**Simulation paradigm**

**Discrete time with agent-based** model was selected as our paradigm for the project. We used discrete-time modelling in our traffic simulation because we need to manage and update the system state in a controlled and precise manner, which is important when dealing with the dynamic nature of traffic flow. This approach allows the simulation to accurately capture the timing of key traffic such as vehicle arrivals, lane changes, and speed adjustments and ensures important details are not missed. By updating the system at distinct intervals, we can ensure that all interactions and transitions within the traffic environment are processed sequentially, minimising errors and improving the reliability of the simulation output. The use of discrete-time modelling hence provides a solid foundation for building a robust and versatile traffic simulation capable of addressing a wide range of traffic management and planning issues. The other paradigm we used was Agents based modelling. We use it so that cars can have proper behaviours on the highway: moving, keeping safe distance, accelerating. For e.g., if the car in front of it slows down, it will also slow down to keep a safe distance following the 2 seconds rule.

**Architecture**

In our traffic simulation, we have 2 simulation files: 1 for early merge and 1 for late merge. In each file, we've created two main classes: Car and Simulation, to accurately model vehicle behaviours and manage the overall traffic system. The Car class manages movement of cars on the highway, the ramp and at the split, keeping a safe distance between two cars based on the 2 second rule and acceleration based on their speed. On the other hand, the Simulation class oversees the wider traffic environment. It handles the arrival of vehicles on lanes A and B, merging, changing lanes and the animation.The class also ensures that lane changes are safe and merging safe and removes vehicles from the highway as they reach their destinations. The model also uses threads to simulate the arrivals of the cars on the upstream and the ramp at the same time. Additionally, we used matplotlib’s FuncAnimation to animate our model. Using a Cartesian coordinate system, the simulation tracks each vehicle's exact x and y positions, updating them at every simulation step to reflect real-time movements and interactions. The animate function updates the positions of the cars on every frame and the run function runs the animation loop. The model also collects data during the simulation for analysis.

**Choices of input distributions**

Uniform distribution: we have used this distribution for generating the speeds of the arriving cars on the highway and on the ramp. Even though there is an average speed for the cars most drivers in the real world go above or below this value due to various factors. However, during this project we did not want extremely fast drivers or extremely slow drivers. Hence, to ensure this variability without extremeness is modelled in our project we used the uniform distribution. The uniform distribution ensures that there is an equal chance of occurrence for the range of speed determined by our mean and standard deviation. Had we used normal distribution when generating the speed, we would have extreme drivers which may make our model complex.

Exponential distribution: we have used this distribution to implement arrivals of cars on the highway and on the ramp. The distribution allows us to capture the randomness and unpredictability of car arrivals in the real-world and model it in our project. The distribution ensures each car arrival is independent but also random

Random distribution: we used the random distribution to decide changing the destination of a car. In the real-world, the decision of changing the destination of a car and changing the lane based on the traffic flow is probabilistic as it is based on traffic conditions and driver behaviour. To model this probabilistic nature into our project and to make our project become a better reflection of the real world we used the random distribution.

**Limitations**

1.      The cars do not accelerate past the split since we had to assign them to the mean speed of the highway to avoid collisions.

2.      The car does not follow the 2-second rule when they reach the end of the lane and perform a sharp turn to merge with the traffic. Instead, they use a safe distance we determined was a better choice to ensure it is different from the late merge and so that they can merge into a smaller gap between the traffic.

3.      Performance challenges when simulating large-scale traffic scenarios with a high number of vehicles and complex road networks.

4.      When the car position is smaller than 790, if the car speed is smaller than 15 m/s, we accelerate it 2 times faster to reach the average speed of the highway. But it could not go beyond the mean speed. Also, for when it is larger than 15 m/s, we also keep accelerating but not beyond the mean speed. That is to restrict extreme speed values, but it is not very close to real life.

5.      When changing the lane or merging from the ramp, in certain instances the distance between the forward moving car and the merging car is very small making it look like they are crashing, upon closer look they are not.

6.   Since we are using thread and since we are new to it, we can only stop the simulation completely when we close all the graphs. When we close the animation the simulation still runs until we close the graphs. And this raises a problem with time. What we did was we ran the simulation for 3 mins (in real life) for both the merge, and we will analyze the statistics we have

**Verification**

The animation provides verification if our simulation is working properly and meets the requirements. It helps with movement of the cars. For eg; it helped us correctly calculate an angle for movement of cars in the ramp and during the merge.

**Calibration**

We calibrated positions for early merging, late merging and for breaking/taking a sharp turn for the cars in the acceleration lane so that we can clearly identify each merge. In addition, we used calibration during the early merge so that if the cars in the early merge position have enough distance from the cars queuing in the sudden merge then they could perform an early merge else they could join the queue. This ensures there are no disruptions to the queue order. The final calibration we used in our simulation is to calibrate an angle for the movement of cars on the ramp, when merging and when splitting to move to their destinations so that they move diagonally. This calibration especially when merging models real-world behaviour of cars as in the real world the cars move diagonally to ensure safety during merging.

**Output and Analysis**

Early Merge:

A screen shot of a computer

Description automatically generated

The above graph shows the average speed of traffic in each lane or the upstream car in the early merge with the color red representing lane A and blue; lane B.

A screen shot of a graph

Description automatically generated

The above graph shows the average arrivals speed of traffic in each lane or the upstream car in the late merge with the color red representing lane A and blue; lane B

A screenshot of a computer

Description automatically generated

Late Merge:

A screenshot of a computer

Description automatically generated

The above graph shows the average speed of cars for each lane of the upstream traffic with the color red representing lane A and blue; lane B.

A graph on a computer screen

Description automatically generated

The graph shows how crowded the traffic is on each lane: lane A, lane B, and on ramp. Red is lane A, blue is lane B, and green is ramp.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Conclusion:

Late merge is better than early merge. Because when you compare the statistics the mean time for late merge is smaller than the early merge even when the time running for late merge is large than the early merge simulation.