University of Illinois
Urbana-Champaign

ECE445

Senior Laboratory Design

V2V Based Network Cooperative Control System

TEAM 22:
Jiazhen Xu
Zihao Li
Yuxuan Jiang
Xinwen Zhu

SPONSOR: Simon Hu
TA: Xuyang Bai

March 24, 2023

Contents

1 Introduction

1.1 Problem and Solution

Intersections pose a significant risk to transportation safety, as they account for a vast majority of severe urban traffic accidents. For example, about 43% of all crashes in the United States occur at or near an intersection [3], about 40% of all casualty crashes in Norway occur at junctions, about 33% of crashes in Singapore. Moreover, these numbers kept increasing over the years[5].

A few major factors influencing accidents at intersections have been unearthed, such as intersection approach conditions, signal timing, curvature etc[3]. In our project, we mainly focus on the intersection approach conditions. Vehicles passing intersection often do not have perception to the traffic condition on other lanes due to lack of attention and visual blocks such as buildings, bridge piles, passing-by vehicles and even pillar blind spots. It would be very helpful if a vehicle can get a holistic view of the intersection conditions for crash avoidance.

Additionally, today, most of the urban intersections are under passive control mechanisms such stop signs, yield signs and traffic lights. Stop signs require vehicles to come to a complete stop, even when there are no other cars at the intersection. This can reduce efficiency by causing unnecessary deceleration. Similarly, passive traffic lights may cause cars to stop when there is no need to do so. According to a very conservative calculation performed by Victor Miller at Stanford University[4], unnecessary traffic stops in the United States can account for 1.2 billion gallon consumption per year, which can satisfy an average American to fill up a 15 gallon tank every other week. Such passive intersection control mechanisms have lead to significant amount of energy waste and call for adaptive control mechanisms.

To address the safety and efficiency issues at urban intersections, we propose to use V2V communication technology. V2V communication can help enlarge the perception field of one single vehicle, allowing it to intelligently understand the intersection's traffic condition and make decisions accordingly.

Our proposed system consists of several subsystems:

1. All vehicles equipped with V2V communication technology will be able to share their current state, including location, velocity, acceleration, and heading, with each other in real-time. This will enable each vehicle to have a holistic view of the intersection's traffic condition and make informed decisions to avoid potential accidents.

2. Computer vision and radar technology will be used to recognize and track approaching vehicles and other road users (pedestrians, motorcyclists, etc.) at the intersection. This subsystem will detect the presence and direction of the vehicles and send this information to the V2V communication subsystem. Besides, this subsystem also use the data from the V2V communication and intersection approach recognition subsystems to control the flow of vehicles through the intersection.

3. The avoidance algorithm running on the server will analyze the data received from the V2V communication and intersection approach recognition subsystems and make decisions on the best course of action to avoid potential collisions. The server will take into account factors such as vehicle velocity, distance, and direction of travel to ensure safe and efficient vehicle movements at the intersection.

We plan to implement the proposed system in two phases. In phase one, we will focus on the collision avoidance algorithm and the V2V communication subsystem. We will use simulation software to test and optimize the algorithms' performance. In phase two, we will integrate the intersection approach recognition and adaptive intersection control mechanism subsystems into the system and perform real-world testing.

1.2 Visual Aid

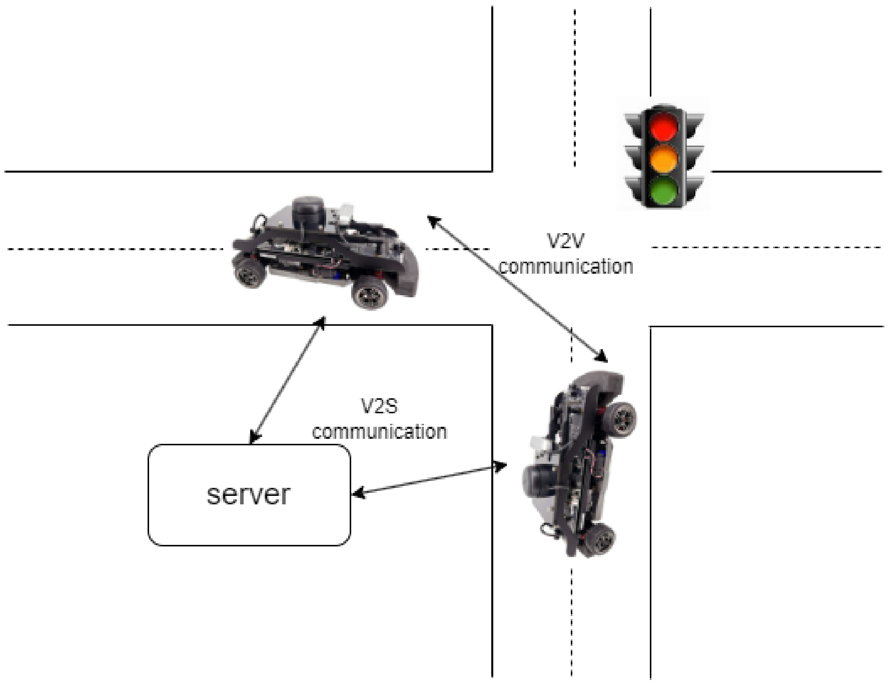

Figure 1: Visual Aid

1.3 High-level requirements

To ensure that our proposed solution is effective and efficient, we have established the following high-level requirements:

1. **Collision avoidance** : The system must achieve a minimum of 95% success rate in simulating collision avoidance at intersections.

2. **Object detection**: The vision-based object detection system must achieve a minimum of 90vehicles and pedestrians at intersections.

3. **Energy efficiency**: The overall energy consumption of the system must be lower than the energy consumption required by traditional traffic control mechanisms, such as traffic lights and stop signs.

By meeting these high-level requirements, we can ensure that our proposed solution addresses the safety and efficiency challenges at urban intersections effectively and sustainably.

2 Design

Block Diagram

The Block Diagram of our design is shown in Figure 2.

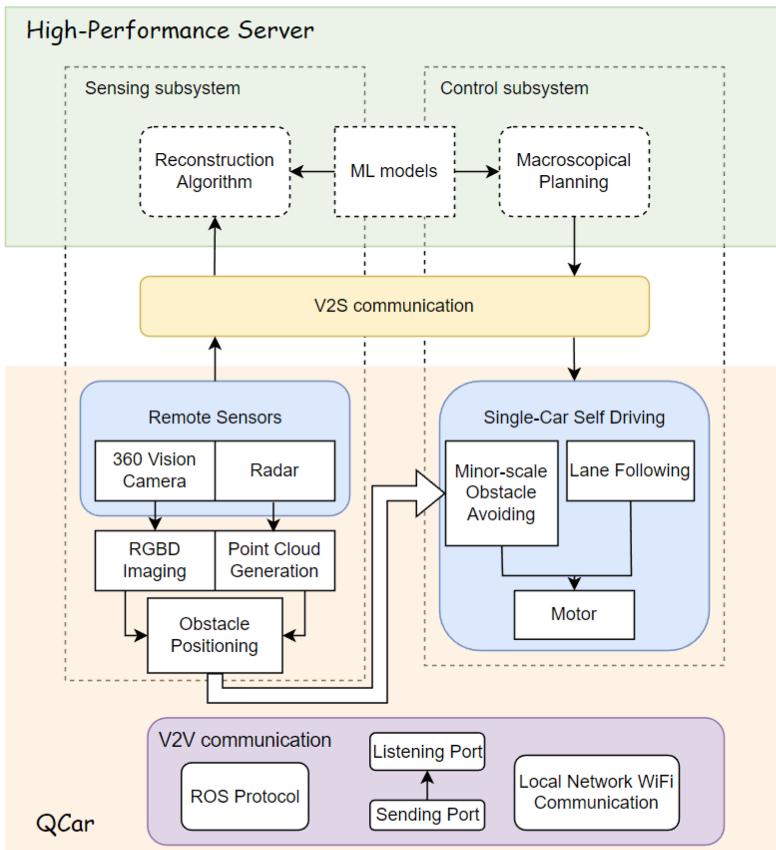

Figure 2: Block Diagram

In the lower level, the Qcar has pre-installed 360 Vision Camera and Radar to monitor its surrounding environment, which could generate the RGBD Imaging and Point Cloud. We use these information to determine the positions of obstacles around the Qcar and send them to our self-driving module. The Qcar itself is responsible for minor-scale obstacle avoiding and navigating. These functions are not well-developed yet on Qcar. The surrounding information will also be sent to a high-performance server to do visualization (reconstruction algorithm) and macroscopical planning. Note that we consider the server not to be a crucial part for safety–even without the server, Qcars can safely navigate through V2V communication and single-car decision making. We treat the server as an augmentation of our system.

Physical Diagram

Our team decides to use the Quanser Car[1] (Qcar) as the experimental car to finish the design. The Qcar Figure 3 is equipped with many sensors, such as a LIDAR, a RGBD camera, and two CSI cameras on the left and right side. Those sensors can capture detailed information about the environment. Besides, we make a detailed investigation on the physical characteristic of Qcar, which will be concerned when deciding V2V/V2X communication rate or setting parameters for mechanical movement control. Figure 4 and Figure 5 illustrate the dimensions of the Qcar. For the dynamic parameter of Qcar, we estimate that the maximum speed can reach 5m/s, and the stop distance will be calculated accordingly given the coefficient of frictional resistance in future experiment.

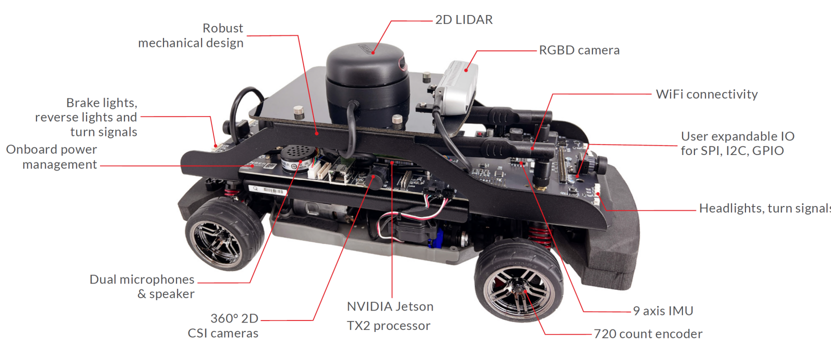

Figure 3: Qcar Diagram

2.1 Communication Subsystem

This communication subsystem, i.e. ROS protocol, is implemented and embedded in the Qcar development suit, so we don't need to realize it again. Our work is primitively to read relevant documents and codes to understand it and use it to transmit information for further control. However, it's an

Figure 4: Qcar Dimensions

Item	Value
weight	2.7kg
Length	0.425 m
Height	0.182 m
Width	0.192 m
Tire diameter	0.066 m
Wheelbase (Figure 4 #1)	0.256 m
Front and Rear Track (Figure 4 #2, 3)	0.170 m
Maximum steering angle	$\pm 30^{\circ}$

Figure 5: Dimensions

critical part in our system since the project is V2V communication-based. Therefore, we will provide a fair and brief description on how the protocol works and contributes to our whole control system.

The Robot Operating System (ROS) [2] is a distributed communication mechanism built on TCP/UDP, and it process communication at the granularity of process. In the ROS communication framework, every process is regarded as a *node*, and messages are passed via logic channels called *topic*s. When a node release a topic, all other nodes subscribe to him will receive that topic. Besides, the protocol of ROS also has its own data format that can characterise the status of the vehicle, including speed, poses, trajectory, and the point cloud module. Therefore, ROS satisfies our requirement for both vehicle-vehicle and vehicle-server communication.

Latency and bandwidth are always problems to consider for communication system, . However, after deeper consideration, we believe both of them won't affect our project because the communication is conducted in local wifi network and we only transmit computed results like positions or velocities that only take less than 100 Bytes. Hence the zero latency and infinite bandwidth are reasonable assumptions for the project. (A fair quantitative analysis)

Our future work for the V2V communication can be summarized as follows: Firstly, a robust ROS communication system is supposed to be built. With vast libraries built on ROS1, we need to support the communication of mixed ROS1 and ROS2 nodes to support full functionality of the ROS protocol.

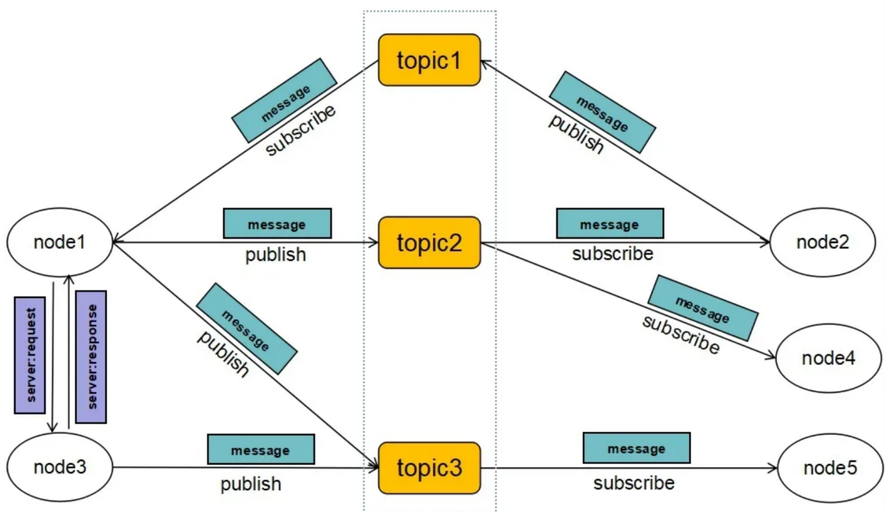

Figure 6: ROS protocol

Specifically, we should build a ROS1-bridge inside each vehicle and server to put the data from ROS1 to ROS2 packages. In addition, The ROS protocol requires that all nodes, i.e. vehicles and servers, stay in the same wifi network, so we must configure them in a local area network as well. Secondly, to fulfill the reliability of our control system, we need to guarantee that timeliness of information. As delayed information may cause a wrong decision of the vehicle, we need to set an upper bound of the latency and amend overtime information in a hearistic way to ensure the safety.

Requirement	Verification
Implement a reliable V2V communication based on existing protocol.	Send basic information between the server and vehicle, and show it on the screen.

Table 1: RV_ROS

2.2 Sensor information Processing

Quanser car contains a 360 camera and an on-vehicle GPU. We can run multi-objects tracking (MOT) algorithm on the car to detect other cars. Once we detect other cars on the camera, we can calculate the angle to detected cars by the position of the object in the image and FOV (field of view) of the camera.

Also, the radar on the car will help generate a point cloud model and position cars in the intersection more precisely. Combining the information of car in camera with 2D-point cloud generated by radar, we can get the precise relative position of detected car. After processing the sensor information, our car will control its motor and steering wheel to avoid collision and send the result of recognition algorithm to other connected cars (or server).

Camera: The QCar platform provides 360° of vision through the placement of four 8MP 2D CSI cameras (Figure 7) at the front, left, rear and right side of the vehicle. Each camera has a wide-angle lens providing up to 160° Horizontal-FOV (field of view) and 120° Vertical-FOV. The corresponding blind-spots have been shown below in Figure 8.

Figure 7: CSI Camera

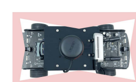

Figure 8: Blind spots of camera

RPLIDAR: RPLIDAR A2M8 is the enhanced version of 2D laser range scanner(LIDAR) (Figure 12).The system can perform 2D 360 degree scan within a 12-meter range(8-meter range of A2M8-R3 and the belowing models). It can take up to 8000 samples of laser ranging per second with high rotation speed. The typical scanning frequency of the RPLIDAR A2 is 10hz (600rpm). Under this condition, the resolution will be 0.45°. And the actual scanning frequency can be freely adjusted within the 5-15Hz range according to the requirements of users.

During every ranging process, the RPLIDAR emits modulated infrared laser signal and the laser signal is then reflected by the object to be detected. The returning signal is then sampled by vision acquisition system in RPLIDAR and the DSP embedded in RPLIDAR starts processing the sample data and outputs distance value and angle value between object and RPLIDAR via communication interface. When drove by the motor system, the range scanner core will rotate clockwise and perform the 360-degree scan for the current

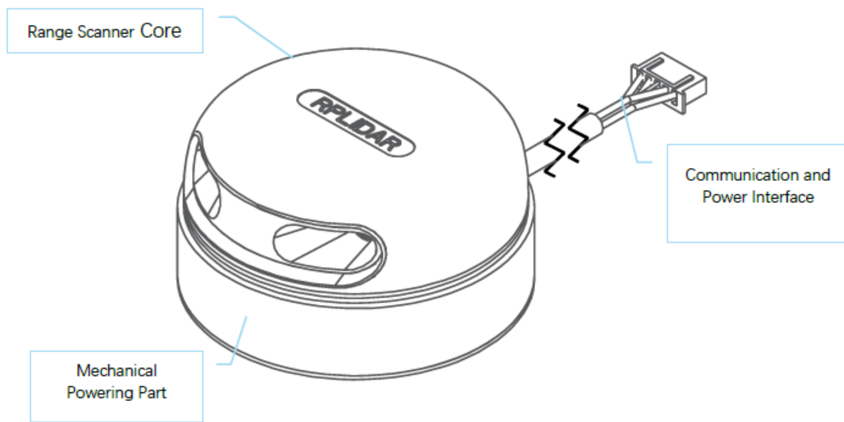

Figure 9: RPLIDAR System Composition

environment.

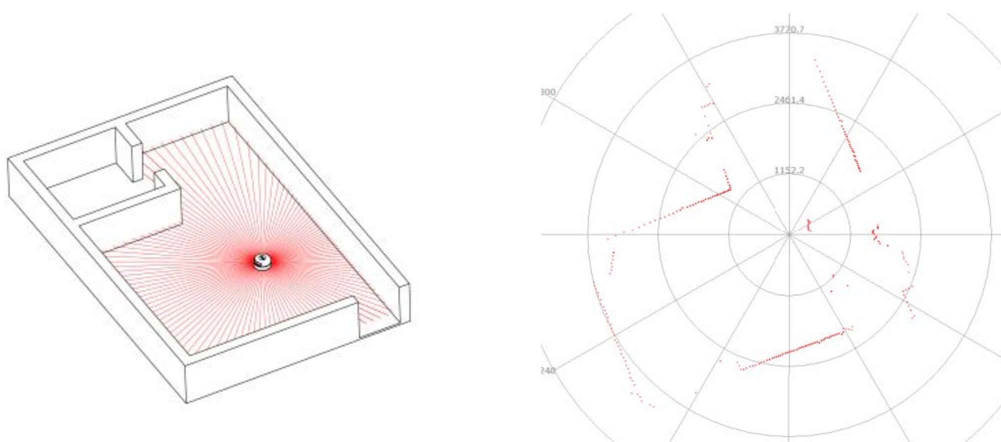

Figure 10: RPLIDAR mechanism

AI performance of Jetson TX2 series: 1.33 TFLOPS

2.3 V2V: Obstacle Avoidance

One of the main objectives of this project is to guarantee the safety of vehicles at intersections through automatic obstacle avoidance. Hence, the obstacle

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Figure 11: Parameters of YOLO models

Requirement	Verification
Radar should be able to detect other cars moving faster than 0.5m/s without the help of camera.	1. Set the speed of the third car to at least 0.5m/s. 2. Mask all output from camera and MOT algorithm. 3. Use the point cloud algorithm to extract the position of moving car. 4. Compare the predicted position with real position. The error should be within $(0.2 \times v_{car})m$.
Frame rate of AI model (MOT algorithm) should be above 10fps (spin rate of radar).	1. Choose an MOT model with $AP^{medium,IoU=.50} > 0.8$ for car. 2. Set the sleep time for each run of MOT algorithm. 3. Check if each run of the AI model can finish in 0.1s.
The detected car position offset should be within 5% in the range of 5 meters	1. Put a normal car at a certain distance from the Qcar. 2. Get the predicted distance to the normal car after processing information from sensors 3. Compare the predicted distance with the measured distance.

Table 2: RV_Sensing

avoidance algorithm is a critical component that utilizes processed sensor information from the V2V network as input, and the output is the movement of the vehicles. Traditional obstacle avoidance methods rely on the perception capability of a single vehicle, which can be blocked by moving objects and road-side buildings, leading to less-efficient and conservative car control. The V2V approach is a natural solution to this problem since it combines the perception field of multiple vehicles, making the obstacle avoidance algorithm less conservative and safer.

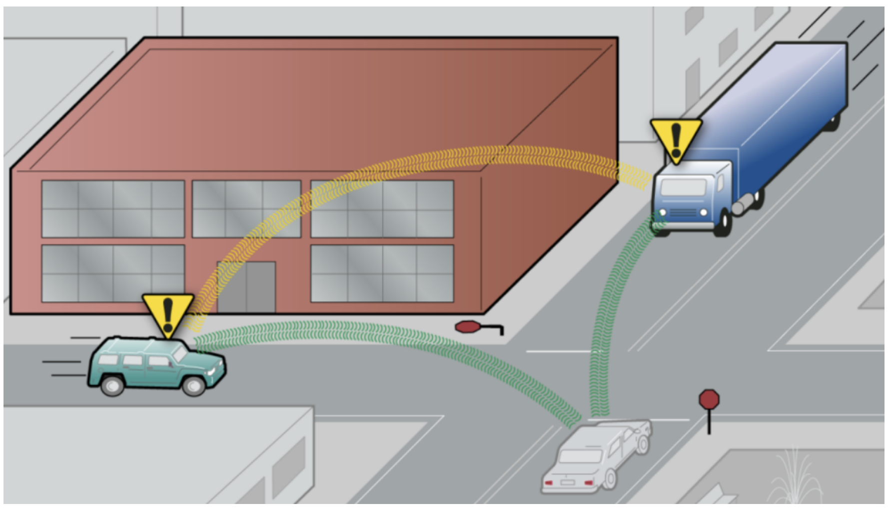

Figure 12: V2V Obstacle Avoidance Example

Designing a V2V-based intersection obstacle-avoidance algorithm involves several steps to ensure its safety and effectiveness. Firstly, the algorithm needs to collect information about the traffic conditions at the intersection from the Sensor Information Processing subsystem and other vehicles through V2V communication. The information includes the location, speed, and direction of other vehicles, pedestrians, and cyclists around the intersection. Each car's onboard computing unit calculates this information separately and then sends it to other nearby vehicles through reliable multicast.

Once the information is collected, the algorithm prioritizes potential obstacles and determines the level of risk posed by each obstacle. Additionally, the algorithm calculates the optimal path for the vehicle to take to avoid each obstacle. Regression algorithms are used to analyze the predicted location of each obstacle when the local vehicle arrives at the intersection, based on time-series data. The algorithm then calculates the appropriate speed change and direction for the car until the risk is resolved, considering the potential obstacles and the current traffic conditions.

Finally, the algorithm takes control of the car and adjusts its movement to avoid any potential collisions. The algorithm prioritizes safety over efficiency, avoiding sudden or erratic movements that could endanger other vehicles or pedestrians.

To ensure the effectiveness of the algorithm, we will test it in a controlled environment using simulated traffic scenarios and optimized to achieve a high success rate in avoiding obstacles and preventing collisions. The details of the requirements can be seen in the RV table.

Requirement	**Verification**
• The algorithm should accurately predict potential collisions at intersections based on information from V2V communication and local sensor processing. • The algorithm should be able to safely navigate through the intersection by avoiding predicted collisions, while not exceeding the physical movement limits of the vehicle.	• Collect a dataset of collision scenarios and test the algorithm using various simulations. • Verify that the algorithm can accurately identify potential collisions in all simulated scenarios. • Verify that the algorithm can safely navigate through intersections and avoid collisions in all simulated scenarios. • Conduct various simulations with the algorithm deployed in a vehicle and verify that the overall accident rate is less than 5

Table 3: Requirements and verification for the V2V obstacle-avoidance algorithm

2.4 Central Traffic Coordinator Subsystem

V2X communication makes the system more efficient. We use central computer and server interchangeably. Our autonomous driving system is a complex system that uses various technologies to allow vehicles to operate without human intervention. In this system, the car is equipped with a range of sensors, such as RGBD and high-resolution cameras, radar, and lidar, that help it gather information about its surroundings. This data is then sent and processed by a central computer, which uses machine learning algorithms to analyze the information and make decisions about how the car should operate in a higher level. That is, considering the possible case where the central computer is down, while the cars have full authority of function itself with the highest priority on safety, they adapt to the commands from the central computer when available.

We describe such functionality of the central as "macro-scale management", compared with the cars' "micro-scale decision-making". In a system where cars can make independent decisions under the control of a server for macro-scale management, the cars are connected to a central server via a wireless network. The server acts as a control center, receiving information from all the cars in the system and using this data to make decisions about traffic flow, route planning, conflict resolving, and other macro-scale management tasks.

On the other hand, each car in the system can operate autonomously on itself, using its sensors and onboard computer to make decisions about its immediate environment. However, these decisions are also influenced by the information received from the central server, when available. For example, if the server detects congestion on a particular route, it may instruct some cars to take an alternative route to avoid the traffic. But if such message from the server does not exist, the car can operate based on its own information and the messages sent from other cars via V2V communication.

The central server, whose representative would be the Ground Control Station (GCS) in the Qcar system, has the ability to communicate with individual cars in the system, sending them specific instructions or alerts as needed. For example, if a car is approaching a hazardous area, such as a construction zone or an accident scene, the server may send a warning message to the car to slow down or take a different route.

Our design is different from those solutions that relies mostly on the server, in the perspective that individual cars also play crucial parts. Some exist solutions for autonomous driving takes the cars only as sensors. They send information to the server, waiting for the response from the server, then do exactly what the server commands. Such system is feasible in some sense, but suffers from the limit of computational power of the central server, and risk from the unreliability of central computer system, the network, with further inability to deal with emergent situations that requires immediate response form the car. Sending information to the server, waiting for it to compute, then send back from central computer to embedded OS on Qcar takes longer time than crash in an emergency.

Overall, an autonomous driving system with independent decision-making capabilities under the control of a central server has the potential to greatly improve traffic flow, reduce congestion, and enhance overall safety on the roads. However, the technology is still in its early stages of development and will require ongoing research and development to achieve its full potential. We hope our work could contribute to the development of such technologies.

Requirement	Verification
A macroscopic map is constructed on the server.	1. Turn on the monitoring module on computer. 2. Turn on the whole system. 3. Check if a visualization is constructed on server.
A macroscopic path planning algorithm is implemented.	1. Choose the start and end position of each car. 2. Turn on the system, run the planning algorithm. 3. Check if the cars can reach their target efficiently.
The system should work safely without the central computer	1. Turn off the planning function of the central computer. Only leave the monitoring function. 2. Turn on Qcars in our scenario. 3. Check if each Qcar runs safely.

Table 4: RV_V2X

2.5 RV Table

To evaluate the work for each subsystem shown above, we conclude those sections and assign points to each requirement item in the following table Table 2.5:

Subsystem	Requirement	Verification	Points
Communication Protocol	Implement a reliable V2V communication based on existing protocol.	Send basic information between the server and vehicle, and show it on the screen.	10
Sensor information Processing	Radar should be able to detect other cars moving faster than 0.5m/s without the help of camera.	1. Set the speed of the third car to at least 0.5m/s. 2. Mask all output from camera and MOT algorithm. 3. Use the point cloud algorithm to extract the position of moving car. 4. Compare the predicted position with real position. The error should be within $(0.2 \times v_{car})m$.	5
Sensor information Processing	Frame rate of AI model (MOT algorithm) should be above 10fps (spin rate of radar).	1. Choose an MOT model with $AP^{medium,IoU=.50} > 0.8$ for car. 2. Set the sleep time for each run of MOT algorithm. 3. Check if each run of the AI model can finish in 0.1s.	5
Sensor information Processing	The detected car position offset should be within 5% in the range of 5 meters	1. Put a normal car at a certain distance from the Qcar. 2. Get the predicted distance to the normal car after processing information from sensors 3. Compare the predicted distance with the measured distance.	5
V2V: Obstacle Avoidance	The algorithm should accurately predict potential collisions at intersections based on information from V2V communication and local sensor processing.	1.Collect a dataset of collision scenarios and test the algorithm using various simulations. 2. Verify that the algorithm can accurately identify potential collisions in all simulated scenarios.	5
V2V: Obstacle Avoidance	The algorithm should be able to safely navigate through the intersection by avoiding predicted collisions, while not exceeding the physical movement limits of the vehicle.	1.Verify that the algorithm can safely navigate through intersections and avoid collisions in all simulated scenarios. 2.Conduct various simulations with the algorithm deployed in a vehicle and verify that the overall accident rate is less than 5%.	5
Central Traffic Coordinator Subsystem	A macroscopic map is constructed on the server.	1. Turn on the monitoring module on computer. 2. Turn on the whole system. 3. Check if a visualization is constructed on server.	5
Central Traffic Coordinator Subsystem	A macroscopic path planning algorithm is implemented.	1. Choose the start and end position of each car. 2. Turn on the system, run the planning algorithm. 3. Check if the cars can reach their target efficiently.	5
Central Traffic Coordinator Subsystem	The system should work safely without the central computer	1. Turn off the planning function of the central computer. Only leave the monitoring function. 2. Turn on Qcars in our scenario. 3. Check if each Qcar runs safely.	5

Table 5: Total RV Table

2.6 Tolerance Analysis

The precision of multiple critical parts in our system is limited:

- The precision of sensors: camera. The camera has high-precision, but we need to scale the image into specific resolution. The lower the resolution, the lower the precision. Each camera has 160° Horizontal-FOV and 120° Vertical-FOV, therefore if we're using 4x:3x resolution, the precision in euclidean space would be:

$$\Delta_{Cam} = \frac{d \cdot rand(160/2)}{4x/2} = \frac{d \cdot 80/180 \cdot \pi}{2x} = 0.6981\frac{d}{x} \tag{1}$$

  Where d is the distance of an object from the camera. For $x = 160$ and $d = 2$ (meters) the precision is 0.008726 (meters). This means an object in 2 meters (4 times the car's length) will have an error less than 1 centimeter.

- The precision of sensors: RPLIDAR. The resolution of RPLIDAR is 0.45°, therefore the precision in euclidean space would be:

$$\Delta_{LIDAR} = d \cdot rand(0.45) = 0.007854d \tag{2}$$

  Where d is the distance of an object from the camera. For $d = 2$ (meters) the maximum error is 0.0157 (meters). This means an object in 2 meters (4 times the car's length) will have an error approximately 1.57 centimeter. The advantage of RPLIDAR than cameras is the speed. Signal navigating and processing time is very fast and can be ignored. Combining both precision of cameras and LIDAR, the general precision of position detection $\Delta_{Pos}$is within our requirement and verification. For a distance of 5 meters the error is at most 4 centimeters (within 5%).

$$\Delta_{Cam} \leq \Delta_{Pos} \leq \Delta_{LIDAR} \tag{3}$$

3 Cost and Schedule

As all infrastructure and the Qcar development suit have been provided in the laboratory, we do not need to purchase anything else. All costs are labor cost as each member in the team will at least spend 10h/week on average. The schedule of our project is shown below Table 6:

Time	Task
March 31	Xinwen Zhu & Zihao Li: Implement multi-object tracking algorithm on camera image of Qcar Jiazhen Xu: Set up a local communication in ROS protocol Yuxuan Jiang: Realize positioning of Qcar itself (Qcar doesn't have positioning module like GPS).
April 7	Xinwen Zhu & Zihao Li: Realize positioning of other cars by camera image and point cloud generated by radar. Jiazhen Xu & Yuxuan Jiang: Realize point cloud fusion. Can transmit location of detected car.
April 14	Allmembers: Realize prediction of Qcar trajectory.
April 21	All members: Realize collision avoidance on Qcar.
April 28	All members: Realize the policy that decide whether Qcar should move on or continue stopping both on Qcar and ground control station.

Table 6: Schedule

4 Ethics and Safety

Ethics and safety are crucial aspects of any new technology, especially those involving autonomous or semi-autonomous systems. Below are some key points to consider:

- **Privacy**: The use of V2V technology raises concerns about privacy, as it involves the exchange of sensitive information between vehicles. It is important to ensure that personal information is properly protected and that the data is only used for the intended purpose.
- **Bias and discrimination**: As with any technology, there is a risk of bias and discrimination in the development and deployment of V2V systems. It is important to ensure that the algorithms and systems are designed to be fair and equitable, and that they do not perpetuate or exacerbate existing inequalities.
- **Accountability and liability**: With the introduction of V2V technology, there may be questions about who is responsible in the event

of an accident or malfunction. It is important to establish clear lines of accountability and liability, and to ensure that there are mechanisms in place for addressing any issues that may arise.

- **User trust and acceptance**: For V2V technology to be successful, it is important that users trust and accept the system. This requires clear communication about how the technology works, what data is being collected and how it is being used, and what the benefits and risks are. It is also important to involve users in the design and testing of the system to ensure that their needs and concerns are taken into account.

- **Regulatory compliance**: V2V technology will be subject to various regulations and standards, both at the national and international level. It is important to ensure that the technology is developed and deployed in compliance with these regulations, and that there is ongoing monitoring and evaluation to ensure that the system continues to meet these standards.

References

[1] Qcar. https://www.quanser.com/products/qcar/.

[2] Ros 2 document. https://docs.ros.org/en/foxy/index.html, 2023.

[3] Dominique Lord, Ida van Schalkwyk, Susan Chrysler, and Loren Staplin. A strategy to reduce older driver injuries at intersections using more accommodating roundabout design practices. *Accident Analysis Prevention*, 39(3):427–432, 2007.

[4] Victor Miller. The impact of stopping on fuel consumption. http://large.stanford.edu/courses/2011/ph240/miller1/, 2011.

[5] Richard Tay and Shakil Rifaat. Factors contributing to the severity of intersection crashes. *Journal of Advanced Transportation*, 41:245 – 265, 06 2007.