Distributed Systems Project

Team 3

Task 1:

1. **Identify which data/signal/events are required for the interaction / communication between the trucks**

1.1 Specify an appropriate protocol

To enable the trucks to communicate with each other, we have to use a wireless communication protocol that has low latency and high bandwidth data exchange. We can choose Wifi for this communication because this communication has:

- Operate in 2.4 or 5 GHz frequency
- Maximum data rate is ranging from 1.73 Gbps to 866.7 Gbps depending on type of Wifi
- Wi-Fi networks can be secured using encryption and other security measures, which protect against unauthorized access and hacking.
- Wi-Fi networks can cover a wide range of distances, depending on the type of router and the environment in which it is used.

1.2 Use State Machines for the model-based specification

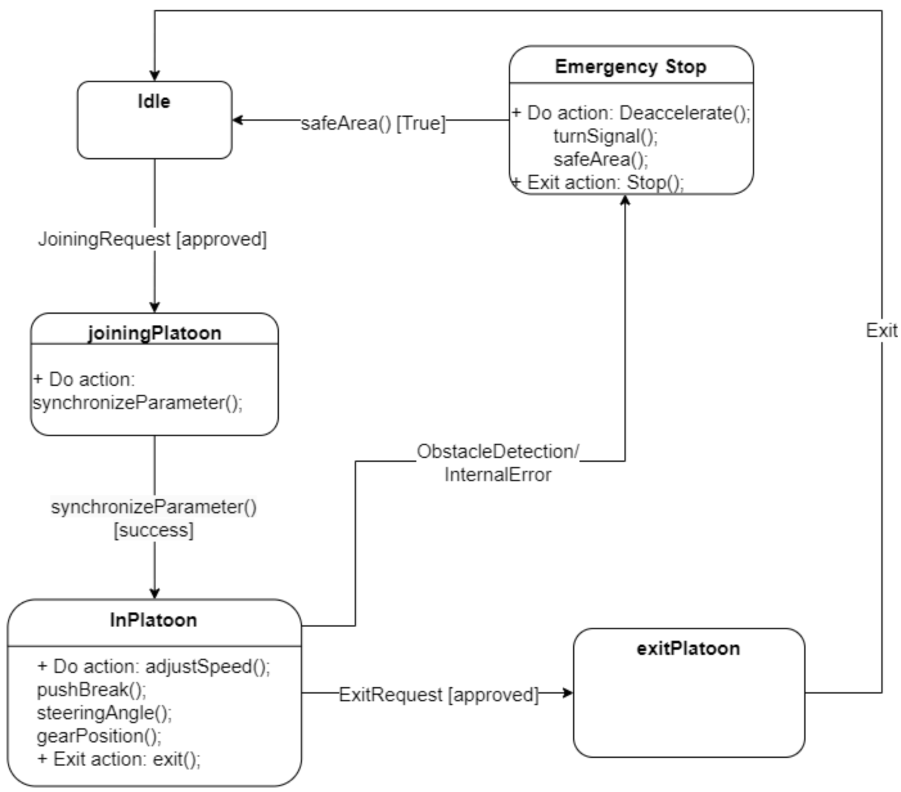
Data:

- *Object detection*
- *Break information*
- *Acceleration information*
- *Deacceleration information*
- *Steering angle*
- *Gear position*
- *The distance from precedence truck*

Communication Protocol:
- low latency
- high bandwidth data exchange
=> Wifi

Data:

- **object detection**
- **break information**
- **acceleration information**
- **deacceleration infor**
- **steering angle**
- **gear position**
- **The distance from precedence truck**



2. Identify the relevant control behaviour for the trucks

2.1 How can the distance to the precedence truck be guaranteed

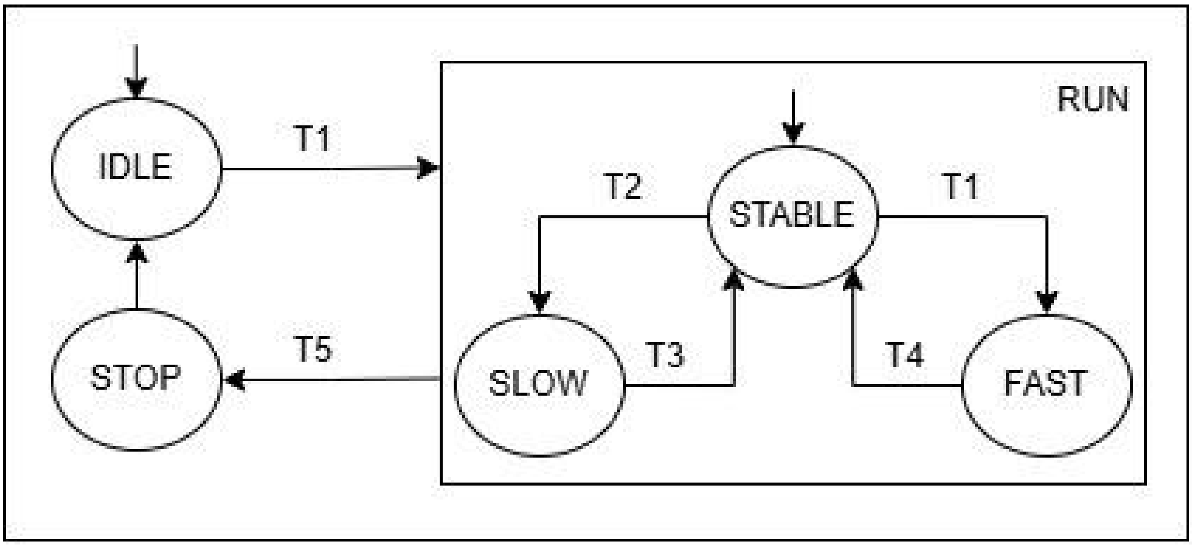
The distance can be guaranteed as follows:

- A sensor will be equipped with each truck (except the leading truck) to detect the distance with the precedence truck.
- There is a server or database or something that can store the status of all trucks (radar value for detection) and the action of the lead truck (such as braking, turn left/turn right).
- Trucks will communicate via Wifi/Bluetooth/... and will access to the database periodically to update latest information.
- Lead truck updates its action to the database.
- A thread hold for distance must be defined so that the latter truck can control their speed (slow down/ speed up). Thread hold value should be calculated to avoid any collision that may occur because of delay.

- If something crosses between the platooning, the truck can stop. That action will be updated to the precedence trucks, those trucks should then stop as well.

State machine of one truck controlling speed to keep distance:

- d: current distance with the precedence truck
- th_max/min: maximum/minimum distance between 2 trucks
- leading_RUN: holds TRUE when the leading truck runs.
- leading_STOP: holds TRUE when the leading truck stops.
- latter_STOP: holds TRUE when the latter truck stops.
- obstacle: holds TRUE when there is something crossing.



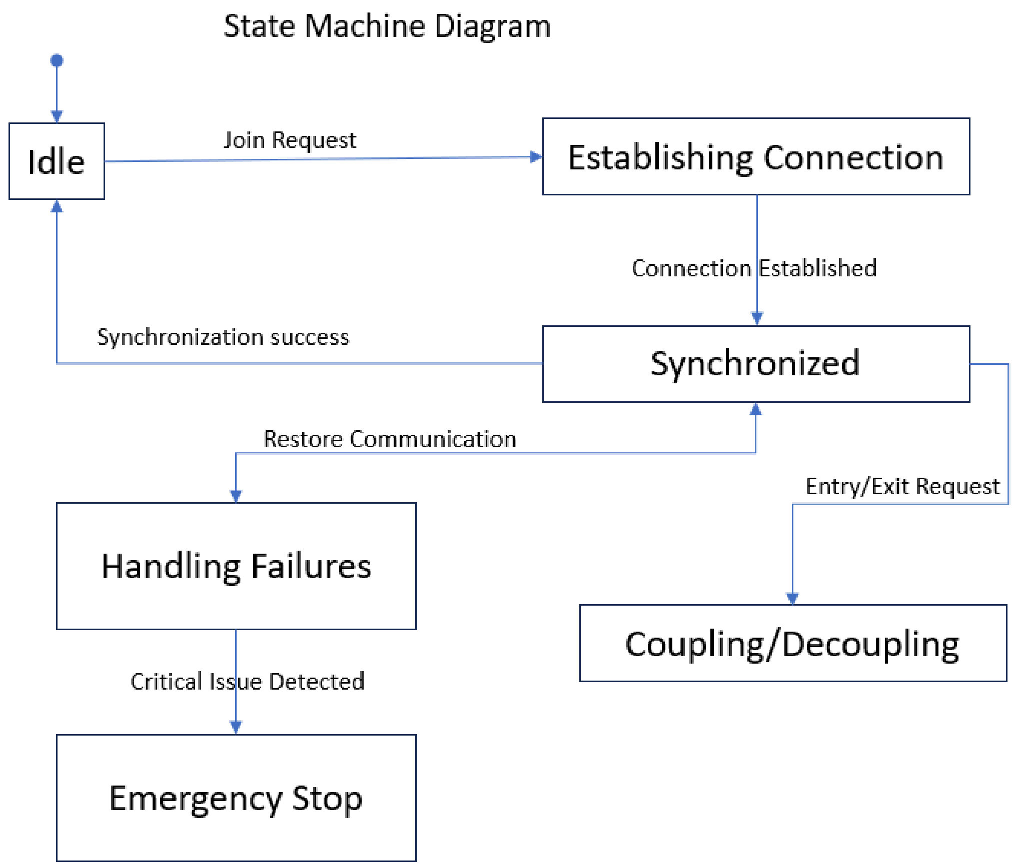
- T1: (d>th_max) && (leading_RUN)
- T2: d<th_min
- T1: d>th_max
- T3: d>th_min
- T4: d<th_max
- T5: (obstacle) && (latter_STOP) && (leading_STOP)

2.2 What happens in cases of a e.g. communication failure - > is your system robust / still stable?

It is important to have mechanisms that ensure the system remains robust and stable during communication failure. The following suggestion could be considered:

- Fallback Mechanisms: Trucks can switch to pre-programmed fallback algorithms that maintain safe distances and speeds based on sensor data alone and runtime average measurement results.
- Backup systems: Implement backup systems that can take control when primary systems fail
- Use multiple communication channels to ensure continuous data exchange. If one fails, the others can take over smoothly.

2.3 Use State Machines (and/or Activity Diagrams) for the model based specification



Activity Diagram