CISC 322/326
Assignment 3: Apollo Architecture Enhancement
April 11, 2022

Sebastian Wakefield	19sslw@queensu.ca
Dominic Guzzo	18dpg3@queensu.ca
Oscar Wojtal	18ow2@queensu.ca
Alice Cehic	17ac121@queensu.ca
Cooper Lloyd	17cel4@queensu.ca
Daniil Pavlov	17dp15@queensu.ca

Abstract

In this report, we propose an enhancement to the ApolloAuto system which would enable the possibility of driving on unmarked roads. This presents passengers with greater flexibility in deciding their route. We propose two approaches for implementing the off-road feature. The first approach, would entail adding a new way to route the car using satellite images and additions to the Map, Routing, Localization, Storytelling, and Planningmodules. The second approach would be to include all new Routing, Storytelling and Planning subsystems and add to the Prediction module. Both approaches involve similar additions to the HMI, Planning, Storytelling and Localization modules and have different effects on our chosen non-functional requirements. We mention the new interactions between modules in our enhancement and what architectural styles would be best suited for this system. Testing methods for the enhancement are also mentioned in detail, as well as the potential risks in creating this feature.

The SAAM analysis is used to determine which of the two approaches proposed is optimal by analysing how each approach affects our non-functional requirements. This is done by comparing the approaches based on the performance, evolvability, testability and maintainability attributes. Each approach and attribute is also analyzed based on the stakeholders: Apollo Users, the development team, Apollo, and Apollo Shareholders. As a result of our SAAM analysis, we have concluded that the first approach is optimal since it has a more efficient use of the map module and find the optimal path much faster. Finally, the benefits of using approach one to implement the enhancement is demonstrated through the use cases of initiating off-road functionality and rerouting once the current rout eis found to be impassable.

Proposed Feature - Off-road autonomy

Overview

The feature we are suggesting is the ability of the Apollo software to perform off-road autonomous driving, so that the vehicle can operate on unknown roads or trails. This will mainly be used for areas where the mapping of the road cannot be identified by the Map or Routing modules and therefore there is a need for off-road improvisation to get to the specified waypoint. The vehicle will be able to move through trails, paths, and non-urban roads using just perception and localization information, without the need for prior knowledge of the roads obtained from map APIs. Given a waypoint in an off-road area, the vehicle should be able to navigate its way to the destination while overcoming most terrains, depending on the severity of the terrain and the ability of the vehicle. If the vehicle is unable to make its way through the path due to a dead-end or impassable terrain, the vehicle should stop, notify the user, and turn back if requested. This feature will provide an enhanced user experience as the passenger can now go to any destination accessible by their vehicle. Not just urban areas that have roads tracked by the map API. This addition will also allow for later use of this software on smaller motor vehicles that often drive off-road, such as dirt bikes and ATVs. It

will also be of great use to those living in more remote locations around the world who seldom drive on city roads.

Functional Requirements

We identified several functional requirements for our architectural enhancement. Once the user selects their waypoint on the HMI, the system should be able to compute whether off-road functionality is needed and if the destination is reachable. Assuming the system believes the destination to be reachable, the system should be able to recognize possible paths and gauge whether it will lead to an optimized route to the waypoint. The system should take into account current terrain while driving and whether the terrain can be overcome by the vehicle. Some of the terrain that the vehicle should be able to recognize is swamps, ponds, fallen trees, and mud. The terrain information for the specific area, especially unexpected terrain data, should be noted and saved for later use. If the terrain cannot be overcome, then it should notify the user through the HMI and suggest secondary destination options in the radius of the current unreachable waypoint. In addition, the system should be able to identify trees, bushes, logs, and common animals, as well as predict their future trajectory and movements.

Non-functional Requirements

We must address various non-functional requirements before integrating our Off-road enhancement functionality into the Apollo software. Reliability is one of the most essential non-functional requirements for our proposed feature. When working with autonomous vehicles, we must be able to trust the software to properly control and maneuver the vehicle in order to prevent crashes, risky maneuvers, and loss of control, so this requirement must be carefully considered. Given that our new feature is an enhancement to the already existing Apollo software, it should not have a significant impact on processing or runtime; the time taken to receive routing information after off-road initiation should be less than 500 milliseconds. Furthermore, the off-roading functionality should not introduce new vulnerabilities; however, we must ensure that our feature is compatible with Apollo's existing vehicle cyber security solutions, such as Secure IVI, IDPS, and CANScan. For the maintainability NFR, our feature is an addition to the existing software and should be smoothly integrated into the existing Apollo software to enable for future maintenance and innovation in next versions of the software. We will thoroughly document our work on the off-roading feature to make it easier to maintain if something goes wrong or it needs to be revised. Reviewing the Managerial requirements, a realistic timeline for our enhancement feature to be complete would be one to two years. This time will be used to thoroughly test and collect data in order to train the autonomous vehicle to negotiate off-road terrain safely and reliably. To avoid anything going wrong, we must also establish and follow strict safety procedures.

The requirements for the reliability of the system are the same as for a conventional automobile, that is, some basic sensors must be operational, they must be able to send valid data to Perception. The speed of the autonomous vehicle's response to changing conditions must be sufficient, that is, the latency of the entire chain from recognition to Control should be continuously measured. If it is impossible to continue movement, the vehicle must safely stop, and it must be

possible to send a signal with its coordinates and information about the current situation to a remote service center.

Thus, the Maintainability of the system changes as follows:

1) Additional sensors for terrain recognition in front and below the bumper.
2) The ability to send information to a remote service center and the ability to receive control commands.
3) The possibility of inertial estimation of the position of the vehicle.
4) The possibility of learning the way out of an incomprehensible situation based on the choice of a pilot.
5) Extended requirements for the recognition system since objects that can be damaged in an off-road environment are much more diverse than in an urban environment.

Testing

Off-road initiation use case: Input an off-road location as a destination parameter to the Routing module and verify that it calls the proper functions and classes to initiate off-road functionality. The system should return numerous possible routes that the vehicle may take or notify the user that the destination is not reachable.

Optimized route detection use case: After returning the possible paths to reach the destination and choosing the most efficient path, we must verify that this is indeed the shortest path to the destination with the lowest possibility of bad terrain. We will have to make sure the system considers both the distance to the destination and the reliability of the terrain.

Detect impassable terrain use case: Drive the car on various terrains to test if the Planning, Storytelling, and Prediction subsystems can detect circumstances such as muddy terrain, swamps, dead-ends, and elevation. These modules should interact with each other to output the reliability of the environment and calculate the possibility that the vehicle can make it through. We will test the accuracy of these predictions using multiple environments with many obstacles to refine the scenarios we check for.

Reroute use case: Giving the Routing module an unreachable destination as a parameter, we should test whether the system notices that it is unreachable and that alternative waypoints are provided to the user via the HMI. These alternative waypoints should be within a reasonable distance from the initial waypoint and be accessible by vehicle. After waiting for user selection, the routing process should commence once more.

Potential Risks

The proposed off-road system must be reliable in the sense that the failure, malfunctioning, or incorrect operation of individual components must not lead to accidents, for example, collision of the autonomous vehicle with obstacles, the vehicle getting into a situation where the vehicle itself, passenger(s), or cargo will be damaged, a situation when the vehicle gets stuck and cannot move

without help, or a situation where nature or property is damaged. This could be prevented with three possible conditions:

1) The vehicle must not continue moving in a situation where the Apollo system cannot identify movement in a certain direction as completely safe. For instance, a situation where Apollo is not able to identify the terrain in front of the automobile because, for example, the sensors cannot indicate the height difference down in the direction of movement (of a cliff, steep slope, or the depth of the water body being crossed). Also, it is necessary to consider the impossibility of automatically determining the characteristics of the surfaces in front of the vehicle that look like solid ones, such as sand, a swamp, a frozen water body, or snow. Such surfaces can be both a threat to the mobility of the vehicle and a potential hazard to the passenger(s) or cargo.

2) The vehicle must return to the starting point if the amount of fuel left is only enough to return to the starting point, and the movement to the endpoint cannot be accurately predicted. The autonomous vehicle must consider the remaining amount of fuel to ensure the safe return of the automobile to the starting point. The vehicle movement must be recorded from the starting point using satellite navigation (GNSS) or inertial sensors (gyroscope, compass).

3) In case it is impossible for Apollo to recognize stationary or moving objects along the route, the system must require a reaction from the pilot (located either in the vehicle or working with the automobile remotely) to decide on the possibility of further movement and, if this reaction is not received, a safe stop or return to the starting point must be made. A useful feature would be remembering the pilot's choice and using that choice in similar situations, but this still could potentially lead to accidents. People along the route, animals, or elements of the cultural landscape (fences, garden beds, etc.) must be clearly recognized.

Proposed Approaches

Both approaches contain similar additions to the HMI, Planning, Storytelling, Prediction, and Localization modules, but have key differences when it comes to Routing and Mapping. Our first approach uses satellite data to construct maps of the off-road terrain which are then used to help find an optimized path to the destination. Our second approach instead relies solely on perception data from the cameras on the vehicle to analyze possible paths as it detects them.

1. Our first approach would include changes to the Map, HMI, Routing, Planning, Perception, Storytelling, Prediction, and Localization modules. The Map module operates by housing a base_map that allows other functions to adapt it for different purposes, such as the routing_map which only retains road geometry. For our enhancement, the Map module will create a new map variant off-road_map that is calculated when the driver initiates an off-road route. Off-road_map will use satellite images along with topographical maps in order to provide the car with better terrain information where available. A forest_map could also be implemented using satellite images of the forest in the winter and in the summer to calculate how wide the trunks of trees are, creating dead zones around the trunks and any obstacles on the floor to later aid in routing through the forest. There is potential in the future to

incorporate machine learning into this process so that the input to forest_map could be as many pictures of the area as available, and the goal would be to effectively “peek” through the forest canopy by piecing together open spots, unique to each photo. HMI will also need to be updated to allow the user to set waypoints off of known urban roads - this will then send destination information to the Routing and Planning modules so an optimized path can be found. A new submodule will have to be incorporated into the Routing module to process the new off-road_map. When the driver places a waypoint, Routing will consult the off-road or forest_map to check if reaching the destination is a possibility. If the destination is impassible, i.e. route is too steep or there is a river, the driver should be notified and proposed plausible alternatives within a certain radius of the original waypoint. Next, Routing will attempt to find a possible route by using the information from off-road_map. If such a route exists, then the journey can continue. If not, the user should be notified that the destination may not be reachable, but should be given the option to continue. The Planning module will have to incorporate new scenarios to fit the scheme of off-roading and traversing through the forest. This might be best implemented through a new submodule since Planning currently houses exclusively city driving scenarios. A new submodule would give Planning the flexibility to eventually add more scenarios enhancing off-road capabilities. The normal off-road scenario will keep track of possible routes as the car is driving, and in the event, a dead-end is reached, the dead-end scenario will route back to the most likely paths planning had encountered on its way to its current position. The Storytelling module will have to adapt to the changes in Map and localization when coordinating cross-module interactions. Prediction will have to be tuned to handle the tighter spaces while driving in the forest, additionally turning off the prediction for trees planted in the ground as they do not move and will only waste precious latency and computing power needlessly. Perception would have a perspective switch when entering an off-road scenario where it will percept its environment into two categories, the first being ground objects and texture so that the car will know how it'll behave when driving over them. Along with perceiving objects on the ground that the car should not drive over objects such as sharp rocks or deep holes. The second category should be objects that the car can collide with, such as trees, bushes, and possibly common forest animals. Localization will have to change the way it perceives the car compared to its surroundings since there is no road boundary. This can be achieved by calculating the car's last known position summed with the vectors it has traversed previously. The vectors are comprised of distance, which is derived from tire rotations and the direction, which is derived through an internal compass. Uphill\downhill calculations can be factored in with a gyroscope (i.e. 200 meters travelled while the car is at a 45-degree angle means 100 meters actual distance). This information in conjunction with GPS and any other data should give localization an accurate estimation.

2. Our second approach to implementing this new enhancement is to include three entirely new Routing, Storytelling, and Planning subsystems for off-road. With additions to other modules such as the Prediction module for the off-road functionality of the system. These new Routing, Storytelling, and Planning modules overlap and depend on each other greatly. So, we believe they should be split into 3 separate folders within a new module called *off-road.*

We would use the new storytelling subsystem to keep track of new scenarios that only apply to off-road driving, such as scenarios where the terrain is too thick to drive through, the path is too narrow, or the ground is not adequately solid. Almost all the current scenarios in Apollo's Storytelling module relate to urban areas where there are intersections, crosswalks, stop signs, and more in *modules/storytelling/story_tellers.* These scenarios would not be useful for the car's autonomous off-road experience; therefore, a new module may be needed to keep track of these new stories so that modules subscribing to it can properly assess the situation at hand. The new planning subsystem will have extra methods to decide certain things like terrain harshness and will not need most of the functionality currently in the Planning module that deals with urban streets, such as traffic lights and stop signs. It will change the way the optimizers are run in *modules/planning/tasks/optimizers*; instead of creating a road graph beforehand using the map module, it will construct this road graph as it moves through the road and attains data from perception devices, thus, it would no longer depend on the Map module. It will also have many added scenarios to handle off-road driving like tracking traction constantly and weight distribution to see if the car can handle the terrain and adjust accordingly. The current Routing module creates a topology graph and optimizes the path toward the destination using nodes on the graph given by the Map module. The new Routing module for the off-road enhancement can no longer rely on this graph since the route will not be recognized by the Map module and the road graph cannot be created ahead of time. Therefore, the autonomy of the car will have to operate in a branch-and-bound algorithmic way where it accounts for the cost of every possible path as it views it using the Perception and Prediction modules. This new module will have to send the car down the current path while it gathers information from the perception devices to create a topology graph of the road. The additions to the prediction module will entail editing the *modules/prediction/evaluator* folder to predict new obstacles, such as common forest animals.

Concurrency & Architectural Style

The additions in each approach are very similar, except for the actual routing functionality, as approach one uses satellite images from the Map module to route and approach 2 uses perception data from the Perception module. After the user selects a waypoint outside the path of known roads on the detailed map displayed by the HMI, off-road routing can be initiated and planned. This waypoint information inputted into the frontend of HMI will be stored in the backend and sent to the Planning module and new off-road Routing module. These two modules will interact and send data back and forth. The Planning module will publish data to help plan the trajectory for the Routing module, and the Routing module will publish data to be used by the Planning module. The Routing module will also subscribe to satellite data coming from the Map module to create paths using approach one. For approach two, the Routing module will subscribe to the output of the Perception module to look for possible paths. The Planning module will need to subscribe to localization and perception information gathered by the system so that it can make accurate decisions. The Planning module will also share this information with the Storytelling module to help create off-road scenarios and plan routes. Perception information of obstacles additionally needs to be predicted, as the vehicle must know what kind of obstacle is in the way and where the object is moving.

Therefore, the Prediction module must subscribe to the Perception module, and the Planning module must subscribe to the output of the Prediction module.

Based on the concurrency of this system, a publish-subscribe architecture will still provide a suitable means of sending and receiving data so that the vehicle can run autonomously off-road. Numerous modules will need to broadcast their outputs so that the new off-road planning module can optimize the path to the destination. Moreover, data that is subscribed to by numerous modules, such as perception data, can be subscribed to by many modules using this architectural style and can support reuse between modules. Publish-Subscribe style also allows for excellent system evolution should it evolve to handle off-road vehicles such as ATVs since other components may replace components without affecting the interfaces of other components in the system.

SAAM Analysis

Attribute	**Approach 1**	**Approach 2**
Performance	Although this approach would initially have a slower response time due to the use of satellite images, the results would be more accurate and there would be minimal processing later. The initial route would have better reliability.	This approach is lacking in that it uses a branch-and-bound routing method and only accounts for paths once the cameras perceive them instead of using a satellite image to assess all paths beforehand. With so many extra calculations and path queries, the accumulated latency would have a bad effect on performance.
Evolvability	Great potential for evolvability, as there can be numerous additions to the type of map used by the Routing module to find routes. Currently, our enhancement has an off-road_map and forest_map, but there is potential for various others that look at different terrains and environments.	Good potential for evolvability other than the routing system inside the new off_road module. This system is limited by an improvised branch-and-bound approach and does not have much room to evolve.
Testability	This approach provides software that is easier to test because it looks at paths using satellite images and thus finds all possible paths beforehand - saving the quality assurance team a lot of time and avoiding the use of excessive perception data.	Testability for this approach is less than ideal because paths are analyzed as the car moves through them and not beforehand like approach one. Therefore, lots of perception data would have to be inputted to the system to ensure proper testing.
Maintainability	Since all there are so many software additions to various modules, the maintenance could become cumbersome.	Based on the localization of the added software in the off_road module, this approach will be easily maintainable.
Security	There are numerous security vulnerabilities in this approach because of the new forest_map and	This approach is less vulnerable to security threats. Regardless, cybersecurity is not optimized for any current autonomous

	off-road_map additions. These new maps could lead to security vulnerabilities if not implemented correctly and rely heavily on external devices.	driving software, as there are many points of intrusion.

Attribute	**Apollo User**	**Development Team**	**Apollo**	**Apollo Shareholder**
Performance	Performance is the most important metric for general users, people will be frustrated.	The performance of the off-road routing system has to match the performance of the normal routing system for consistency.	Apollo is expected to make high-quality autonomous driving software that beats competition around the globe. Performance is a key metric to compare themselves against the competition.	Performance heavily impacts the popularity of the software as well as the value of the company. High performance could mean being a step closer to becoming the go-to autonomous driving solution.
Evolvability	It is important for the user to notice new additions and software updates so that they believe the value of their purchase is increasing.	There must be room for evolvability in the software so improvements can be made and performance improved.	Apollo must continue to improve its system to keep up with the growing competition new discoveries in the industry.	Evolvability of the system is important for long-term industry domination and consistent profits.
Testability	Testability is vital when issues arise with a user's system. Good testability ensures the issues are solved efficiently without long wait periods.	Testability is very important for quick development time and scheduling since each feature needs to be tested thoroughly.	Testability ensures that the software has minimal flaws and is safe for any user. This also makes sure that any issues that come up can be found and solved quickly.	Testability of the system ensures that new features can be outputted quicker and it is easier to catch up with the competition in the industry and maintain market share.
Maintainability	Maintainability allows for new features to be added with ease on top of old features. This creates consistently added	Maintainable software means that the development team can add new features a lot faster and save time with	As software is added to the system over many years, a lack of maintainability will lead to hard to find bugs and will	Research and Development will be slowed with low maintainability and sped up with easily maintainable code. R&D is a vital

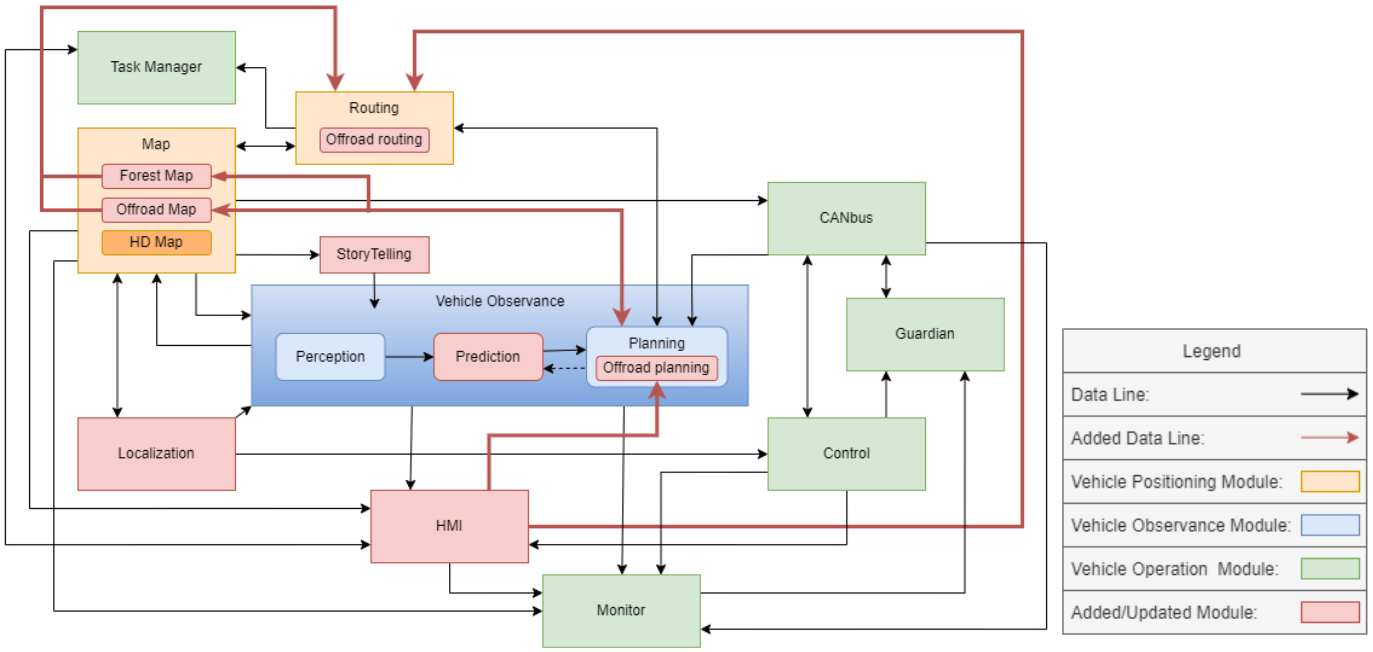
	value for the purchaser.	new projects built upon legacy code.	cause long-term development issues. This will hurt productivity and incur a lot of refactoring.	sector for making profits and increasing publicity in the market.
Security	Security is among the most important factors for users, especially tech-savvy users. This is sometimes even more important than the performance of the software depending on the user.	Cybersecurity is a huge concern for the development team and lots of testing will be done by quality assurance analysts to ensure the software cannot be penetrated.	Apollo must uphold a reputation of creating secure products to keep customers interested. Customers tend to avoid products that could lead to their private information being leaked.	Security issues with the system can create negative news about the company and thus decrease the value of Apollo's share.

Chosen Approach

Approach one allows for faster performance as the implementation of offroad_map allows the system to make faster decisions when already on the trail. It can cross-reference the map's obstacles and impassable areas with potential paths while en route in order to know which alternative paths would not work and which ones are most likely to succeed as it passes them. Approach two uses a branch and bound method where the vehicle will only know the outcome of a path once it has taken it. In approach one, the disadvantage of importing and processing the off-road maps at the beginning of an off-road session were negligible as this would allow the systems resources to focus on latency and response times when it is needed later in the route. Approach one also provides better evolvability than approach two as the revised modules will incorporate submodules to house off-roading scenarios and information. This would allow for easy future expansion into the domain as scenarios can easily be appended when it is time to expand the system. Approach two is similar in this aspect. Though, it is hindered by its routing modules branch-and-bound algorithm as this will be difficult to rework when the system evolves and will have to be entirely replaced. The testability of approach one is superior to approach two as the team will be able to test the Map and Routing modules virtually, meaning the developers can start seeing where problems are emerging early, and refine it much more before real-life tests. This is not possible with approach two as paths are analyzed as the car moves through them, meaning the team would have to create entire virtual settings if they would like to test the feature before real-life scenarios. Approach two provides better maintainability as most of the code changes are compartmentalized into the off_road module. Approach one does not carry the same benefit as many modules will have to be modified, whose complexity will negatively affect the maintainability. Approach one's greatest drawback is the vulnerabilities associated with importing photos for its offroad_map feature. If not implemented correctly such as verifying that the images come from a credible source, it would only serve as another point of intrusion. This would usually greatly impact

our decision, but as long as basic precautions are met, it does not make a big difference in our scenario as cybersecurity is not optimized for any current autonomous driving software.

Affected Directories and Files



Map module

The Map module (*apollo/modules/map*) will include two additional directories to help with off-road routing called *offroad_map* and *forest_map*. Both maps will be constructed using satellite images along with GNSS to attain high-definition images of the terrain and area so that routes to the destination can be found. Offroad_map will be used to analyze routes and the environment in open spaces, while forest_map will be mainly used to find paths through dense forests and find dead zones. The satellite images and perception devices will help the system use these maps in conjunction to create an accurate portrayal of the environment and help the Routing module construct paths.

Routing module

The Routing module (*apollo/modules/routing*) will need the same inputs of map data and start and end locations, but will have an additional folder to help with off-road routing. The current Routing module uses a folder (*apollo/modules/routing/topo_creator*) to help create a topographical representation of the routes using the HD map. For off-road, we will need an additional folder to construct this graph using the offroad_map and forest_map instead. The type of input data will change significantly and therefore, we need a new folder to construct the nodes and edges of the topology graph.

Planning module

The Planning module (*apollo/modules/planning*) will need additions to detect and plan for new off-road environments as well as additions for route planning. Many factors for off-road driving are not accounted for in *apollo/modules/planning*/*scenarios*, such as muddy terrain, swamps, thick bush, etc. There needs to be major upgrades to this folder to handle these new scenarios and plan accordingly. Moreover, planning navigation will also need to change, as navigation planning also needs to consider the reliability of each route in terms of possible environmental issues. It needs to analyze these extra elements along with the length of each path to make strategic decisions on path choices. Finally, there needs to be changes to the current *apollo/modules/planning*/*reference_line* folder which helps localize the car in relation to the current road boundary. Since the road boundary detection and consistency is much different in an off-road scenario, forest_map and offroad_map need to be used to help find these boundaries instead of the HD map.

Localization module

The Localization module (*apollo/modules/localization*) will need minor modifications to localize the car during off-road autonomy. Localization will have to change the way it perceives the car compared to its surroundings since there is no road boundary. This can be achieved by calculating the car's last known position summed with the vectors it has traversed previously. The vectors are comprised of distance, which is derived from tire rotations and the direction, which is derived through an internal compass. Uphill\downhill calculations can be factored in with a gyroscope (i.e. 200 meters travelled while the car is at a 45-degree angle means 100 meters actual distance). This information in conjunction with GPS, IMU, GNSS, and data from the Map module should give localization an accurate estimation.

HMI module

The HMI module (*apollo/modules/dreamview*) will have adjustments in the way that a user can select a destination on the dashboard interface. Instead of solely having address input, the user should be able to select an off-road location on the map using the touch screen.

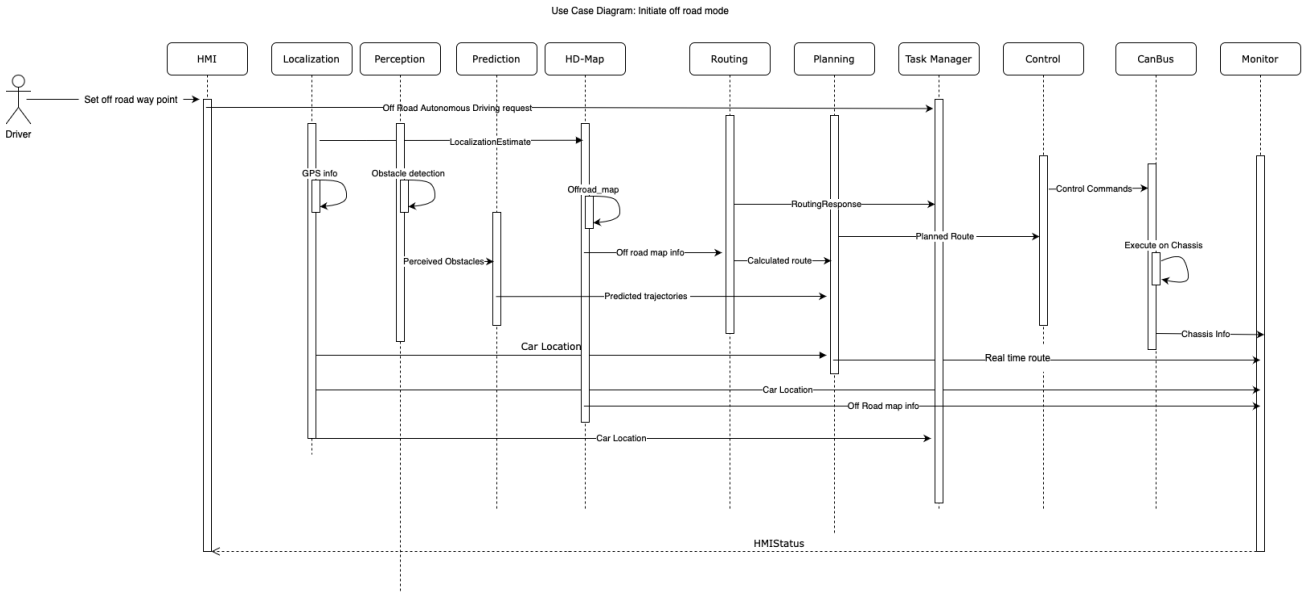
Prediction module

The Prediction module (*apollo/modules/prediction*) will need additions to predict the movement of more objects that could become obstacles during off-road driving. It will also have to be tuned to handle the tighter spaces while driving in the forest, additionally turning off the prediction for trees planted in the ground as they do not move and will only waste precious latency and computing power needlessly.

Use Cases

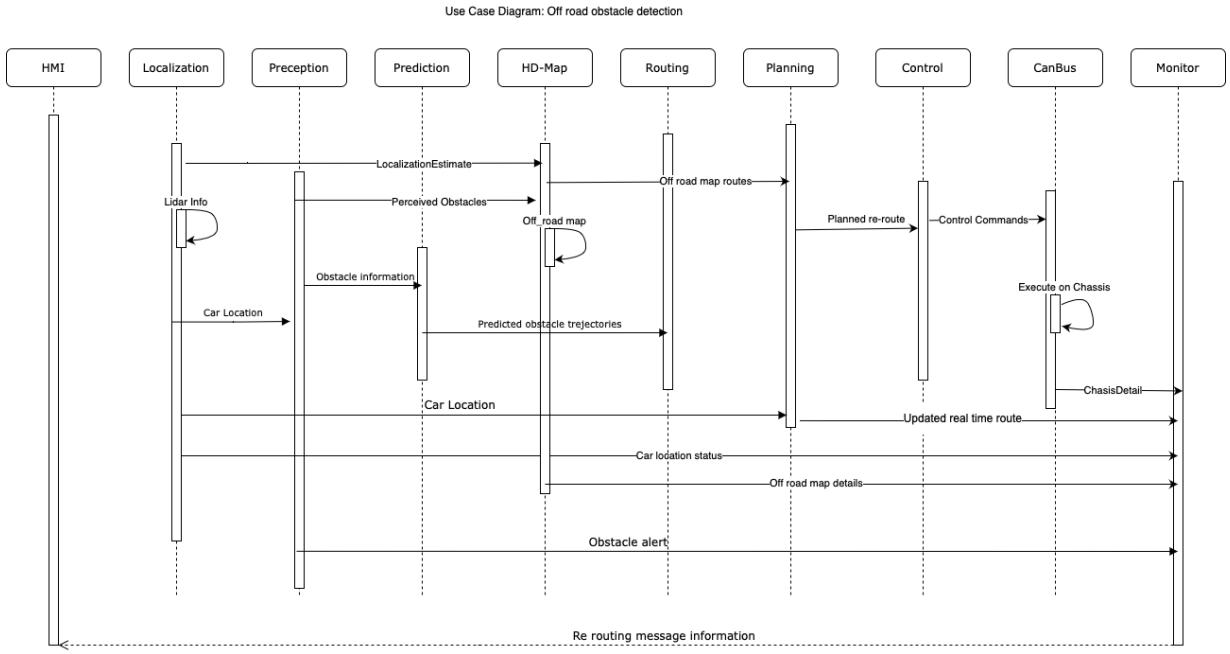
Off-road mode initiated

Description: Once the driver selects a waypoint on the HMI that is outside of an urban area where there are no known roads, off-road mode is planned and initiated. The system finds a list of possible paths to the off-road destination after being given waypoint information and selects the optimized path to send the car.



Detection of impassable terrain while on route

Description: During an autonomous drive down an optimized path towards an off-road waypoint, the vehicle encounters some harsh terrain, such as mud or a swampy area. This terrain is analyzed by the system and decided if the car can make it through. If the car is not able to overpass the terrain, it alerts the user via the HMI and reroutes to the next best path.



Data Dictionary

Smarter Eye	“A driving assistant system which provides users with automobile autopilot systems. It uses different technologies such as three-dimensional depth of vision, image processing, video analysis, pattern recognition, and data mining.” (tracxn.com)

Publish-Subscribe Architecture	An architectural style that allows independently implemented system components to register themselves as a publisher or subscriber for a particular event or message channel. That is, the publisher produces an event or message to the channel, and the subscriber receives that information.
Conceptual Architecture	Describes the system in a general way, without going into implementation details, but describing the requirements for the system and approaches to solving them. For example, a description of a set of system modules and the nature of their interaction without a specific description of the implementation of the modules and details of their interaction, such as data structures.
Ultrasonic Radar	A device for determining distance using reflected ultrasonic waves. The Ultrasonic Radar devices are significantly cheaper than LiDAR devices but give significantly less accurate results.

Naming Conventions

GPS	Global Positioning System
HMI	Human Machine Interface
LiDAR	Light Detection and Ranging
GNSS	Global Navigation Satellite System
RTK	Real-Time Kinematics
CANBus	Controller Area Network Bus

Limitations & Lessons Learned

Limitations:

- There were limited resources on our subject matter (autonomous off-road driving) for our team to reference. This made it very challenging to think of plausible approaches to implement this feature into Apollo's system. The only resource we could find was a research article by NASA that was a bit out of our scope but had some great ideas we could build off of.

- Some of our additions to the software are out of the scope of our understanding, such as the machine learning algorithms needed to detect different terrain types. We were not able to go into detail about how a lot of the planning and prediction functionality would be implemented because of our lack of background knowledge of computer vision.

Lessons Learned:

- We learned how to create a SAAM analysis through this project and how to use it to assess different architectural approaches. Initially, our group thought the simplest approach would be the best, but after completing the SAAM analysis for all our NFRs,

we realized that this was not the case. The analysis forced us to look at factors we had not considered and gave us a rich view of what risks and benefits each approach had.

- While doing research on which enhancement autonomous cars needed, it became clear to our team that there were many things holding autonomous driving back. We learned about why autonomous driving cars were not yet fully autonomous and what some of the barriers were to this full autonomy - one of the main ones being security [2]. We also found that there are 6 levels to full autonomy (levels 0 to 5) and that Apollo was only level 4 which signifies high-driving automation but not full autonomy [3].
- Most importantly, our team learned how important collaboration was in finding an optimized approach. We initially had a single group member creating both approaches and found that both approaches were very similar to each other and not very efficient solutions to our off-road enhancement. Then, we had another member join in on the implementation approach by brainstorming and questioning the original creator. This discussion sparked a totally new way of routing using satellite functionality, and it took collaboration and communication to get there.

Conclusion

Our proposed feature will help passengers to travel greater distances on difficult terrains beyond the urban settings already programmed. Two approaches were suggested to implement this feature, both demonstrating similarities and differences. While the second approach was better in terms of maintainability, the first approach ultimately proves to be more efficient in areas considering performance, evolvability, and testability. Finally, the enhancement feature will aid the passenger's user experience while travelling to any destination accessible by their vehicle, and no longer limited by map APIs and urban roads.

References

(1) "Barriers to Autonomous Vehicle Adoption." *AUTOCRYPT*, 31 Aug. 2021, autocrypt.io/barriers-to-autonomous-vehicle-adoption/.

(2) Greicius, Tony. "NASA's Self-Driving Perseverance Mars Rover 'Takes the Wheel'." *NASA*, NASA, 1 July 2021, www.nasa.gov/feature/jpl/nasa-s-self-driving-perseverance-mars-rover-takes-the-wheel.

(3) "The 6 Levels of Vehicle Autonomy Explained." *Synopsys Automotive*, www.synopsys.com/automotive/autonomous-driving-levels.html.