A+ Hardware Testing & Critique

Do Not Remark from CO145

Lab: Autonomous Vehicle Challenge

Name: [redacted]

Student Number: [redacted]

Team Number: Thursday 14:10-16:00 Team 1

Team Name: The Karen Project

Abstract

In April 2015, The Karen Project (TKP) began the task of creating an autonomous vehicle capable of following a line, navigating a line maze, and then solving a walled maze with moving obstacles. The robot was also to meet other design and project management criteria to fully satisfy the brief given by the ENGR101 AVC.

The ENGR101 Autonomous Vehicle Challenge (AVC) is designed to replicate a real-world engineering project. The key principles behind this replication include being placed into a team randomly, working to a deadline and budget, and meeting a set of vital, non-negotiable requirements. It also includes documenting the project. These factors are the primary motivation for this challenge, as it forces teams and individuals to manage the project as though they are working where such constraints exist. It is vital that all engineers are able to work in a team where roles are delegated and team benefit comes before individual preference.

Developing documentation skills is an often overlooked but critically important part of preparing for the engineering workplace. Future graduates need to understand the environment they will be working in before they are in it, and this project provides this experience and understanding.

The Karen Project was unable to fully complete its AV before the deadline, satisfying just 45% of the intended specifications, and this report discusses the key issues which caused this failure. Poor delegation and accountability, and a lack of sound planning were two of the main contributing factors. Many aspects of the robot were well designed, but the overall performance of the device was hindered by bad practices in the engineering process used. There were also a number of factors which, despite not directly contributing to the eventual performance issues of the robot, could be improved in future projects of a similar nature; such as more comprehensive time-based planning and communicating the strengths and weaknesses of team members more honestly. Team members from The Karen Project learned a lot from this experience, and if the process was undertaken again there would certainly be a greater ability for the robot to be completed fully with the time and resources available. This report also looks into the design of the robot.

Introduction

This report focuses on the engineering approach taken by The Karen Project to complete the Autonomous Vehicle Challenge set out in the ENGR101 course at VUW. It is written to aid in understanding the strengths and weaknesses of the approach, and develop a conclusion about what could be improved in future to make the outcome more ideal. Some of the specific topics analysed are The Karen Project's methodology for splitting the workload between team members, ensuring that all team members are working to the best of their ability and contributing, and monitoring the quality of work produced by team members. Beyond this, it also looks at the design of the robot itself; what worked and what didn't, how design decisions were made and on what basis. Most importantly it analyses how each of these factors contributed to the outcome of the whole project.

The motivation behind the challenge is to develop skills which are of value in the workplace, and "to give (a) taste of what engineering projects are in real life". Part of the B.E.Hons course is work placement, and the experience of teamwork to satisfy a prescribed engineering goal is greatly beneficial to all of the students who are undertaking this degree. In addition to this, the challenge develops attributes and approaches which apply to future coursework. The writing of this report is a key part of this continual development, as it encourages focusing on what made the project successful or, in the case of The Karen Project, unsuccessful.

Although the aim of the challenge itself is to create a robot which fulfills the requirements set out in the project brief, its greater goal is acquiring knowledge about the engineering and technology process which can be applied to all kinds of future work, as described above. This is greatly beneficial when undertaking any future coursework or employment in the field of engineering. The knowledge of what worked well and what didn't should allow team members to further their own understanding of contributing to an engineering team.

Background

The design brief provided is the key source of information regarding the AVC, and outlines the rules and guidelines of the challenge, primarily giving the specifications which the robot should meet. The main points of the design brief are listed below:

- You work in a team that is not of your own choice.
- You have a deadline (week 11 of trimester 1)
- Your budget is limited (Parts Bazaar $100)
- Your robot should stay within the range and go from start to the finish position.
- The testing range is 2x2 m square made out of 4 quadrants.
  1. Follow the white line (on black background)
  2. Solve the white line maze
  3. Solve maze with walls
  4. Solve maze with walls and obstacles
- Points are awarded based on how far your robot went through the night range.
- Robot should be positioned at start line when testing commences
- You have 15 minutes of customer attention.
- Robot should be totally autonomous and no human intervention is allowed.
- You can take the robot back to the start line. 15 minutes timer keeps running.
- 20 points are given for accurately reporting robot position over wireless link to the basestation
- Robot should not change range as it moves through

Additionally, points can be removed for lack of recyclability, poor aesthetic design and failure to complete weekly blog posts discussing individual contribution to the project.

The Karen Project consisted of five team members randomly selected from the Thursday 14:10-16:00 lab section.

This report does not focus just on the design of the robot. While the design itself will be covered, it would be unwise to ignore the non-design factors of the approach to this challenge which caused a poor performance at the end of the project. Instead, this report focuses on how the project went on the whole, while also containing a high level of detail about the robot.

Method

The paragraphs below discuss the personnel and design decisions made about the autonomous vehicle by The Karen Project.

Design and Procedures

The first steps taken were to assign roles to team members so that everyone would be involved in the build. The initial roles were two programmers, one primary and one secondary; two designers, one primary and one secondary; and one member who would do coordination work and manage the project to some degree. The team made these decisions based primarily on individuals' experience, and when little experience was available (such as with CAD), individual preferences and interests came forth. These roles were not strictly set out, but rather aimed to keep members working on the correct part of the project and distribute labour evenly across tasks.

A prototype was produced using a CD as a base, and some pre-made motor mounts. This allowed for an instant ability to write and test code with a very similar

The team then made some basic design decisions to base the robot on:

- Wheels vs tracks
  - Wheels were chosen for their increased maneuverability over tracks, and lack of sufficient reason to choose tracks (no limits on ground pressure). Wheels also require less power to complete turns as there is less friction.
- Robot shape (Figure 1)
  - A two-layered design allows width and length dimensions to be kept to a minimum, while still allowing all components to fit without excessive modification to compress the design. This design also allows for easy access to all of of the components.
  - Rounded corners minimise snagging on walls.
  - Wheel cutouts reduce surface area of the robot, allowing it to make it through the narrow corner gaps between walls.

Figure 1: 3D Model of baseboards with motor mounts

- Sensors
  - A QTR light sensor is necessary regardless of anything else, as this is the component which allows for line detection.
  - Distance sensors for side walls, able to sense whether something is a short distance away from the robot, useful for the walled parts of the course
  - Lever switches for the front of the robot, able to sense if the robot physically comes into contact with a wall or obstacle.
- Miscellaneous
  - Large rectangular cutouts in the upper baseboard which would allow plenty of options for cable routing and the cable ties for sensors.
  - All of the necessary screw holes pre-cut in the CAD drawing.

- Pre-made M3 Support pillars to hold the upper baseboard above the lower board.
- Motor mounts designed to screw into lower baseboard with M3 screws.

Based on these decisions all of the necessary parts were 3D designed by the primary mechanical designer and confirmed by the secondary designer.

The motor mounts were 3D printed and the baseboards laser cut at VUW. All parts were sourced from VUW. The whole robot was then able to be assembled. 3D printing was chosen for motor mounts as it is relatively fast for small parts (~40mins) and the lesser strength of parts is not an issue. The baseboards, conversely, needed to be very large (but thin) and strength is more important for these parts.

The AV itself is powered by a 6.6V Li-Fe battery. This provides a very large battery life ideal for long periods of testing. The battery is secured into the robot by means of two screws, which act as posts, and a rubber band going around the battery and the screws The brain of the robot is an Arduino Uno board, which includes a number of pin header inputs and outputs directly connected to a microcontroller. An onboard USB port allows for direct connection to a computer for programming. As the Uno cannot provide enough output current to drive motors, a Vicmoto motor driver board shield was used to interface between the motors and the Arduino. This board also provided power inputs for use with the analog distance sensors. A Zigbee transmitter shield was used to transmit the quadrant information to a nearby PC terminal with a Zigbee USB receiver attached.

A Gantt chart was produced to guide the project in terms of time, as shown below (Figure 2).

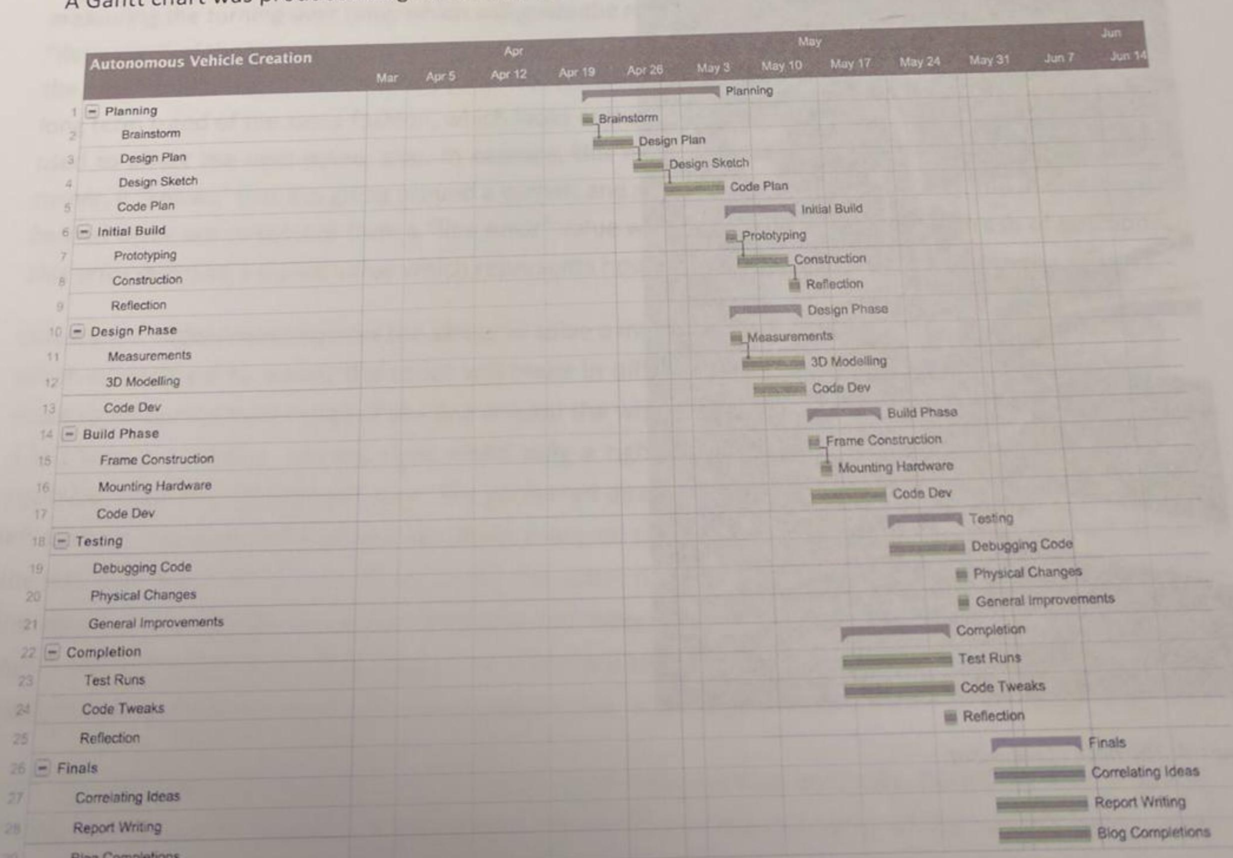

Figure 2: Gantt chart of expected progress by time.

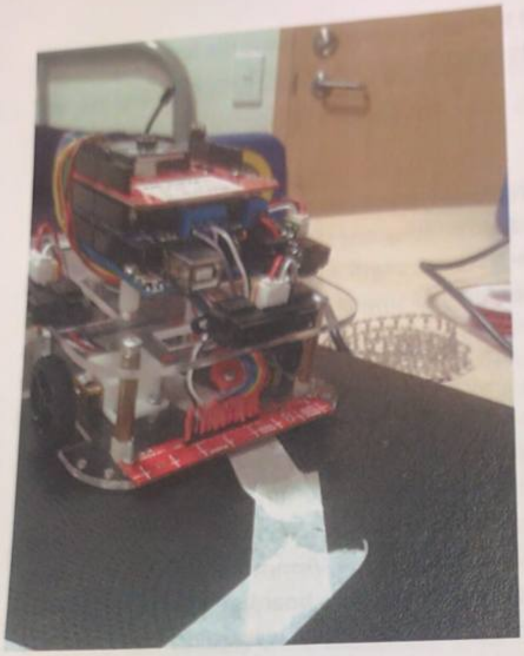

Figure 3: The front of the robot.

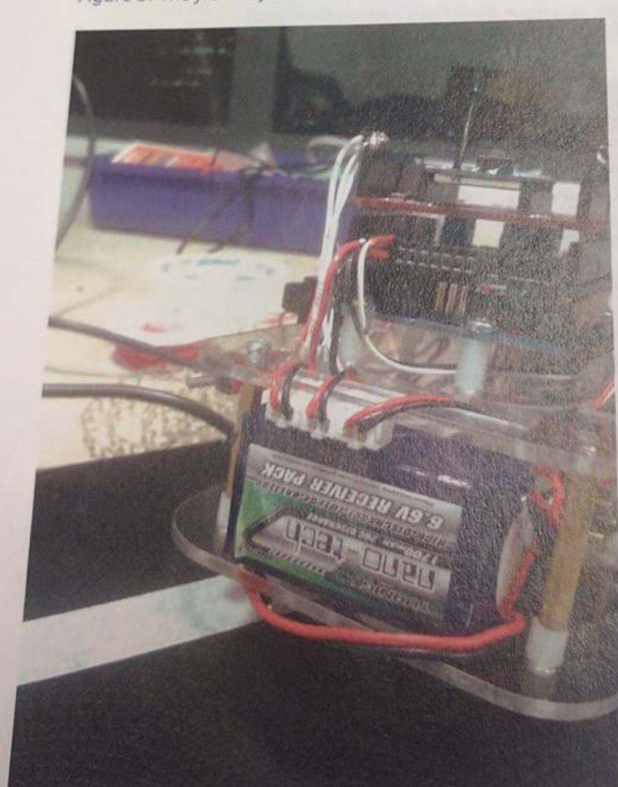

Figure 4: The back of the robot.

Programming

The program design of the robot is potentially the most vital part of the whole system, as it ties together the operation of the robot with its mechanical design. Even the smallest errors in code can cause big problems in operation.

The Karen Project's primary programmer wrote the majority of the code which the robot ran. Some parts of it, namely the Zigbee transmission methods, were written by the secondary programmer. The robot's code is written in the C language (which just has some simple modifications for Arduino).

The code is based around a section for each of the four quadrants which implements the necessary checks. The loop begins by checking which quadrant the robot is known to be in, and then calling the update state method which is relevant to that quadrant.

In quadrant 1 and 2 where there is a line, the measurements which the robot makes are just the values from the line sensor. There is a library file which aids in these making these measurements and returning a state to the program regarding the position of the line. It is more simple to make decisions (if statements) in the actual program code based on a human-interpretable description of where the line is, eg. isTapeLeft and isTapeRight rather than 8 raw sensor values. The most important attribute of the robot necessary in these quadrants is that the robot must be able to follow the line in an efficient manner. Simply checking where the tape is, moving forward, checking again and compensating is not an efficient method of doing this. For example, this makes turning painfully slow as the robot violent swings from left to right. Rather than this, it is advisable to implement a way of measuring the turning over time, which will guide the robot's next move. This is done in the form of a "derivative" of the turning speed which means that if the last move was a left turn, the robot will bias the next move towards the same left turn. This enables smooth cornering. An integral also achieves a long term trend of the same fashion, which looks at the change over a large period of time and this is used to affect the next move, also. In essence, this means that when the robot is going around a corner, it "knows" that it is going around a corner, and not simply making a random jump in direction. Both of these are calculated from a "line error" value which is measured at every refresh of position. This error refers to a signed value which represents how far the line is from the center to the sensors.

Quadrant 2 additionally requires the ability to solve a maze, which is achieved by a set of if statements which correspond to moves the robot will make in different situations. The basic idea is to get the robot to follow the outer edge of the line around the whole maze. This involves turning left when only a left turn is available, turning right when only a right turn is available, and when two turns are available, making a "preferred" turn. The preferred direction used by The Karen Project's code was a left turn, as this provided the best results in practical testing (due to the nature of the course). If no line was found (the robot was off the track), the robot performed a half turn (180°) to try and find the line again). In quadrant 2 the error derivative and integral are not used, because all lines are at right angles to each other and so the only turns that need to be made are in multiples of 90°. Getting the robot to turn exactly this amount is not a simple task, however.

Some clever programming was used to enable consistent turning amounts, based on motor speed, the ratio of electronic signal to motor speed (Figure 5) and the diameter of the wheels being used. Using all of these factors it is possible to make precise turns. This means that changing the battery voltage (such as a battery replacement) does not alter the turning performance of the robot.

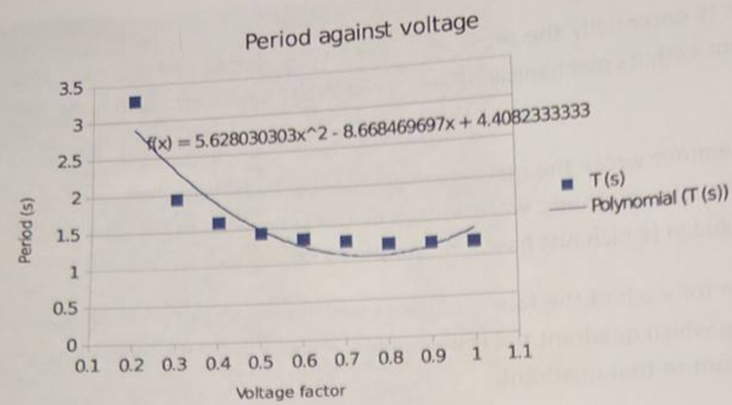

Figure 5: Describing the relationship between motor speed/turn duration and the applied voltage (relative to max level of 1)

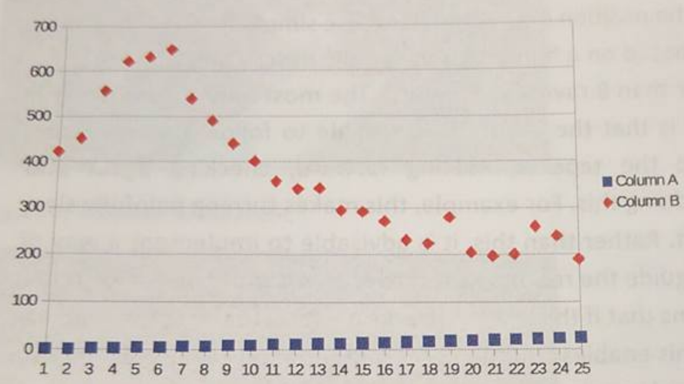

Figure 6: The output of analog distance sensor for a range of distance values

The above graph allowed code to be written which determined a "critical" distance between the robot and the wall at which point the next move would need to respond and move away from the wall. This is the method which the robot uses to navigate through quadrants 3 and 4. Much the same approach using the error of the distances to side walls is used, and if there is a wall ahead and walls on both sides, the robot turns around completely.

Some of the other features implemented after all of the original design considerations are an additional physical speed control (which alters the maximum motor speed), and a power switch (which simply disconnects the battery). These were simple and effective ways to increase the functionality of the robot, and make testing easier.

Testing was accomplished both in the lab on the actual course and on temporary courses with similar characteristics built by the primary programmer in his flat. Testing involved letting the robot run through a course and optimising the code to gain better results. Common modifications to the code were changing the differential and integral coefficients to optimise the smoothness of turning without causing the robot to under-compensate and go off the track. Most testing was completed by the primary programmer, as they had the most comprehensive understanding of the code and how it would need to be modified. The majority of testing was done in the primary programmer's flat.

Results

The Karen Project's autonomous vehicle was not able to complete the specified course in under 15 minutes. During the allocated time slot, changes were made to the programming in order to get the robot to go past the first quadrant without turning around.

At first, the robot reached the perpendicular white line at the end of quadrant one, and then turned around and travelled back towards the start line. This happened a number of times even with code changes which were intended to allow it past this line. The code modification regarded increasing the delay time after it sensed the white line before it treated it as a corner, rather than a marker line. This was not successful, and only produced very unusual behaviour from the robot, such as turning on the spot and/or driving off the course entirely.

Ultimately the robot was able to get past this line and begin to solve the line maze. This was achieved by forcing it to make only right turns after the first left turn. Unfortunately this also meant that it would not be able to solve the maze. It was a decision made out of desperation rather than a proper fix; a kludge. This meant that the robot was only able to complete approximately half of quadrant two, and so was awarded 45% of the available marks.

The other requirements of the robot were met to a high degree. All parts were either screwed or cable-tied to the structure, resulting in a high level of recyclability. The Zigbee transmitter worked correctly and was able to send location messages to a nearby receiver. Aesthetically the AV was tidy, with all wires neatly placed and routed.

The Karen Project conducted a post-assessment team meeting to determine some of the factors which contributed to the poor performance of the team, and these are documented in this report.

Discussion

Despite a high standard of technical engineering on many levels, from the code quality to the mechanical design of the robot, The Karen Project's AV was let down by a lack of understanding. There was confusion and lack of communication regarding the rules of the challenge, the requirements, and the individual progress of each team member. There was also no proper testing routine, which would would have solved the issues with the robot before the day of performance.

Overall, the mechanical design of the robot proved to be excellent. Aesthetically, the robot was not a mess of tangled wires, but all cables were routed purposefully and methodically to allow maximum access to all parts of the robot. The 3D modeled parts allowed the robot to be quickly assembled, and all parts were able to fit together. Only one error occurred in this modeling, due to the unfortunate miscalculation of some values originally given in inches and needing to be converted into mm. This meant the line sensor would not screw directly into the lower baseboard, and new holes had to be manually drilled after the laser cutting. Something that should have been more carefully considered beforehand was the line sensor's distance to the ground. The documentation on the sensor was consulted whilst building the prototype, but the outcome of this (that the line sensor would end up far too high off the ground) was not considered when building the actual robot. Ultimately, a small spherical wheel was used to hold the rear of the robot above the ground, and thus the sensor at the front was closer. This could have been decided earlier on in the design, however, and it would have been easier to implement (with screw holes and a small cutout, for example). In terms of the physical design, the AV was everything it was intended to be.

The code for the first quadrant worked perfectly and produced a very good outcome (a smooth traversal of the course), the Zigbee transmitter worked, and the speed control was a useful part of the robot. It was only coming into the second quadrant that the code failed and produced unwanted results. In terms of the code, it is not clear why the robot did not function as intended.

This, however, does not put the blame on the robot. In fact, it actually reveals a lapse in the attention of The Karen Project – proper testing procedures. While it is fine for code to not work initially in projects, as there are most often going to be bugs that haven't been fully resolved by the first test, it is not okay for bugs to go unnoticed and unresolved in the long run. The fact that such huge bugs were not addressed before the day of performance indicates that testing procedures had not been sufficient. It also indicated that there had not been sufficient support given to or received by the primary programmer.

Looking back over the project, The Karen Project neglected to initially decide on when or how a fully working model would be decided. In essence, there was no deadline for a fully functional model to exist, and this proved to be a large downfall. For example, the primary programmer was the only one who really knew how the code worked, and as such was the only team member testing, observing, and making alterations based on those observations. It turned out that the majority of testing did not occur on the actual course, but instead on made up courses. The programmer had taken over the code entirely in the last weeks of the project, and there was no way of the team helping. This issue is one of accountability; making sure that each team member is actually doing what they say they are doing, and similarly monitoring the quality of their work. It is almost certain that if another team member had stepped in alongside the primary programmer, the issues with the code could have been resolved and proper testing could have occurred before the demonstration. Even during the demonstration, code was still being altered, and this was only able to be completed by the primary programmer. In summary, the accountability of team members is paramount to making sure every task gets the attention it deserves, and testing needs to be defined at the start of a project. That doesn't mean every test should be fully defined before prototyping. It means testing needs to be specified throughout the project, with certain deadlines which must be met, and there needs to be more than one person validating that testing and performance is correct.

The initial delegation of roles within the team proved to be a far more important issue than was expected. Without properly defined roles at the beginning of the project, team members will not get the correct work done. The Karen Project found it difficult to split up the workload well, and this meant that throughout the project some team members were not working to their maximum efficiency, while a couple of team members, such as the primary programmer, were working far beyond what an individual member should have to. This could have been alleviated if all of the work had been set out before the project. Similarly, individuals in the team should have made their individual strengths and weaknesses more apparent to the rest of the team for the purposes of this delegation.

A very relevant mistake to the poor delegation that occurred in the challenge was not reading the rules and guidelines before starting work on all of the challenge. No one in the team was aware that such rules existed, but it was still poor practice to not obtain a full written outline of what was to be achieved. This caused a number of flow-on issues:

- The testing date was thought to be over a week later than it actually was, and so the sense of urgency was not as high as it should have been.
- The requirement for blog entries and a Gantt chart was not understood, and so these were not fully produced.
- The work to be done was not fully known at the time of assigning work to each team member, and so some members did far more or less than others.

Eventually, the rules were found and some of the issues were able to be fixed, but ultimately the team was left with significantly less time than was originally thought, and there was far less documentation than required by the challenge at this point. The Gantt chart, completed retrospectively, also highlighted just how far behind the team was compared with an ideal scenario.

Ultimately, all of these issues can be summarised by a complete and overriding point: communication. Communication between client and engineers, communication between team members, communication between team and governing body (staff). All of these are important because they ensure that

1. The team knows what they are aiming to achieve, when it is needed, and how they will make sure they do it.
2. The client knows the engineers understand the problem in enough depth to solve it.
3. The higher engineers know the team is capable of solving the problem in a technically and ethically correct fashion.

The Karen Project failed to address any of these points, and ultimately it led to a disappointing but unsurprising outcome. Within the team; members didn't ask for help when they needed it, some team members worked tirelessly and others barely worked at all, and everyone assumed without basis that the project was being completed and going on without them. Some of the most vital aspects of the project were left to just one person, such as the coding, while small tasks were inefficiently handled multiple people. But it was never discussed or highlighted, and so the whole project fell apart.

If completing the project again, The Karen Project would put a lot more emphasis on communication, planning and testing. There are some very important lessons learned through this project, and if they were implemented in a future project there would be a tremendous improvement in the quality of the outcome. Some specific points which would be changed include:

- Fully recognising the documentation of the challenge before beginning the project
  - Always refer back to the project requirements throughout
  - If unsure, seek clarification rather than making gambles
- Constantly monitoring team member input and quality of work
  - In a team environment, pointing out an issue is not a matter of personal pride but helps everyone
  - Make sure team members are working to their own strengths and abilities before learning a completely new task
- Testing should be at least partially defined before the project begins
  - Know what should be working completely at different points in the project timeline
  - Make sure that testing corresponds to the real application of the project, and be sure of the properties of the demonstration course

If this project had involved a real client, it would have been an extreme embarrassment to the team's company and it would be very hard for the reputation of the team to recover from such a failure. While it is likely that, given more time, The Karen Project could have brought the robot to complete the course, this was not the biggest point of the whole exercise. The main point of the exercise is to look at what went well, and what went wrong, and further understanding on how these affected the outcome. In this case, it seems far more things went wrong than right, but this allows for many lessons to be learned.

Conclusion

The Karen Project's robot was unable to fully satisfy the brief given by the ENGR101 AVC, only achieving 45% of the goals set out. The Karen Project's methodology for ensuring that all team members were working to the best of their ability, splitting the workload and monitoring the quality of work compared with the expected outcome were all vital components of an engineering project which TKP failed to fully realise. While a technical failure, the primary motivation for this challenge is a taste of reality, and going through the process of hindsight and seeing what should have been done is not what was done is beneficial. Team members from The Karen Project learned a lot from this experience. More comprehensive time-based planning, communication and understanding of the task would have all played a large part in allowing The Karen Project to fulfill its aims, and in future projects individuals from the team will certainly draw on the experiences of the AVC.

The overall performance of the AV was hindered by bad engineering practices and processes. There were some excellent parts to the robot, such as the mechanical design, but the code let the whole project down, and this is not the fault of just one person but the lack of support from the team. A whole swathe of issues are responsible for this oversight of support and help, and they are tied up in how The Karen Project approached this task. This report has looked into the strengths and weaknesses of the approach, and how these contributed to the total project outcome.

The Karen Project appreciates that beyond the degree to which the project was completed, the AVC challenge is rooted in developing skills which are applicable to engineering in the real world. Additionally, it grows attributes and uncovers approaches which apply to future coursework. Engineering is not a one-off, it is a repetitive cycle of working towards better understanding how to improve oneself, a team, and the world around. This report has looked into what can be learned from the experience of the AVC. Indeed, if this challenge was completed again, the ability of The Karen Project to meet all of the requirements and in the set timeframe would be far greater. The Karen Project has become more resourceful because of this exercise.