UNIVERSITY OF Southampton

Design Journal Summary
FEEG6013 Group Design Project

11
Automotive Driving Robot
Design, Development, and Modelling of an Automotive Driving Robot

---

ID Number: 29418194
Name: Thomas Green

Primary Supervisor: Dr. Anil Kunnappillil Madhusudhanan
Co-Supervisors: Prof. Suleiman Sharkh

Submitted on: 10/05/2022

1. Summary of individual contribution

The mechatronic nature of the project meant that our time and resources would be best allocated by dividing the team into two sub-teams: a mechanical team and an electronic team. Due to my breadth of experience in hardware implementation and software development I opted to join the electronics team, focussing on being an intermediary between the software side (Mahmoud) and the hardware side (Osa). This eventually evolved to me leading the hardware side of the project and providing continuous assistance to Mahmoud on the software side. In addition to this I was the innovation lead for the project, this involved me researching existing products, presenting new ideas to the team, and guiding the design process with an alternative perspective.

Throughout the project I also found that I was able to contribute to the mechanical side, this was through the design and manufacture of mechanical components, such as the potentiometer mount, which would directly interact with electronic components. I also undertook additional responsibilities that were not stated explicitly in the project brief objectives, however, would prove beneficial to the advancement of the project, these included the creation of a GUI.

2. Roles, activities, and outputs

2.1. Objectives and Team Roles

In the early stages of the project, it was important that we established an overview of what we wished to achieve in the project as a team and assign members of the team roles which would cater to their strengths. Using a SWOT matrix, I was able to determine my strengths in the team would best be focussed on the electronics side, bringing together the hardware and software sides of the project into one complete package. We also segregated the team into sub-teams, one electronics team encompassing three members, and one mechanical team with the other two team members.

At this early stage we also set out our objectives that we wished to achieve during the project, these included short-term aims, long-term aims, and stretch goals. These goals are set out on Page 2 of Design Journal 1 (DJ1). Despite setting these objectives early on it was found that throughout the project the actual aims shifted, some areas became irrelevant (building a prototype pedal) whilst other areas would become more central to the success of the project (PID development and tuning).

2.2. Initial Research

This project contained themes that I was already versed in, the use of Arduino and the making of a GUI, as well as other themes in which I had considerably less experience in. In order to solve this knowledge gap, I thoroughly researched a few of the areas of the project in which I would be most active. The knowledge a gained in a few of these sections is laid out below.

2.2.1. Existing Solutions

When first starting the project, it became apparent that none of us in the team had any previous experience with pedal robots. Due to this the first area researched was existing pedal robot solutions, both commercially and in academia. From this I was able to gain information on the mechanical designs of existing solutions, the approximate electrical and electronic configuration, and the performance specifications (Page 2, DJ2). There were very limited commercial solutions, and many of the academic solutions were either incomplete or performed different functions to what we were looking for in this project.

This research was used to inform my initial designs (explored further in Section 2.3.), particularly in relation to the actuator type and the actuating motion that would be required. All the solutions found during my research were found to far exceed the budget we had for our GDP and so this led to me looking at how we could imitate their solutions but at a significantly lower cost.

2.2.2. Electronic Throttle Pedal

The original plan for the project was to create a prototype pedal on which we could test our pedal robot, and for this understanding how the pedal in the Toyota Prius worked would prove useful. Despite not requiring making a prototype pedal in the end, the discoveries made during this research were used throughout this project. Particularly the workings of the throttle by wire system, the two integrated hall effect sensors (one main sensor and one back-up sensor), and the relationship between throttle valve position and throttle pedal position (Page 3, DJ1).

2.2.3. Electronic Actuators

The actuator is the keystone to this project, and as the team member with the most experience with actuators I took leadership and responsibility for the selection of the actuator. As such I performed in depth research into all the available actuator options, including the price/performance metrics.

I was quickly able to decide that using a DC actuator would be much more suitable for the is project than an AC actuator principally because the lack of an AC power source in vehicle without using an inverter. Subsequent to this I researched different DC motor technologies, including servo motors and stepper motors (Page 3, DJ2). From my research I was able to make an informed decision of which motor to use which is expanded on further in Section 2.7.2.

2.3. First Designs

I, as well as the other members of the team, independently brainstormed and each presented two potential designs for the pedal robot. For each design we compiled a list of strengths and weakness, as well as critiquing others presented designs. The chosen design from these would form a basis for the design we would implement.

2.3.1. Design 1

My first design (LHS Page 4, DJ1) drew inspiration from the AR1 accelerator pedal robot from AB Dynamics. It would use a rotary actuator connected to the pedal via a series of mechanical linkages, these would translate any rotational motion of the motor shaft to an arc motion that would actuate the pedal. This design was very simple in comparison to others that were presented and had a proven track record (many of the commercial and academic solutions used a similar design).

A major advantage of this design is that it did not have to be situated next to the pedal and instead could be offset closer to the driver's seat to avoid getting in the way of the driver. It would also be easier to make this design modularly so that it could be easily adjusted and switched between different vehicles.

2.3.2. Design 2

My second design (RHS Page 4, DJ1) used a linear actuator instead of a rotary one and would not require any linkages between the actuator rod and the pedal. At first this design seemed simpler, however there were issues with the actuator interfering when the vehicle was been driven normally (the actuator would get in the way of the drivers' foot) and a support structure would have to be designed to hold the actuator in position.

2.4. Prototype

Once we had settled on a design (Page 3, DJ2) I began work on a small-scale prototype, this would me and other members of the electronics side of the team to work on the software side of the project in parallel to the mechanical team manufacturing the robot. The small-scale prototype (Page 4, DJ2) I designed, and 3D-Printer gave us a model to test our code on, most importantly the open-loop and GUI code that we would then base the final robot software on.

I also reverse engineered the Toyota Prius pedal sensor connector (Section 2.5) to allow me to use the current pedal position as an input to the small-scale prototype. By decoding the hall effect sensor output using a mapping algorithm I was able to find the throttle percentage of the full-size pedal at any position and replicate it on the small-scale prototype.

Another advantage of producing this prototype is that it created a much easy path to communicate the workings of the robot to the mechanical team, I was able to provide them with a better representation of what we needed the robot to do and so this influenced how they went about designing the robot.

2.5. Reverse Engineering the Pedal Sensor Connector

Instead of creating a prototype of the Toyota throttle pedal we purchased the exact model of pedal that was in the car and firmly mounted it on a stand. The purchased pedal came from the supplier with no documentation but included a small portion of the wiring harness that originally connected the pedal to the ECU of the Toyota Prius. Researching online revealed a few details about the connector but did not give much information about the sensors within the pedal, or the connector pin-out.

In the Toyota Prius manual, there was reference to the sensors inside this pedal model being two hall-effect ICs, one main sensor and one back-up sensor in case the main sensor fails. The ICs in question each had 3 pins: a ground pin, a sense pin, and common voltage pin (in this case 5 volts). By suppling 5 volts and connecting the ground up I was able to monitor how the sense pin output varied with different pedal position inputs. From this I was able to create an algorithm to map the sensor voltage to a pedal position, and I used the back-up hall effect sensor to confirm this pedal position.

This connector was used throughout the rest of the project: values from the sensors was used to tune the PID; it was used in the force experiment described in Section 2.6; it was used to generate input profiles to test the robot; and it was used to confirm the linear relationship between the motor shaft position and throttle pedal position.

2.6. Force Measurements

I devised an experiment that would use a calibrated load cell placed on the pedal to determine the static force required for different pedal actuation positions. The data collected from this experiment could then be used in the motor selection process to ensure our chosen motor had sufficient torque.

I wired a low-cost strain gauge load cell and HX711 amplifier in the configuration shown in Figure 3, DJ3. When the strain gauge is under load there will be a voltage differential, this differential is then amplified by the HX711 amplifier so that is detectable by the Arduino. The Arduino was acting as a data logger in this scenario using a small program I created to output the load cell data stream to a CSV file. I was also able to use the pedal sensor connector to output the pedal position to the CSV file, this was thanks to the previous work I did which is explained in Section 2.5.

Once I had debugged and tested the Arduino and load cell system, I created a 3D-Printed housing for the load cell (Figure 4, DJ3) by modifying an existing load cell enclosure from GrabCAD. This was required as the load cell needs to be supported around the edges, with the central portion of the load cell free to flex and so undergo strain. Once the enclosure was printed and assembled, I calibrated the load cell using a series of known masses placed on the enclosure pressure plate. By doing this I was able to systematically relate the voltage outputs from the strain gauge (as read by the Arduino) to a particular input force to the pedal.

Once calibrated I began to take measurements with the load cell mounted on the throttle pedal (Figure 5, DJ3). When conducting the experiment the pedal was firmly mounted to the pedal stand designed by the mechanical team. This was to ensure there was no unwanted movement in the pedal with the aim to replicate the real pedal motions and conditions as closely as possible. Further to this I ensured to sit in a position as if I was in the drivers seat of a car when actuating the pedal. This yielded the results shown in Figure 7 of DJ3.

From the results I found that the minimum force to begin moving the pedal was 12N, whilst the minimum force to actuate the pedal to full throttle was 26N. The minimum full throttle actuation force increased to 42N when accounting for the extra force required to overcome the tactile bump in the pedal that occurs to signify to the driver that the pedal is fully depressed. These force measurements were used to inform the motor selection process (Section 2.7.2).

2.7. Component Selection

Component selection was a key stage in our project, Mahmoud and I took responsibility for selecting the components. We both took the lead for researching and selecting different components, I focussed on the ECU, motor, and battery, whilst we collaborated in the selection of the motor controller and other components.

2.7.1. ECU

When selecting the ECU there were two clear options: a Raspberry Pi microcomputer, or an Arduino microcontroller. The three main criterion I used to select the ECU were the power requirements, the reliability, and the practicality.

In terms of power requirements, the Arduino was found to have a much lower peak power draw and a more versatile input voltage range which allowed easier integration with the 12-volt battery that was used. The Arduino was also found to be more reliable for our use case and provided the ability to handle analogue inputs without any additional hardware, which would be useful for the analogue sensors used in this project. These reasons, explored further in Page 2, DJ3, led us to select the Arduino Uno Rev3 as the robot ECU.

2.7.2. Motor

The motor was the most expensive component in this project and the most amount of time was spent research the exact requirements before selecting the component. During the selection process I considered the torque required, the input voltage, the style of motor (brushed vs brushless, permanent magnet, etc.), and the peak output speed.

To determine the torque, I used the force data collected in the load cell test (Section 2.6). This data was used in conjunction with the crank length range found by the mechanical team to determine what the minimum torque at the motor shaft should be. With a crack length of 12 centimetres and a maximum pedal force required of 42N, I calculated that the required torque would be 5.2Nm once the contact angle with the pedal had been accounted for (an early version of these calculations is shown on Page 3, DJ2). To account for any other system losses due to friction I compensated the torque value using a safety factor of 1.2 giving a final required torque of 6.2Nm.

Using data collected from the data collection experiment we found that the pedal should be able to be fully actuated and released at a bandwidth of 3Hz. I calculated that this would correlate to a motor output speed of 20rpm and so this was set as the minimum output speed.

Using these criteria and the accounting for the budget restraints I selected the Micromotors P205-24-100 planetary geared brushed DC motor, capable of 7Nm torque and an output speed of 32rpm. This exceeded our required specifications and amounted to £198.

2.7.3. Motor Controller

The motor controller had to be matched to the motor, meaning that it had to support 24V, be able to deliver up to 2A, and have a connection interface for the Arduino. This led to only a few options for the controller, from which we selected the Cytron MD10C as it was economical and extensive documentation already existed about the use of the controller with Arduino. The exact requirements and reasoning is explored further on Page 3, DJ3.

2.7.4. Battery

The choice of a sealed lead-acid battery was largely due to cots and safety reasons. Lead acids are considerably cheaper than lithium polymer batteries and pose less of a fire risk. The sealed nature of the battery ensured that we would not have to worry about the mounting orientation of the battery or the venting of toxic gasses into the car cabin.

A 12volt, 9Ah Yuasa battery was chosen for its large capacity, and an ideal voltage to be charged from the car DIN4165 port. Information about charging the battery and more detailed analysis of the costs is on Page 3, DJ3.

2.8. Wiring

To choose the correct wire cross section for our system I stress tested all the components individually to find their peak current draw and confirm that these corresponded with the values presented in the component specification sheets. It was also important at this stage that I considered how the use of voltage regulators affected these peak current draws, and so these were included in the circuit when measuring the currents.

Through this I found that the combination of the motor controller and the motor drew a maximum of 2.6A from the 24V battery under full load conditions (the motor was briefly set to 100% speed whilst actuating the pedal). With an estimated 0.2 metre cable run from the battery to the motor, and a desired voltage drop of less than 3% I found that wire with a cross section of 0.13mm^2 would meet the criteria.

To leave extra leeway for expansion of the system, especially the ability for the robot to function correctly on pedals that offered more mechanical resistance, and to reduce the voltage drop across the system I settled on a slightly larger wire of 0.22mm^2. This would offer the current carrying ability of 5A at the rated 24V, with a reduced voltage drop of 2%.

I choose to use stranded wiring for our robot primarily due to the increased flexibility simplifying wire routing. Female spade connectors were used for the battery and motor connections, whilst the connections to the voltage regulators were soldered directly to the circuit board. The wiring configuration is shown in Figure 1, DJ4.

2.9. Manufactured Components

Despite my focus not being on the mechanical side in this project, there were some areas in the project where I designed and manufactured components. These mainly involved components that would affect sections I worked on, such as the potentiometer mount, the motor to potentiometer coupler, and circuit board offsets.

The offsets were very simplistic, being cylindrical sections through which a screw was threaded, however different lengths of these were used in the mounting of the circuit boards to ensure that all boards were mounted at the same level. This had two main benefits: it made wiring the boards easier and allowed me to run wires underneath the boards chelping with wire management. In addition, it ensured that there was a lesser risk of short circuits by creating a significant air gap between the exposed pins on the bottom of the circuit boards and the metal base plate.

The potentiometer mount was needed to ensure that the potentiometer shaft and the motor shaft were coaxial. This was a simple construction (Figure 1, DJ5) that had a central cylindrical cut-out for the potentiometer shaft, and an additional cut-out for the potentiometer locating pin to stop unwanted rotation. Whilst this component would undergo very little force, it was important it was stable and so the mount was waterjet cut from 3mm aluminium and secured using an adjustable bracket.

The coupler was used to ensure parity between the motor shaft position and the potentiometer shaft position. This was required so that we could accurately track the position of the motor shaft and use this to inform the PID (Section 2.12). This component would operate at very small torques and so I used plastic additive manufacturing to produce the part due to its reasonably complex internal geometry (Page 1, DJ4).

2.10. Open Loop Code and Initial Closed Loop Code

The open loop code was the first stage of implementing software for the robot, this code would operate without any feedback from the potentiometer about the motor position. Instead, I established a relationship between the motor input variables (the rotation speed and rotation duration) and the degrees of rotation (Table 1, DJ4). I used this to determine a formula that would allow me to input a value for the number of degrees of desired rotation, this would then turn the motor at a fixed speed for a certain duration to achieve this correct angle rotation.

The open loop code allowed us to begin testing the capabilities of the motor, ensure all the mechanical components were of sufficient quality, and served as a basis for the closed loop to be built upon. Information I learned at this stage, such as the of sign of a the PWM signal determining the direction and the absolute magnitude determining the speed when in locked anti-phase mode, would be used throughout the rest of the project.

The closed loop code built upon the open loop code and introduced feedback from the potentiometer. The code worked in a similar manner; however, the function would be continuously cross referencing the expected current position and the actual current position, given by the potentiometer, and adjust accordingly.

2.11. Calibration Function

The calibration function was introduced for a couple of reasons; it allowed the interoperability of the pedal robot between cars with different pedal travel lengths and actuation forces, and secondly it allowed the robot to readjust it's its range of motion automatically whilst we were tinkering with the robot.

The calibration function works in two stages, firstly the minimum force needed to push the pedal is determined, next the ranges are found by actuating the pedal at this minimum force in both directions until resistance is met. To determine the minimum force the motor is given a PWM value and the change of shaft position over a set period of time is found. The PWM value is incremented until the rate of change of motor shaft position is deemed sufficient (this was found to be 15degrees in 500 ms). The maximum limit is found by actuating the motor at this minimum PWM value and monitoring the change in motor shaft position, once the motor shaft position stops changing then the pedal is assumed to have reached is maximum depression and so this is set as its max position. The same is repeated in the reverse direction to find the minimum position. This algorithm is shown in the following two flow charts.

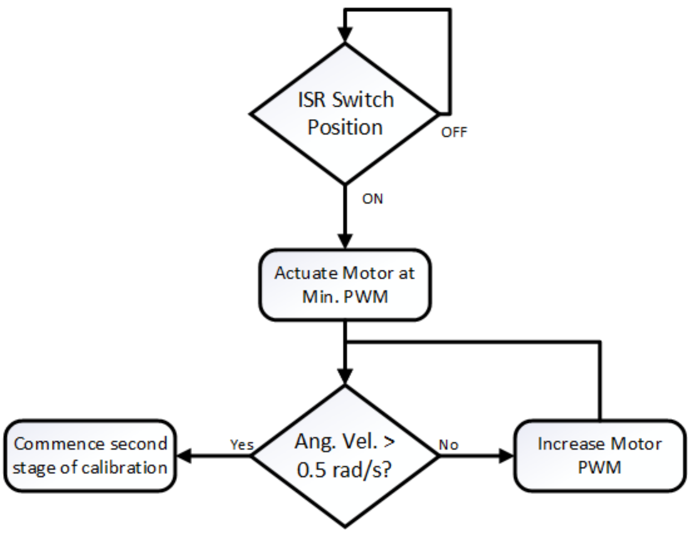

Figure 1: The first stage of the calibration algorithm.

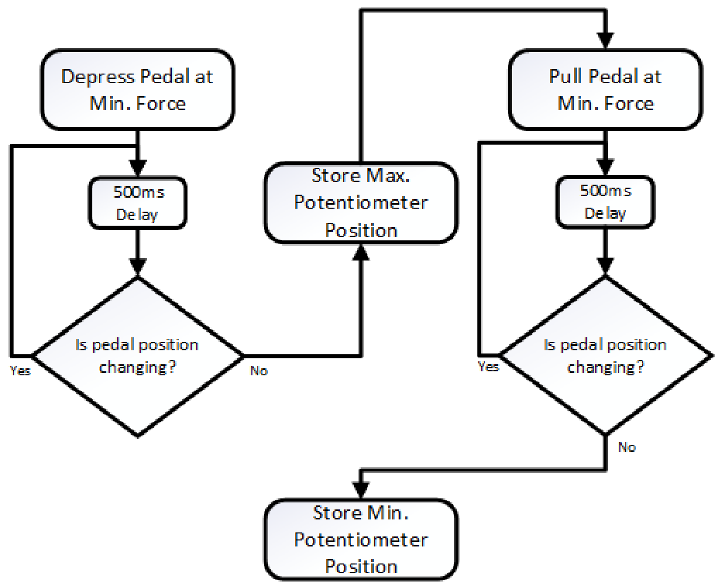

Figure 2: The second stage of the calibration algorithm.

2.12. PID

The PID was an evolution of the closed-loop code, functioning in a similar way however with the benefits of a more universal implementation and the ability to tune the PID for increased performance. Mahmoud led the development and implementation of the PID and I took a supporting role, assisting with testing the PID, tuning the PID, and implementing the PID with the existing hardware.

We set performance aims for the PID that we wished to achieve, these included an overshoot of less than 10% and settling time of less than the average human reaction time (~200 ms). These criteria were assessed using step responses throughout the tuning process as well as continual monitoring of the maximum robot bandwidth. During the tuning of the PID it was discovered that even slightly altering the differential parameter would lead to chaotic behaviour and so we adapted our focus onto the proportional gain and integral parameters.

At this stage I discovered there was the large amount of backlash in the system, discussed further in Section 2.16. This backlash was significant enough to cause significant issues with the settling of the PID and so we implemented a position threshold of 2% to prevent oscillations as the robot approached a steady state. This threshold would later be removed once Jason and I had fixed the issues with the crank with the use of a motor hub.

2.13. System Identification

A key aim in this project was the developing a dynamic model of the driving robot using system identification, this model could then be used for further tuning of the PID controller. Mahmoud and I both worked extensively on this area of the project, with Mahmoud acting as the lead for the PID portion. For the system identification there were three stages: the collection of identification and validation data sets, the model identification using the identification data set, and the validation of the model using the validation data set.

To collect the identification data sets we generated a series of random input data that would be passed to the robot. The robot output, i.e. the throttle pedal position, would then be monitored to see how the robot responded to each random input. To generate the random input signals, we used a pseudorandom binary sequence (PRBS) that would vary the time and speed that the robot would move in any given location.

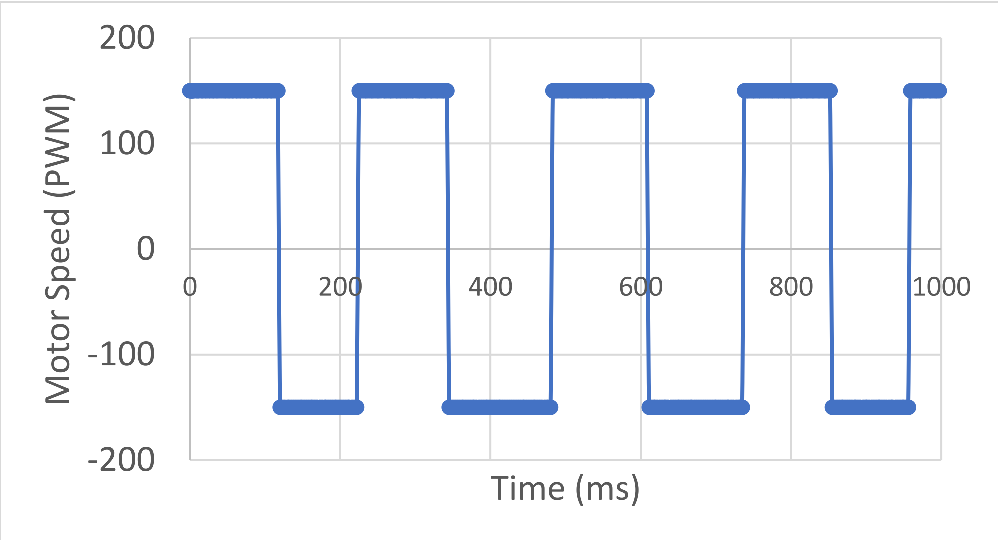

Figure 3: Example of the PRBS signal, a negative value indicates that the crank was travelling clockwise and vice versa.

The original PRBS signals we used were leading the robot to actuate too far in any given direction, this led to the robot reaching the limits of the pedal. This would not give a representative model of the robot and so the PRBS signal had to be tuned so that all motion would be within the bounds of the pedal's range of motion. The response to these signals was recorded and then pre-processed before use in the MATLAB System Identification Toolbox.

After collecting the data from the robot, we noticed that the Arduino analogue read function we were using was leading to an inconsistent sampling rate. The time between samples would vary by up to 0.005ms which could lead to problems when doing the system identification. Therefore, I designed an algorithm and implemented a program to batch resample all the data we had collected to a single consistent sampling rate (Page 3, DJ6).

This resampled data was then used in the MATLAB System Identification Toolbox, one set was used as the input identification data and another dataset was used as a verification set. The system identification toolbox also required the delay of the system, this is the time taken for the robot to respond to any given input. This delay was to be expressed as a multiple of the sample rate and was found to be 7 sample rates (or 20ms) using the step response data we collected when assessing the performance of the original PID (Section 2.12).

Further to this the toolbox required an estimation of the number of poles and zeros of the system, the exact value of these was unknown and so we gave a reasonable first estimate (3 poles, 2 zeros) and tuned the model from there until we achieved satisfactory autocorrelation and cross corelation values. The best model of the robot we achieved had 9 poles and 6 zeros, using the validation dataset it was found to have a fit of 95.37% with the autocorrelations and cross correlations largely within the required bounds (Page 4, DJ6). This obviously yielded a very complicated transfer function however proved useful for the further tuning of the PID.

2.14. GUI

The GUI was a non-essential goal in this project however would be a useful interfacing method to communicate with the robot. I performed extensive research into the software packages of existing commercial pedal robots and found which features they offered that would also be useful for our robot.

The first stage to making the GUI was deciding on the layout and selecting what data would be included in the GUI. Using the research of existing pedal robot GUIs and previous experience in GUI design I produced an initial GUI template (Figure 6, DJ2). This template was not final and continued to evolve throughout the project as it was found it would be more useful to replace some of the data elements.

The GUI was programmed in Java using the Processing graphical library and integrated development environment (IDE). In addition to this the ControlP5 library was used due to its extensive list of included controller and GUI elements, such as buttons, sliders, and input fields. Processing was used due to its easy integration with Arduino and the capability to handle analogue serial inputs directly to the frontend without any backend reprocessing.

The final GUI had three main interface screens: the first being an information screen that gave the user extensive information about any safety precautions that should be taken before utilising the robot (Figure 4). It also notified the user of the safety mechanisms onboard the robot in case of failure. The following screen initiated the calibration function (discussed further in Section 2.11) and informed the user of the calibration movements the robot was currently making.

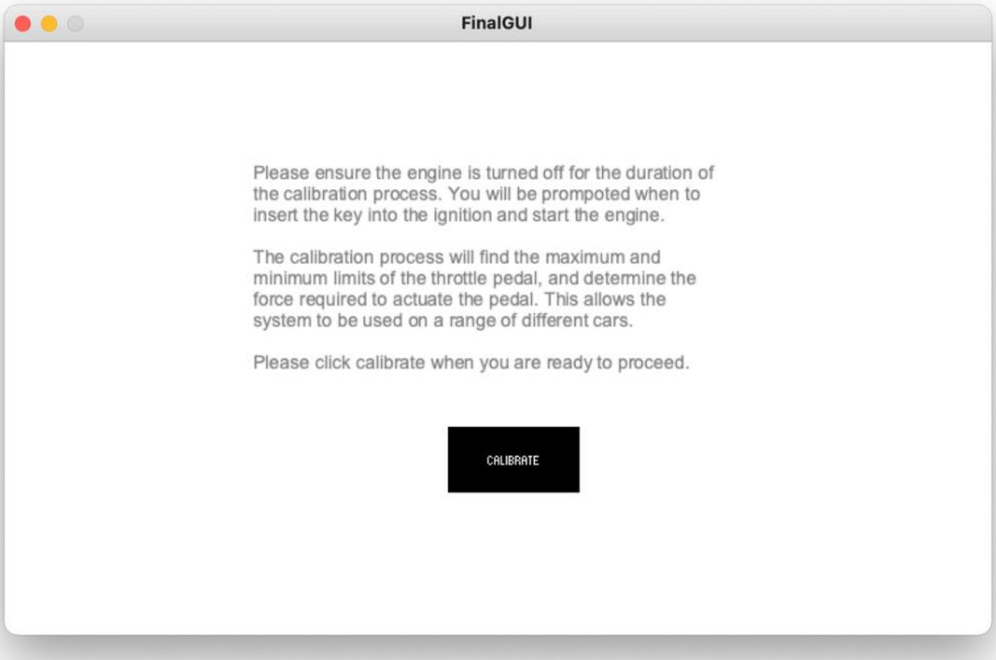

Figure 4: Opening screen of the GUI explaining the following calibration stages.

The final screen would persist throughout the use of the robot, this screen contained information about the state of the robot, such as the current throttle pedal position the PID setpoint (Figure 5). Also on this screen was a user input section where the user could override the default operation of the robot and provide a desired motor speed, pedal position, or alter the PID tuning characteristics.

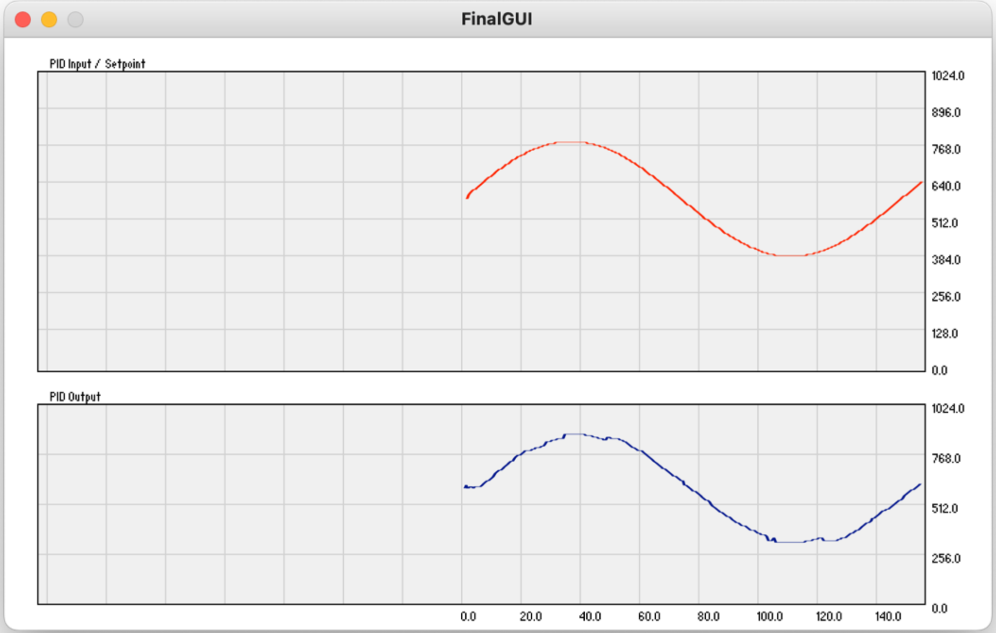

Figure 5: The final GUI screen showing the PID setpoint and the resulting robot position (PID Output). For both graphs the x-axis is time and the y-axis is relative motor position.

The GUI ended up being the most time constrained portion of the project, not least because it was non-essential, so my focus was on the completion of the core robot for much of the product span. However, it was largely completed to a level with which I was satisfied with and can be iterated further in future years if additional functionality is required.

2.15. Safety mechanisms

At a late stage of the project, I realised that the pedal would not return to a neutral throttle position when actuating force had been removed. It was originally assumed that the force from the compressed spring in the pedal would push the motor and crank back to its neutral position when the power was cut off. Whilst this did occasionally occur, on other occasions the pedal would get stuck at approximately 50% throttle. As the pedal returning to a neutral position was not consistent or reliable, I concluded that it would be imperative for the safety of the operator to introduce a system to always return the pedal to a state of 0% throttle. After brainstorming with the mechanical team about what solutions we could add at this late stage we settled on a manual override, this would work in conjunction with the ISR switch and emergency stop button to return the pedal to a safe position. The mechanical team were then given the task to manufacture and implement this.

2.16. Testing

The final month of the project was spent testing the robot, ensuring that it was able to meet all the required specifications and that it was able to work reliably. This was done in plenty of time to ensure that we would be able to correct any issues with the robot, and still allow us to fully assess the robot in its final state.

It was at this stage that I discovered the issues with the crank D-shaft mount (addressed on Page 5, DJ5) as well as the gear teeth degradation. I proposed some solutions to the mechanical team and worked with them to implement new mechanical components, in addition to software restrictions, to limit any further damage to the gearbox and create a more suitable crank mounting mechanism.

With the help of Mahmoud, I tested the maximum achievable bandwidth of the robot using a variable frequency sine wave signal input. Experimentation showed that a maximum pedal actuation frequency of 4.5Hz was achievable before attenuation, the signals are shown in Figure 2, DJ6. We also assessed the PID performance, focussing on its ability to exceed the performance of a human driver, this is expanded further on Page 2, DJ6.

3. Key Achievements

Throughout the project the majority of my effort was put towards the electronics side, in particular the software and hardware. My key achievements are as follows:

- The design and build of a small-scale prototype that was used to test the feasibility of early code in lieu of the completed robot. This allowed the project to progress at a faster pace due to parallel working.
- Integrated an Arduino and load cell onto the pedal and subsequently performed experiment with the aim of confirming the pedal actuation force requirements. The data gained through these experiments would later inform the motor selection process. During this I reverse engineered the pedal so that we could harness the onboard Hall effect sensors to determine pedal position.
- I performed thorough research into the available components for the robot and made informed decisions on the selection of the components from this research supplemented with additional experimentation.
- Assembled and wired the robot using appropriate wiring diameters and connectors found via research. In addition to this I designed and manufactured the mount for the potentiometer and a PLA coupler to connect the motor shaft and potentiometer shaft.
- Creation of the initial open loop code and subsequently building on this to create the closed loop code. Further to this I collaborated with Mahmoud on the development of the PID using elements of my closed loop code.
- In conjunction with Mahmoud, we developed a calibration function that would determine the pedal limits and actuation force requirements of any given pedal. I focussed on the force determination side, whilst also provided assistance and insight to Mahmoud who was working on the pedal limit side.
- Designed and programmed a graphical user interface that would allow the users to interact with the robot. The GUI included a live graph of the PID setpoint and the PID output.

4. Critical Review

During this project the team was able to successfully design, manufacture, and build a pedal robot that was able to effectively actuate a throttle pedal. We were able to meet the vast majority of the project aims, and continuously adapted our expectations and objectives to respond to any unforeseen circumstances that arose. Due to collaborative working, we were able to build a system that married mechanical and electronic component into a complete mechatronic system. The resulting product met or exceeded all the specifications that we defined early on in the project and serves as a good basis for future GDP teams to build upon and refine.

As a whole the project had many successful elements, however, there were also other areas that both as a team and as individuals we could learn from and improve in. The benefits and drawbacks of our final design proposal, in addition to an assessment of the success of our team in reaching this final proposal are elaborated on below.

4.1. Innovation

This project was not an advancement of a previous project, instead we started from scratch and used original novel ideas, in conjunction with research of existing products, to contrive a solution for a throttle actuating robot. This was achieved through an iterative design process, combining the strengths of our collective ideas and iterations to form a final design proposal. Throughout the project we were wary of the budget constraints and so we took different approaches to achieve a functional robot at a low cost. One such example was our own development of a servo motor, thus avoiding the ned to purchase a commercial servo motor which would have come at a much greater cost.

On the software side we developed a calibration function that would be able to automatically adapt the actuation range and actuation force to any vehicle. We were able to implement this feature without the need of any additional hardware or any additional information about the vehicle the pedal robot was installed in.

Through consultations with our supervisors, we discovered that the project would be continued in subsequent years with other GDP groups, these groups would work to add in additional functionality such as brake pedal control. As a result, it was important that we thoroughly documented all our processes and any code so that the project could easily be understood and further iterated. We also ensured that the robot would be expandable to allow the addition of any extra hardware that would be required in these later years.

4.2. Process

In the initial weeks of the project, we established set phases of the project and the stages encompassed in each phase. From this we were able to create detailed team and personal Gannt charts that would ensure all members and the team were on track to achieve our goals. Throughout the project we adapted the upcoming tasks to best fit the direction of the project and consolidated all the knowledge we had individually gained on the robot to ensure all members were well-versed and up to date on all developments of the robot.

During our design process we used research and experimentation to establish an initial design, from this we performed testing to assess the performance of our design and inform any further modifications that we would make. This method had many benefits, not least being the early creation of first iteration provided us with a full-scale robot to test our code on. This allowed us on the electronics team to develop and debug any software functions whilst the mechanical team worked on improvements to the robot. Through analysis of all the robot component we were able to estimate the lifecycle for each part., from this we would be able to focus on improving the robustness of the weakest parts. This focussed improvement approach meant that we would be able to quickly introduce improved components, and we would not be wasting material and time on improving areas that were already sufficient.

4.3. Communication

This project required continuous effective communication to numerous parties, for example, the communication of the project design process to our supervisors during bi-weekly meetings to update them on our process and receive their valuable insight. Without this communication channel we would have struggled to overcome some of the more complex issues that arose during the project life cycle.

Much of the final couple months of this project were dedicated to the production of the final deliverables, such as the group report and video. This was particularly important for this project as it was important that all the lessons we had learnt, and discoveries we had made were effectively communicated to future students working on the continued project. This would save them having to waste time repeating work that had already been done, instead allowing them to focus on the improvement of the system.

4.4. Sustainability

One of the key long-term aims in this series of GDP projects was the reduction of energy consumption in electric vehicles, and so it was important that we maintained this sustainability aim throughout the execution of our project. We limited our material waste by ensuring to use scrap metal where appropriate and choosing the correct material specifications for the load conditions.

An important aim for our project was the interoperability of the robot in a wide range of cars, this was achieved by using adjustable mechanical components and the aforementioned calibration function. This negates the need for the production of multiple pedal robots, and so avoids the associated economic and environmental costs.

4.5. Conclusion

During the undertaking of this project, we were able to achieve the objectives we defined in the first couple of weeks. Whilst there are still areas for improvement the final design proposal, we produced was more than capable for the required purposes and will provided a good basis for further development.

Our team operated to a good standard, with members taking ownership for their individual sections and readily providing assistance to others when needed. I felt that was able to provide my team with a wide range of technical insight, hardware knowledge, and software skills. The project also enabled me to further improve many of my team working, communication, and organisational skills.