Automated Driving Using External Perception

Conceptual Design Review Report - Fall 2022

December 13, 2022

Team E

Atharv Pulapaka
Dhanesh Pamnani
Jash Shah
Ronit Hire
Shreyas Jha

Sponsor: Nissan Automotive Inc.

Liam Pederson, Najam Baig, & Viju James

Carnegie Mellon University

Contents

1 Project Description **1**

2 Use case **1**

3 System Level Requirements **3**
3.1 Mandatory requirements . . . 3
3.1.1 Functional requirements . . . 3
3.1.2 Performance requirements . . . 4
3.1.3 Non-functional requirements . . . 4
3.2 Desired requirements . . . 5
3.2.1 Desired performance requirements . . . 5
3.2.2 Desired non-functional requirements . . . 5

4 Functional Architecture **6**

5 System and Sub-system level trade studies **7**
5.1 System level trade study . . . 7
5.2 Perception Sensors . . . 8
5.3 RC cars . . . 8
5.4 Object detection Alogrithm . . . 9

6 Cyberphysical Architecture **10**

7 Subsystem Descriptions **11**
7.1 RC cars . . . 11
7.2 Track . . . 12
7.3 Infrastructure sensing . . . 13
7.4 Decision making system . . . 14
7.5 Testing and Validation . . . 14

8 Project Management **16**
8.1 Work Plan and tasks . . . 16
8.2 Schedule . . . 17
8.3 System Demonstrations (Spring and Fall) . . . 19
8.4 Team Responsibilities . . . 21
8.5 Provisional Budget . . . 21
8.6 Risk Management . . . 22

9 Appendix **24**
9.1 CyberPhysical . . . 24
9.2 Trade studies . . . 25
9.3 Schedule and milestones for Fall 23 . . . 27
9.4 Gantt Chart . . . 28
9.5 Risk Management . . . 29
9.6 RC car and track requirements . . . 32

1 Project Description

Automobile manufacturing facilities are an extremely busy place where vehicles are manufactured and assembled every other minute. The long queue of assembled vehicles at the end of the assembly line is a huge problem for every automaker. To clear this queue these assembled vehicles need to be marshaled to a shipping yard usually located a few miles away. Traditional marshaling of non-autonomous vehicles using human drivers is a repetitive, laborious, costly, and slow process.

Automated driving using external perception is a potential solution to automate this repetitive process. With OuterSense, the aim is to develop an automated, and reliable way of marshaling non-autonomous vehicles from one point to another. Following the timeline of the MRSD Project Course, OuterSense shall demonstrate a proof of concept of this solution on a scaled down model. For this project, OuterSense shall drive RC cars through external perception and decision making in a loop without any collisions with other cars on a test track. There are a few major challenges in achieving this. Since external cameras will be used to track the vehicles, there is an inherent parallax error in estimating the lateral pose of the vehicle. Furthermore, since perception, pose estimation, motion planning, and computation of motion cues are performed off-board the controlled vehicles, there is an induced latency in the driving. Consequently, if the vehicle speed is too high, it may drive off the lane and collide into other vehicles; if the pose estimation is delayed (and thus inaccurate), the car may oscillate violently within the lane (Figure 1). Thus, to achieve safe and automated driving, absolute synchronization between the various autonomous functions is essential. In this project, we shall address these challenges and develop a solution that can enable safe automated driving using external perception on a given track.

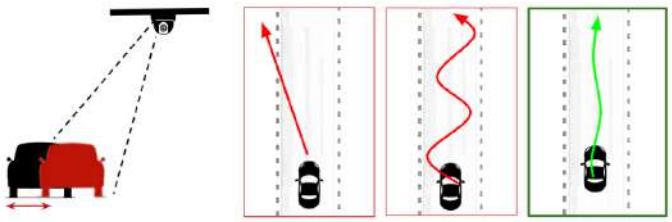


Figure 1: Challenges in automated driving using external perception

2 Use case

Nissan is a global automobile OEM with numerous vehicle manufacturing facilities in the United States. The vehicle assembly facility in Smyrna Tennessee spans 884 acres and manufactures over 820 vehicles per day. These assembled vehicles need to be marshaled to the loading station located 5 miles away from the assembly station which enables efficient distribution to the widespread dealerships across America. Human drivers are employed to drive these finished vehicles to the loading stations, and then make their way back to the assembly line to repeat the on-going process (Figure 2). The process to marshal cars from the assembly plant to the loading stations (located merely 5 miles away) using human drivers costs an automaker over 25 million dollars year on year. Additionally, shortage of labor and absence of drivers further creates a bottleneck in the supply chain.

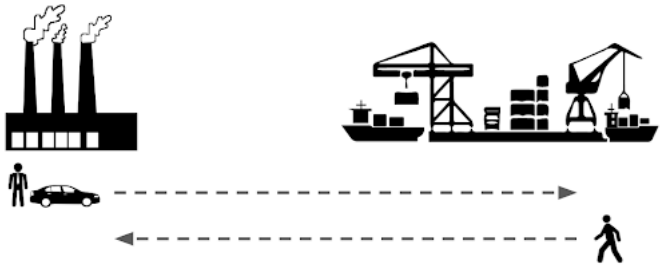


Figure 2: Marshalling using human drivers

To eliminate this recurring cost and bottleneck in the supply chain, infrastructure sensors are placed along the path (Figure 3), to enable automated and safe driving of the vehicles by ensuring that they accurately maintain their lane and avoid any obstacles on the road.

Figure 3: Infrastructure sensors setup along the road

These sentries track the cars, estimate their pose, and communicate this information to a cloud based decision making system. Further, the cars themselves communicate rudimentary odometry data like vehicle speed and steering angle to the decision making system. The decision making system contains an intelligent algorithm that uses the information from the numerous sentries, integrates it with the vehicle odometry data to localize the vehicle and compute motion cues to drive without any collisions. These motion cues are sent to the cars which enables them to traverse safely through the path. The system can drive multiple cars at once, and ensures that there is no collision between the controlled vehicles, the controlled vehicles and any rogue vehicle, or between the controlled vehicles and pedestrians by ensuring that the vehicles stay in lane boundaries. Further, as an additional level of safety an emergency stop button can be used to override all motion cues and stop all vehicles safely at any given point.

Thus, the newly assembled vehicles are marshaled to the shipping yard without any human driver via the OuterSense system (Figure 4), saving time, labor, and money for the automaker.

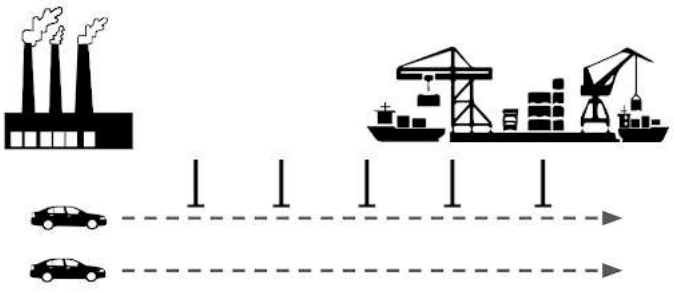


Figure 4: Automated marshalling using infrastructure based perception

3 System Level Requirements

System requirements shown below, are developed based upon a thorough understanding of the problem statement, use case, system objectives, and discussions with stakeholders.

3.1 Mandatory requirements

3.1.1 Functional requirements

To demonstrate the proof of concept on a scaled down model and to solve the problem mentioned in the aforementioned use case, Table 1 highlights the list of functional requirements of the system with their corresponding IDs (FRx).

Mandatory Functional Requirements (FR)		
SN	**Description**	**FR ID**
1	Drive vehicles using external perception	FR1
2	Avoid Collisions	FR2
3	Stop vehicles in case of emergency	FR3
4	Track vehicles	FR4
5	Localize vehicles	FR5

Table 1: Mandatory functional requirements

FR1: Drive vehicles using external perception
Using perception sensors mounted on ambient infrastructure lies at the core of OuterSense. Sensing and control of the vehicles required for automated driving are performed externally. The system shall perceive the data coming from the external infrastructure sensors, plan the vehicle path, and communicate motion cues to the vehicle.

FR2: Avoid collisions with any controlled vehicle
The system requirement for avoiding collisions with other vehicles is crucial for the system to demonstrate safe driving. The system shall monitor surroundings and constantly check for presence of other controlled vehicles. If other vehicles are present the system alters the trajectory as per the data obtained from the infrastructure sensors, and accordingly computes and sends motion cues to avoid collisions.

FR3: Stop vehicles if there is an emergency
System shall be capable of stopping all the vehicles at any time during an emergency. Any vehicle can cause safety hazards and hence, it is crucial for the system to be able to stop the vehicles when required. On giving an emergency stop command, the system shall send motion cues to bring the controlled vehicles to a controlled halt.

FR4: Track vehicles
The system shall identify and track vehicles throughout its journey. The data obtained from tracking is later used for pose estimation of the vehicles.

FR5: Localize vehicles
To ensure the vehicle is following the desired trajectory and maintaining the lane it is crucial

for the system to localize the vehicle and get its position and orientation. The system shall use tracking data coupled with its knowledge of the environment (map and camera parameters) to estimate pose of the vehicle regularly.

3.1.2 Performance requirements

Performance requirements are derived from the functional requirements and quantify the outcome of a functional requirement. Table 2 shows the performance requirement for the system with their corresponding IDs (PRx).

SN	Mandatory Performance Requirements (PR)	PR ID
1	Ensure 0 collisions amongst controlled vehicles	PR1
2	Drive controlled vehicles within 12.5% of lane boundaries	PR2
3	Stop all controlled vehicles within 2 seconds of E-stop or communication loss	PR3
4	Pilot controlled vehicles at an average speed of 12.5 cm/sec	PR4

Table 2: Mandatory performance requirements

PR1: Ensure 0 collisions amongst controlled vehicles
One of the important functionalities of the system is to demonstrate successful operation with other controlled vehicles. The system should ensure the vehicle does not collide with other vehicles on track and drive the vehicle safely throughout the journey.

Corresponding functional requirements: FR2, FR4, FR5

PR2: Drive controlled vehicles within a region of 12.5% of lane boundaries
Another important functionality of the system is to control lateral displacement of the vehicle. System should ensure the vehicle stays within 12.5% of the lane boundaries. This is an important metric for system performance as the system should be able to keep the vehicle within the defined region irrespective of collision avoidance.

Corresponding functional requirements: FR1, FR4, FR5

PR3: Stop all controlled vehicles within 2 seconds of E-stop or communication loss
When the system is encountered with an emergency, software glitch, or communication loss the controlled vehicle should come to a stop within 2 seconds of giving an emergency command or by following an initial velocity profile that slows down the vehicle and brings it to a stop.

Corresponding functional requirements: FR3

PR4: Pilot controlled vehicles at an average speed of 12.5 cm/ sec
The system should be capable of driving the vehicle at an appropriate speed. To replicate real world speed the system will drive the RC cars at a scaled down speed of 12.5 cm/sec.

Corresponding functional requirements: FR1

3.1.3 Non-functional requirements

Table 3 shows the non-functional requirements for the system with their corresponding IDs (NFRx).

SN	Mandatory Non-Functional Requirements (NFR)	NFR ID
1	Demonstrate proof of concept on a scaled down model	NFR1
2	Have minimum number of infrastructure sensor units	NFR2

Table 3: Mandatory Non-functional requirements

To demonstrate the solution the system has 2 non-functional requirements:

1. The system replicates the real world set up in a scaled down model. It is important to use an appropriate scale to accurately represent scaled down versions of vehicles, track, infrastructure sensing units to successfully demonstrate the proof of concept.

2. The system is to be designed in a way to ensure utilization of a minimum number of infrastructure sensing units and minimum overlap between these units

3.2 Desired requirements

The below sections describe the requirements that are desired by both the team and the stakeholders. Table 4 and 5 shows the desired performance requirements and desired non-functional requirements for the system with their corresponding IDs (DPRx and DNFRx) respectively.

3.2.1 Desired performance requirements

SN	Desired Performance Requirements (DPR)	DPR ID
1	System shall control 4 vehicles	DPR1
2	System shall drive vehicles at maximum 20 cm/sec speed	DFR2
3	System shall have 0 collisions with pedestrians and rogue on road	DFR3
4	System will identify selected vehicles 100% of the time	DFR4

Table 4: Desired Performance Requirements

3.2.2 Desired non-functional requirements

SN	Desired Non-functional Requirements (DNFR)	DNFR ID
1	System will accept start and end points for vehicles and control vehicles accordingly	DNFR1
2	Demonstrate proof of concept on complex tracks	DNFR2
3	System will accept user input to select vehicles and their destinations	DNFR3
4	System will give user regular updates	DNFR4

Table 5: Desired Non-Functional Requirements

4 Functional Architecture

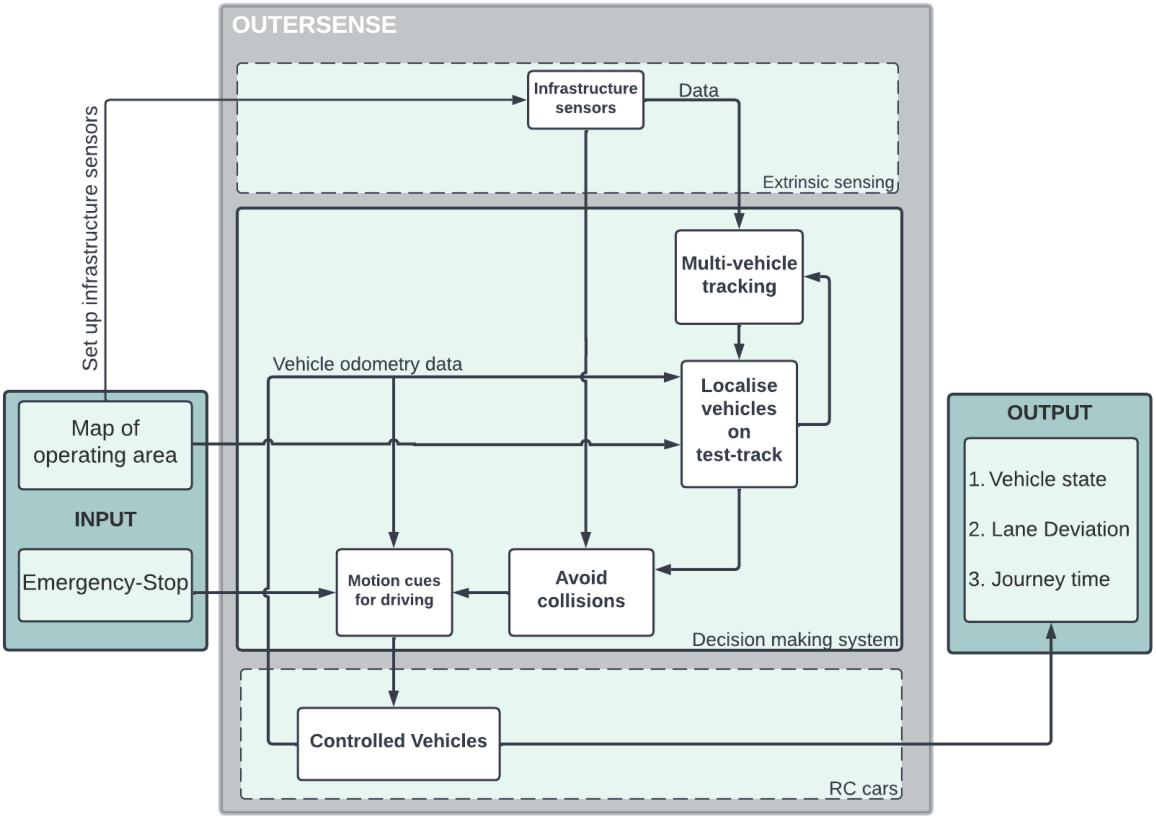


Figure 5: Functional Architecture

The functional architecture describes the high level system overview of OuterSense. It demonstrates how the system takes inputs in the form of a map and an emergency stop button to perform safe and automated driving of non-autonomous vehicles. The output of the system is defined only in terms of information; namely the vehicle state, lane deviation information and the journey time.

As a precursor, Team OuterSense takes in an operating area and decides where to position infrastructure sensors along the route such that a minimum number of sensing units are used to span the route. These units house the perception units and the required embedded hardware. The longitudinal position of these sentries are decided based on the field of view of the perception units. Any two infrastructure sensors are placed in a way such that the intersection of the fields of view of the cameras on those units is minimized.

After the sensors are placed considering the aforementioned factors, visual information is used as feedback to drive the controlled vehicles within the lane boundaries and detect any potential collisions. To ensure that the vehicles controlled by the system stay within the lane boundaries, the system tracks each vehicle, localizes them within the given map and plans their path. Additionally, the system will ensure the vehicle avoids collision with obstacles. For the scope of this project, obstacles for the system are limited to only the controlled cars. System monitors the surroundings of the controlled vehicle and looks for any obstacles around it. If there are

other vehicles around the controlled vehicle the system will plan the trajectory for the car and accordingly generate motion cues. The system alters the path or reduces the speed of the vehicle to ensure there are no collisions and communicates these updated motion cues to the vehicle by planning the trajectory.

In accordance with the trajectory, motion cues in the form of steering and velocity profile are computed for each vehicle controlled by OuterSense and sent to the respective vehicles. Using these two inputs, the vehicles utilize an on-board controller to achieve these desired trajectories. Further, the vehicles continuously send odometry data collected from the intrinsic sensors on board, to the decision making system which is used by localization of the vehicles. The system then fuses all odometry data from the vehicles and all the tracking information to estimate the pose of the vehicles on the track. Once it has localized the cars on the map, the system checks again for obstacles and repeats the above loop perpetually. Maintaining this loop with minimal lag will ensure safe and automated driving of the non-autonomous vehicles.

To warrant an additional level of safety, an emergency stop button is included in the system. This command overrides all motion cues and is communicated to the on-board controllers of the cars to bring all controlled vehicles to a controlled halt.

Thus, this functional architecture showcases how the key functional requirements come together to achieve safe automated driving using external perception. A few key trade studies are shown in the sections below following which a more detailed architecture highlighting the technologies used to achieve the aforementioned functions.

5 System and Sub-system level trade studies

5.1 System level trade study

The system level trade study is conducted on decision making compute architecture. To solve the problem a centralized, clustered or distributed architecture design can be adopted. Computing system design is thoughtfully chosen by comparing different parameters and finally considering project timeline and scope.

Centralized System Architecture For a centralized architecture, all sensor units send data to a single computing system that assesses all relevant information and controls the vehicles.

Cluster System Architecture In the clustered case, the area of interest is divided into multiple regions. For each region there exists a dedicated decision making system that uses data from the sensor units in that region and control the vehicle.

Distributed System Architecture In the distributed architecture, each sensor unit has decision making capabilities where it uses its own data and controls the vehicle in its field of view.

It is important to note that, to minimize latency, each sensor unit in central and clustered architectures has edge computing capabilities. Before sending data to the decision making system, the sensor unit will pre process data and compute certain functions like creating a bounding box on vehicles and send this data to the computing system

SN	Criteria	Weight	Centralized		Cluster		Distributed
			Edge	No-edge	Edge	No-edge	
1	Least latency	30	**3**	1	8	3	10
2	Hardware cost	10	**8**	10	3	5	1
3	Reliability	20	**1**	2	5	6	10
4	Timeline	40	**10**	10	3	3	1
Total		100	**5.9**	5.70	4.9	3.8	5.5

Table 6: System Level Trade Study

5.2 Perception Sensors

To successfully tackle the challenges described in section 1, choosing the right perception sensor is of paramount importance. Cameras offer an ambiguous but a richer view of the environment as compared to LiDARs and considering the plethora of processing algorithms available for them, camera as an extrinsic sensor was preferred over LiDAR. After the initial pruning, 4 cameras were shortlisted.

The Microsoft Kinect 2 is a 3D camera with a wide field of view and a large depth resolution, making it an ideal contender for OuterSense. It has a 30 fps frame rate and a wide range of depth detection, up to 8 meters. The Intel D435i Realsense,also, is a depth camera with an IP67 rating and a wide range of depth detection, up to 8 meters. It has a 30 fps frame rate and supports motion tracking while it can be used in a variety of weather conditions. An advantage of this sensor is that there is a plethora of online support which facilitates debugging. The Orbbec Astra performs strongly under most conditions, has depth detection of up to 8 meters and also has a differentiating factor in the power consumption which is minimal compared to the other cameras. The ZED stereo camera offers a 120 degree field of view which no other camera on this list has, which makes it an excellent competitor for the application of OuterSense. Even though each camera has features that individually suit the system well, considering a weighted average of all the parameters, the Intel D435i Realsense offers the best compromise for the project. The low frame latency of this sensor can capture small movements of the RC car on track, making downstream control decisions in the pipeline accurate. Apart from its technical capabilities, ease of use and prior team familiarity also make this sensor an apt choice. A detailed version of this trade study can be found in the appendix.

5.3 RC cars

As the aim of the project is to demonstrate the Proof of Concept on a scaled down model, it is important to understand what the term scale means. Scale in the RC vehicle world is defined as the amount by which a real sized car is scaled down in dimensions to obtain the RC vehicle. The commonly available scales are 1/24, 1/18, 1/12, 1/10. The track we build will be a replica of a real world track where the lane width is roughly 2 times the car width. The track design therefore, depends on the RC vehicle size considered for the project (Figure 6(a)). Thus, the selected RC vehicle cannot be too big as it would drastically increase the space required for the project. On the other hand, if a smaller RC car is selected the customization and mounting of components will be difficult.

Now that the scale term has been delineated the following section talks about the RC car

options considered for the trade study and the parameters these options were weighed against.

Latrax Prerunner: This car is a 1/18 scale 4 wheel drive RC vehicle with 2.4Ghz radio system equipped with 7.2v NiMH battery. It is an Ackerman steered RC vehicle with a drag link. It comes with plastic servos for steering and a plastic spur gear in the drive train. Horizon Axial: The SCX24 Deadbolt is 1/24 Scale 4 wheel drive RC car with Axial AX-4 and 2.4GHz 3-channel radio system. It comes equipped with a 350mAh 7.4V LiPo battery. Traxxas XL-5: The Traxxas XL - 5 is 1/10 scale RC vehicle with 2.4 GHz radio system and 3000mAh, 8.4V, 7-cell NiMH battery. It comes with a metal servo and adjustable suspensions Roboworks: This 1/10 scale RC vehicle comes with ready to plug slots for ROS controller, LiDAR and Camera. It is equipped with onboard Ubuntu, ROS1 and STM32 drivers. It is the most holistic research RC vehicle platform that is currently available for use. RC vehicle available in inventory : This is a 1/12 scale RC car available in MRSD inventory. It has various modifications done and some sensors that have been retrofitted. The RC cars available however are not exact replicas of each other. Based on the trade study parameters shown in the appendix the latrax prerunner was the optimal middle ground.

5.4 Object detection Alogrithm

The efficacy of tracking and localization of RC cars is directly affected by the upstream object detection algorithm. To accurately and reliably track the vehicles, a robust and fast algorithm is required. The cars present on the track don't look like standard road vehicles and are equipped with colored markers on the top to enable detections. Thus the job of the detector is to find these colored regions in the image space. Also, the cars will be the only foreground object in the image and everything else can be considered as background. This reduces the traditional object detection problem to performing image segmentation and constructing bounding boxes around the foreground objects.

A literature review on the state of the art segmentation algorithms and a review of some of the classical methods led to the consideration of the following algorithms:

Mask RCNN: In this architecture, objects are classified and localized using a bounding box and semantic segmentation that classifies each pixel into a set of categories. Every region of interest gets a segmentation mask. A class label and a bounding box are produced as the final output.

U-Net: U-Net is a convolutional neural network originally developed for segmenting biomedical images. When visualized its architecture looks like the letter U and hence the name U-Net. Its architecture is made up of two parts, the left part – the contracting path and the right part – the expansive path. The purpose of the contracting path is to capture context while the role of the expansive path is to aid in precise localization.

Threshold Segmentation: A non-learning based approach where the only object color is used to cluster the foreground objects together. Although the algorithm is computationally cheap, the quality of detections highly depend on the illumination conditions.

Taking into account the simplicity of the track and the fact the detections will be performed in a controlled environment, a CPU based Threshold Segmentation algorithm would suffice. In case a high failure rate is encountered, the fallback is to rely on U-Net for bounding boxes around the car. A detailed version of this trade study can be found in the appendix

6 Cyberphysical Architecture

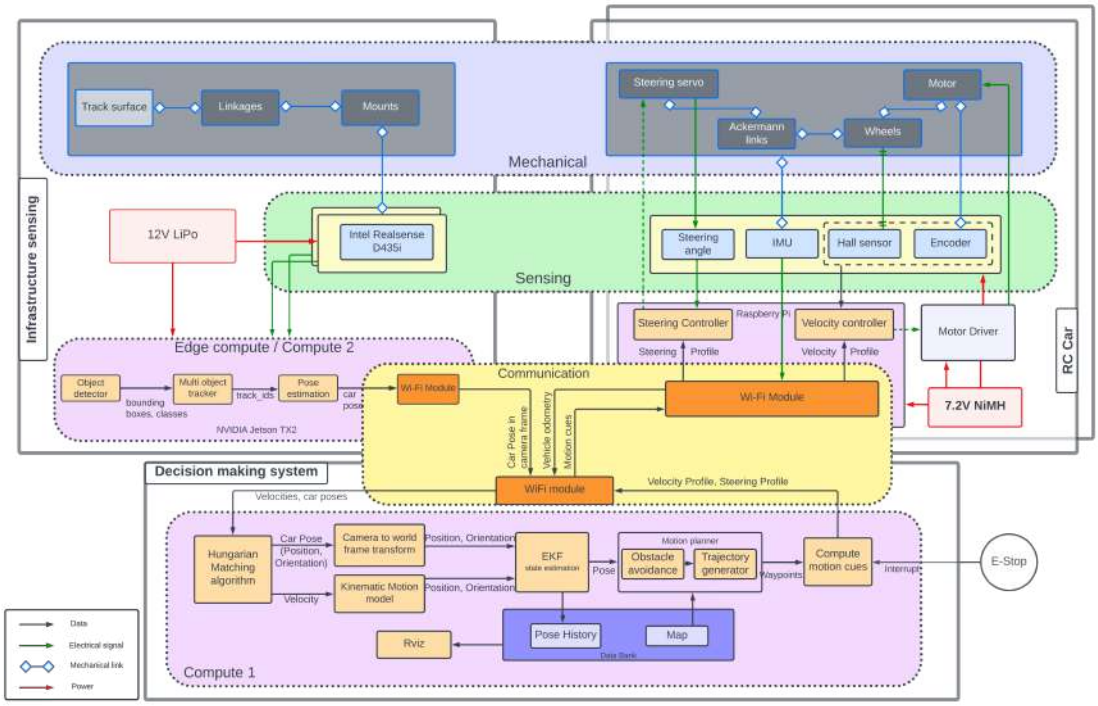


Figure 6: CyberPhysical Architecture

The cyberphysical architecture for Outersense is shown in Figure 6. Mechanical hardware, sensing, compute and communication are the core building blocks and are distributed across the 3 substems - Infrastructure sensing units, RC cars and the Decision making system. The cyber physical architecture draws strong parallels with the functional architecture shown in Figure 6.

Mechanical: The mechanical block encompasses the track surface, ground linkages and sensor mounts on the infrastructure as well as the chassis of the RC car which houses the steering links, wheel assembly, and the vehicle drivetrain.

Sensing: Introspective sensors on the RC car and extrospective sensors on the infrastructure combine to form the sensing ecosystem. RGB-D cameras (Intel Realsense D435i) mounted at regular intervals on the infrastructure as per the track layout capture the pose of the vehicle and obstacles. The introspective sensors on the chassis of the RC car (Hall sensor and encoders) capture vehicle odometry data for low level feedback. In addition, an IMU mounted on the vehicle chassis is used as a source of acceleration data.

Compute: A central decision making system supplemented by edge computing nodes on the infrastructure fulfill the computational needs of the system. Raw data captured by the infrastructure sensors goes through a perception block consisting of object detection, multi-object tracking, and pose estimation algorithms to determine an initial estimate of the pose of the car. The central decision system integrates this pose data with dead reckoning from odometry data to generate accurate heading of the vehicle and localizes the vehicle on the track. Pose information

along with the track layout (map) is used to determine the deviation of the vehicle from the lane. The waypoints are fed into a motion cue generator which is a high level planner that generates velocity and steering profiles to correct for lane errors. At each time step, the newly estimated poses are stored in a data bank for visualization, metric generation and debugging purposes.

Communication: The communication block consists of wifi modules from each of the three constituent blocks of the architecture. The wifi modules on the infrastructure units transmit estimated vehicle position and orientation whereas the controller on the RC cars transmits odometry data (steering angle and vehicle velocity) over the local network. The receiver on the decision making system collates this data and transmits motion cues corresponding to each vehicle. The vehicles receive these motion cues and feed them into the low level steering and velocity controllers. All communication is performed via ROS2 nodes pertaining to the DDS server for OuterSense.

Control: Two levels of control systems are required. First, there is a high level control architecture implemented through the elements of the central decision making system that computes the motion queues for each vehicle in order to stay within the lane boundaries and avoid collisions. Second, there is a lower level PD control deployed on each RC Car cascaded with the motor controller to execute the desired velocity and steering profile communicated to each car from the decision making system.

7 Subsystem Descriptions

7.1 RC cars

To demonstrate PoC of the system on a scaled model, RC cars are aptly suited as they mimic real world vehicles to a great extent and satisfy most of the mechanical constraints. Trade study performed above highlights Latrax Prerunner as the best choice for the basis of this subsystem. The Latrax Prerunner is a 1/18th scale model and comes equipped with 4 wheel drive, Ackerman steering and macpherson strut suspension. It also houses a 12 V brushed DC motor and a radio receiver to receive input commands.

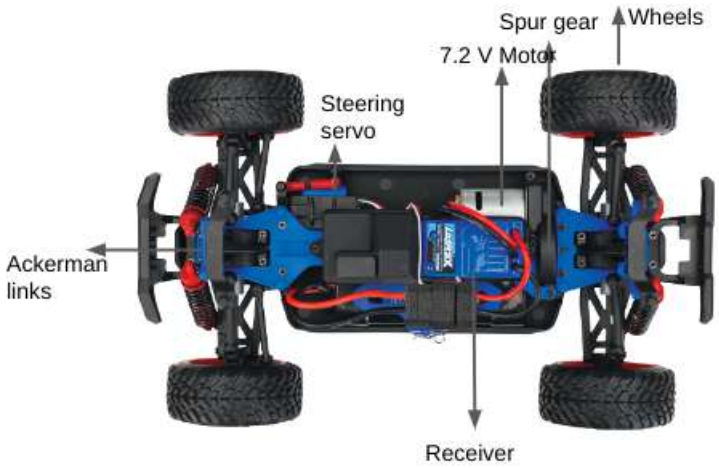


Figure 7: RC car

The following hardware customizations are done to ensure robustness and controllability:

1. Metal geared servo: Off the shelf, the RC car comes with a plastic gearbox servo as an actuator for the steering mechanism. While reliable and accurate in the short run, to ensure reliability in the long run it is replaced with an identical metal geared servo.

2. Metal spur gear: To determine vehicle velocity, the existing spur gear driven by the motor is replaced with a metal spur gear and coupled with a Hall sensor to extract motor RPM and estimate vehicle velocity using the final drive ratio.

3. Rotary encoder: For robust and precise velocity control, a secondary sensor in the form of a rotary encoder is installed on the pinion of the driving motor to get the motor RPM

4. Onboard electronics: In addition to the Hall sensor on the spur gear and a rotary encoder on the pinion, an IMU will be integrated with the vehicle chassis to estimate vehicle acceleration. The standard radio receiver is replaced with a Raspberry Pi to implement low level control and acts as a communication interface between the car and the decision making system.

Hall sensor, IMU and Encoders complete the introspective sensing suite of the car and with these in place, feedback control loops for steering angle and velocity are implemented. The steering controller uses the potentiometer reading from the servo as feedback for PD control of the steering angle and the velocity controller uses both the hall sensor as well as the rotary encoder for PD control of vehicle speed. For each control loop, the steering and velocity profiles from the central decision making system are treated as the reference signals.

7.2 Track

The test track is a purely mechanical subsystem with dimensions, layout and surface as its defining features.

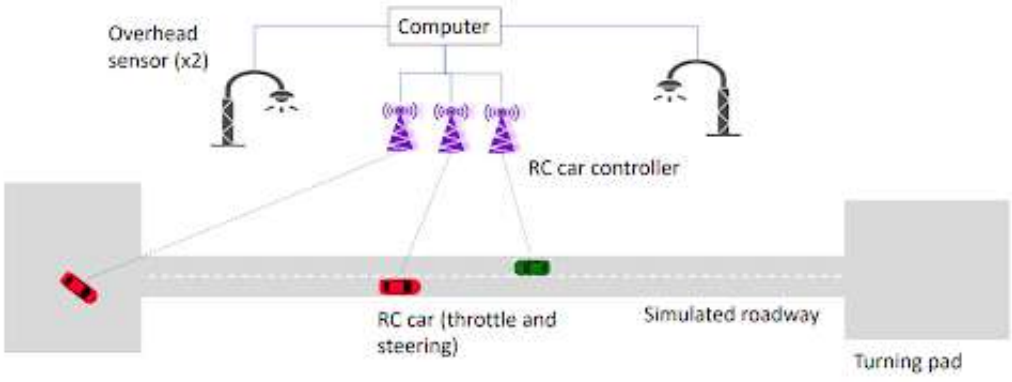


Figure 8: Track layout

Track dimensions: Track dimensions are a function of RC car size and turning radius. Lane width, and in turn track width, is dictated by the constraint of maintaining lane to vehicle width ratio of approx 1.8-2 as found in the real world. Additionally, the performance requirement on

speed [PR3] influences the track length while the turning radius of the car affects the overall compactness of the track.

Track layout: Layout for the first iteration of the track is shown in the figure above. The layout is simple enough to require minimal manufacturing effort and complex enough to test all system nuances. The layout and the performance requirement [PR2] on lane keeping together govern the drivable area for the RC cars.

Track surface: The track surface is a sufficiently rough and leveled interface that prevents undesirable tire slip and also allows for repeatable testing by reducing unevenness of the ground underneath.

The track is designed to be modular and in addition to serving as a level ground for the RC cars, it also provides mounting points for infrastructure sensors. Two overhead cameras (part of infrastructure sensing) are mounted on the track surface using ground linkages and their placement is such that the union of the two camera FOV's capture the entire track.

7.3 Infrastructure sensing

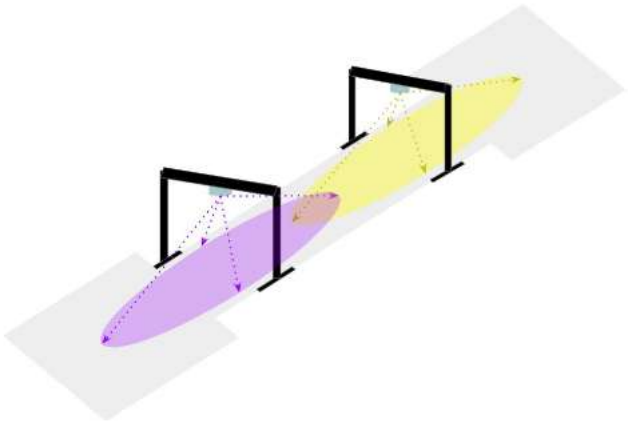


Figure 9: Overhead sensors

The infrastructure sensing subsystem comprises all the Intel Realsense D435i cameras mounted overhead on the track and the associated NVIDIA Jetson TX2 as an edge compute device. The hardware in the form of sensor mounts is designed to be easily reconfigurable to allow for quick recalibrations keeping the modularity of the track in mind.

Each sensing unit is responsible for locating the vehicle in its FOV and communicating its estimated pose to the central decision system. This is achieved with as follows:

The baseline object detection algorithm takes in the denoised RGB image as input from the camera and applies threshold segmentation to detect the presence of a RC car. The car is assumed to be the only foreground object in the image and a rectangular bounding box is drawn around it. Alternative deep learning approaches to object detection can be applied in case this simplistic method fails.

A multi-object tracker is initialized from the output of the object detector and candidate tracks are created for each detection. If an object is recurring in consecutive frames, it becomes a tracked object and a tracking is associated with it. Otherwise the candidate track is dropped.

The pose estimation algorithm processes the object bounding box and based on planar world assumption and using camera calibration parameters, estimates the orientation and position of the car. This estimate along with the associated tracking id is transmitted by the communication module to the decision making system. A fallback is to use a side-mounted Velodyne 16 LiDAR instead of the Intel Realsense camera if the pose estimates from the cameras are not accurate or if the parallax errors are too high.

All baseline algorithms on the edge system are classical non-learning based to minimize computational latency. However, having Jetson TX2 with GPU capabilities allows for deployment of learning based algorithms and achieving similar runtimes.

7.4 Decision making system

The decision making subsystem consists of algorithms for data association, state estimation, localization, planning and motion cue generation. The incoming stream of data from infrastructure sensing and RC vehicles passes through a series of processing modules to output a control strategy in the form of steering and velocity signals.

First off, the incoming data passes through a Hungarian Matching algorithm to determine the correct correspondence between the infrastructure sensor data, vehicle and the vehicle odometry data. This results in accurate mapping of multiple sensor readings for each vehicle.

The velocity from the matched odometry data is augmented with orientation information and passed into the motion model which performs dead reckoning for state estimation. This estimate is updated with environment measurements using EKF on the sensor model which converts camera pose to world pose.

This pose estimate along with the track map is fed to a motion planner which determines the deviation of the car from the lane centerline and comes up with a sequence of waypoints to correct for lane errors. To meet the desired performance requirement PRx.x, the planner will eventually also take into account the poses of the identified rogue vehicles and modify waypoints to avoid potential collisions.

Motion cue generator uses these waypoints and produces steering and drive commands to follow this trajectory. The drive commands are artificially constrained so as to maintain safe distance between all controlled vehicles. Additionally, the generated steering and velocity profiles are appended with a “halt profile” to ensure safety of controlled vehicles in case of communication loss. The algorithm also handles interrupt from the E-stop button and overrides any generated velocity profile with a halt profile and fast-forwards to transmission.

7.5 Testing and Validation

This subsystem defines a list of activities and documents to iterate over the design and implementation decisions taken in other sub-systems.

Unit and subsystem testing: Post design finalization and initial assembly of RC cars a component level testing will be done. Basic tests for the controller and servos interface will be done to validate steering and speed commands. Sensor and controller connections will be made and sensor data will be validated. Post these activities, controllers, motors and sensors will be mounted on the RC car. A low level controller will be programmed and tested to ensure the RC vehicle follows speed and steering commands accurately. A communication module will be set up and tested by sending commands and receiving sensor data.

Once perception sensor design is finalized, the sensor will be tested to analyze the data. Controllers will be programmed to pre process the sensor data and compute functions to form bounding boxes on vehicles and estimate the pose based on perception sensor data. This data will be sent over the communication module.

Unit level testing for each component of the decision making system will be done. The component level testing will happen on:

1. Data association and communication
2. State estimation
3. Localization
4. Planning
5. Motion cue generation

On successful completion of every function in the decision making system it will be integrated with other subsystems.

Subsystem Integration and testing: The integrated testing of perception sensor and RC vehicles will be done after subsystem testing is completed. The perception subsystem will sense vehicles,compute functions and send data to decision making system.The data sent over the communication module will be validated by verifying the data packet size and accuracy of vehicle estimate. The decision making system will receive data from the RC vehicle and perception unit and perform further computation. Small test cases will be performed on the vehicle to validate the decision making system's capabilities.

On Track Testing: On successful system integration and testing, the following tests will be conducted to validate system functionalities:

1. One vehicle will be tested on track and sensing will be done by only one sensor unit. The vehicle will be controlled to drive in a straight line.

2. One vehicle will be tested on track and sensing will be done by two sensor units. The vehicle will be controlled to drive in a straight line.

3. One vehicle will be tested on track and sensing will be done by two sensor units. The vehicle will be controlled to drive in a full loop. In the full loop test the RC vehicle will complete the entire track (Track shown in Figure 9) taking complete turns and running in a loop.

4. 2 vehicles will be tested on track and sensing will be done by only one sensor unit. The vehicles will be controlled to drive in a straight line

5. 2 vehicles will be tested on track and sensing will be done by two sensor units. The vehicles will be controlled to drive in a straight line

6. 2 vehicles will be tested on track and sensing will be done by two sensor units. The vehicles will be controlled to drive in a full loop.

8 Project Management

8.1 Work Plan and tasks

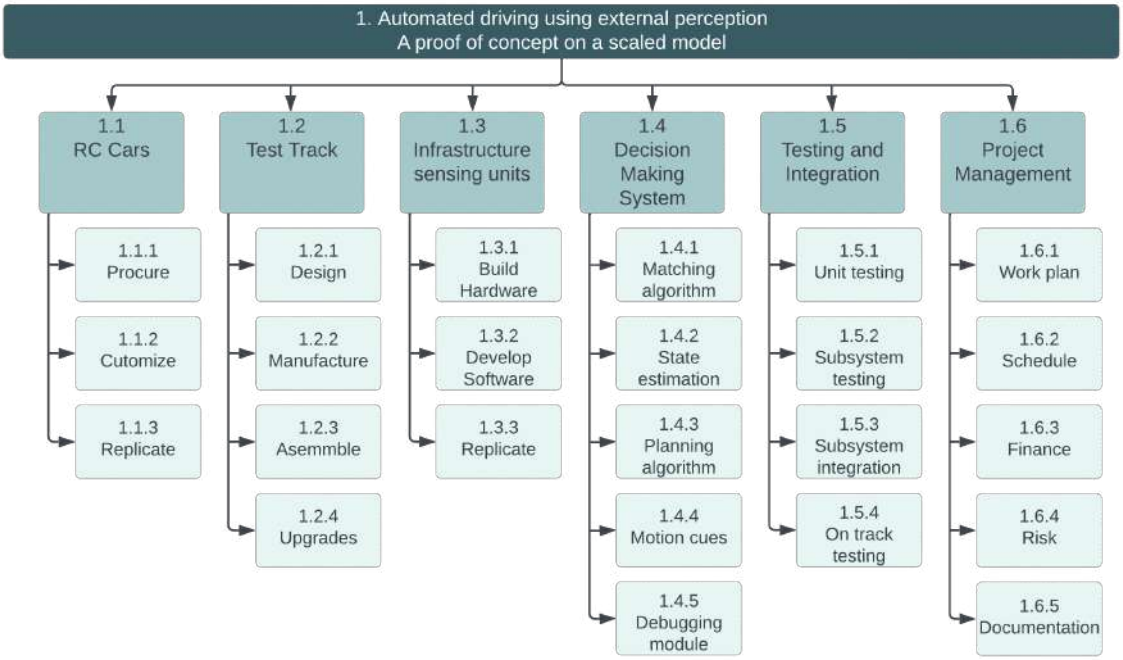


Figure 10: Workbreakdown structure

At the highest level of the work breakdown structure(WBS) lies the primary objective: demonstrate the proof of concept of automated driving using external perception on a scaled down model. A fusion of the product and process oriented WBS will be followed containing four products and two processes. The 4 major elements to meet the primary objective will be treated as products i.e. the RC cars, test track, infrastructure sensing units, and, the decision making system. Whilst developing these four subsystems the continuous process of test-integration and management shall be executed. This is aimed to be in sync with the V-model of systems engineering, which given a clear set of requirements from the stakeholder allows the team to detail, integrate and validate at every step whilst driving the project to completion within a stipulated time frame. A more detailed work breakdown structure is available in Appendix.

For the RC Cars to be developed as per the derived requirements (refer Appendix), the team shall focus on procurement as prescribed by Risk Mitigation plan. Post procurement, mechanical and embedded hardware upgrades need to be incorporated and tested. On achieving a robust mechatronic system, on board car software containing the low level control systems and communication modules shall be developed and deployed. Finally, on successfully passing defined subsystem tests, the design and build procedure are freezed and documented. This same process (procurement, customization, and a subsystem test) will be applied to 3 more RC cars required for the core demonstrations, with a spare to factor in associated risks. The test track is a critical product required to demonstrate the proof of concept on a scaled model. The derived requirements for the track are defined in Appendix. The dimensions of the track is dependent on the scale of the RC Car. The track design involving CAD modeling and rapid prototyping can only be completed once the team finally procures the RC Car. On successful unit tests for the

track design, the manufacturing phase shall begin. In order to utilize the CMU machine shop, team members need to successfully complete the machine shop course. Post machine shop authorization and sourcing of the raw materials, the fabrication process begins. Simultaneously, the mechanical team shall design the mechanical components for the infrastructure sensing units ensuring their coupling with the track and capabilities to house Intel RealSense D435i and/or VLP-16, and the other electronic components required at the infrastructure sensors. At the same time, the embedded and electronics team sets up the electronic components and brings initial structure to the edge software as described by the middleware and software architecture team. Post successful unit tests of the mechatronic systems on the infrastructure sensors, the design and build procedure are freezed and documented. This will be used to build 3-4 more infrastructure sensing units as the project progresses. As a note, the replication of RC Cars and Infrastructure Sensors into more identical copies (as shown in the cyberphysical architecture) are required only prior to the on-track testing phase and are scheduled accordingly. All the replicated RC cars and sensor unit will undergo the same unit level and subsystem level testing.

By Week 10 of 2023, the hardware elements of the subsystems are scheduled to be developed and validated, and the teams shall now be primarily focused on developing the perception, localization and controls functionalities. The middleware and software architecture team shall be working on the visualization and debugging block. None of the software modules can be deemed to be successfully completed unless the prior module is deemed complete and this is captured below in the schedule for Spring ‘23.

On successful completion of these work packages and the associated unit, subsystem and integration tests, the team shall replicate the RC Cars and infrastructure sensors as required to move into the sequence of on track tests. The management team shall follow the agile model and continuously strive to define and track lower level work products to track the schedule. It will also manage the project budget and hold periodic meetings dedicated towards risk mitigation. Finally, the management team is also responsible for documentation as the team progresses into developing and demonstrating OuterSense.

8.2 Schedule

Internal and external milestones for Spring 2023 are captured in Table 7, and the corresponding schedule is represented using a Gantt chart in Figure 11.This is derived from the work plan defined in Section 8.1 and associated dependencies of the tasks considering team member responsibilities in various phases of the project.

S.N.	Milestone type	Date	Title	Description
1	Internal	28 Feb, ’23	RC Car customization ends	Successful subsystem test for RC Car.
2	Internal	11 Mar, ’23	Hardware for test track ready for integration	Test track manufactured and hardware for one infrastructure sensing unit built
3	External	13 Mar, ’23	Preliminary Design Review	Compliance and quality as per requirements of PDR; to showcase preliminary system design.
4	Internal	1 Apr, ’23	Critical software components developed	Successful unit tests for each critical functional software module.
5	External	19 Apr, ’23	Spring Validation Demo	Elucidated in SVD demo section
6	External	26 Apr, ’23	Spring Validation Demo - Encore	One vehicle autonomously driven around the full test track spanned by two sensor units.

Table 7: Spring 2023 Milestones

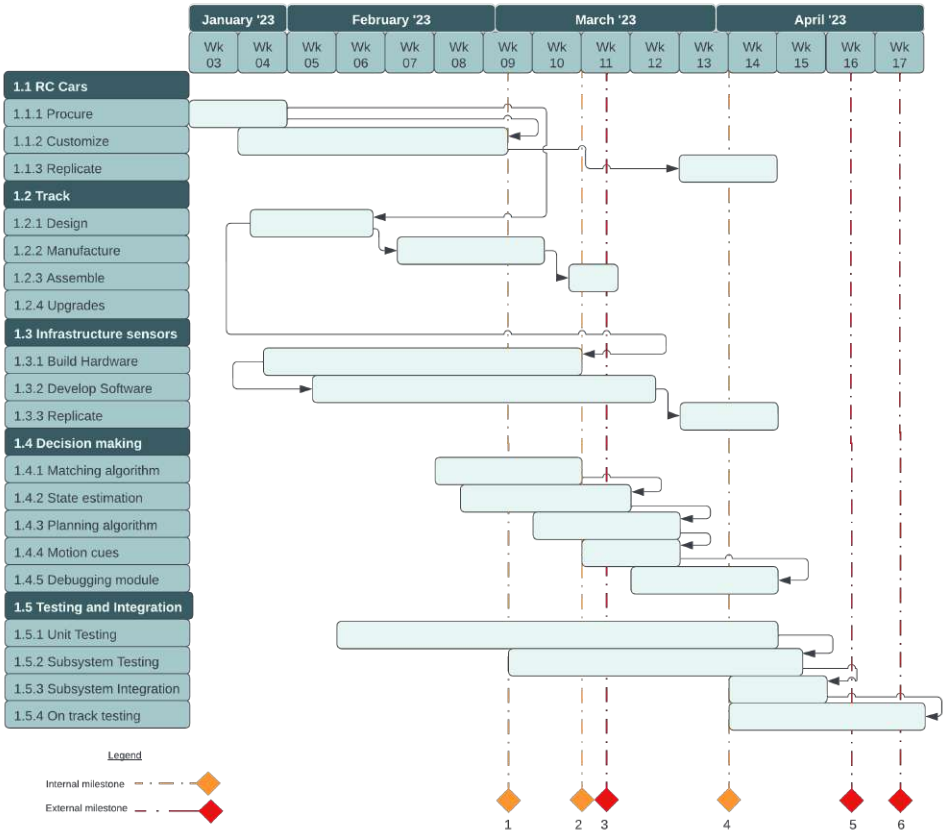


Figure 11: Gantt Chart

The schedule and milestones planned for Fall '23 are available in Appendix.

8.3 System Demonstrations (Spring and Fall)

Our project aims to demonstrate a proof of concept of automated driving using external perception on a scaled down model. On this front, we will demonstrate 2 RC cars going in a loop around a test track that has 2 external infrastructure units that house the camera. These infrastructure units will be adjustable in height and position to meet the non functional requirement of having minimum overlap between sensing units and the track will be dumbbell shaped as shown in the subsystem description. The team will have two main validation demonstrations over the coming year. The requirements mentioned in Tables 1 and Tables 2 are met slowly over the two semesters. These will be accomplished in various increasing levels of complexity, starting with driving one car in a straight line using one infrastructure unit and ending with the fully integrated automated driving system using external perception. Dates and requirements met align with the project scheduled above. All tests will have written test plans and SOPs, and will be rigorously documented with photo, video, and ground truth measurements.

At the end of the Spring 2023 semester, our Spring Validation Demo (SVD) will feature the demonstration of key subsystems with progress towards select functional requirements. The Fall Validation Demo (FVD) at the end of the Fall 2023 semester will showcase all high level system functionality, addressing all mandatory functional, performance and non-functional requirements. A more detailed breakdown is provided below.

Spring Validation Demo

Demo conditions

- **Location:** Newell-Simon Hall B level using the Modular Test Track
- **Equipments:** Disassembled Test track, 2 x Infrastructure unit, RC car, adjustment blocks, Tool box, external computer, Monitor showing outputs

Objective of Demonstration with corresponding requirements

- SVD.1: Detect, Track, and estimate pose of 1 RC car
- SVD.2: Compute and Communicate Motion cues to 1 car
- SVD.3: Low level control on RC car to meet motion cues
- SVD.4: Drive one RC car in a straight line (PR2, PR4)
- SVD.5: Maintain lane and stay within 12.5% of lane boundaries (PR2)
- SVD.6: Drive one RC car at average speed of 12.5cm/second (PR4)

Procedure

1. Assemble the test track according to the numbers lazer cut on each modular unit following the steps in the Standard Operating Procedure Document (SOP).
2. Use a Level gauge indicator to check for flatness and leveling shims to make necessary adjustments.
3. Assemble the Infrastructure sensing units on the track at the appropriate position as per the SOP.
4. Adjust the height and measure at 3 points to meet the required height as in SOP
5. Check power and readiness of the RC car by running it remotely
6. Power the infrastructure units
7. Calibrate cameras after track assembly as per the SOP
8. Set up the central Decision Making System
9. Run system checks for communication, perception, and low level control as described in SOP
10. If no errors start with the demonstration, follow debugging procedure in SOP if errors
11. Place the RC cars at the start line
12. Reset the clock to measure average speed of the vehicle in straight line
13. Repeat steps 11 and 12
14. Document all errors built up during the test and all other data obtained

Success Criteria

- One RC car is driven in a straight line within 12.5 % of lane boundaries at an average speed of 12.5 cm/second

For the FVD the team will demonstrate our systems ability to meet all our mandatory performance and non-functional requirements. In doing so, we will highlight the progress we have made during the fall semester, and in particular, detail our progress with integrating all our subsystems.

Fall Validation Demo

Demo conditions

- Location: Newell-Simon Hall B level using the Modular Test Track
- Equipments: Disassembled Test track, 2 Infrastructure unit, 2 RC cars, adjustment blocks, Tool box, external computer, Monitor showing outputs, Spare parts

Objectives of demonstration with corresponding requirements

- FVD.1: Detect, track, and estimate pose of 2 RC cars
- FVD.2: Match data, compute, and communicate motion cues to respective cars
- FVD.3: Drive 2 RC cars continuously in a loop at average speed of 12.5 cm/s PR4

- FVD.4: Maintain lane and stay within 12.5% of lane boundaries PR2
- FVD.5: Avoid collisions with other controlled vehicles PR1
- FVD.6: E-stop that can stop the vehicles within 2 seconds PR3

Procedure

The procedure that needs to be followed will be similar to the procedure that would be followed for the SVD. The only change is listed below.

- Place the RC cars in opposite ends of the loop and repeat the procedure

Success Criteria

- Two RC cars are driven continuously around a loop on a scaled down test track demonstrating proof of concept of automated driving using external perception.
- The cars stay within 12.5% of lane boundaries.
- The cars travel at an average speed of 12.5 cm/second.
- The cars have 0 collisions with other controlled cars throughout the duration of the demonstration

8.4 Team Responsibilities

Primary and secondary owners have been assigned for each major subsystem. This ensures that each domain expertise required for the project has an additional owner who can take charge in case the primary owner faces any issue that may temporarily inhibit them from working on the project. This is an important risk management strategy and the responsibilities have been tabulated in table 8.

Domain	Primary Owner	Secondary Owner
Mechanical	Dhanesh Pamnani	Ronit Hire
Embedded and Electronics	Atharv Pulapaka	Shreyas Jha
Middleware and software architecture	Jash Shah	Shreyas Jha
Localization	Shreyas Jha	Jash Shah
Perception	Ronit Hire	Dhanesh Pamnani
Planning and Controls	Atharv Pulapaka	Jash Shah
Management	Dhanesh Pamnani	Ronit Hire

Table 8: Team Responsibilities

8.5 Provisional Budget

The budget presented in the table below is not a fully comprehensive parts list and would develop as the project progresses. The goal of the table is to allocate a budget to different components within subsystems. The Cost to Team considers availability of parts in inventory.

S.N.	Part Name	Sub.	Qty.	Inv.	UC	TC	CTT
1	Raspberry Pi	RC cars	1	Yes	150	150	0
2	Teensy 3.6	RC cars	4	Yes	30	120	30
3	RC cars	RC cars	4	No	450	1800	1800
4	NVIDIA Jetson TX2	Infra	4	Yes	304	1216	0
5	Aluminium Sica rods	Infra	10m	No	500	500	500
6	Wood	Track	-	No	1000	1000	1000
7	3d print material	Track	-	Yes	200	200	0
8	Incremental Quadrature hollow shaft encoder	RC cars	4	No	25	100	100
9	Hall Sensor	RC cars	4	No	15	60	60
10	Adafruit 9-DOF IMU	RC cars	4	Yes	35	140	0
11	PCB	RC cars	4	No	40	120	120
12	Ethernet switch	Infra	4	No	17	68	68
13	LED Studio Lights	Track	2	No	75	150	150
14	LAN Cable	Infra	75ft	No	14	14	14
15	Wifi Transceiver	Infra	4	No	50	200	200
16	Leveling Shims	Track	1	No	50	50	50
17	E- Stop button	Misc	1	No	50	50	50
18	Fasteners	Misc	-	Yes	200	200	0
19	Battery-12V Lipo	Infra	4	Yes	25	100	0
20	Monitor,keyboard, mouse	RC cars	1	No	275	275	275
21	VLP-16	Infra	2	Yes	8000	16000	0
22	External SD card	Infra	1	No	170	170	170
23	Intel RealSense D435i	Infra	4	Yes	399	1596	0
24	ROS online course	Misc	5	No	60	300	300
	Total					**24,579**	**4,887**

Table 9: Rough order of Magnitude Budget

8.6 Risk Management

Risk Management is essential to identify and prepare for uncertainties that might hinder the-completion of the project within the set schedule, budget or scope. It is important to continually assess possible risks and determine the likelihood and consequences of them occurring. For our project, we have identified 5 major risks and outlined their mitigation strategies in Table below. These risks and their risk reduction strategies are further detailed in the Appendix.

Risk	Title	Risk Type	Mitigation	Likelihood	Consequence	Severity
1	RC Vehicle getting damaged whilst testing	Technical, Schedule	-Build cushioned fence - purchase spare parts	4	5	High
2	Unforeseen issues in building track	Technical, Schedule	-Prioritize track building -simplify design -train team members	4	5	High
3	Unforeseen issues in building RC vehicle	Technical, Schedule, Cost	-Purchase and not build - identify vendors to outsource	5	5	High
4	Latency or communication issues	Technical, Schedule	-Send sequence of motion inputs	5	5	High
5	Open source libraries incompatible	Technical	-Use libraries with minimum dependencies -Adopt libraries with maximum community support	3	4	Medium

Figure 12: Risk Management Table

References

1. https://www.intelrealsense.com/wp-content/uploads/2022/11/Intel-RealSense-D400-Series-Datasheet-November-2022.pdf

2. https://latrax.com/products/prerunner

3. K. He, G. Gkioxari, P. Dollár and R. Girshick, ”Mask R-CNN,” 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

4. https://www.ri.cmu.edu/app/uploads/2021/05/Traffic4D/Longitudinal/iv2021.pdf

5. https://www.researchgate.net/publication/324387691/YOLOv3/An/Incremental/Improvement

9 Appendix

9.1 CyberPhysical

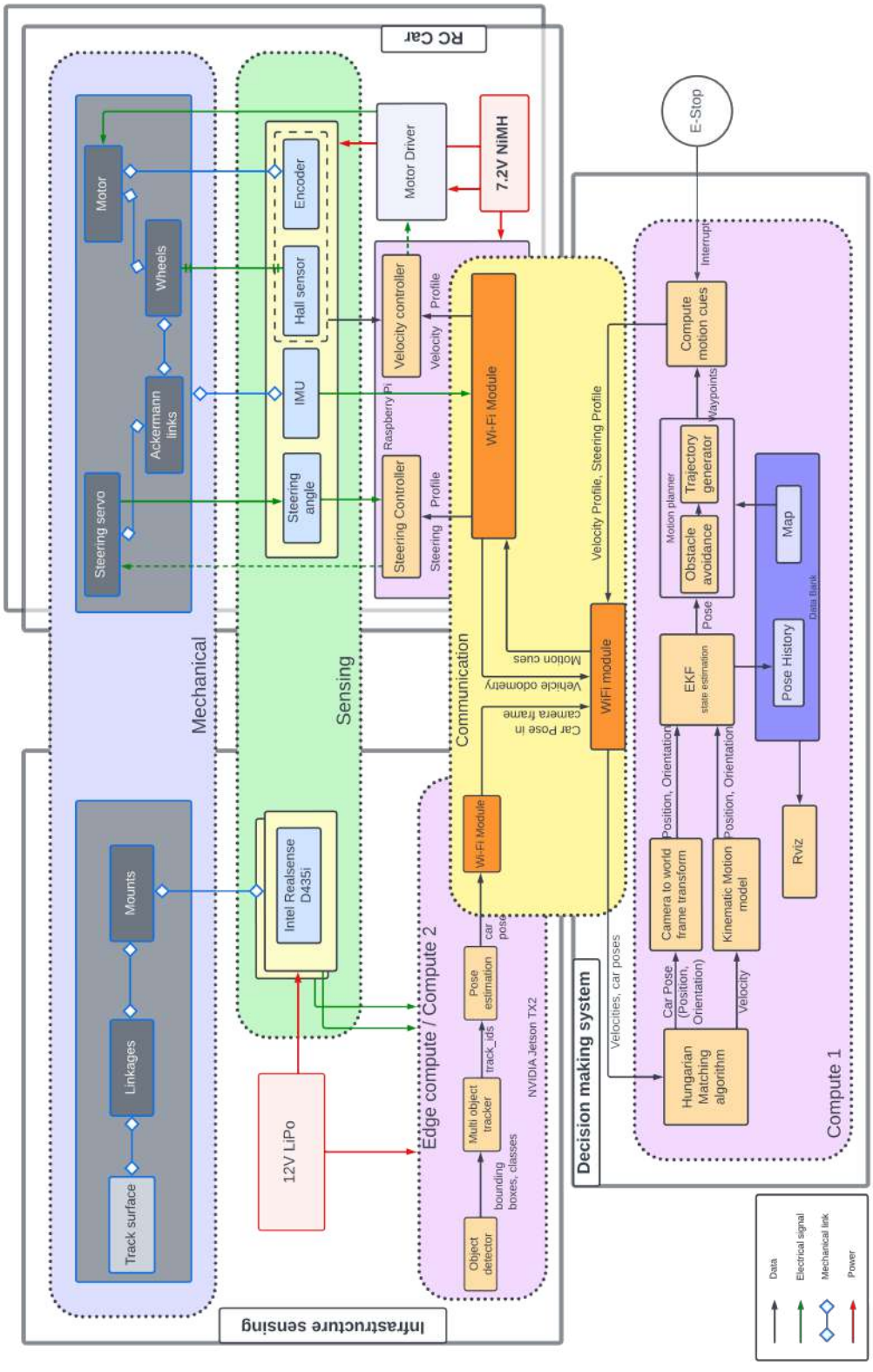


Figure 13: CyberPhysical Architecture

9.2 Trade studies

S.N.	Criteria	Weight	Centralised		Cluster		Distributed
			Edge Computing	No Edge Computing	Edge Computing	No Edge Computing	
1	Least Latency	30	3	1	8	3	10
2	Hardware Cost	10	8	10	3	5	1
3	Reliability	20	1	2	5	6	10
4	Timeline	40	10	10	3	3	1
Total		100	**5.90**	**5.70**	**4.90**	**3.80**	**5.50**

Table 6: Computing architecture trade study

Figure 14: System level Trade Study

S.N.	Criteria	Weight	Microsoft Kinect II	Intel D435i realsense	Orbbec Astra S	ZED Stereocam
1	FOV	30	6	**8**	8	10
2	Frame Rate	20	6	**6**	6	6
3	Frame Latency	10	4	**8**	8	6
4	Resolution	10	4	**6**	6	4
5	SDK, documentation, support	30	8	**10**	4	6
Total		100	6.4	**8**	6.2	7

Table 7: Trade study for perception sensors

Figure 15: Perception Sensor Trade Study

S.N.	Criteria	Weight	Latrax Prerunner	Horizon Axial	Traxxas XL-5	Roboworks ACS	Inventory
1	Scale	15	**8**	10	1	1	3
2	Steering	15	**10**	10	10	10	10
3	Replicability	12	**10**	10	10	10	3
4	Inbuilt-sensors	12	**1**	1	1	8	6
5	Extensibility	12	**8**	1	8	6	8
6	Cost	9	**8**	8	5	1	10
7	Workability	9	**5**	1	8	10	6
8	Modularity	9	**6**	1	8	10	8
9	Spare Parts	7	**10**	10	10	3	1
Total		100	**7.36**	6	6.48	6.66	6.28

Table 8: RC car Trade study

Figure 16: RC car Trade Study

S.N.	Criteria	Weight	Threshold Segmentation	Mask RCNN	UNet
1	FPS	40	**10**	7	7
2	Computational Requirement	25	**10**	2	5
3	Illumination Invariance	25	**1**	8	8
4	Ease of implementation	10	**10**	7	3
Total		100	**7.75**	6	6.35

Figure 17: Object Detection Trade Study

9.3 Schedule and milestones for Fall 23

On moving into Fall '23 and post successfully achieving the scheduled milestones for Spring '23, the team shall begin on track testing with multiple RC cars being controlled simultaneously through external perception. This requires building upon the planning block to first accommodate and for multiple controlled vehicles and then also be robust to obstacles in the environment. The team shall also work upon improving the perception stack deployed on the edge for higher localization accuracy and lower latency through faster performance. In mid-October, the team shall work on building a more complex track and making the required upgrades. Finally, in November, the team shall be solely focused on building robustness in the system through rigorous integration and on-track testing.

S.N.	Milestone type	Date	Title	Description
7.	Internal	23 Sep, '23	Autonomous control of multiple RC cars	Two vehicles autonomously driven around the full test track spanned by two sensor units.
8.	External	17 Oct, '23	System Development Report	Compliance and quality as per requirements of the system development report
9.	Internal	27 Oct, '23	Extension to multi-agent planning and dynamic obstacle avoidance	Succesul unit test &subsystem test for decision making system. Two vehicles autonomously driven around the full test track spanned by two sensor units.
10.	Internal	12 Nov, '23	Complex track and corresponding infrastructure sensors built	Successful unit and mechanical integration tests - required for DFR 2.
10.	External	21 Nov, '23	Fall Validation Demonstration	Elucidated here.
11.	External	28 Nov, '23	Fall Validation Demonstration - Encore	Demonstration of all desired performance and non-functional requirements.

Figure 18: Fall sschedule and milestones

9.4 Gantt Chart

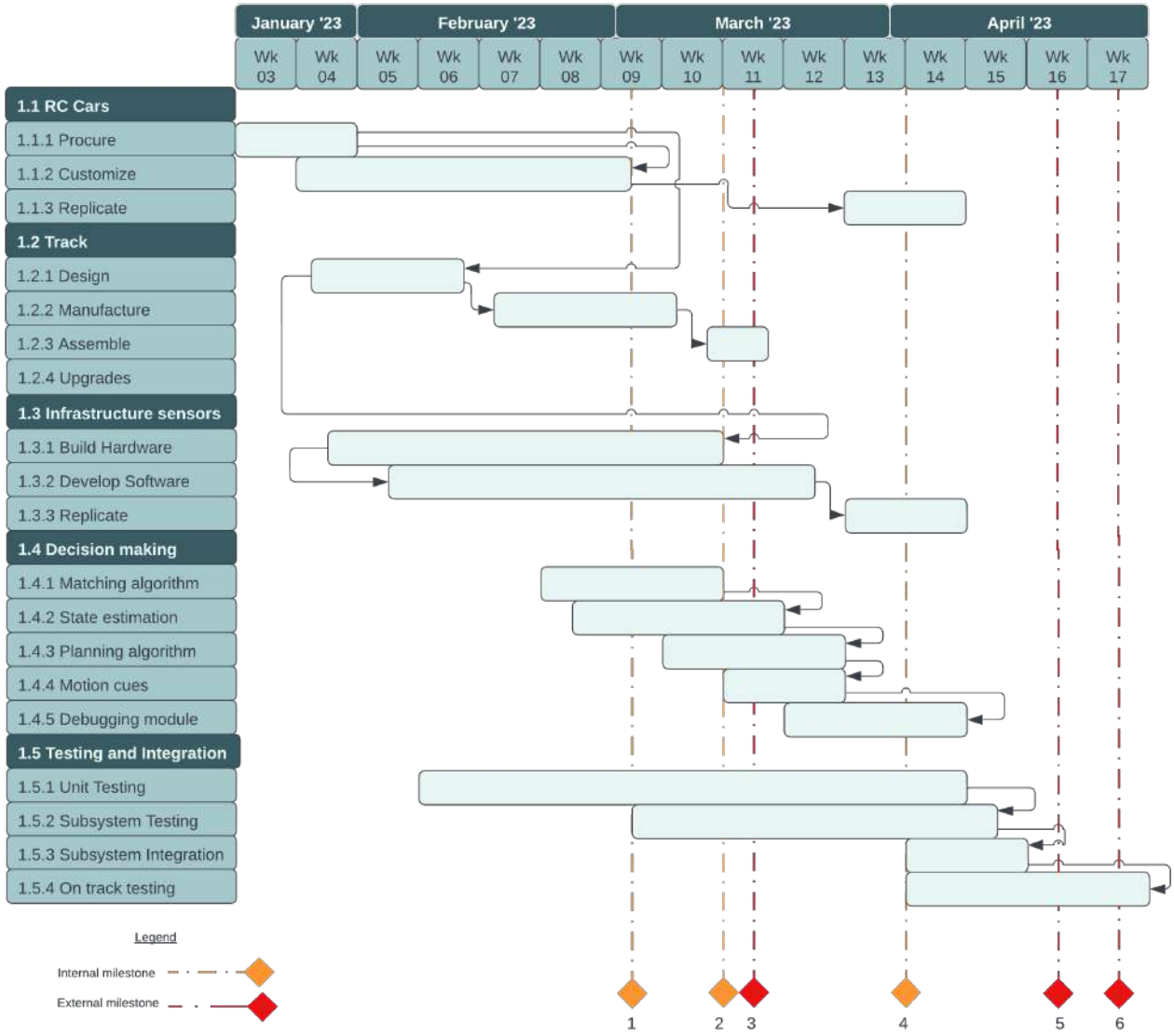
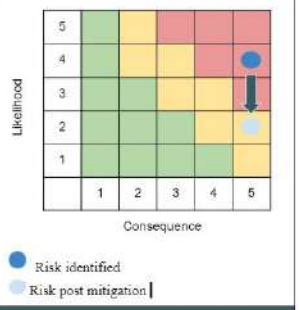


Figure 19: Gantt Chart

9.5 Risk Management

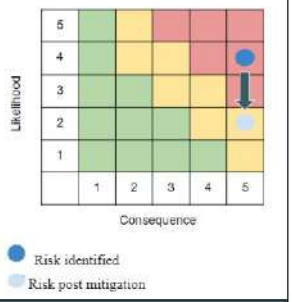
Risk Title	RC vehicle getting damaged whilst testing				
Risk Owner	Mechanical, Embedded and electronics	**Date submitted**	12/13/2022	**Date Updated**	12/13/2022
Risk Description		**Risk Type**			
RC vehicle going off track and crashing with infrastructure unit or onlookers causing damage to RC vehicle that can stop further testing and delay the project		Technical Schedule			



Risk Reduction plan			
Action No.	**Date**	**Action**	**Comments**
1	05/10/2023	Build cushioned fences along track boundaries to protect cars from falling/getting damaged	Testing will not be done until this is ensured
2	02/06/2022	Purchase Spares	Spares will be included when components are purchased

Figure 20: Risk 1

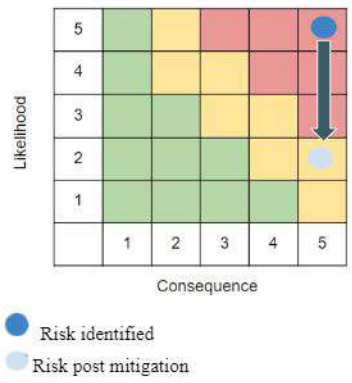
Risk Title	Unforeseen issues in building track				
Risk Owner	Mechanical	**Date submitted**	12/13/2022	**Date Updated**	12/13/2022
Risk Description		**Risk Type**			
Issues faced while building the track. Problems include Inaccuracies in track design development ,delay in procuring items or finalizing location for track		Technical Schedule			



Risk Reduction plan			
Action No.	**Date**	**Action**	**Comments**
1	01/15/2023	Prioritize track building	Team members and resource will allocated to complete this activity on priority
2	01/15/2023	Simplify design	All efforts will be taken to not make this activity complex and complete it quickly
3	01/30/2023	Team members will be trained in machine shop	Team member are skilled to complete tasks in short time

Figure 21: Risk 2

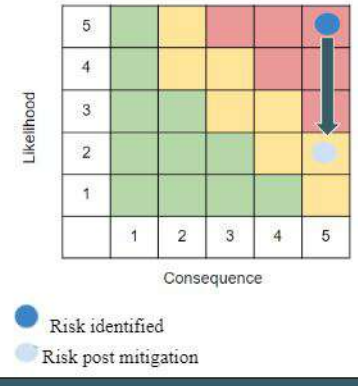
Risk Title	Unforeseen issues in building RC vehicle				
Risk Owner	Management	**Date submitted**	12/13/2022	**Date Updated**	12/13/2022
Risk Description		**Risk Type**			
Issues faced while building the RC vehicle. Problems include difficulty in making modifications on RC vehicles or delay in procuring components. Since multiple vehicles are required cost has to be considered such that it is within budget		Technical Schedule cost			



Risk Reduction plan			
Action No.	**Date**	**Action**	**Comments**
1	01/15/2023	Purchase RC vehicle and not build	Purchase RC vehicle that meets all the requirements and spend less effort in building
2	01/15/2023	Prioritize purchasing and Identify vendors	Decide on components and check for lead time, accordingly start procuring process
3	01/30/2023	Out source modification activities	Identify RC shops that will make the required modifications

Figure 22: Risk 3

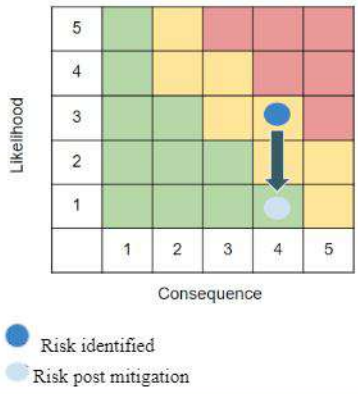
Risk Title	Latency or Communication issues				
Risk Owner	Planning and Controls	**Date submitted**	12/13/2022	**Date Updated**	12/13/2022
Risk Description		**Risk Type**			
Communication issues when data packet is lost and commands are not updated or due to some issue there is delay in computing can cause RC vehicle		Technical			



Risk Reduction plan			
Action No.	**Date**	**Action**	**Comments**
1	03/20/2023	Send a sequence of motion inputs at every time instance.	Commands to stop will be sent every instant, if command is not updated the vehicle will come to a stop.

Figure 23: Risk 4

Risk Title	Open source libraries incompatible				
Risk Owner	Middleware and software architecture	**Date submitted**	12/13/2022	**Date Updated**	12/13/2022
Risk Description		**Risk Type**			
Open source libraries might be incompatible with different components and can cause delay in projects .		Technical Schedule			



Risk Reduction plan			
Action No.	**Date**	**Action**	**Comments**
1	02/10/2023	Use libraries with minimum dependencies	Check all libraries required and their dependencies and according finalize
2	02/10/2023	Adopt libraries with maximum community support	Good community support can help in problem resolution

Figure 24: Risk 5

9.6 RC car and track requirements

RC Car requirements

- Ackermann steering
- Wheel speed sensor
- Steering sensor
- On-board communication
- On-board controller
- Differentiate by colors/ queue position/ markers
- No fiducial markers IMU

Track Requirements

- Lane width = 2 x Car Width
- Max. height of installation = scaled light pole height
- Two turning pads
- Should be spanned by 2 sensor units
- No track markings apart from lanes
- Have track markings to collect ground truth
- Modular design for adaptability
- Leveled surface
- Infrastructure support units to mount and position sensors
- Illumination support structures
- Barricades and cushions for safety