TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

A
PROJECT PROPOSAL
ON
AUTONOMOUS VEHICLE/ROBOT AI(SECOND HALF OF THE AUTONOMOUS VEHICLE PROJECT)

SUBMITTED BY:
PRAMESH SHRESTHA (PUL076BEI025)
PRASHANT KARN (PUL076BEI026)
TRIPTI SHARMA (PUL076BEI047)

SUBMITTED TO:
DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

June, 2023

Acknowledgements

We would like to express our heartfelt gratitude to all those who have encouraged and inspired us in the work on this project.

We would like to thank the **Department of Electronics & Computer Engineering** of IOE-Pulchowk Campus for providing us with an opportunity to work on this project. Similarly, We would like to express our sincere appreciation to the **NIU Corporate Team Nepal**, for encouragement and providing us with the equipment necessary for our project.

We would also like to thank our other classmates who provided valuable feedback for our project concepts. Their willingness to share ideas and perspectives helped us to develop a more nuanced understanding of the subject matter. Also we would like to express our sincere gratitude to the **Robotics Club** for promising us to provide with the necessary resources and space to build the mechanical and electronics system for the Vehicle. The club's commitment to fostering a culture of innovation and experimentation has been a significant factor to encourage us to undertake this project. We appreciate their continued support and encouragement.

Lastly, we are thankful to our seniors as well who guided and give feedback to our concept paper. Their experience and knowledge were crucial to us for designing this project.

Abstract

Autonomous vehicles have been in rapid development in the recent decades, and have just about perfected their performance in the open road. However, they still struggle with the famous Last-Mile-Problem, where self driving vehicles struggle with navigating the unpredictable, unmarked, and complicated networks of paths within neighbourhoods and university grounds. We decided to take on this challenge, and look at the problem not in terms of vehicular transport, but in terms of robotics. We propose to build an autonomous vehicle capable of navigating around Pulchowk Campus all with its own ability, while carrying a passenger. We embark to make this a highly modular and flexible project, with its components highly layered, abstracted and separated, with communication able to be swappable and usable in a wide range of applications. For this half of the two-team collaboration, we set out to create the Artificial Intelligence aspect of the vehicle, designed to have four distinct layers, for sensor/actuator processing, quick reflex actions, 3D mapping and pathfinding, and logical extrapolation respectively. We have created a robust project management and quality assurance scheme to accomplish this project of such a scale. We wish to opportunitise ourselves to learn the various problem solving and programming aspects to overcome this challenge, as well as to try out new, innovative ideas. Our goal is to build a foundation for research and deployment for not just the field of transportation, but for robotics and Artificial Intelligence in general.

Contents

List of Figures

List of Tables

List of Abbreviations

1. Introduction

1.1 Background

1.1.1 Historical Background

The incredibly impactful event of the invention of the Automobile had an ironic twist - its replacement of the original autonomous vehicle. Despite the car having overwhelming advantages over them in aspects such as speed and cleanliness, there was one thing horses could do that cars still cannot - drive themselves. The world must have reminisced about this loss, as evidenced by the immense advancements in the technology in the past two decades of the self-driving car, or more generally - the autonomous vehicle. These advancements combine state-of-the-art technologies in image processing, machine learning, and control systems to create outstanding results, paving the way for the field to mature to the point of products being almost commercially available, and more interestingly to us, the technology being available to optimise, modify and expand upon. In our project, we look back at the horse, noting its closer similarities to robotics rather than vehicles, and try to find the correct direction to leap towards for the next generation of autonomous vehicles.

1.1.2 Context and Introduction

As part of our Engineering degree course (Bachelor's in Engineering for Electronics, Communication and Information - BEI), we are expected to complete a two-part Project module in our final year, in 2023 AD. For our project, we created a two-team collaboration to hopefully accomplish a project of a relatively large scale - The Autonomous Vehicle Project. Here, we put together our engineering skills - from mechanical and electronics skills to analysis and programming skills - to build an autonomous vehicle capable of navigating around Pulchowk Campus all with its own ability, while carrying a passenger. We embark to make this a highly modular and flexible project, with its components highly layered, abstracted and separated, with communication able to be swappable and usable in a wider range of applications. For this half of the two-team collaboration, we set out to create the Artificial Intelligence aspect of the vehicle, including lower level sensory and actuator processors and higher level problem solving agents. We wish to opportunitise ourselves to learn the various problem solving and programming aspects to overcome this challenge, as well as to try out new, innovative ideas (see section 4.5). Our goal is to build a foundation for research and deployment for not just the field of transportation, but for robotics and Artificial Intelligence in general.

1.2 Problem statements

The problem statements we tackle for our half of the two-team collaboration are as follows:

- Can a vehicle autonomously transport a passenger from one place to another given the locations, with the passenger being completely passive?
- Is layering and modularity possible and furthermore advantageous in the AI aspect of such a vehicle?
- Is the direction towards traditionally structured AI agents on top of deep learning layers the correct direction to go?

These questions are integral to the challenge of building this project. The first two reflect core ideas for implementation and testing, while the third represents a new idea for which the project will be a playground environment. Scope and vision of the problem statements are explained in the sections below.

1.3 Objectives

1.3.1 Collaboration Objectives

The collaboration objectives state the goals of the two-team collaboration, covering the entire project including the hardware, electronics, control systems and software. The can be stated as follows:

- To build an autonomous vehicle capable of driving a passive passenger from one location to another given the locations.
- To have the vehicle deal with obstacles, stay on route, and make decisions about changing routes both on a small and large scale in order to cause least damage to itself and discomfort to the passenger.
- To create a modular, layered and abstracted system that on principle opens possibilities of the modules' usage in a much wider variety of applications in the field of robotics and AI agents.
- To demonstrate an appropriate project management scheme covering all the important aspects of project management and work record keeping.

Further specifics from this point on can be found in the respective project objective sections for either half of the collaboration.

1.3.2 Objectives of This Project

This half of the two-team collaboration includes the Artificial Intelligence part of the project.

- To build an AI system that is layered to work in a better leveled interaction pattern with the loosely corresponding control system layers of the project.
- To provide possibilities to train the different layers and modules in different, more tailored environments to improve efficiency
- To create good abstraction between the modules to expand their working possibilities
- To provide an experimentation environment for new ideas based on combining traditional AI agent systems with modern day machine learning

The first two points relate to the structure of how the different components of the project fit together. For example, the AI module responsible for quick immediate reactions may only communicate with the lower level, more responsive module of the hardware system, and may make a decision without talking to the more powerful higher level decision making AI modules, to save on time.

An example for the third point can be that the higher level decision making AI module may not need to know how the movement is actually being made, only where it is taking the entity. This allows it to work equally well with legged robots, flying robots, and even virtual agents.

The final point refers to the fact that projects such as this one have high risk when trying to implement novel concepts and ideas. Our high modularity allows us to actually experiment with these concepts, provided redundancy from other modules tasked with the same goals.

1.4 Scope

1.4.1 Collaboration Scope

The collective scope of the entire project refers to both the problem statements and the objectives to define the coverage and limits set for our goals in order to make the project well defined and achievable. They can be stated as follows:

- Regarding the transportation of a passenger from one location to another
  - The number of passengers will be limited to one
  - The area for which the locations may be defined will be confined within Pulchowk

Campus

  - Obstacles will be within the context of the campus (hence, some common elements present in the usual context of vehicles may not be considered, such as traffic lights)
  - The speed off the vehicle will be limited to 20 km per hour

- Regarding layering and modularity
  - The number of layers and modules will be kept low to accommodate the workflow for our small team and low resources

1.4.2 Scope of This Project

For this half of the two-team collaboration, the following scope will be considered:

- Regarding the capabilities of the AI agent
  - The agent will be able to navigate around static and slow moving common objects such as people, parked vehicles, etc. but not objects with high speeds (noticing which its response will be to stop)
- Regarding the information provided to the AI agent
  - Optical, depth and ultrasound sensors will provide live data to the agent
  - The agent will have prior knowledge about the layout of the campus, the permanent objects in it, etc. as a 3D environment

2. Literature Review

2.1 Related work

Openpilot, an open source driving assistant which was developed by Comma, aims for the replacement of Original Equipment Manufacturer (OEM) advanced driver-assistance systems with the objective of improving visual perception and electromechanical actuator control. WIth the objective of making driving safer and accessible to a wide range of vehicles, Openpilot has a great community working to improve to enhance the driving experience for everyone.

2.2 Related theory

For an overview, Openpilot reads data from various sensors (such as camera, IMU, Global navigation satellite system (GNSS), etc), processes their data and relays it to a large neural network. It then converts the output from the NN into actionable commands for the vehicle to operate.

Driving into more technical aspects, the main system of openpilot can be divided into Sensors and Actuators, Neural Network, Localization and Calibration, and Controls. The services controlling the sensors are responsible for managing communication between various peripherals including GNSS, infrared LED, Cameras, gyro sensor, magnetometer, and light sensors. Now using these data along with a desired input, one of the NN models runs the supercombo model, which outputs the desired driving path along with other metadata including lane lines, lead cars, road edges, and more. One of the salient features included in Openpilot is driver monitoring which uses a camera installed in the driver camera and makes sure if the driver is paying attention to the road or not before giving the control over to the driver. The input of the NN model is warped into the calibrated frame, which is aligned with the pitch and yaw of the vehicle. This normalizes the image stream to account for the various ways in which people mount their devices on their windshields.There are also services which account for localizing the car in the world. This localizer outputs the position, orientation, velocity, angular velocity and acceleration of the car. Now before actually controlling the car, Openpilot plans its action. Planning is separated into two parts - lateral planning (steering) and longitudinal planning (gas/brake). Both use an Model Predictive Control (MPC) solver to ensure the plans are smooth and optimize some reasonable costs. It takes the fused (neural network + radar) estimates of lead cars and the desired set speed, feeds it into an MPC solver, and computes a good acceleration profile for the next few seconds. And finally after receiving the plan, in the form of curvatures and velocities/accelerations, it converts it to control signals. These vehicle agnostic targets (acceleration and steering angle) are then converted to vehicle specific CAN commands that work for that car's API, through a closed loop control system running at 100Hz. It also parses the raw CAN data from the car and publishes it in a canonical format. [1]

Undoubtedly, each product using Comma.ai's technology has its cons

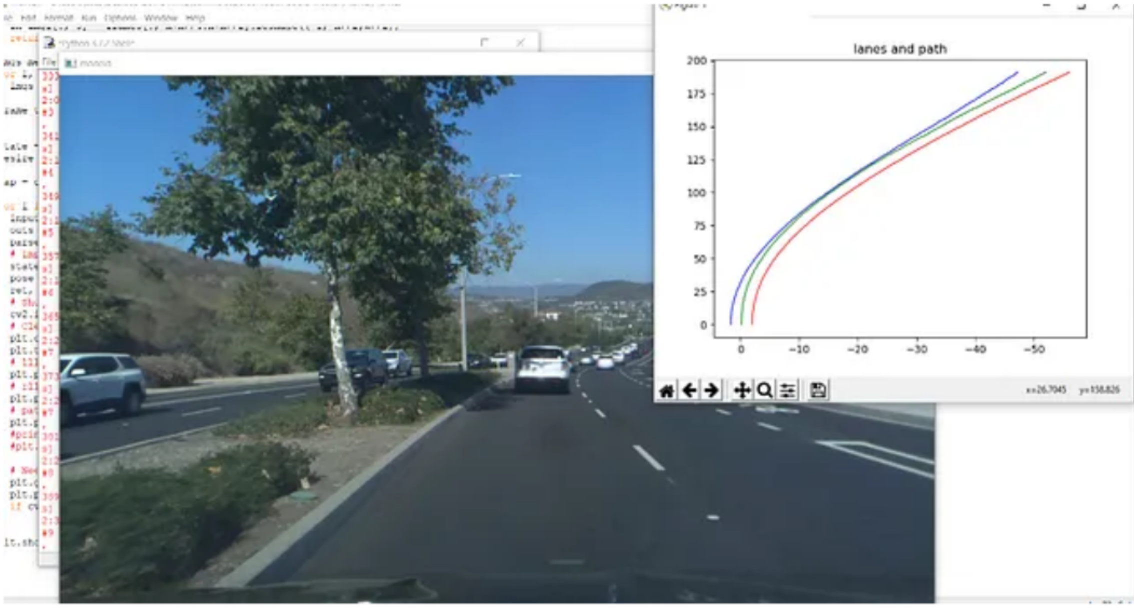
- Limited availability: Advanced Driver Assistance System (ADAS) is only available for selected makers and car models. It means not all drivers will be able to use their technology.
- Dependence on software updates: its functionality depends on regular software updates, which may not be available for older vehicles.
- Legal concerns: Comma.ai's ADAS is not yet fully approved for use in all states and countries, so users may face legal repercussions if they use the technology in areas where it is not yet supported. [2]

Super Combo Model

At the core of the Openpilot we have the supercombo model, an elegantly crafted, yet lightweight pretrained self-driving car model to predict lane lines. It take 2 Consecutive image frames in YUV format, State of the Recurrent Cells (being a recurrent neural network), and Desire (actions to be performed) as an input. It then predicts the following Outputs:

- Path (Trajectory output to be followed)
- Left Lane
- Right Lane
- Standard Deviations of Path and Lanes
- Lead Car information
- Longitudinal Accelerations, Velocities and Displacements for n future steps
- Pose(the predicted translation and rotation between the two input image)

The Path, Left Lane and Right Lane are the 'Predicted Trajectory' and the lane lines in Top Down / Bird's Eye View coordinate system relative to the ego vehicle.



Path and Lane outputs of supercombo

The model predicts the curve on the road to the right.

The one other thing to see here is that lane lines also fade when no lane lines are visible, or simply when the model is less confident about its prediction. This 'confidence' score is basically predicted by the model Bayesian Neural Network and is simply the 'standard deviations'. These standard deviations are predicted for each of the path, left lane and right lane and many other outputs including pose, lead outputs, etc. [3]

3. Proposed Methodology

3.1 Project Management

This project is expected to be on a scale large enough to require quick and efficient task division, work tracking, careful planning in all time scales, etc. As a result we have decided to build a robust project management scheme for our two-team collaboration, giving the role of Project Manager to one of the six members. The project manager shall be conducting all of the aforementioned tasks as well all levels of integration, planning, controlling and monitoring, etc. as mentioned in and in accordance with **Project Management Institute (PMI)'s Project Management Body of Knowledge (PMBOK)**. The team member with the project manager role assigned has drafted a Project Management Plan laying out the planning, controlling and monitoring schemes that will be used during the development and progress of the project. The Project Management Plan document has been attached to this proposal. The project management aspect of the project will be made as official and as close to the actual industry as possible in order to provide legitimate project managing and team-work experience holding proper weight for applications in the industry.

3.1.1 Quality Assurance

With a robust project management scheme, a robust Quality Assurance (QA) scheme is also required. The purpose of this scheme will be to estimate, enforce and monitor the quality of entities and modules built, having a well defined plan for measuring and tracking the quality. This information will be used to alter the plan for the project to keep expectations realistic as well as the workflow to keep quality acceptable. For example, a 3D map of the campus will be required for building the virtual environment for early training and testing (see below), the workflow for which is for the most part unique and improvisational. QA will be responsible for defining the quality of the 3D mapping required, and will monitor the work efficiency, noting successes and failures, to help in the future. Since a large portion of the project is defining layers, modules, their abstraction, their interfacing etc. their quality will be defined as how flexible they are with their abstraction, how universal their inputs and ouputs are, etc. All these definitions will be managed by QA.

QA will be integrated with the project management activities, and the role of *Quality Assurance Manager* will be given to one of the six team members. This role (not necessarily the assignment) has been separated from the project manager role to stress its importance and build a separate abstracted scheme. Common practice is to have a dedicated team for QA, however, since our team is small, this will be limited to one of the roles of a single member, with flexible coordination with other members for any problem solving and information gathering involved.

3.2 Virtual environment training and testing

The project is a combined effort of two teams working on hardware and software simultaneously. As such, in order to have a platform for the AI model to train until a physical platform is created, we will be using a virtual environment for the initial training. We will be using unity for the environment to create an approximate layout and model of the college area of Pulchowk Campus with approximate placements of obstacles. As well as a model of the vehicle to test AI models on. We will be using barracuda, a lightweight cross-platform Neural Networks inference library for Unity which uses Open Neural Network Exchange (ONNX) as a format for ML models. To make a close approximation to the actual real environment setup, we will be using a depth map along with the camera for the input for the model.

3.3 Real environment adaptation

Once the physical model of the vehicle is ready we will be running our AI models into the hardware setup. The area will cover the roads within the campus ground of the Pulchowk campus with input from the stereo camera, depth map, GPS and heat maps. There will be a proper feedback system from the vehicle to have an accurate output movement.

3.4 Testing new ideas

The system we propose is a modular system with different layers. As such there will be a lot of reductant tasks that different modules will perform. This system will provide a better service for testing out other higher risk modules, which will allow us to try out other modules without worrying about the failure of the entire system.

4. Proposed System Design

4.1 Layer based design

4.1.1 Purpose and advantage of layering

The main purpose of layering for our particular project is to create better defined roles for each aspect of the product, along with the appropriate abstraction, in order to make it more easily integrable in other types of products. Our goal is to maximise this flexibility and build a foundation for a large variety of future directions for the components of this project. For example, our upper level AI layers will be separate and abstracted from the lower level ML processing layers, with flexibility in the type of inputs and outputs they can process and generate. The upper level AI layer may be in charge of controlling the general direction of the movement, while the specifics of how that movement is conducted are left for the lower layers, making it compatible with all types of vehicles, robots, and even virtual agents. The lower level ML processing layer may provide its output to a variety of AI units based on the purpose.

Another important purpose of layering is to provide better interaction capability with the hardware layers of the vehicle. For example, the lower level reflex layer will be directly interacting with the actuators and control system network, and thus will benefit from having no direct connection to the decision making AI layers above in order to work faster, be independently develop-able, replaceable, expandable, and work with fewer connections required.

4.1.2 Layer hierarchical layout

Four overall layers have been defined for this particular project. Further sub-layers and interfacing layers have been omitted for simplicity in the following diagram:

[I'll doodle the diagram first]

The arrows represent the flow of information between the layers. This flow is not straightforward, since the Reflex layer only interacts with the layer below, and the Instinct layer skips a layer while sending information downwards. The purpose

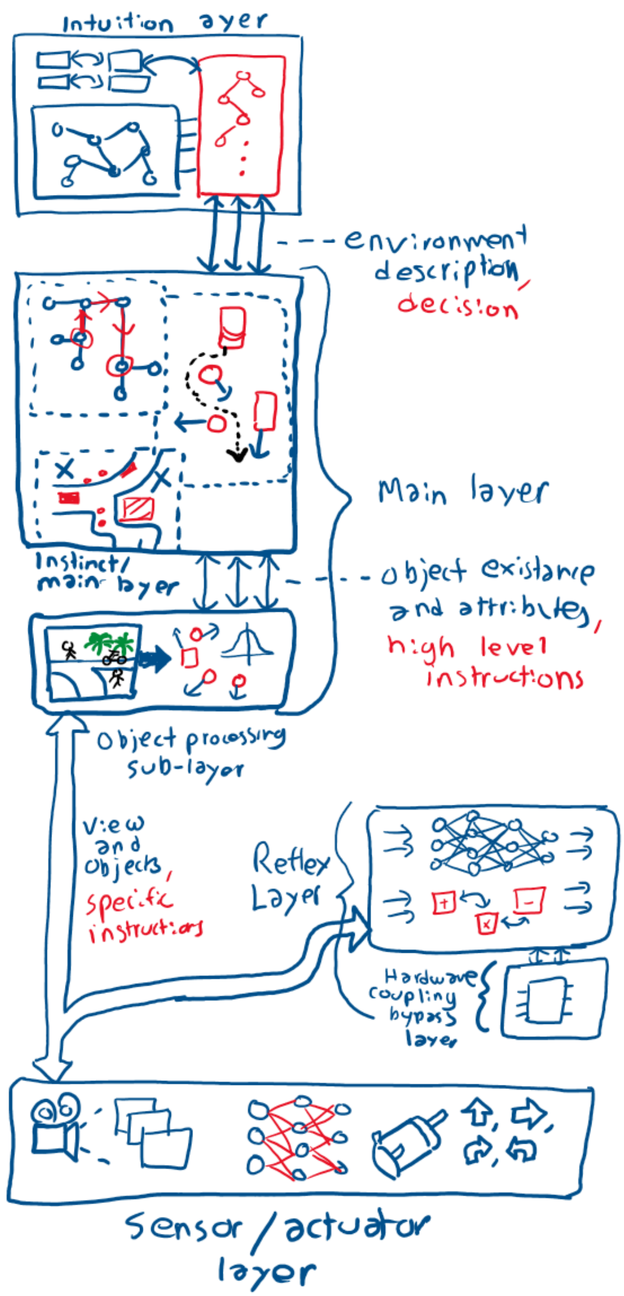


Figure 4.1: Layers of the AI system

4.2 Sensor/Actuator layer

It includes ML or non-ML based algorithms to perform quick processing of raw information provided by the sensors, as well as algorithms to properly instruct the actuators for movement. It may contain several different units running on different boards and devices for closer interaction with the sensors and actuators. The types of work they may do includes image processing, object recognition, short term instruction sequence generation, etc. For example, one of the optical sensors we may be using is a triple-camera that can detect the depth of the surroundings using parallax, and this layer will be responsible for converting the parallax information to a depth map. Image recognition may be performed using the depth map as well as the colour maps to detect where the road is, if there are obstacles in the scene and where they are, etc. This information will be provided in addition to (or in isolation if appropriate) to the upper levels.

An example for output could be a movement instruction from the above layers, such as moving to a coordinate ahead or turning about 70 degrees to the right being fed to this layer, from which it generates a sequence of steps to perform that movement based on the current state of the vehicle. If the vehicle is already moving in that direction, it may provide no output. If the vehicle is static and facing another way, it may provide a sequence of instructions to first start turning, then stop turning, then start moving and then stop moving. These instructions will be further broken down into actuator signals through algorithms in the control system network.

Information provided to this layer: Raw data from the sensors such as the camera view, ultrasonic sensors, user input, etc.

Information provided to the upper layers: Depth map, recognised objects, buffer distance, etc.

Output of this layer: activation sequence or instruction sequence for an actuator

4.3 Reflex layer

This layer includes fast response processing. For example, it may use shallow neural networks or hard coded algorithms for processing camera view, distance maps, recognised objects, etc. from the input to directly make a decision in the output for braking, turning, etc. This is useful for situations where there isn't enough time for the slower, upper level decision makers to respond. This layer may also have multiple units working in multiple boards, for tight coupling with the hardware. Basic obstacle avoidance by stopping upon sensing a close object in the ultrasound proximity sensors is considered part of this layer, despite that containing very little information processing.

Information provided to this layer: Raw information about the environment, along with some processed information from the sensor/actuator layer wherever appropriate

Output of this layer: Instruction for stopping, evading movement, etc. (which will be converted into an activation sequence by the sensor/actuator layer)

4.4 Instinct/Main layer

This layer will contain a single instance in the system, and is responsible for higher level processing and decision making. Being made of a large collection of ML and non-ML algorithms, many of the working in parallel, it will take in information from the sensor layer including camera view, recognised objects, their distances, the vehicle's position, orientation, etc. to build a 3D reconstruction of its immediate environment. It will use statistical algorithms to define the validity of the environment's elements' existence as well as their physical extent. All this processed information will be crucial to provide data for processing and generating both small and large time scale plans for movement. It will provide a relatively high level of instructions to the sensor/actuator layer, which will further break it down into more specific instructions. Examples of these instructions may be movement destination coordinates, orientation target, etc.

The goal of the resultant autonomous vehicle in this project differs from the goal of OpenPilot. While OpenPilot focuses on lane detection and keeping the vehicle in the lane, our working environment does not include open roads, but rather a complex layout of buildings and paths. Our focus is thus more on source to destination movement rather than controlled high speed movement. Hence, our approach takes inspiration from but differs from OpenPilot in many crucial ways.

For example, OpenPilot focuses on end-to-end processing, creating a single black box that takes in input and outputs signals to turn the steering, adjust the throttle, etc. This allows them to have easier training, simpler design, and less danger of error propagation.

Our case is more similar to Tesla's approach, with different modules tasked with different parts of the processing. There are not much details available for the layout of such a system, however, hence we are taking on the challenge of designing our own system.

Information provided to this layer: Whatever is provided by the Sensor/Actuator layer Output: High level instructions for action Information provided to the upper layer: A description about the 3D structure of the current surroundings, speeds of moving objects, etc. as well as output (formatted as instinctual information)

4.4.1 Working Principle

The information from the environment is used to fill in the awareness environment of the entity. The awareness environment consists of a 3D reconstruction of the environment in terms the entity understands. This may includes objects such as Available Path, Obstacle, Fast-Moving-Obstacle, Slow-Moving-Obstacle, Person, etc. each with their own attributes such as speed, position, etc. This information is kept realistic and static by statistical analysis of what it is observing. For example, if an object is detected for only a few disparate frames, it can be inferred that it must be an error and does not actually exist.

With the awareness environment created, algorithms can be used to determine the roughly estimated path to the destination, based on coordinates provided. More precise path estimations can be done for more immediate sub-destinations, requiring dodging obstacles and predicting their movement. This is expected to be a mix of traditional and Machine Learning algorithms, mostly based on the types of training data that can and cannot be gathered, simulated, augmented, or generated.

The architectures of the algorithms and machine learning models involved will be determined along the development of the project, as well as through testing and experimentation.

4.5 Intuition layer

The process of thinking and logical extrapolation is not yet understood for the human (or for that matter, any animal) brain. However, the differences between a human brain and most AI agents are to some extent visible. For example, AI agents usually do not have separate memories and imagination spaces, while humans seem to. Inspirations can be taken from such aspects and differences to build a rough emulation of how the thinking process works for agents such as humans.

This is the inspiration for the development of such an AI agent, making heavy use of graph neural networks and clausal logic. The hope is for this system to be able to solve problems as well or even better than the Instinct/Main layer.

4.5.1 Working Principle

It will work more similar to a language model rather than an image processing model - with its own vocabulary as a graph data structure, with definitions constructed from linking the nodes using logical flows such as “and”, “or”, “implies”, etc. These logical connections will have a strength value each, which will act as trainable parameters to modify its understanding of the world and flow of logic using something similar to reinforcement learning. It will have a context memory which will be used to gather input about the surroundings (in terms of its own vocabulary), for processing that information by traversing along the logical connections and making probabilistic conclusions, and for describing a solution as the output (again, in terms of its own vocabulary). Translator sub-layers will be used for interfacing this vocabulary based information from and to the various other formats the lower levels will work with.

5. Proposed Experimental Setup

The experimental setup will include both the environment for testing and development of the vehicle, and the scoring systems for QA. The former will be used to make decisions about whether or not to include, change, or expand upon the current implementation of any particular module, while the latter will be used to decide what the module should accomplish for a good contribution to the entire system.

5.1 Real environment layout

The working environment of our autonomous vehicle has been limited to the confines of Pulchowk campus. This is for both permission reasons, and to better suit the goal of the vehicle - to be able to find good paths around complex layouts of buildings and roads, rather than to stay in a controlled high speed state in a single lane.

The environment consists of roads without lanes acting as pathways to various buildings for different departments. Obstacles include people walking, other vehicles such as cars and motorcycles moving, trees, animals such as dogs, sudden height differences, puddles of water, etc. The area is around 2 square kilometers, consisting of two sections separated by a main road (which we are not allowed into).

The roads are relatively smooth and easy to navigate around, and the people are receptive to experimental setups, and usually give way. This should make it easier for us to build and develop a system, and cut out many of the deal breaking edge cases (such as high speed mishaps)

The vehicle will have prior knowledge about the layout of this environment, which we will acquire and build from 3D scans, satellite mapping, etc. and will be able to position itself and other objects more accurately than on the unknown open road. This should also make it easier to make the project work successfully.

5.2 Virtual environment layout

The virtual layout (referred to in the previous sections as the 'awareness environment') is the representation of that prior knowledge the vehicle has of the environment. Before the vehicle is actually built and in working condition, we will need to develop the working algorithms, which will be done to some extent in the virtual environment. This environment will be set up in terms of objects the agent will understand, such as paths, obstacles, fast moving objects, slow moving objects, etc. which should be available to generalise and simulate and hopefully prevent over-fitting. The training process in such an environment is described below.

The different layers of the AI will be subjected to different levels of detail of the virtual environment. This makes it easier to build training schemes and keep the size of training data manageable, as well as make training separate for the different layers, which would be more possible to divide to different computers, Google Colab accounts, etc.

5.3 Training and testing

5.3.1 Sensor/Actuator layer

The majority of training of the Sensor/Actuator layer can only be performed when the vehicle is in working condition in the real environment. Thus, this is expected to be done in the latter half of the project development cycle.

The inputs to be dealt with by this layer is difficult to simulate. However, prior training will be performed in the approximated 3D scan of the environment to provide a good starting position for the real environment training, due to limitations in training times, physical presence, etc.

The outputs given by this layer may include its inputs. As a result, the Instinct/Main layer has been divided into two sub-layers for further abstraction, and will be explained below.

5.3.2 Reflex layer

The reflex layer will mostly consist of non-ML algorithms which do not need to be trained. For training the rest however, it is not expected to require a lot of data, hence the real environment training situation should be enough.

Since this layer does not provide complex information to the above layers, and can override their decisions, it can be trained separately with flexibility with timing.

5.3.3 Instinct/Main layer

The object processing sub-layer of this layer will be trained in a similar way to the Sensor/Actuator layer, with the focus being on the real environment, but with good prior training in a virtual approximation.

The main sub-layer will be provided relatively high quality simulated training data, and has potential for large scale thorough training. This will likely be performed before the training and development of the lower layers, with extensive testing and research.

5.3.4 Intuition layer

Since the Instinct/Main layer will be capable of driving the vehicle on its own, the Intuition layer will be given its decisions as high quality training data for training.

The development of the language model and vocabulary of this model will be performed in a generalised, virtual environment, with the help of guesswork and reinforcement learning. The reinforcement learning may be aided with existing language models to provide good quality feedback.

6. Timeline

The various works needed to for the completion of this project are divided into 10 months duration starting from June of 2023. These tasks can be viewed in the Gantt Chart below:

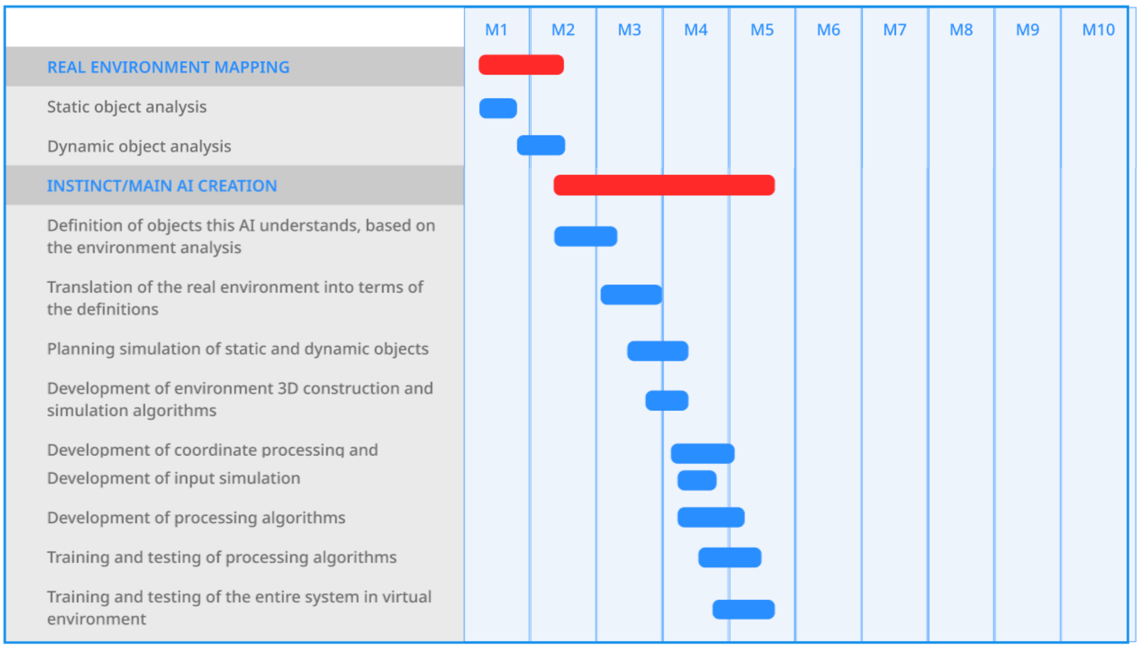


Figure 6.1: Gantt Chart (First Half)

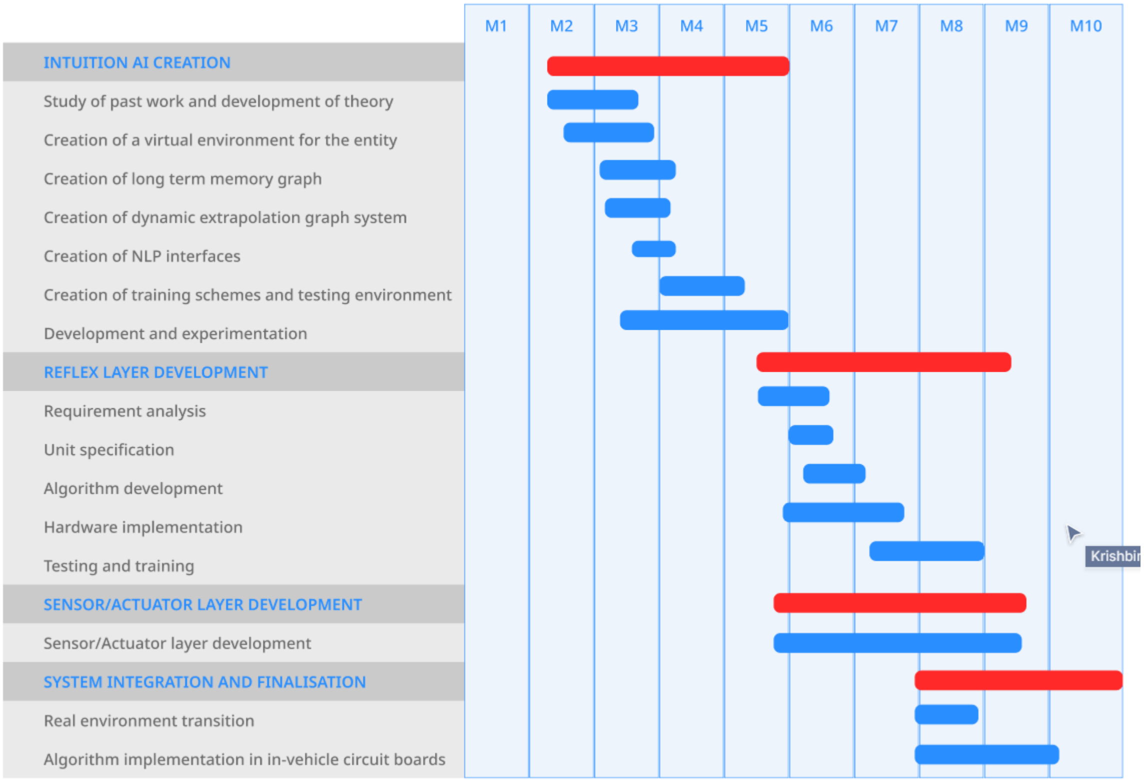


Figure 6.2: Gantt Chart (Second Half)

References

[1] How openpilot works in 2021. https://blog.comma.ai/openpilot-in-2021/, 2021. [Accessed 20-Jun-2023].

[2] Review of comma.ai- what are the pros and cons of ai for automobiles. https://aiplainly.com/blog/comma-ai-review/, 2023. [Accessed 20-Jun-2023].

[3] Mankaran Singh. End-to-end motion planning with deep learning. https://mankaran32.medium.com/end-to-end-motion-planning-with-deep-learning-comma-ais-approach-5886268515d3, 2020. [Accessed 20-Jun-2023].