National College of Ireland

Project Submission Sheet – 2020/2021

Student Name:	Omkar Ratnoji Tawade		
Student ID:	19232136		
Programme:	MSc. Data Analytics (MSCDAD_A)	**Year:**	2020-21
Module:	Modelling Simulation and Optimization		
Lecturer:	Rejwanul Haque		
Submission Due Date:	16-05-2021		
Project Title:	CA		
Word Count:	3940 words		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Omkar Tawade
Date:	16-05-2021

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Modeling, Simulation and Optimization

CA – 60%

Omkar Ratnoji Tawade
Data Analytics
National College of Ireland
Dublin, Ireland
x19232136@student.ncirl.ie

***Abstract*— This document is for the purpose of CA Project for the Modeling, Simulation and Optimization module for the National College of Ireland, MSc in Data Analytics. The goal of this project to create a 3 km segment of motorway that reduces after 2km from 3 lanes to 2 lanes and based on this lane structure we must answer three questions such as optimal average travelling time in free flowing (t_{opt}), optimum throughput when average travelling time is 20% longer than t_{opt}, maximum throughput when average travelling time is twice the optimal average travelling time (t_{opt}). We will have to use fully electrical vehicles with different acceleration and deacceleration factors and automated cars or partially automated cars in our project. This project will help us to understand the behavior of different cars on motorway using average travelling time and optimal throughput.**

Keywords—simulation, average traveling time, optimal throughput, vehicle, motorway

I. Introduction

In this project we will be a creating a full simulation model right from generating single lane or multiple lanes to adding a single vehicle or multiple vehicles on motorway. This project is built to make traffic system more efficient and effective. Before proceeding with the simulation, we need to understand the real traffic system and how to drive on the motorway and certain rules. So, first rule is that the speed limit on motorway is 120km/hr. Certain categories of vehicles are not allowed on motorway which are no learner drivers, vehicles under 50 cc vehicle with speed limit equal to 50 km/hr, invalid carriages, pedal-cycles, pedestrians, and animals. Vehicles are not allowed to reverse on motorway and vehicles are not allowed to stop and park on motorways unless in case of emergency. As vehicle is prepared to join the motorway it must make sure that it gives way to traffic on the motorway and use slip lane as long as possible because this allows to traffic behind to see for longer and it also allows vehicle speed upwards before vehicle join the left lane.

On motorway the primary lane is a left lane and right lane is used for overtaking or allowing traffic to merge. The minimum distance between vehicles should be maintained. Drivers need to constantly check the vehicle on the back and front. Driving on the motorway requires lot of concentration. It requires a lot of regular mirror checks and it requires blind spot in which driver moves eyes across shoulder to see the side way traffic. This is the basic structure of motorway and rules of vehicle which should be followed while running on the motorway. In our project we will try to simulate these real scenarios of motorway and will try to add vehicles and model this vehicle such that the data will look like that vehicle have been operated by humans and in next case we will add automated vehicles.

Figure 1. Motorway Road

There can be congestion on the road if too many vehicles are driving on the road and due to this congestion, there can be accidents on the road. But what are the reasons behind the congestion? We will discuss few of the following reasons which leads to congestion on the motorway. Braking causes the high chance of congestion on the motorways. If a driver on a motorway applied a brake for 2 or 3 second, then the vehicles behind those will also apply the brake and this causes the chain reaction and this will ultimately reduce the distance between the cars and result of this chain reaction is a traffic. Second scenario that leads to congestion is changing of lane by vehicles unexpectedly which is considered as bad driving can also cause accidents on the motorways. There are manual ways to regulate this traffic efficiently. Braking should be avoided on the motorways, but this doesn't mean that you should not apply brake in an emergency. On motorways you have enough distance between the vehicles so that can deaccelerate the vehicle without using brakes by just not using the accelerator. There also other effects of braking which will relate to drivers, like if drivers are applying brakes more often then will reduces the fuel economy of vehicle as well as it reduces the exhaust emission. Rubber necking which means taking eyes off the road for more than 3 seconds can make driver to apply the brakes immediately if another vehicle comes near to this vehicle for 3 seconds. Our project will consider the braking scenario and will also alert the crash situation it happens on the motorway model. We can adjust few parameters on the motorway model to optimize the flow of traffic on the motorway.

II. Literature Review

Bin Yu et.al [1] analyzed the traffic simulation of running speed in a connected vehicles environment. They discussed about the connected vehicle first, in which they explained how connected vehicle exchange information like speed, acceleration, deacceleration and location between each other. They developed this model on VISSIM simulation in which they designed a two-lane road of 400m with maximum velocity equals to 70km/hr. Further they add road segment division. They collected the speed of vehicles of first 400m of road and observed the data of speed, acceleration and deacceleration of vehicles. After 400m, they observed the change in these values obviously due to lane change. They applied average speed optimization after 400m segment and recorded the behavior of vehicles. In this simulation researchers not allowed for lane change function in first segment of 400m. They found that the speed of each vehicle was close to maximum value after few iterations and the speed of all vehicles at last 2 seconds are equal. Due to use of average speed optimization after 400m vehicle maintained safe distance between them but the speed of vehicle was reduced, and it was not near to the maximum value. Researcher also implemented the mixed traffic flow in normal and connected states. They observed the fluctuation in average running speed and delay when connected vehicles rate was increased and this helped them to conclude that connected vehicle rate have significant impact on the results and found that 0.6 was the optimal rate of connected vehicle to improve the running and delay time.

Miguel C. Figueiredo et. al [2] proposed an approach to simulate autonomous vehicles in urban traffic scenarios. In this paper, they studied about the art of driverless cars simulations. In their proposed research their simulator is supported with many other simulators such as pedestrian simulator, driver agent, infrastructure agent. They used the 3D simulation viewer to view the simulation which is connected to proposed simulator. The proposed simulator is connected to MAS-T2er Lab's microscopic traffic simulator and to the pedestrian simulator. The proposed simulator will get idea about the traffic from these simulators and proposed simulator can also send the data back to these simulators so that they can use this data to calculate the result of their simulation. Various agents are also connected to the proposed simulator and characteristics of these agents are extracted such as driver agent will give the characteristic of car and infrastructure agent will give characteristic of traffic lights. The proposed simulator has collision detector which constantly tracks the vehicle-to-vehicle movements. Agents interface with the simulator will help user to directly control a vehicle within the simulator. Real vehicle reaction to virtual sensor responses, it is the ability of vehicle to avoid the virtual obstacles. Obstacle agent is used to connect to the simulator, to give information about the broken vehicle or emergency traffic sign. The proposed simulator connection to the external microscopic traffic simulator gives the ability to the simulator to load road network map from external traffic simulator or geographic information system (GIS) database.

Andreas Tapani [3] presented a traffic simulation model for rural roads and driver assistance systems. In this paper, researcher considered microscopic traffic simulation of rural roads and used traffic simulation for evaluating driver assistance system. They used rural traffic simulator to simulate traffic on tow lane rural road and on rural roads with separated oncoming traffic lanes. The delay in travel time on rural road are more due to interaction between the vehicles and infrastructure but the travel time on rural are prominently affected by the road geometry. Further they discussed about the driver assistance system. This system helps for driving routine. ADAS is used to build the fully automated roads. Further they discussed about the examples of ADAS like adaptive cruise control, intelligent speed adaptation and lane departure warning to driver vigilance monitoring, pre-crash vehicle realistic separation and parking aid. ADAS is developed for human machine interfaces. Adaptive cruise control can control the acceleration and deacceleration of the vehicles. They have also used RutSim model which is time base stochastic model which is used to simulate the lanes. It also found to be produced traffic properties with separated traffic. They also considered the simulation for different region, country, social or economic conditions. They observed that increased modeling detail can create a need for different modeling assumptions regarding driver behavior for various applications. They also included the overtaking assistant. This assistant is again modeled in RuT simulator. This assistant assisted the driver in the judgement of overtaking scenario based on the time gap between the vehicles.

Ding Ding [3] proposed a method for modeling and simulation of highway traffic using a cellular automation approach. They discussed about the tree traffic models which are microscopic traffic model, macroscopic traffic model, and mesoscopic traffic model. Further they analyzed the traffic flow and used cellular automata model. In microscopic model the car following model describes the relation between vehicle location and velocities. This model helps to determine how vehicles follow one another by keeping specific time and distance between them. In macroscopic model, the traffic flow is determined by using fluid dynamic differential equation. In mesoscopic traffic model, it considers vehicle as gas particles so simulation using gas kinetic dynamic is easily fit in mesoscopic traffic flow. Researchers proposed cellular automation model and categorized it as microscopic traffic models. In this model, a road is constructed like a checkboard and time is also discredited. The algorithm of this model was that each vehicle will move at velocity v. If there is a vehicle at car space ahead then they increased the velocity of vehicle by 1 and if there are car ahead then the velocity of vehicle is decreased by 1. In this model, the street is divided into cells and each cell is almost occupied by one vehicle. They used this cell to increase or decrease the velocity of the vehicles. If vehicle is moving forward to a cell and if the cell is not occupied by any vehicle, then the current vehicle can increase the velocity by 1. To simulate the realistic behavior, they proposed an algorithm in which at first step all the vehicle whose speed did not reached v_{max} can increase their speed by 1. In second step, it will check the space ahead of the vehicle and will decide whether to increase and decrease the speed and in last step it determines the new position of the vehicle.

III. Methodology

Our program creates the simulation of real-life traffic on the road as well as it will analyze the behavior of automated vehicles on the motorway. There are many functions in the programs such as lane creation, vehicles, surround, and recorder. We will understand the flow of the program sequentially.

A. Lane:

Lane class is created and always the lane id is initialized to zero. Lane function accepts the two parameters. First parameter is the length of the motorway and second parameter is maximum velocity of the vehicles on that lane. It also the list of vehicles. Further four variables were initialized which are previous lane, next lane, lane to the left and lane to the right. We can program to build a single lane as well as multi lane. Linked list method is used to create such lanes as it will help to join the lanes with the help of ids. In multiple lanes these ids are helpful. If we create a 3-lane motorway then center lane will have the left lane id and right lane id. Left lane will be appended with center lane id also right lane will be appended with left lane.

```
LANE_ID = 0
VMAX = 120/3.6
c = Lane(1000, VMAX)
while c.totalLength()<2000:
    c.extend(Lane(1000, VMAX))
l = c.widenLeft()
r = c.widenRight()
print("Left Lane:  ", l)
print("Centre Lane:", c)
print("Right Lane: ", r)
```

```
Left Lane:   [2 1000m R:0]-[3 1000m R:1]
Centre Lane: [0 1000m L:2 R:4]-[1 1000m L:3 R:5]
Right Lane:  [4 1000m L:0]-[5 1000m L:1]
```

Figure 2. Two Lane Motorway Simulation

There are many functions inside the lanes class. Following are the list of functions

1. getLane: This will help to know which lane it is depending upon the direction. If direction is slow then lane will be always left and if direction is equal to fast, then lane will be right.
2. attachLeft: This function takes the input current lane and attached a lane left to the current lane.
3. attachRight: This function takes the input current lane and attached a lane right to the current lane.
4. widenRight: This function help to add a lane segment to the right of the current lane of equal length. It also considers the maximum velocity of current lane to lane which will be widen to its right.
5. widenleft: This function help to add a lane segment to the left of the current lane of equal length. It also considers the maximum velocity of current lane to lane which will be widen to its left.
6. Extend: This function will extend the lane to given kilometer or meter road segment.
7. totalLength: This function will calculate the total length of lane segment.
8. enter/leave: This function helps vehicle to enter or exit a lane using vehicle and lane function.

B. Vehicles:

Vehicle class have more functions compared to the lane class also the parameters in the vehicle functions are more compared to the lane class. Starting lane, starting position, time, position, velocity, change in velocity, which is acceleration, change is acceleration are the parameters which are accepted by the vehicle function. Acceleration and Deacceleration have the limits to simulate the real-life scenario in the project. Following are the functions used in the vehicle class.

1. updateOnly: This is the main function which helps move the vehicle forward as well as it help to enter the lane and leave the lane.
2. update: This function help to use the surround function inorder to overtake vehicle by changing or moving to the right lane and then coming back at left lane.
3. setTarget: It take the array of change of time and velocity at that time.
4. Process: This function helps to start braking when two cars are very near to each other also it helps to change the lane explicitly
5. emergencyBraking: It estimate the time for braking for autonomous cars and for manual driving cars it uses the change in time or use random element.
6. adjustVelocity: This function smoothly adjust velocity over time and helps to increase or decrease the velocity at give time depending upon the threshold of velocity is passed for given interval.
7. crash: when car comes close to each other crash function returns the vehicle id, position, and time.

```
VMAX = 120/3.6
env = simpy.Environment()
rec = SimpleRecorder(env, 0, 5, 1)
l = Lane(500, VMAX)
r = Lane(500, VMAX)
l.attachRight(r)
v = Vehicle(env, rec, startingLane=l, dx0=20, t=[1, 3], v=[20, 'R'])
#v.traceSurround=True
v.traceOvertake=True
rec.run()
print("Left Lane: ", l)
print("Right Lane:", r)
```

```
t=    5.0s Overtaking v0 returns to slow lane at x=  106.0m
Left Lane:  [0 500m R:1 (0)]
Right Lane: [1 500m L:0 (0)]
```

```
rec.getData()
```

	t	x	v	a	id	lane	oldLane	pos	event
0	0	0	20	0	0	0	None	0	enter lane
1	0	0	20	0	0	0	None	0	timer
2	1	20	20	0	0	0	None	20	timer
3	2	40	20	0	0	0	None	40	timer
4	2	40	20	0	0	0	None	40	change fast
5	2	40	20	0	0	1	0	40	enter lane
6	3	60.5	21	1	0	1	0	60.5	timer
7	4	82	22	1	0	1	0	82	leave lane
8	4	82	22	1	0	1	None	82	done change fast
9	4	82	22	0	0	1	None	82	timer
10	5	103.55	21	-1.11111	0	1	None	103.55	timer
11	5	103.55	21	-1.11111	0	1	None	103.55	change slow
12	5	103.55	21	-1.11111	0	0	1	103.55	enter lane

Figure 3. Vehicle on motorway

IV. THE SIMULATION MODEL

We will now implement the model to observe the throughput, optimal average time and density. We will start with the lane creation first. We have created a two-lane model which is 3km in length. We have initialized a left lane for 3000km and a right lane for 3000km and we have used “attachRight” function to create a two-lane road.

```
l = Lane(3000, VMAX)
r = Lane(3000, VMAX)
l.attachRight(r)
```

Figure 4. Created Two Lane

We will be running simulations by changing the number of cars and by changing the interarrival time. For interarrival time we are using random expo-variate function to generate the interarrival times using the inter arrival time initialized earlier. So, we will be changing the initial inter arrival time which will ultimately change the inter arrival times array. The random expo-variate function is used to simulate the huma behavior as it will create an array of time interval by using random number which are exponential. Further number of time interval and speed interval is controlled by cycle variable which will be a random integer between 4 to 8. Cycle variable will decide the number of intervals in time and speed array. Speed array is calculated using random normal variate function which simulates the human behavior in which speed is calculated using random normal variate function in which speed is randomly generated but these values are normally distributed. We have divided our simulation in two parts in first half of simulation we will be using random normal variate function to generate the speed and in later half of the simulations we will use free-motorway speed generation function. Normally vehicles will run on the left lane but in case of overtaking the vehicle will change its lane to right lane which is also known as fast lane. In vehicle function we have passed the starting lane, initial time, initial speed, time array and velocity array. We will iterating this function to the number of cars in each simulation, so in our first care we have iterated it over 5 times. We will be recording the simulations by changing the number of cars. We have recorded all necessary parameter in a csv file which we will use to interpret our results. We have changed the number of cars from 5 to 10 to 20 to 40 to 80 to 100 in each simulation by keeping the inter arrival time constant which is used to calculate the interarrival time array. In further simulations we changed the inter arrival time to 15 basically we are reducing the interarrival time and respective we are again increasing the number of vehicles on the road from 5 to 100 in same manner. By changing the number of car and number of inter arrival time value we almost computed and recorded the 20 simulations, and we will interpret the results in the further section. We have used the data frame to store the simulation result. Now using this data frame, we will be filtered out the necessary variables which are required to calculate the throughput, density, speed and optimal time average. Speed is calculating by averaging out the speed of each vehicle. Throughput is calculated based on difference of time at which first car leave the lane or ends and time at which last car leave the lane or ends. Optimal time is calculating by averaging out the time taken by each vehicle to cover the 3km road.

```
df['v'].mean()
```

```
30.462772179732323
```

Figure 5. Average speed on the road.

```
df[(df.id == 0) & (df.event=='leave lane')]
```

	t	x	v	a	id	lane	oldLane	pos	event
832	109.0	3012.96	27.6465	-0.0137599	0	0	None	3012.96	leave lane

Figure 6. First Car Leave Time

```
df[(df.id == 4) & (df.event=='end')]
```

	t	x	v	a	id	lane	oldLane	pos	event
845	110.0	3011.07	34.4959	-0.0196867	4	None	None	3011.07	end

Figure 7. Last Car Leave Time

Since our first simulation was for 5 cars so we had passed the condition id equals to 4. Otherwise depending upon the number of cars we will need to change this value.

```
import numpy as np
id_ = df['id'].unique()
time_ = np.zeros(shape=(len(id_)))
for i in range (0,len(id_)):
    df1 = pd.DataFrame(df[(df.id == i) & (df.event=='enter lane') & (df.pos==0) ])
    time1_ = df1['t'].to_numpy()
    df2 = pd.DataFrame(df[(df.id == i) & (df.event=='end')])
    time2_ = df2['t'].to_numpy()
    time_[i] = np.subtract(time2_, time1_)
print(sum(time_)/len(time_))
```

```
99.33013433525706
```

Figure 8. Average Time

We have recorded all these values at each simulation and recorded in the csv file. We also performed simulation for different vehicles. We have considered four types of vehicles which are family car, electric car, truck and bus. We have created added car length object in the class vehicle and also added threshold of speed difference required to overtake. For family car the length of car is equals to 4 and for truck we have kept the length of vehicle equals to 9. We have used different vehicles in the same simulation and recorded the parameters which will be required to calculate throughput and density. We have used freeway motorway speed to generate the speed array and for interarrival time we have used random uniform function in this simulation.

V. RESULTS AND INTERPRETATION

Now we will observe the results of our simulations. We took normal vehicles and introduced the human behavior by using the speed function using random expatriate function. We iterated by increasing the number of vehicles and by decreasing the inter arrival time.

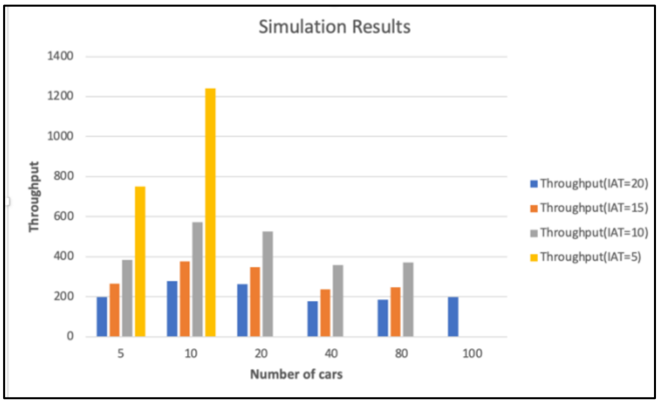

Figure 9. Simulation Result

Figure 9 is a plot between the throughput and number of cars at different inter arrival times we will try to identify the relationship between these two parameters.

We can see that when the inter arrival time is high, throughput is very low. Also, at same inter arrival time, when IAT=20 we can see that throughput increases at very slow rate when number of cars increased but again it decreased when the number of cars were too large which suggested free flowing traffic even though number of cars were more.

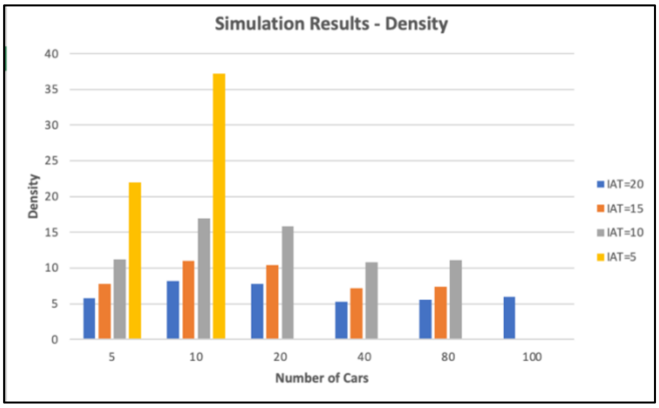

Figure 10. Density vs Number of Cars across different IAT

Figure 10 gives the clear representation of traffic on the road. If we observe the general trend of the traffic, we can see that when the number of vehicles is increased and inter arrival time is decreased then more traffic is observed. As we can see when number vehicles are 10 and IAT=5 then there is a maximum traffic present on the road and further we can observe that yellow bar are absent in the graph for number of vehicles equals to 20, 40, 80 and 100 which is due to the crashes of vehicles occurs at that interval. We have not recorded the density for the iteration where crashes occurred. Now we will observe the result of throughput and density when we used freemotorway speed function to create the speed array. We have followed the same steps which we had followed in previous method.

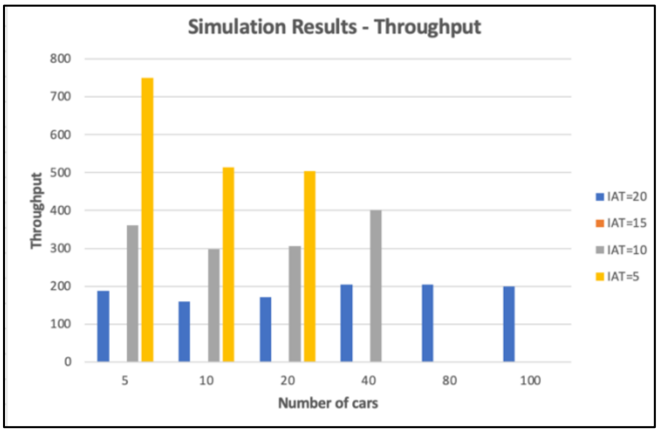

Figure 11. Throughput vs Number of Cars using freeway motorway speed

Figure 11 suggest that when we use freeway motorway speed for the vehicles, we are observing high throughput on the road when the number of vehicles are low and interarrival is also low. This is the opposite scenario of previous case.

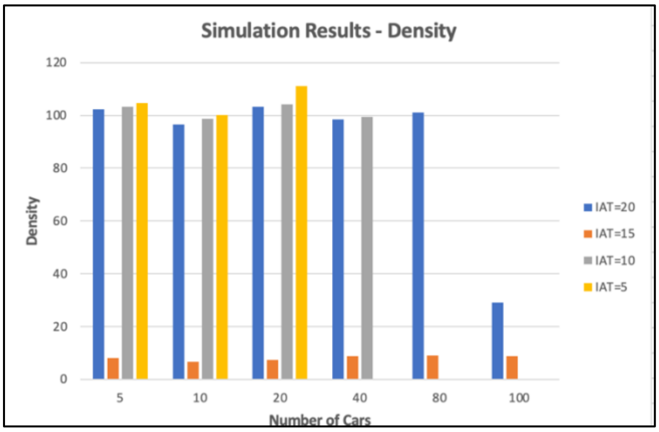

Figure 12. Density vs Number of Car using Freeway Motorway Speed

Figure 12 suggest that interarrival time equals to 15 is best for any number of vehicles as the density on the road at that interval is very low compared to other intervals.

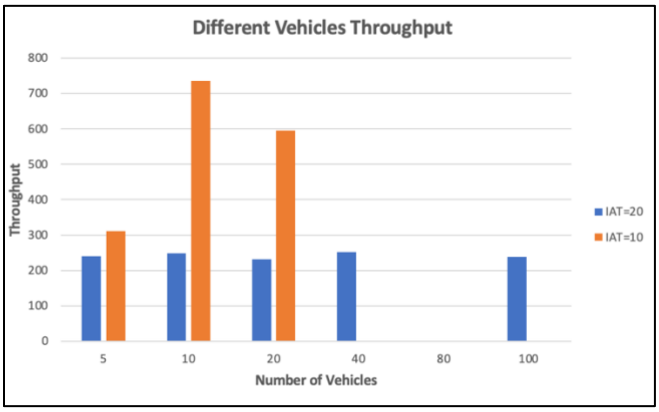

Figure 13. Different Vehicles Throughput

Figure 13 suggest that throughput is increasing at greater rate when compared to the throughput of similar vehicles. Also we can observe that crash scenario is occurring early when compared to one type of vehicle simulation result.

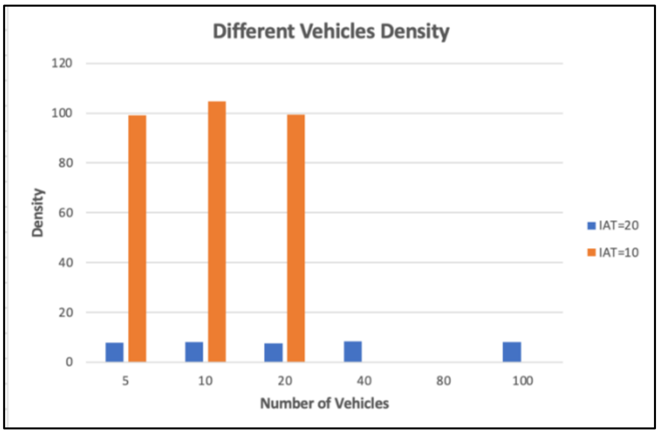

Figure 14. Different Vehicles Density

Figure 14 suggest that when inter arrival time is more than then density of traffic is less. Also when number of vehicles are less then crashing scenario too are less

VI. Conclusion and Future Work

We simulated over 40 simulations using two speed function and based on the result of these simulations we can conclude that model is performing excellent when number of cars are less and interarrival time is more, but model performance is degraded when number of cars are increased and interarrival time is decreased. In future work, we can improve the model performance in this area as well as we can introduce the traffic simulation inside this simulation as well as we can consider the motorway lane for city and rural roads.

References

[1] B. Yu, M. Wu, S. Wang, and W. Zhou, "Traffic Simulation Analysis on Running Speed in a Connected Vehicles Environment," *International Journal of Environmental Research and Public Health*, vol. 16, no. 22, p. 4373, Nov. 2019.

[2] Figueiredo, Miguel & Rossetti, Rosaldo & Braga, Rodrigo & Reis, Luís. An Approach to Simulate Autonomous Vehicles in Urban Traffic Scenarios. 1 - 6. 10.1109/ITSC.2009.5309524. (2009)

[3] Tapani, A.. Traffic simulation modelling of rural roads and driver assistance systems. (2008).

[4] Ding Ding. Modeling and simulation of highway traffic using a cellular automaton approach. (2011)