Intelligent Vehicles Final Project
Shockwave Traffic Jam

B03202027 Chih-Yu Hsu
R07922035 Yi-Ting Lee

January 3, 2019

Abstract

In this report, the causes of traffic jams in a single-lane circular road will be discussed. How the density and reaction time of drivers will affect the speed and flow of cars will be shown in figures. The performance of 3 kinds of cars: human drivers, autonomous cars, and connected cars will be compared as well. Finally, some methods to increase the speed and flow of cars will be mentioned.

1 Introduction

Traffic jams are a waste of time and unbearable in severe conditions. Sometimes it might be due to accidents or roadwork. However, traffic jams can form without these causes. One of the models to explain this kind of traffic jam is the shockwave model. Cars are described as fluid floating in tubes. When the speed of a car changes slightly, since cars are not coordinated, it causes the following cars to brake and finally forming a traffic jam. [1]

As a result, we would like to discuss four problems in our project by implementation and quantifying the vehicle efficiency:

1. Causes of traffic jam caused by human drivers
2. Comparison of human drivers and connected cars
3. How to improve the efficiency
4. Special cases: accident removal and shockwave traffic jam

In section 2, in order to improve the rationality of the experiment, we conclude the basic idea and important parameters of some paper related to traffic jam. In section 3, we list our implementation settings and experiment scenario. In section 4, there are several links of our demo video. In section 5, we display our results. And we have some conclusions in the end of the report.

2 Related work

Our shockwave traffic jam model mainly comes from reference [1]. They performed an experiment with 22 cars on a 230 meters circular road. In the beginning, each car is at roughly 30 km/hr separated with equal spacing. Then each car tries to maintain its speed while keeping the safe distance. Small fluctuations in speed is considered. In a while, there will be a cluster (traffic jam). The front car of the cluster will accelerate and leave the cluster, while cars behind the cluster will join the cluster.

The speed distribution of cars comes from reference [2]. It suggested that the speed distribution be given by the noise term in acceleration. For example, cars at 56 km/hr will have an acceleration noise of 0.03g, where g is gravitational acceleration. The maximum acceleration and deceleration comes from reference [3]. The maximum acceleration and deceleration is roughly 0.5g and 0.3g respectively.

Reaction time of our model comes from reference [4] and [5]. The perception and reaction time (PRT) is composed of detection, identification, decision and response time. It requires roughly 0.5 seconds for the detection process, and it is reasonable to estimate the whole PRT as 2.5 seconds. Details about the reaction time are found in reference [6]. Suppose a group of people with half males and half females, average age 45 years old ranging from 20 to 70 years old. According to the formula of ADRT (acceleration and deceleration response time) and BRT (braking response time), the average ADRT is 1.27 seconds with standard deviation 0.14 second, and the average of BRT is 0.65 seconds with standard deviation 0.01 second. Since detection requires 0.5 second, in our experiment the μ in reaction time for acceleration and deceleration is 1.77 and 1.15 respectively. It also mentions that drivers are expected to decelerate at 5-8.5 m/s^2 in an emergency situation. That is how we set the braking deceleration to avoid collision (when the following car is right after the leading car).

The idea of not-so-aggressive mode comes from reference [7]. It suggests that staying in the middle of the leading car and the following car prevents forming traffic jams. This is because it gives human drivers more time to react.

3 Implementation

In our implementation, we use Python3 and VPython7 to simulate and visualize the traffic jams and do statistics in different situations. We model our environment as an one-lane circular. Several cars are moving in the circular simultaneously, and there are three types of cars in our experiments:

1. human drivers
2. autonomous cars without connectivity
3. autonomous cars with connectivity

3.1 Definition

First of all, the definition of each term and the global settings are listed as followed:

Table 1: Definition of global settings

Variable Name	Definition	Value range
num_cars	Number of cars in the simulation	10-40
circular_length (m)	Total distance in the simulation	200-400
min_gap (m)	The minimum distance between any two cars at all time	5
gap_front (m)	The gap between each car and its leading car	
save_gap (m)	If gap_front is smaller than save_gap, the car will decide to decelerate	
dt (s)	Time of one iteration for simulation	0.002
initial_mean_speed (km/hr)		40-60
max_speed (km/hr)	Maximum speed of all cars	108
min_speed (km/hr)	Minimum speed of all cars	0
max_decel (m/s^2)	The max deceleration of each car	
max_accel (m/s^2)	The max acceleration of each car	
initial_speed (m/s)	Initial speed of each car	
brake_decel (m/s^2)	Deceleration to avoid collision	5-8.5

Also, some parameters are different due to different types of cars. The detailed explanation is from section 3.3 to 3.5.

Table 2: Differences among three types of cars

Variable Name	Definition	Human Drivers	Auto	Auto + Connected
decel (m/s^2)	Deceleration of each car	N(1.5, 1.5×0.125)	max_decel	max_decel
accel (m/s^2)	Acceleration of each car	decel × 0.5	max_accel	max_accel
max_idle_time (s)	Since the reaction times of human drivers have randomness, set a max idle time to avoid some unreasonable values	5	1	1
reaction_time_decel (s)	The time from the drivers/cars decide to decelerate to the drivers/cars start to decelerate	N(2, 2 × 0.125)	0.5	0.3
reaction_time_accel (s)	The time from the drivers/cars decide to accelerate to the drivers/cars start to accelerate	reaction_time_decel × 2	0.5	0.3

3.2 Implementation settings

3.2.1 Circular

The circular road of our model is implemented using a ring object, centered at (0, 0, 0). Cars can only move in the middle of the circular in counter-clockwise and cannot surpass each other. Each car has its own position expressed in degree θ_i, with i from 1 to num_cars. Thus, by defining circular_radius, we can set the position of each car as

$$position_i = circular\_radius \times (cos(\theta_i), sin(\theta_i), 0)$$

on the circular.

3.2.2 Cars

Cars are modeled as a "sphere" object with radius = 1.5 to simplify visualization. All cars have their own position, speed and acceleration, which are bounded by max_speed, min_speed, max_accel and max_decel. Also, they have their own acceleration/deceleration strategy, "compute_safe_gap" function and reaction time. We also add "color" on the cars to visualize its speed during the simulation. Cars with speed larger than 80 (km/hr) as labeled as "blue" cars, and cars with speed less than 20 (km/hr) as "red" cars. We also have "green", "yellow", and "orange" for middle situations. Finally, we color 'cars'white" when they are under "avoid collision" mode.

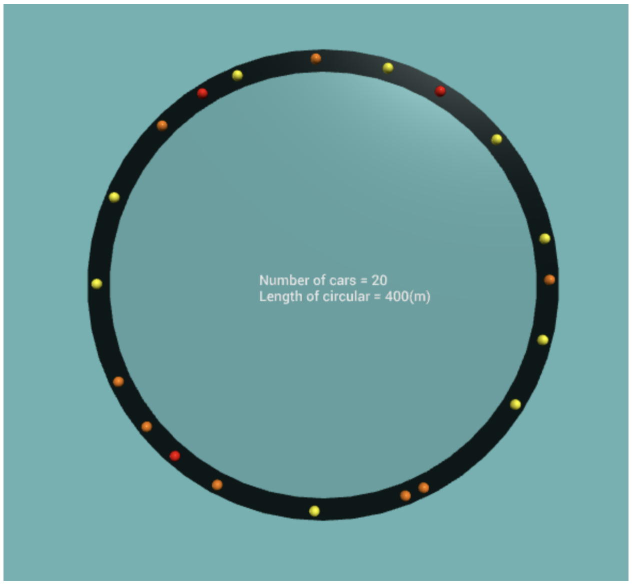

Figure 1: Visualization

3.3 Implementation details

3.3.1 Compute save gap

The "compute save gap" function for human drivers and autonomous cars are different.

- For human drivers,

$$save\_gap = max(2 \times self.speed, min\_gap)$$

- For autonomous cars (with and without connectivity),

$$save\_gap = 0.1 \times self.speed + \frac{self.speed^2}{2 \times self.decel} - \frac{front.speed^2}{2 \times front.decel} + min\_gap$$

All the cars only consider the gap between its leading vehicles.

3.3.2 Acceleration and Deceleration Strategies

There are two strategies in our implementation.

1. When the human drivers or autonomous cars detect that the gap_front is larger than save_gap, they will try to accelerate.
   When they detect the gap_front is smaller than save_gap, they will try to decelerate.

2. When the human drivers or autonomous cars detect that the gap_front is larger than save_gap $\times c_1$, they will try to accelerate.
   When they detect the gap_front is smaller than save_gap $\times c_2$, they will try to decelerate.
   Where c_1 is larger than 1 and c_2 is smaller than 1.

All accelerate and decelerate behavior will take place after reaction_time_accel and reaction_time_decel once the drivers or autonomous cars make the decisions.

3.3.3 Initialization

- Initial position: All cars are scattered throughout the circular and have a bit of randomness.
  $$init\_\theta_i = \frac{2 \times \pi}{num\_cars} \times (i + uniform(-0.05, 0.05))$$

- Initial speed:

$$init\_speed = Gaussian(\mu = initial\_mean\_speed, \sigma = initial\_mean\_speed \times 0.125)$$

- Initial acceleration = 0

3.3.4 Loop

For every iteration, each car follows these steps:

1. Compute save_gap based on its "compute_save_gap" function

2. Decide to accelerate or decelerate with their acceleration/deceleration strategy

3. Update its next_action_array by its reaction time

4. According to the first index of next_action_array to execute acceleration or deceleration

To model the differences between human cars and autonomous cars, all the cars will start to execute accelerate or decelerate motions after their respective reaction time. In general, human drivers have much longer reaction time than two autonomous cars w/o connectivity, and autonomous with connectivity cars has the shortest reaction time.

3.4 Next_action_array

In order to implement different reaction time, we have a next_action_array in our program.

The length of next_action_array in the first dimension is num_cars, and the length of the second dimension of next_action_array is (max_idle_time/dt + 1). The elements in the next_action_array is {-1, 0, 1}. "1" means acceleration, "-1" means deceleration and "0" means remaining the same speed.

In the beginning, all values in the next_action_array are set to zero. And in each iteration, we update next_action_array by three steps:

1. Once a car decide to take action to accelerate or decelerate, some elements (according to its reaction time) in its next_action_array will be set to 1 or -1, and flush all elements behind to zero.

2. The car will take action according to the first element of next_action_array.

3. In the end of the iteration, all the elements in the next_action_array are moved forward for one index.

In order to explain why we design next_action_array like this manner, let us consider a simple example:

- Assume that there are num_cars cars in our simulation, dt (time interval for each iteration) is 1 second, and the max_idle_time among all cars is 5 seconds. Thus, we initialize next_action_array with shape = (num_cars, 5/1+1) = (num_cars, 6) and all values are zeros.

- The reaction_time_accel of the i-th car is also 5 seconds and the reaction_time_decel is 3 seconds.

- At t = 0, suppose the i-th car detect that the leading cars ((i-1)-th) is far away and decide to accelerate. We update next_action_array[i][5] to 1, since reaction_time_accel/dt = 5. Thus,

$$next\_action\_array[i] = [0, 0, 0, 0, 0, 1]$$

  In the end of this iteration, move forward each element for one step, and

$$next\_action\_array[i] = [0, 0, 0, 0, 1, 0]$$

- At t = 1, suppose the i-th car detect that the leading car ((i-1)-th) is braking so the gap between the leading car becomes smaller. The i-th car decide to decelerate. We update next_action_array[i][3] to -1, since reaction_time_decel/dt = 3.

$$next\_action\_array[i] = [0, 0, 0, -1, 1, 0]$$

  At the same time, we need to flush all elements with index larger than 3. That is, we can simply do

$$next\_action\_array[i][3 + 1 :] = next\_action\_array[i][3 + 1 :] \times zero$$

  Thus,

$$next\_action\_array[i] = [0, 0, 0, -1, 0, 0]$$

  And in the end of this iteration,

$$next\_action\_array[i] = [0, 0, -1, 0, 0, 0]$$

- At t = 2, assume the i-th car decides to do nothing, and in the end,

$$next\_action\_array[i] = [0, -1, 0, 0, 0, 0]$$

- At t = 3, assume the i-th car decides to do nothing, and in the end,

$$next\_action\_array[i] = [-1, 0, 0, 0, 0, 0]$$

- At t = 4, assume the i-th car decides to do nothing, and

$$next\_action\_array[i] = [-1, 0, 0, 0, 0, 0]$$

  However, since the first index is "-1", the i-th car executes "real" deceleration. And in the end of this iteration,

$$next\_action\_array[i] = [0, 0, 0, 0, 0, 0]$$

4 Simulation Demo

Some videos were uploaded to YouTube with the following links.

1. Circular and 1 car
   https://youtu.be/XpP19UtTy0A
2. 20 human drivers
   https://youtu.be/4WN80Z-9mrw
3. 20 autonomous + connected cars
   https://youtu.be/1IhYrUssqpA

5 Results

The following experiments are performed over 200 seconds for 5 times (seed(0)-seed(4)) in each cases. The average speed of cars is calculated for every 2 seconds.

5.1 Human drivers: Density

The common settings are as follows: Maximum speed = 108 km/hr. Initial speed = random.gauss(μ = 40 km/hr, σ = 0.75). Braking deceleration = random.gauss(μ = 6.74 m/s, σ = 0.58).

5.1.1 Setting A

- Cars are distributed over a 230 meters circular road.
- The maximum acceleration and deceleration are = 0.3g and 0.5g respectively.
- The distribution of acceleration and deceleration are random.gauss(μ = 0.3g/2, σ = 0.03g) and random.gauss(μ = 0.5g/2, σ = 0.03g).
- The reaction time for acceleration and deceleration are random.gauss(μ = 1.77, σ = 0.14) and random.gauss(μ = 1.15, σ = 0.01) respectively.

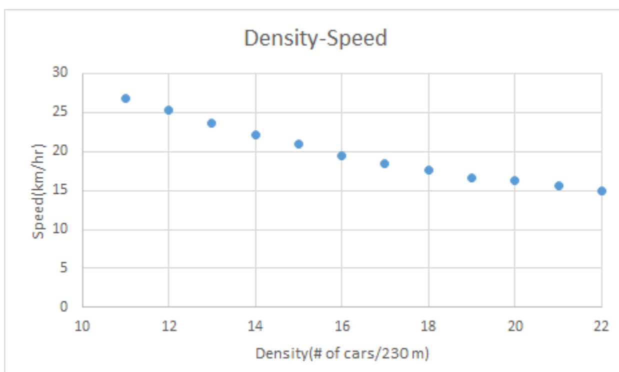

Figure 2: Setting A: density

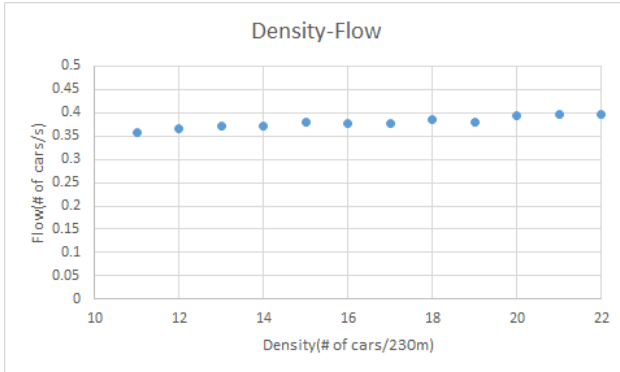

Figure 3: Setting A: flow

5.1.2 Setting B

- Cars are distributde over a 400 meters circular road.
- The maximum acceleration and deceleration are = 0.3 and 3 m/s^2 respectively.
- The distribution of deceleration is random.gauss($\mu = 1.5$, $\sigma = 0.1875$) while the acceleration of each car is half of its deceleration.
- The reaction time for deceleration is random.gauss($\mu = 2$, $\sigma = 0.25$) while the time for acceleration is half of the time for acceleration.

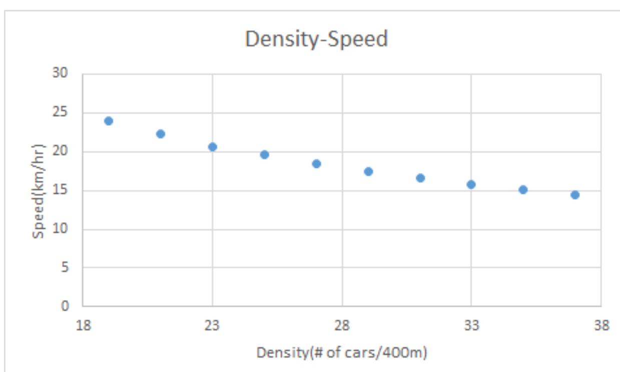

Figure 4: Setting B: density

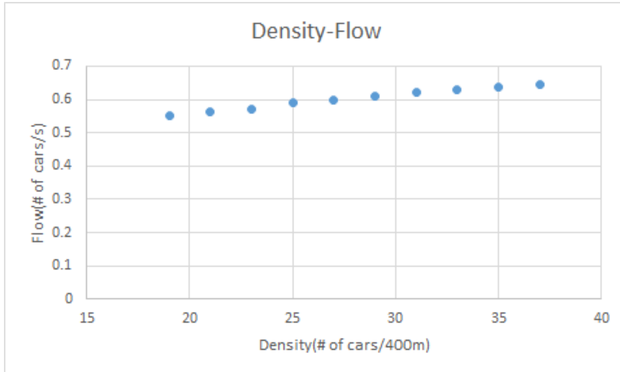

Figure 5: Density B: flow

Both settings shows that as the density of cars increases, cars goes slower, which makes sense. However, the flow of cars stays roughly the same in setting A, while the flow in setting B increases slightly. Since the upper bound of acceleration and deceleration in setting B is smaller, the result may imply that a less aggressive setting can improve the flow of cars.

5.2 Human drivers: Reaction time

The reaction time ranges from 0.5 to 2.5 seconds. Jammed degree is defined as the number of cars below 10 km/hr.

5.2.1 Setting A

- 11 cars are distributed over 230 meters circular road.
- Reaction time is from 0.5 to 2.5 seconds without randomization. Reaction time for acceleration and deceleration are the same.
- The rest setting is the same as the setting A mentioned above.

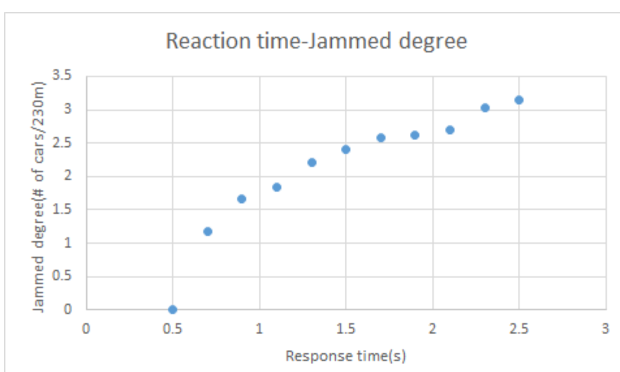

Figure 6: Setting A: reaction time

5.2.2 Setting B

- 20 cars are distributed over 400 meters circular road.
- Reaction time is from 0.5 to 2.5 seconds without randomization. Reaction time for acceleration and deceleration are the same.
- The rest setting is the same as the setting B mentioned above.

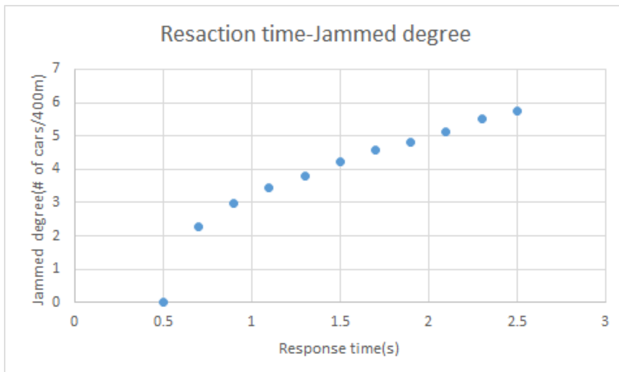

Figure 7: Setting B: reaction time

In both settings, as the reaction time increases, cars are jammed in a more severe degree, which makes sense. As the figure shows, as the reaction time goes down to 0.5 second, no car is jammed. However, since the number of jammed cars is calculated every 2 seconds, we cannot make sure if there is no jammed condition indeed. Besides, according to the paper reviews above, it is quite hard for human drivers to respond in 0.5 second.

5.3 Autonomous v.s. Connected cars

20-40 cars are distributed over 400 m circular road for autonomous car and connected cars respectively. The reaction time for autonomous car is 0.5 second, while for connected cars it is 0.3 second. For automatic cars, the reaction time is the time difference between the moment the car senses other cars and the moment it starts to accelerate(or decelerate). For connected cars, the reaction time is the time difference between the moment the car receives information from other cars and the moment it starts to accelerate(or decelerate). We think the reaction time for human drivers, automatic, and connected cars are as follows: human driver ¿ automatic ¿ connected cars. That is why we set 0.5 and 0.3 second in our experiment.

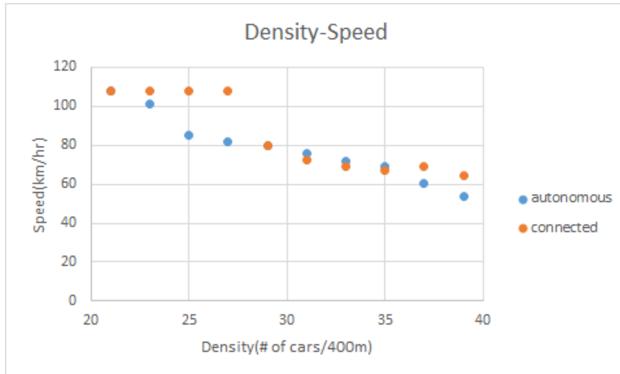

Figure 8: Autonomous v.s. Connected: density

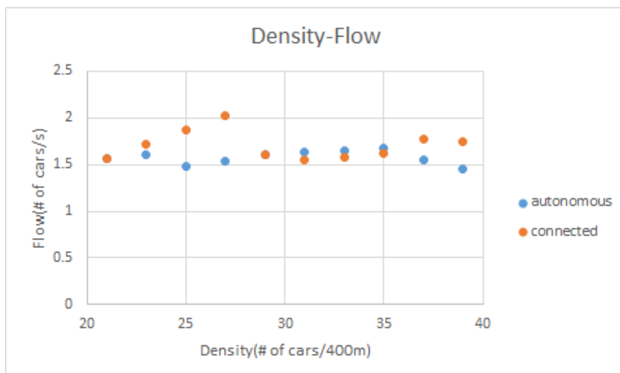

Figure 9: Autonomous v.s. Connected: flow

Since the maximum speed is set to 108 km/hr, when the density of cars is low enough, connected cars can drive at the highest speed. Both autonomous cars and connected cars show that as density increases, the average speed of all cars is decreased. As the figure shows, when the density is low or high enough, connected cars are faster than autonomous cars. This makes sense because connected cars have shorter respond time in our model. However, the speed of connected cars is a bit lower than autonomous cars when there are 29 to 35 cars. This may imply that connected cars might work better when the density is either low or high enough.

The flow of connected cars increases in low density since they can drive at the highest speed. When density is higher than 29 cars/400 meters, flow is increased as well, which is the same as setting B of human drivers. For autonomous cars, flow does not increases steadily like connected cars. This might imply that cars with a shorter reaction time can perform better as density increases.

If we set the upper limit of connected cars to 1000 km/hr, we find that connected cars can reach a speed of roughly 800 km/hr when there are only 2 cars on the circular road. We are not sure if this is practical in reality since the reaction time of connected cars are only 0.3 seconds.

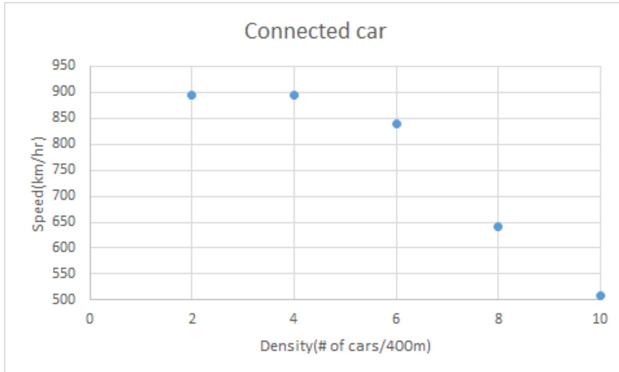

Figure 10: Connected without speed limit

5.4 Penetration rate

Penetration rate is defined as the ratio of connected cars to all cars (human drivers and connected cars).

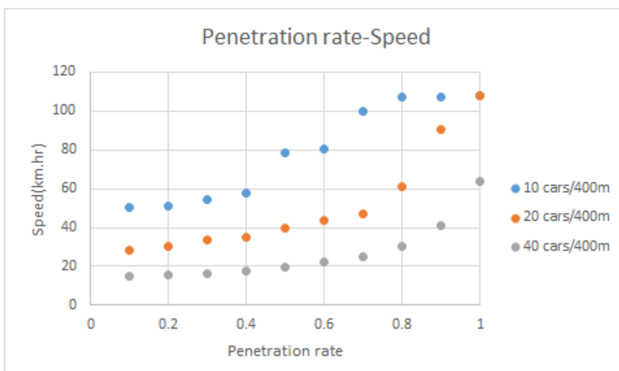

Figure 11: Penetration rate: density

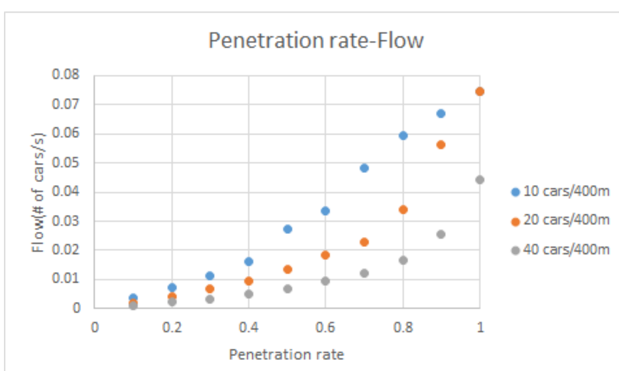

Figure 12: Penetration rate: flow

When the density is low, cars can drive at the maximum speed if connected ratio is high. Also, connected ratio results in a jump in speed. It might show that when the density is low, human drivers is a bottleneck to the overall speed. However, we do not have to care much about it since the flow shows a steady increase. When the density is higher, penetration rate shows a significant improvement in speed (beyond linearity). The flow also shows a steady increase as penetration ratio increases.

5.5 Condition simulation

5.5.1 Accident removal

The experiments below are performed over 200 seconds for 5 times (seed(0)-seed(4)) in each cases. In the beginning, 10 cars are still on 1/10 of 4000 meters of circular road with equal spacing. Then the first car drive at 100 km/hr to leave all the others behind. We measure the time for all cars to reach 60 km/hr under 3 conditions: all human drivers, all autonomous cars, and all connected cars.

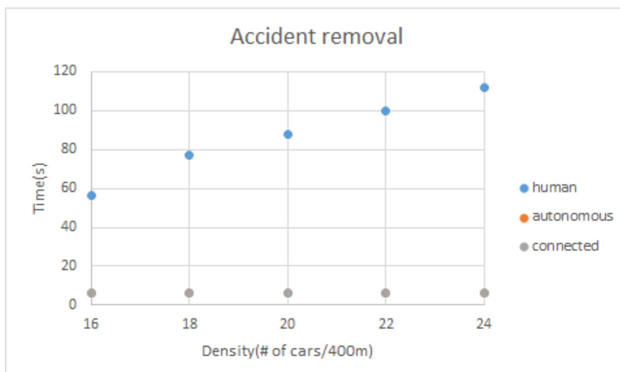

Figure 13: Accident removal

For human drivers, as the density increases, it requires more time for the following cars to regain their speed. For autonomous and connected cars, the increase in time is much less compared to human drivers as density increases. (Autonomous cars and connected cars have roughly the same speed, so the orange dots and gray dots are overlapped and not easy to see.)

5.6 Less aggressive condition

5.6.1 Human driver

In former sections, cars are in either accelerating or decelerating mode. Here we add an constant velocity mode, which makes cars not so aggressive. Under this condition, a long cluster of traffic jam forms. Suppose the constant speed mode is activated when the gap between cars is at α to β times of safe distance, and 22 human driver cars moving on a 230 meters circular road using setting B, then there is a 16-car-long cluster moving backwards in a speed of 2.4 cars per second. The animation can be seen at https://www.youtube.com/watch?v=d0R20Ki5pTY&feature=youtu.be.

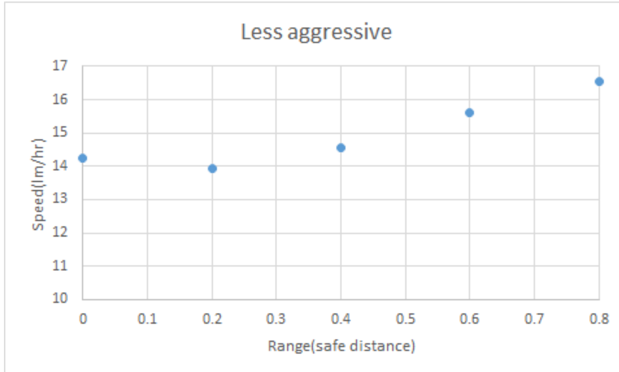

Figure 14: Human driver

Cars are in constant mode when its gap is α to β times of its safe distance, where the range in x-axis is $\beta - \alpha$. The result shows that when the constant speed mode is adopted in a larger range, the average speed is higher.

5.6.2 Autonomous car

For autonomous car, each car is forced to drive in the middle of its leading car and its following car.

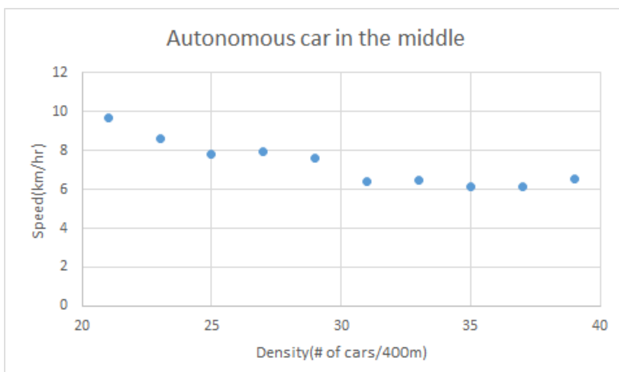

Figure 15: Autonomous car

Compared to the graph in Autonomous v.s. Connected car section, autonomous cars are not faster when they stay in the middle. The idea of staying in the middle is to help drivers have sufficient time to react, but the reaction time of autonomous cars is already low enough. Therefore this strategy does not improve the performance of autonomous cars.

6 Conclusions

In this experiment, causes of traffic jams are discussed and different kinds of cars are compared with each other. The causes of traffic jam of human drivers can be due to car density and reaction time. Connected cars have better performance than human drivers, so as penetration rate increases, performance is better. To improve efficiency, the non aggressive approach works for human drivers but not for autonomous cars. In accident removal condition, connected cars and autonomous car have better performance. The shockwave traffic jam is observed in our model as well. The cluster forms in the video provided above.

References

[1] Y. Sugiyama *Traffic jams without bottlenecksexperimental evidence for the physical mechanism of the formation of a jam*

[2] R. Kuhne, P. Michalopoulos *Continuum flow models*

[3] R. Herman *Traffic dynamics: analysis in stability of car following*

[4] James L. Pline *Traffic Control Devices Handbook, Institute of Transportation Engineers, 2001*

[5] Nicholas J. Garber, Lester A. Hoel, *Traffic and highway engineering, 2nde., 1997*

[6] A. Mehmood *Modeling reaction time in car following behaviour based on human factors*

[7] https://www.youtube.com/watch?v=iHzzSao6ypE&fbclid=IwAR3w6zdqjV8eMg5jJKvUEdyqIqLNry0yAL-U3rsvZYHc1CcJHYdzlEpga4Q