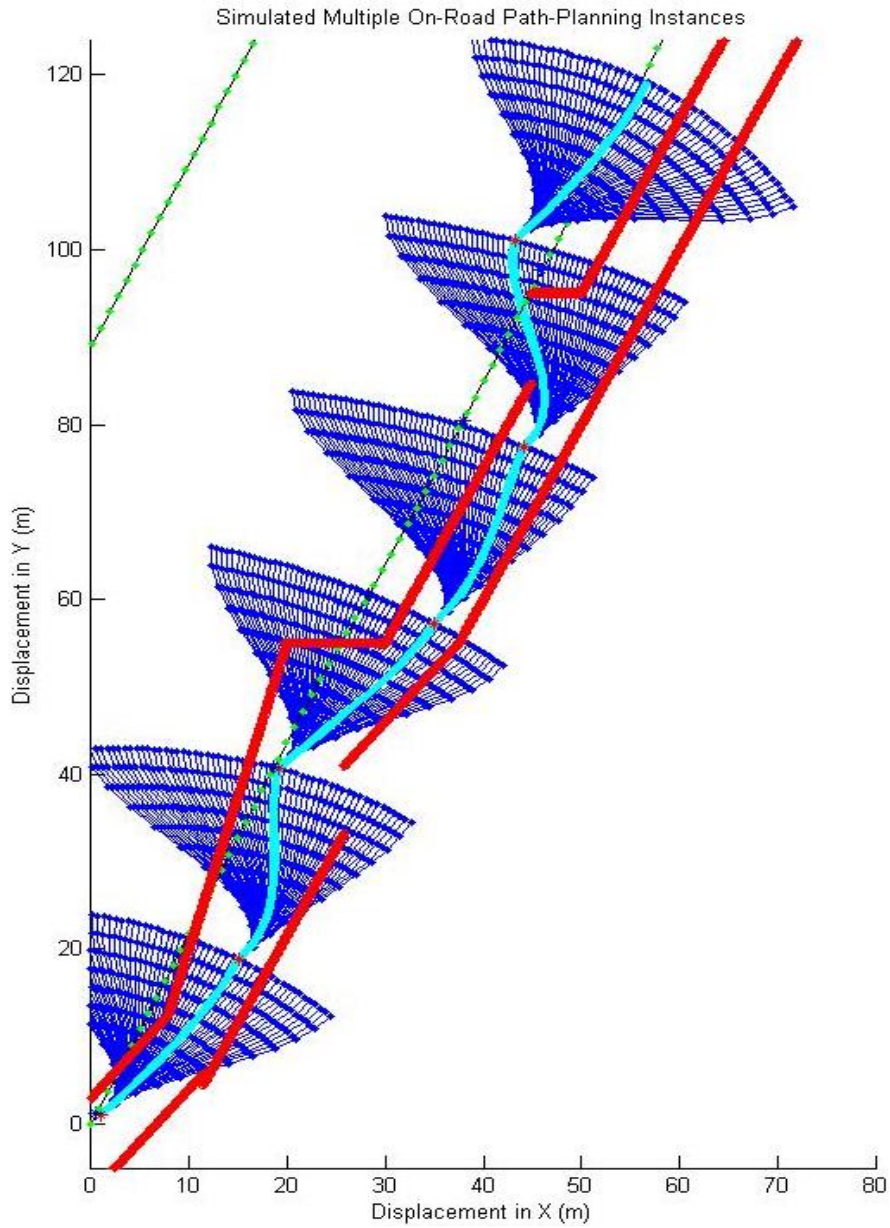

Figure 8: Multiple simulated path-planning instances. Potential paths in blue, selected paths in cyan, detected road edges in red and base frame in black and green.

Note that in the simulation depicted in Figure 8 the selected path leaves the road when there is a gap in road edges (to simulate an intersection) that allows the vehicle to more closely follow the ideal path (base frame), but not when there is a gap that would take it further away. This implies that the vehicle will choose whichever exit from an intersection is closest to its ideal path to the next waypoint, but the vehicle is unable to consider whether or not that path will take it to the next waypoint in the long term.

It was determined through simulations that the Road Edge Avoidance Radius had to be set to a value at least that of the granularity (the distance between the points along each path that are checked for collisions) used by the path-planning algorithm; otherwise it was possible for a potential path to go straight through a road-edge but have none of the checked points lying within a road-edge avoidance radius of the road-edge, as is displayed in Figure 9.

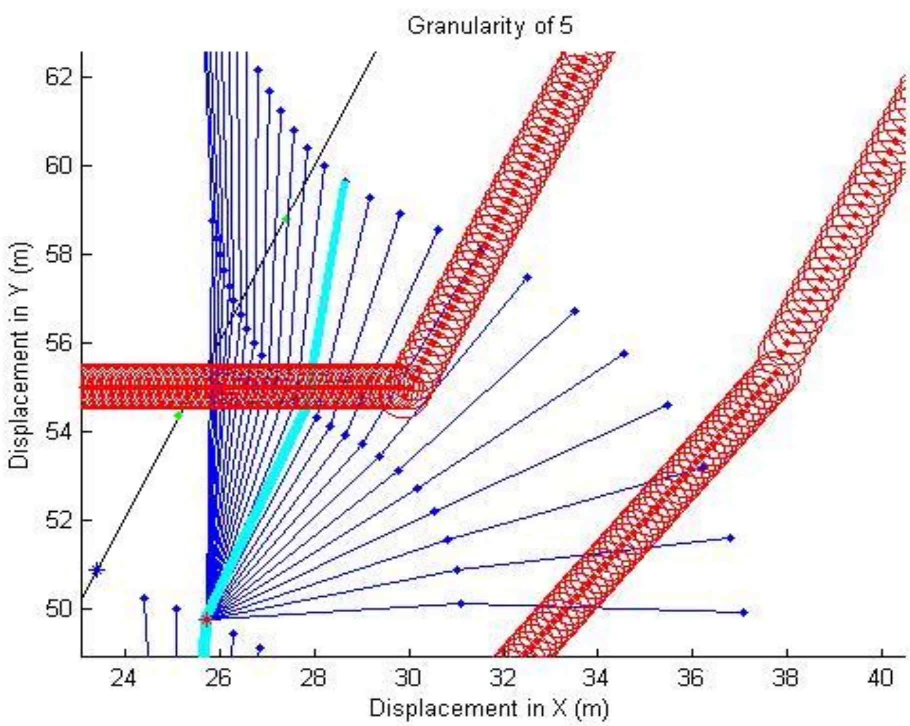

Figure 9: Simulation with high granularity

There was also the case whereby a potential path slightly intersects the radius around a road-edge and as such was not determined to have a 'collision' with the road-edge, the choosing of this path and paths like it is prevented if the Path Estimate (or Manoeuvre) Granularity (the lateral distance between potential paths) is sufficiently small, then the slightly intersecting path should have neighbouring path(s) that intersect the road-edge and due to the use of a Discrete Gaussian Convolution over the collision matrix in determining the safety cost of paths, this would result in this slightly intersecting path receiving a (relatively) high safety cost, preventing it from being chosen. It is demonstrated in the simulation depicted in Figure 10 that the path-planning will favour paths that are not adjacent to paths that intersect with a road-edge.

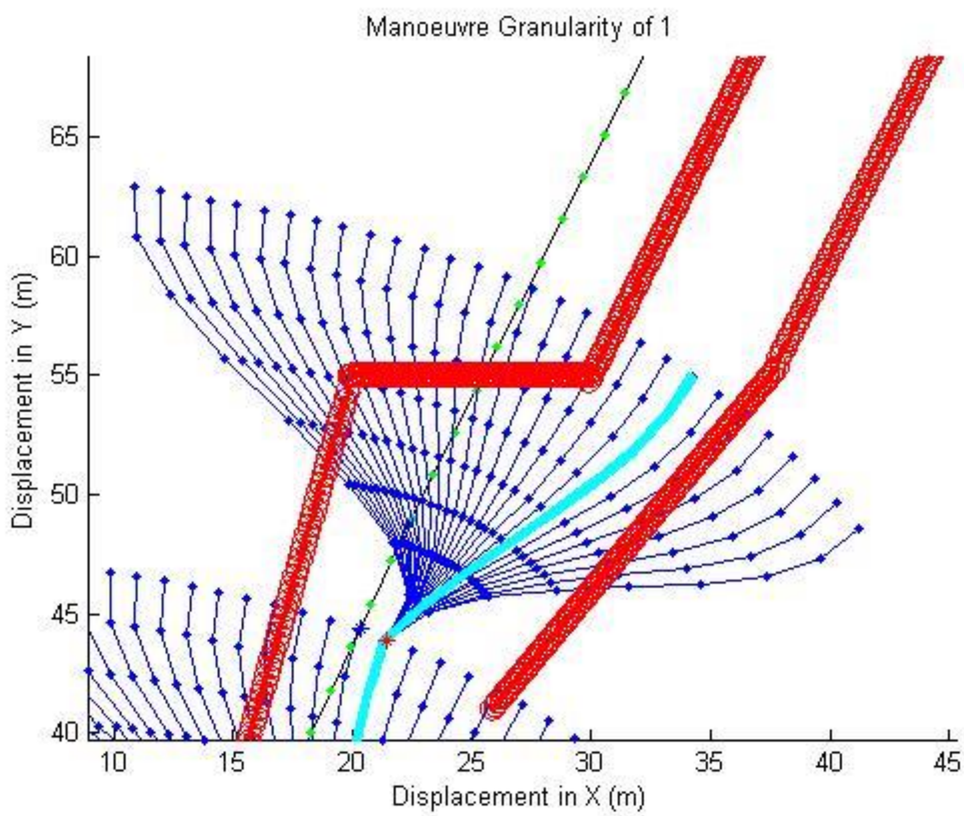

Figure 10: Relatively small manoeuvre granularity

Lane-Keeping

An implementation of the Lane-Keeping algorithm described in this paper was coded and tested alongside the road-edge avoidance algorithm, however the particular implementation that was used proved to be prohibitively expensive in terms of time taken to calculate the costs relative to the standard road-edge avoidance algorithm and it was not as accurate as would be necessary in real life urban environments.

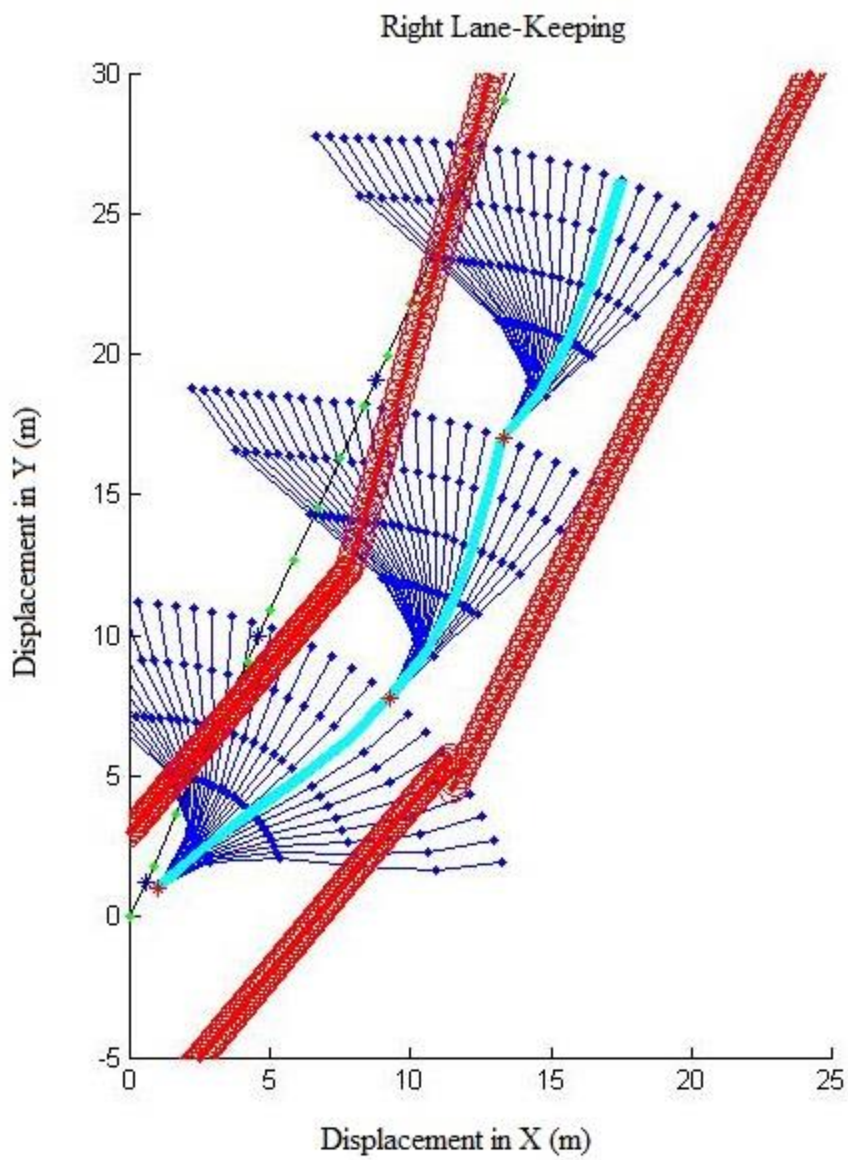

Figure 11: Example of Lane-Keeping algorithm, attempting to keep right

Conclusion

Discussion of Results

The project has achieved its goal of improving the vehicle's road-edge detection through the extension of road-edge detection across multiple scan layers with vehicle attitude adjustment, and has shown that it can be successfully applied for the mapping of traversed road-ways. The vehicle is now also able to autonomously drive itself without deviating from the road - except at intersections marked by gaps in the roads edge, and now no longer requires tediously entering waypoints along the length of every curved section of road.

Unfortunately, the implementation of the Lane-Keeping algorithm that was tested in this project was found to be both too time expensive as well as not accurate enough for integration in to the vehicle. Optimisation and further development of this algorithm is unfortunately now outside this scope of this project due to time restrictions but there is sufficient scope for a future student to build on this algorithm.

Overall, the vehicle is now more accurately able to navigate when driving on roads or paths, this allows for significant research to be explored in new areas regarding autonomous driving in urban environments as well as the high-performance and competitive environments for which the car is suited.

Future Work

In terms of potential research extensions to the road-edge detection ability of the vehicle, the extension of the algorithm to intelligently make use of the multiple scan layers through the use of a planar regression instead of the linear regression currently used. This would allow for the detection of 'road corridors' rather than road segments and could allow for overall more accurate mapping of the road.

In order for the lane-keeping algorithm to become feasible to operate in real time it would be necessary to replace the hardware on which the vehicle's control systems operate (at present, a Raspberry Pi 2), one potential hardware solution would be the use of a GPU (graphics processing unit) as "GPUs are optimized for taking huge batches of data and performing the same operation over and over very quickly…" [60], such operations form the majority of the vehicle's control software, from incoming LiDAR data analysis and the road-edge detection algorithm to the vehicle's path-planning algorithm. The modification of the source code to operate in such a hardware environment would likely be a significant undertaking and would be of sufficient scope for at least one engineering student's final year project.

There is also the potential for the inclusion of visual sensors such as cameras as these are recently seeing increased prevalence in autonomous vehicles. This would potentially allow for research into the creation of a road-edge detection algorithm based on a fusion of camera and LiDAR data. In fact, Oxford University uses a fusion of camera and LiDAR data for vehicle localisation, as well as Object Detection [61], a similar approach could be integrated in to the vehicle's road-edge mapping to allow for maps of sufficient quality to allow for the use of SLAM (Simultaneous Localisation and Mapping) algorithms, reducing the need for GPS which are relatively inaccurate. In addition, there has been significant research undertaken in using visual sensors to detect lane-markings, some research even being completed within UWA [62] this research could be implemented onto and incorporated into the vehicle and its control algorithms, perhaps in conjunction with the lane-keeping algorithm described in this paper, thus allowing it to lane-keep to a standard necessary for safe traversal of an urban environment.

As of the conclusion of this year, the vehicle's localisation and path-planning have improved, however there is significant scope for improvements to be made to the vehicle's control loops, as at present while the vehicle can accurately determine its present location and a path to get to its desired location, its ability to accurately follow this path is limited. Improvements could be made to the PID control loop (as it is presently not tuned correctly) that controls the vehicle's steering, and several further potential control improvements could be explored.

There is significant scope for potential improvements that could be made to the Advanced Path-Planning algorithms: from changing the length of planned manoeuvres based on the vehicle's speed - to maximise accuracy and efficiency by not over planning while preventing the vehicle from overshooting the planned path before the next path-planning instance; the design and integration of new cost functions more suited to high speed racing environments – such as deciding how best to approach corners; and the design and implementation of modifications to obstacle detection and avoidance so as to allow the vehicle to react to and potentially avoid or overtake moving obstacles such as other vehicles or pedestrians.

References

[1] "The REV Project," The REV Project, 2015. [Online]. Available: http://therevproject.com/videos-photos/videos-photos.php . [Accessed 24 October 2015].

[2] T. Drage, 'Final Year Presentation: Autonomous Control of a Formula SAE Car', UWA, 2013.

[3] "The REV Project," The REV Project, 2015. [Online]. Available: http://therevproject.com/vehicles/sae2010.php [Accessed 24 October 2015].

[4] M. French, "Advanced Path Planning for an Autonomous SAE Electric Race Car" BE Thesis, UWA, 2014.

[5] M. Webster. "Mechanical Actuation and Low-level Control for a BMW X5 Automatic Safety System". The University of Western Australia, 2011.

[6] N. Randell. "Design and Implementation of a Motor Drive for Steering a BMW X5 Vehicle". The University of Western Australia, 2010.

[7] A. Oagana, 'Mercedes-Benz to Introduce Autobahn Pilot Assistant in Two Years - Video', autoevolution, 2013. [Online]. Available: http://www.autoevolution.com/news/mercedes-benz-to-introduce-autobahn-pilot-assistant-in-two-years-video-70731.html [Accessed: 24- Oct- 2015].

[8] S. Lawson, 'Cars will drive themselves in three phases: First, traffic jams', Computerworld, 2015. [Online]. Available: http://www.computerworld.com.au/article/569309/cars-will-drive-themselves-three-phases-first-traffic-jams/ [Accessed: 24- Oct- 2015].

[9] G. Garvin, 'Automakers say self-driving cars are on the horizon', Tampa Bay Times, 2014. [Online]. Available: http://www.tampabay.com/news/business/autos/automakers-say-self-driving-cars-are-on-the-horizon/2171386 [Accessed: 24- Oct- 2015].

[10] R. Hotten, 'Carmakers face challenge from Google and Apple - BBC News', *BBC News*, 2015. [Online]. Available: http://www.bbc.com/news/business-31720645 [Accessed: 24- Oct- 2015].

[11] K. Korosec, 'Elon Musk: Tesla's autopilot is rolling out everywhere (except Japan)', *Fortune*, 2015. [Online]. Available: http://fortune.com/2015/10/23/tesla-autopilot-goes-global/ [Accessed: 24- Oct- 2015].

[12] The Milwaukee Sentinel, ''Phantom Auto' Will Tour City', p. 4, 08-Dec-1926.

[13] J. Schmidhuber, 'Prof. Schmidhuber's highlights of robot car history', Jürgen Schmidhuber's Homepage, 2015. [Online]. Available: http://people.idsia.ch/~juergen/robotcars.html [Accessed: 19- May- 2015].

[14] Mercedes-benz.com.au, 'Mercedes-Benz Australia - September 2013 - Mercedes-Benz S-Class INTELLIGENT DRIVE', 2015. [Online]. Available: http://www.mercedes-benz.com.au/content/australia/mpc/mpc_australia__website/en/home_mpc/passengercars/home/passenger_cars_world/news/september_2013/s-class-intelligent-drive.0007.html [Accessed: 20- May- 2015].

[15] Cs.cmu.edu, 'The Carnegie Mellon University Autonomous Land Vehicle Project (NAVLAB)'. [Online]. Available: http://www.cs.cmu.edu/afs/cs/project/alv/www/ [Accessed: 20- May- 2015].

[16] S. Russell, 'DARPA Grand Challenge Winner: Stanley the Robot!', Popular Mechanics, 2006. [Online]. Available: http://www.popularmechanics.com/technology/robots/a393/2169012/ [Accessed: 21- May- 2015].

[17] Viac.vislab.it, 'Intercontinental Challenge', 2015. [Online]. Available: http://viac.vislab.it/ [Accessed: 21- May- 2015].

[18] Riotinto.com, 'Rio Tinto improves productivity through the world's largest fleet of owned and operated autonomous trucks - Rio Tinto', 2015. [Online]. Available: http://www.riotinto.com/media/media-releases-237_10603.aspx [Accessed: 21- May- 2015].

[19] Riotinto.com, 'Mine of the Future™ - Rio Tinto', 2015. [Online]. Available: http://www.riotinto.com/ironore/mine-of-the-future-9603.aspx [Accessed: 21- May- 2015].

[20] G. de Poloni, 'Rio Tinto to test driverless trains in the Pilbara', ABC News, 2014.

[Online]. Available: http://www.abc.net.au/news/2014-04-28/rio-tinto-to-test-driverless-trains/5415292/ [Accessed: 21- May- 2015].

[21] Faculty.washington.edu, 'Park shuttle automated driverless vehicle pilot project - Netherlands', 2015. [Online]. Available: http://faculty.washington.edu/jbs/itrans/parkshut.htm [Accessed: 21- May- 2015].

[22] R. Burn-Callander, 'This is the Lutz pod, the UK's first driverless car', Telegraph.co.uk, 2015. [Online]. Available: http://www.telegraph.co.uk/finance/businessclub/technology/11403306/This-is-the-Lutz-pod-the-UKs-first-driverless-car.html [Accessed: 21- May- 2015].

[23] Nissan-global.com, 'NISSAN | Nissan Research Center | On the Research Front Lines | Hideaki Inoue', 2015. [Online]. Available: http://www.nissan-global.com/EN/NRC/FRONTLINES/VOL3/#page3 [Accessed: 21- May- 2015].

[24] Edition.cnn.com, 'CNN.com - Toyota unveils car that parks itself - Sep. 1, 2003', 2015. [Online]. Available: http://edition.cnn.com/2003/TECH/ptech/09/01/toyota.prius.reut/index.html [Accessed: 21- May- 2015].

[25] Toyota Australia, 'Toyota Camry - Blind Spot Monitor', 2015. [Online]. Available: http://www.toyota.com.au/camry/features/safety/blind-spot-monitor/ [Accessed: 21- May- 2015].

[26] R. Kiino, '2011 Infiniti M first drive - Motor Trend', Motor Trend Magazine, 2015. [Online]. Available: http://www.motortrend.com/roadtests/sedans/112_1001_2011_infiniti_m_prototype_drive/ . [Accessed: 21- May- 2015].

[27] AutoTrader.com, '2013 Ford Fusion Will Get Smart About Lane Drift Warning - Autotrader', 2015. [Online]. Available: http://www.autotrader.com/research/article/car-news/142395/2013-ford-fusion-will-get-smart-about-lane-drift-warning.jsp . [Accessed: 21- May- 2015].

[28] Volvoclub.org.uk, 'Volvo Collision Warning with Auto Brake', 2015. [Online]. Available: https://www.volvoclub.org.uk/press/releases/2007/collision_warning.shtml .

[Accessed: 21- May- 2015].

[29] A. Kessler, 'Elon Musk Says Self-Driving Tesla Cars Will Be in the U.S. by Summer', Nytimes.com, 2015. [Online]. Available: http://www.nytimes.com/2015/03/20/business/elon-musk-says-self-driving-tesla-cars-will-be-in-the-us-by-summer.html. [Accessed: 21- May- 2015].

[30] ieee.org, 'IEEE News Releases', 2015. [Online]. Available: http://www.ieee.org/about/news/2012/5september_2_2012.html . [Accessed: 21- May- 2015].

[31] J. Seigemund, U. Franke and W. Forstner. “A temporal filter approach for detection and reconstruction of curbs and road surfaces based on Conditional Random Fields”. 2011 IEEE Intelligent Vehicles Symposium, pp 637-642, Jun. 2011.

[32] C. Oh, J. Son and K. Sohn, 'Illumination robust road detection using geometric information', 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012.

[33] P. Shinzato, D. Gomes and D. Wolf, 'Road estimation with sparse 3D points from stereo data', 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014.

[34] J. Li et al., 'A 3-D Real-Time Road Edge Detection System for Automated Smart Car Control', 2006 IEEE International Conference on Networking, Sensing and Control, 2006.

[35] K. Guo et al., 'Road Edge Recognition Using the Stripe Hough Transform From Millimeter-Wave Radar Images', IEEE Trans. Intell. Transport. Syst., pp. 1-9, 2014.

[36] M. Nikolova and A. Hero, 'Segmentation of a road from a vehicle-mounted radar and accuracy of the estimation', Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511), 2000.

[37] W. Zhang. “LIDAR-Based Road and Road-Edge Detection”. 2010 IEEE Intelligent Vehicles Symposium, pp 845-848, Jun. 2010.

[38] B. Fardi et al., “Multi Model Detection and Parameter-based Tracking of Road Borders with a Laser Scanner”, IEEE IV2003 Intelligent Vehicles Symposium, pp.

95-99, Jun. 2003.

[39] W.S. Wijesoma, K. R. S. Kodagoda and A. P. Balasuriya. “Road-Boundary Detection and Tracking Using Ladar Sensing”, IEEE Transactions on Robotics and Automation, vol. 20, no. 3, Jun. 2004.

[40] G. Zhao and J. Yuan, 'Curb detection and tracking using 3D-LIDAR scanner', 2012 19th IEEE International Conference on Image Processing, 2012.

[41] W. Yao, Z. Deng and L. Zhou, 'Road curb detection using 3D LIDAR and integral laser points for intelligent vehicles', The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems, 2012.

[42] R. Li, et al., 'LIDAR/MEMS IMU integrated navigation (SLAM) method for a small UAV in indoor environments', 2014 DGON Inertial Sensors and Systems (ISS), 2014.

[43] J. Han et al., 'Enhanced Road Boundary and Obstacle Detection Using a Downward-Looking LIDAR Sensor', IEEE Transactions on Vehicular Technology, vol. 61, no. 3, pp. 971-985, 2012.

[44] S. Rodriguez et al., 'Visual confirmation of mobile objects tracked by a multi-layer LIDAR', 13th International IEEE Conference on Intelligent Transportation Systems, 2010.

[45] P. Lindner et al., 'Multi-channel LIDAR processing for lane detection and estimation', 2009 12th International IEEE Conference on Intelligent Transportation Systems, 2009.

[46] S. LaValle, Planning algorithms. New York, NY [u.a.]: Cambridge University Press, 2006.

[47] Y. Liu and Y. Sun, 'Mobile robot instant indoor map building and localization using 2D laser scanning data', 2012 International Conference on System Science and Engineering (ICSSE), 2012.

[48] S. Jia et al., “Mobile Robot 3D Map Building Based on Laser Ranging and Stereovision,” Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation August 7 - 10, Beijing, China.

[49] M. Fu et al., 'Multiple map representations for vehicle localization and scene reconstruction', 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014.

[50] D. Cole and P. Newman, 'Using laser range data for 3D SLAM in outdoor environments', Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, 2006.

[51] C. Brenneke, O. Wulf and B. Wagner, 'Using 3d laser range data for slam in outdoor environments', Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), 2003.

[52] S. Thrun et al., 'Stanley: The robot that won the DARPA Grand Challenge', J. Field Robotics, vol. 23, no. 9, pp. 661-692, 2006.

[53] M. Montemerlo et al., 'Junior: The Stanford entry in the Urban Challenge', J. Field Robotics, vol. 25, no. 9, pp. 569-597, 2008.

[54] S. Kuthirummal, A. Das and S. Samarasekera, 'A graph traversal based algorithm for obstacle detection using LIDAR or stereo', 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.

[55] H. Zhu et al., 'A path planning algorithm based on fusing lane and obstacle map', 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014.

[56] H. Chen and M. Dong, '3D map building based on projection of virtual height line', APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, 2008.

[57] J. Gillula and J. Leibs, 'How to teach a van to drive: an undergraduate perspective on the 2005 DARPA grand challenge', IEEE Control Systems Magazine, vol. 26, no. 3, pp. 19-26, 2006.

[58] G. Welch, G. Bishop (2006, Jul.). “An Introduction to the Kalman Filter” [Online]. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[59] C. Keonyup, L. Minchae, and S. Myoungho, "Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles," Intelligent Transportation Systems, IEEE Transactions on, vol. 13, pp. 1599-1616, 2012.

[60] A. Hesseldahl, 'FTC Presses On with Intel Probe', *Businessweek.com*, 2009. [Online]. Available: http://www.bloomberg.com/bw/technology/content/dec2009/tc2009122_478796.htm [Accessed: 25- Oct- 2015].

[61] J. Condliffe, 'The Reason We Won't Have Autonomous Cars Any Time Soon', *Gizmodo*, 2015. [Online]. Available: http://gizmodo.com/how-to-teach-an-autonomous-car-to-drive-1694725874 [Accessed: 25- Oct- 2015].

[62] J. Fong, 'Final Year Presentation: A lane detection algorithm for use with vehicular CMOS camera system-on-chip', UWA, 2015.