Report on Tollgate-1
System Modeling
Group - 3

Our project is about autonomous vehicles travelling in a platoon in 2040 in the scenario of a pandemic. The goal is to demonstrate the way robots change the lanes and react at the intersection, at the traffic lights, and in the case of an emergency vehicle (EV) appearance. The report consists of two principle parts – functional and non-functional requirements of our system.

Functional requirements

To achieve our goal, we will use various kinds of sensors: encoders to measure speed and distance travelled by the robot, and for more precise control of the motors; IR Light sensors to follow black tracks and detect the intersections; camera to detect traffic lights, obstacles and other vehicles; accelerator to measure the acceleration of the robot; ultra-sonic sensor to detect the distance to objects.

Also, we have 2 boards in the robot (Fig.1): Redbot main board which is based on Arduino Uno, and Raspberry Pi 4 Model B. As suggested by Fig.2, we plan to use Redbot main board to control driver motors, process IR and ultrasonic sensors' information. We are going to use Raspberry Pi to process camera's and accelerometer's data.

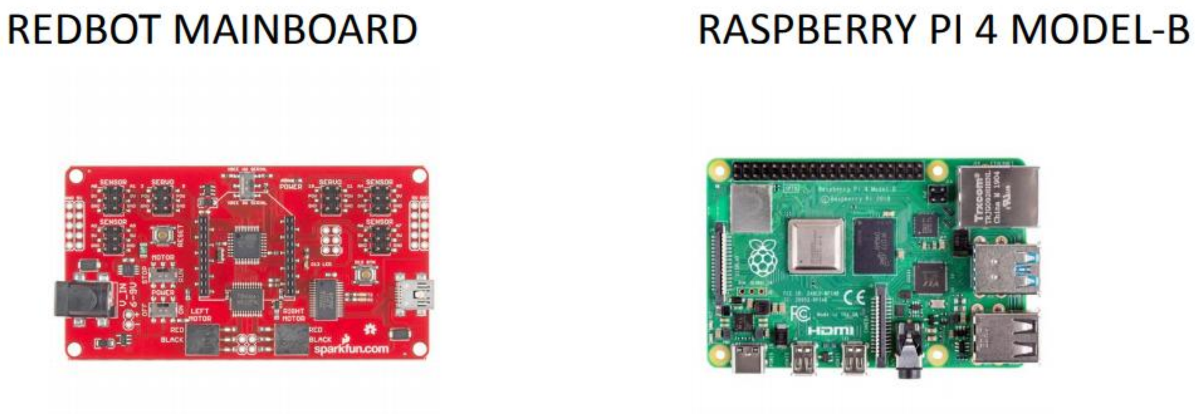

Fig.1. Controller boards

ROBOT ARCHITECTURE:

Figure 2 shows the basic block diagram of robot architecture.

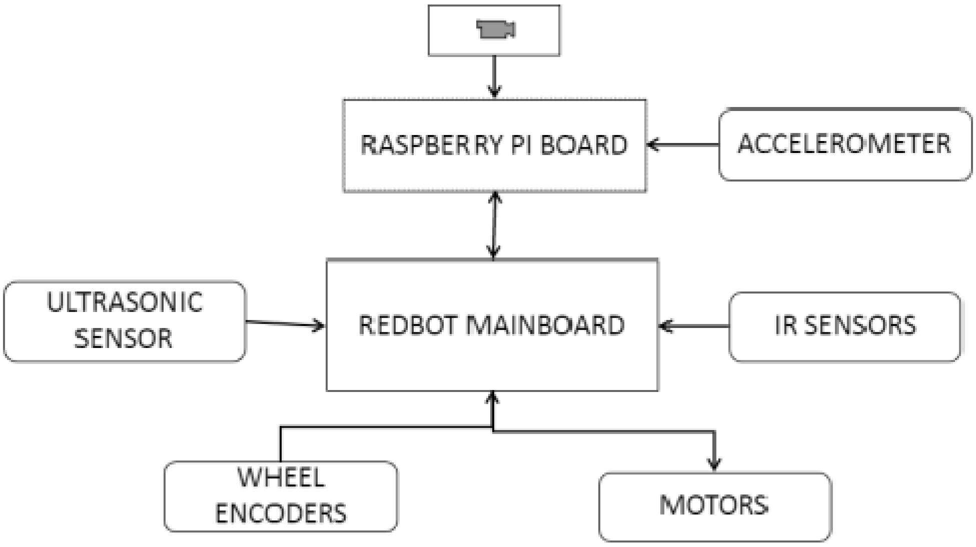

Fig.2. Robot architecture

BASIC ENVIRONMENTAL SETUP:

We can see the environmental setup (which is borrowed from DEIS 2017 project) which has elliptical track design with some buildings like hospitals as well as regular and emergency vehicles.

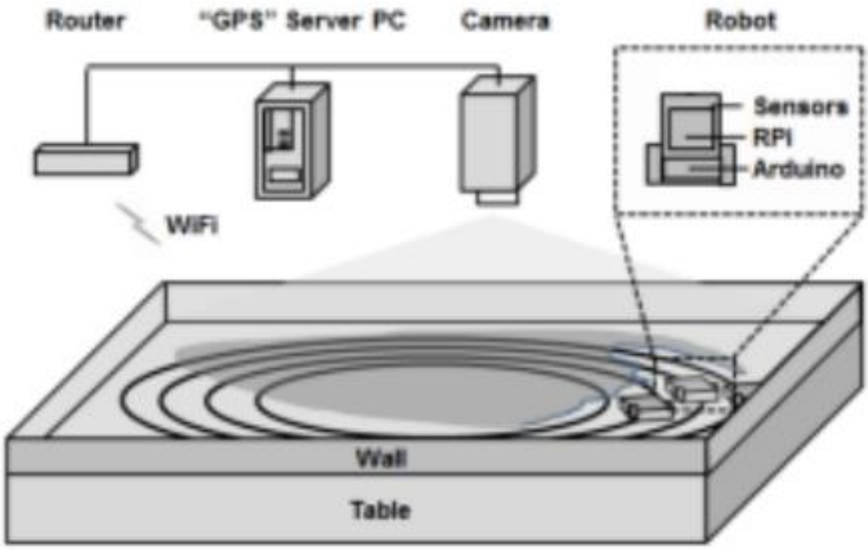

Fig.3. Basic environmental setup

Functional requirements of our system are discussed including basic scenarios.

AVOIDING COLLISIONS:

- **Avoid static obstacles**:

The robot will stop or change the lane when there is an obstacle on the lane.

- **Avoid moving obstacles**:

The robot will either stop or accelerate or decelerate depending on the situation. It happens mostly in case of an emergency vehicle as well as other regular robots.

- **React to traffic lights**:

The robot will maintain same speed if it is green light. The robot will decelerate and eventual stop at signal if it is red.

PLATOONING:
The assembling and disassembling of robots in platoon are decided by the local server (can be a laptop or leader, it will be decided later as project progress). The vehicles may either assemble or disassemble based on the situation. The vehicles should maintain safe distance between themselves and others.
The vehicles in the platoon should break or change lanes if the leader asks them to.

On Fig.4, sequence diagram of breaking the platoon, is presented.

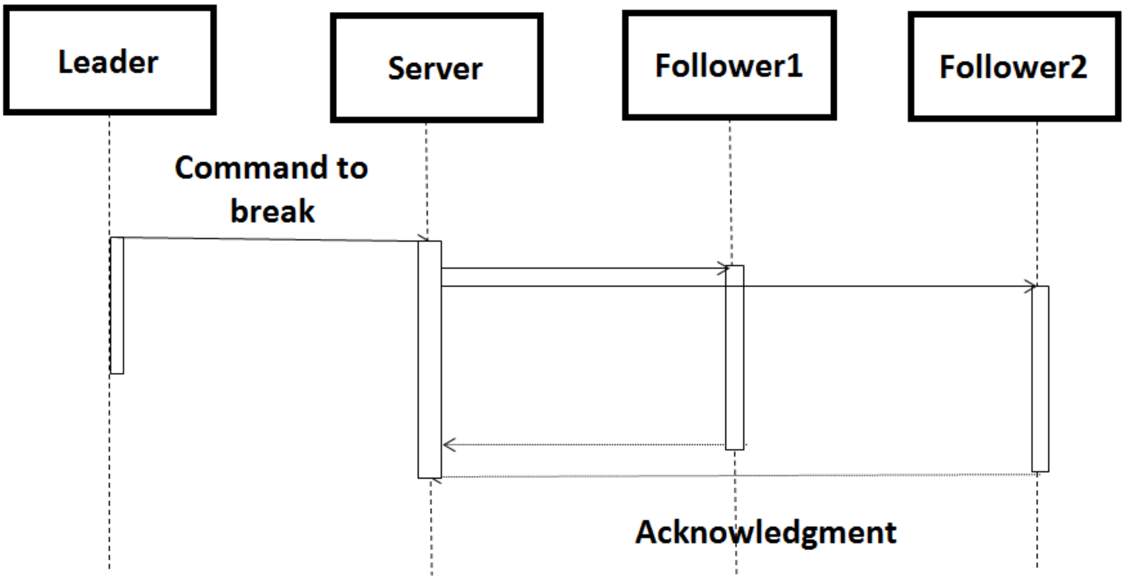

Fig. 4. Breaking the platoon.

For simple understanding of leader and follower, you can see the below (rough) class diagram. Basically, leader robot and follower robot are inherited version of regular robot but with additional properties.

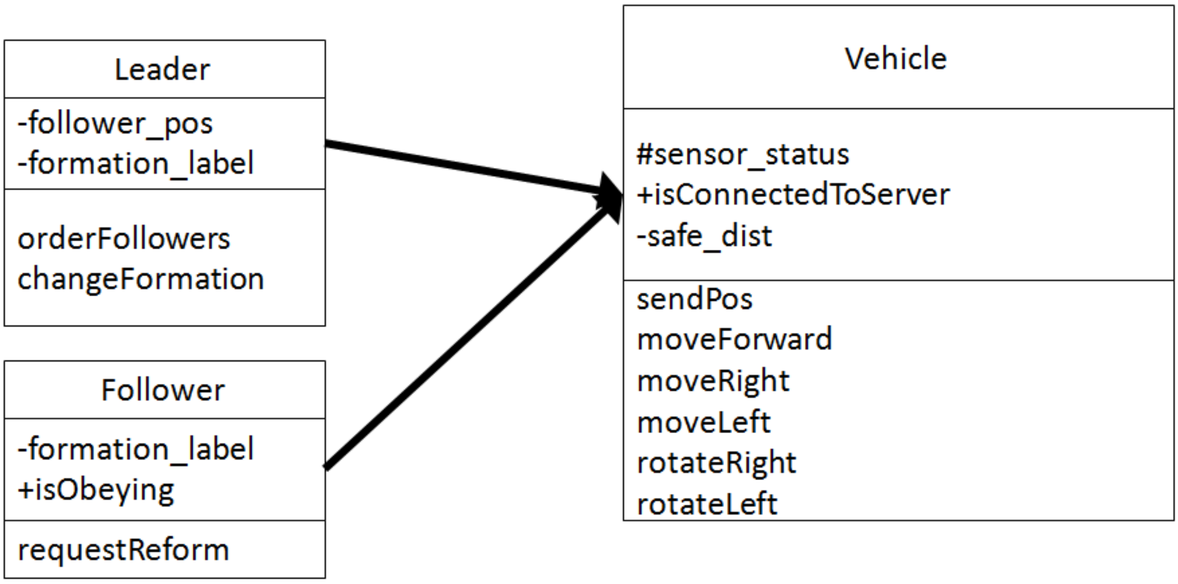

Fig. 5. Class diagram of leader and follower and regular vehicle.

Non-functional requirements of our system are presented.

1. **Usability** - Our system can help the EVs to reach the destination by managing the platooning structure using different reform/ deform strategies.

2. **Portability** - The system can be portable in software as well as hardware modules.

3. **Scalability** – The system can scalable to maximum possible level considering the environmental conditions such as congestion on the road, interference in communication, etc.

4. **Reliability** - Our system can perform efficiently throughout its lifetime under controlled environments.

5. **Availability** - The availability of the system is directly proportional to the battery life of the robot. When the battery is discharged, robot will go out of order and system will be unavailable. The system is available to deliver its potential when environment parameters (GPS server, communication medium) are active.

6. **Maintainability** - When maintenance is needed the system maintenance can be done in a garage and then the system is restored after validation.

7. **Interoperability** - The system can work correctly in the presence of other robots.

8. **Data Integrity** - The system can hold accuracy and consistency in data over its execution cycle. (data encryption techniques are not discussed in this phase of implementation; it may be taken into consideration as project progress in implementation).

9. **Safety**- The system will be designed in such a way that it should maintained safe distance between vehicles and avoid the collision with other objects.

Individual Contribution in project implementation:

Galina: She will be managing ROS2 communication between robots; creating different test scenarios; generating mocks in the scene (virtual objects like pedestrian, other robots, etc.) when physical objects may not be available.

Renu: Redbot main board interfacing with IR sensors and ultrasonic sensor. Implementation of entire development and test codes.

Vinay: Raspberry Pi interfacing with an accelerometer and a camera. These sensors are useful in inclinations or various terrains, object detection (for obstacles, traffic light, making platoon structure, keeping safe distance).

Akhil: Redbot main board interfacing motors. Implementation of entire development and test codes. It is based on wheels encoders, so dead reckoning will be developed based on likely and unlikely cases (like wheel slippage) to maintain the correct direction and motion of a robot.

[***Note***: The distribution of work on use of drone, communication methods, sensor fusion algorithms, localization, data encryption, and other important tasks will be decided later stage of the project.]