Check Your Gap or Get Scrapped: An Investigation of a Car Following Model

Kaeshav Danesh and Kevin Phan

May 3, 2022

Abstract

Traffic flow dynamics is generally split into either macroscopic models or microscopic models. For this paper, we will focus on car following models which is a microscopic model. We first introduce the assumptions and the mathematical description car following model and the numerical scheme used to compute them. Then, we give the Full Velocity difference Model. Finally, we examine different scenarios such as bottlenecks and lane changes and analyze the behavior of cars.

Contents

1 Introduction

How does an obstacle affect a series of vehicles? What impact does changing a road from multiple lanes to one lane have on a series of cars? To answer these questions, our paper will describe a basic microscopic car following model from traffic flow theory, implement these scenarios, and analyze them.

Traffic flow models can be categorized as microscopic or macroscopic models [[van+15]]. Microscopic models describe vehicles on the individual level by treating them as an entire unit. Meanwhile, macroscopic models treat vehicles as a continuum. Further classifications can be done through the model equations such as partial differential equations, discrete equations, discrete variables, or continuous variables. Some applications of traffic flow models include navigation systems, traffic light calibration, and calculating carbon emissions from traffic.

Our paper will focus only on the microscopic model called the Full Velocity Difference model. Using this model, we will explore scenarios including obstacles, and lane changes. From this, we will analyze state variables including position and velocity, and other metrics such as density and flow rate.

2 General Model

2.1 Mathematical Formulation

We follow the mathematical formulation of car following model in §10.2 of [TK13]. Suppose that there are n vehicles in the simulation using the car following model. We index the 1st vehicle by 1, the 2nd car by 2, the αth car by α, and so on. The state variables of vehicle α are position x_α and velocity v_α. Furthermore, we are also assuming that the vehicles in the model have a length l_α. The position x_α of car α is defined as the front bumper of the car. Another useful variable to define is gap. We define the gap of car α by the difference in distance between the back bumper of car $\alpha - 1$ and the front bumper of car α. Mathematically, the gap s_α is defined as

$$s_\alpha = x_{\alpha-1} - l_{\alpha-1} - x_\alpha \tag{1}$$

where $x_{\alpha-1}$ is the position of car $\alpha - 1$, $l_{\alpha-1}$ is the length of car $\alpha - 1$, and x_α is the position of car α. We note that the gap is not defined for a vehicle with no vehicles in front of it.

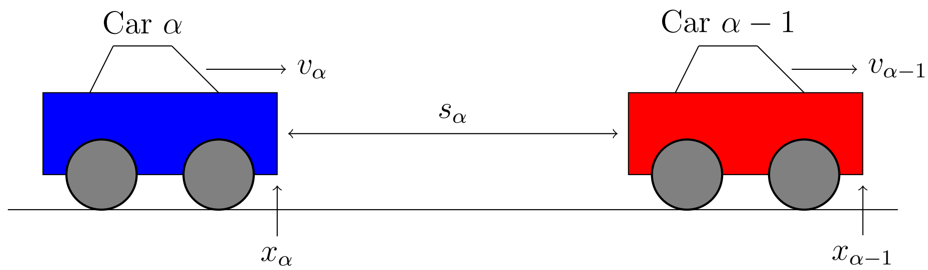

Figure 1: Defining index, position, velocity, and gap of a car.

For simpler notation, we will now refer to the vehicle in front of vehicle α by the leader vehicle l. For a single lane, car l is vehicle $\alpha - 1$. However, it is not necessarily true that vehicle l is vehicle $\alpha - 1$ for multiple lanes.

Taking the time derivatives of $x_\alpha(t)$ and $v_\alpha(t)$ lead to the general coupled differential equation describing velocity and acceleration respectively:

$$\frac{\mathrm{d}x_\alpha(t)}{\mathrm{d}t} = v_\alpha(t), \tag{2}$$

$$\frac{\mathrm{d}v_\alpha(t)}{\mathrm{d}t} = a_{\mathrm{mic}}(s_\alpha, v_\alpha, v_l). \tag{3}$$

Each car following model has a specific acceleration function: a_{mic}[1]. For the simulation, we will use the Full Velocity Difference Model (FVDM) which is described in section (3.1).

2.2 Numerical Scheme

The standard way of numerically solving a system of coupled differential equations would be to use a fourth order Runge-Kutta method. However, higher order methods all assume higher orders of smoothness in the differential equation and its solution [Ste11]. Our model is only smooth to first order because sudden changes in acceleration are common, for example, during lane changes. So, using a high order method could be worse than a simple first order method.

Among first order methods, our options are either the forward or backwards (implicit) Euler methods. The main difference between these methods

[1] We use a_{mic} for the acceleration function and a to describe the acceleration of a car.

is that the backwards Euler method involves solving an implicit equation and is generally more stable.

We are using the forward Euler method because we expect our solution to be stable and solving an implicit equation every timestep would add extra complexity to our implementation.

Implementing the forwards Euler method scheme for a car following model gives us two coupled differential equations for each of our state variables:

$$v_\alpha(t + \Delta t) = v_\alpha(t) + a_{\text{mic}}(s_\alpha(t), v_\alpha(t), v_l(t))\Delta t, \tag{4}$$

$$x_\alpha(t + \Delta t) = x_\alpha(t) + \frac{v_\alpha(t) + v_\alpha(t + \Delta t)}{2}\Delta t, \tag{5}$$

where Δt is the time step and a_{mic} is the acceleration function defined by the car following model used [TK13]. This acceleration function is described in section (3.1).

When solving the equations above numerically, we must also consider the interactions between cars. The velocity equation uses $v_l(t)$, the velocity of the leading car at time t. So, when calculating the state of a car at each timestep, it would be simpler to start from the backmost car and work up. This allows us to use the velocity value at time t from the leading car.

3 Car Following Model

3.1 Full Velocity Difference Model

We followed the details of the Full Velocity Difference Model in §10.6 and §10.7 of [TK13]. The Full Velocity Difference Model (FVDM) is given by the acceleration function:

$$a_{\text{mic}}(s_\alpha, v_\alpha, v_l) = \frac{v_{\text{opt}}(s_\alpha) - v_\alpha}{\tau} - \gamma \Delta v \tag{6}$$

where v_{opt} is the optimal velocity function, τ is the speed adaptation time, γ is the speed difference sensitivity, and $\Delta v = v_\alpha - v_l$ is the difference of velocities between car α and the leading car.

A reasonable choice for v_{opt} is

$$v_{\text{opt}}(s) = \max\left(0, \min\left(v_0, \frac{s - s_0}{T}\right)\right) \tag{7}$$

where v_0 is the desired speed, s_0 is the minimum distance gap, and T is the time gap.

There are properties that an optimal velocity function should have if it were to model traffic flow. These properties are

$$\begin{aligned} v'_{\text{opt}} &\geq 0, \\ v_0 &= 0, \\ \lim_{s\to\infty} v_{\text{opt}}(s) &= v_0. \end{aligned}$$

Typical parameters for highway traffic for the FVDM is given in the table below.

Parameter	Value
v_0, desired speed	$33.\bar{3}$ m/s
s_0, minimum distance gap	3 m
T, time gap	1.4 s
τ, speed adaptation time	5 s
γ, speed difference sensitivity	0.6 s^{-1}

Figure 2: Typical parameters for highway traffic for the FVDM. From page 170 in [TK13].

We now examine how different parameters of the optimal velocity function affect the graph.

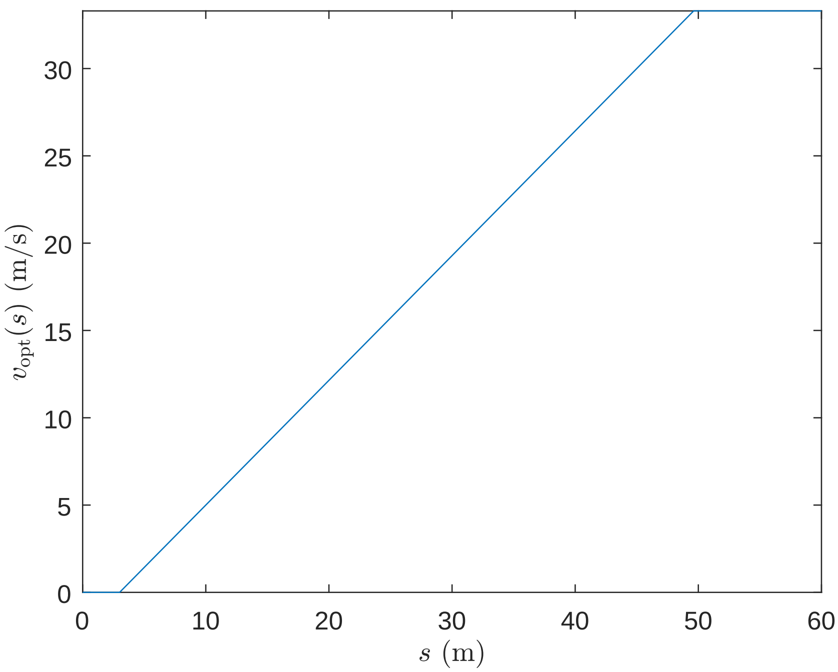

Figure 3: Graph of the optimal velocity function over a range of gaps

Since we are taking the minimum of v_0 and $(s - s_0)/T$, v_{opt} attains a maximum of v_0. Hence, the optimal velocity of a vehicle is the desired speed v_0 for large gaps. Furthermore, v_{opt} is 0 on the interval $0 \leq s \leq s_0$ which means that a vehicle will not move if the vehicle's gap is s_0 or less. Lastly, the time gap T determine the slope of the line. Higher values of T means that the car's desired speed will be reached for higher values of s.

Analyzing equation (6), $v_{\mathrm{opt}}(s) - v_\alpha$ is positive if $v_{\mathrm{opt}}(s) > v_\alpha$. So, the car has not reached its optimal velocity and acceleration is positive. Similarly, if $v_{\mathrm{opt}}(s) < v_\alpha$, $v_{\mathrm{opt}}(s) - v_\alpha$ is negative, hence the car is decelerating to reach its optimal velocity. If $v_{\mathrm{opt}}(s) = v_\alpha$, then $a_{\mathrm{mic}} = 0$. The adaptation time τ determines how fast the car accelerates or decelerates, affecting the time it takes for the car to reach its optimal velocity. The term $-\gamma \Delta v$ is positive if $v_\alpha < v_l$ and negative if $v_\alpha > v_l$. This leads to more acceleration if the car is trying to catch up to the leader car and less acceleration if the car is trying to slow down due to the leader car's slower velocity.

3.2 Implementation

We first initialize initial conditions for the state variables of the vehicles. We store the vehicles in an array and iterate through them. As described in section (2.2), we iterate through the vehicles starting with the last vehicle with respect to position and go through in ascending order. We update the state variables of the cars using equations (4), (5), and (6). We repeat this process for the number of iterations required.

Algorithm 1 Simplified algorithm for FDVM

Require: Initial state variables for each car at $t = 0$.
Require: carArr, an array of cars.
  for $i = 1$:numsteps **do**
    for $j =$ length(carArr):-1:1 **do**
      State variables of jth car $\leftarrow$ Update jth car by a timestep.
    end for
  end for

We still have not addressed how to update the state variables of the first car. Equation (6) needs values for the car's gap, s_α, and the velocity of the leader car, v_l. However, the first car does not have a leader car. To resolve this, we impose a destination for the cars. We use this value in the calculation of gap s_α by computing $x_\alpha - x_{\text{destination}}$. For the situation of the velocity of the leader car, it might be reasonable to set it to set $v_l = 0$. However, if $v_l = 0$, then the term $-\gamma\Delta v$ blows up. In other words, this car is too wary of crashing into the destination. To resolve this problem, it is best to set $v_l = v_\alpha$ which ensure that the term $-\gamma\Delta v$ is 0. However, we note that this will cause a small problem in our simulation. We will address this in section (4.1.2).

4 Examining Different Scenarios

4.1 Homogeneous Traffic

This is a simple demonstration of our model where a line of cars travel on a single lane. We use the same parameters in Figure (2). Furthermore, we set the length of the cars to be 5 meters. At $t = 0$, the initial conditions are $x_\alpha = 200 - \frac{200}{9}(a-1)$, $v_\alpha = 0$, and $a_\alpha = 0$ for $1 \leq \alpha \leq 10$.

TODO: Add $x_{\text{destination}}$. Remove $a_\alpha = 0$.

4.1.1 Data

We present graphs of position, velocity, acceleration, and gap. These graphs will be used to compare the graphs we get from later sections.

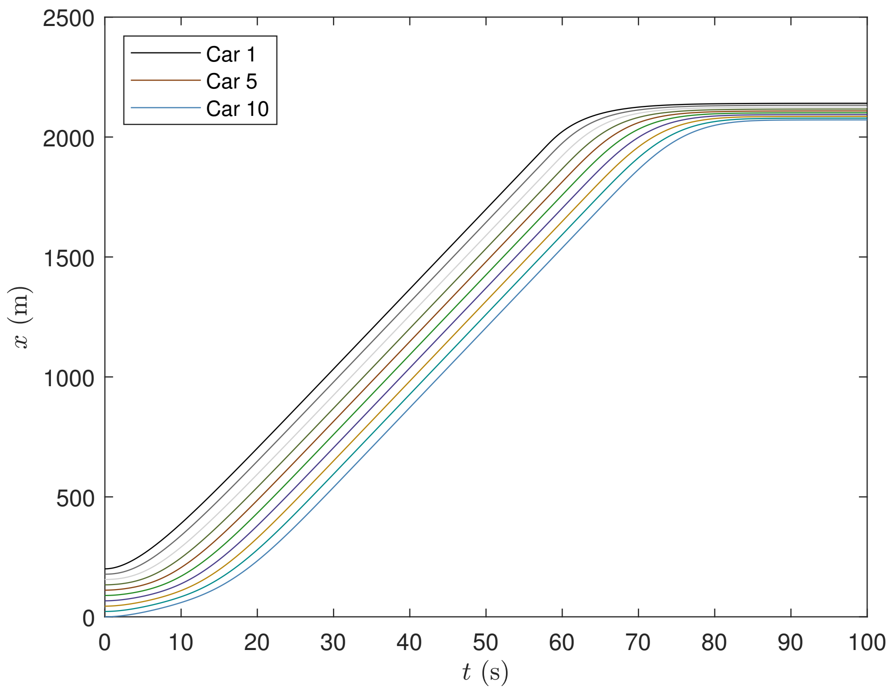

Figure 4: Position versus time graph.

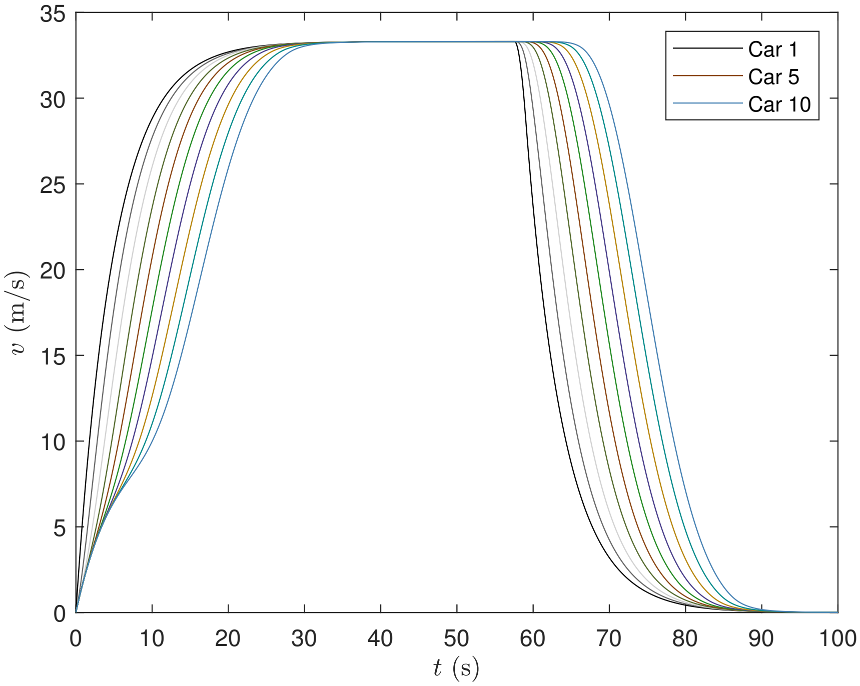

Figure 5: Velocity versus time graph

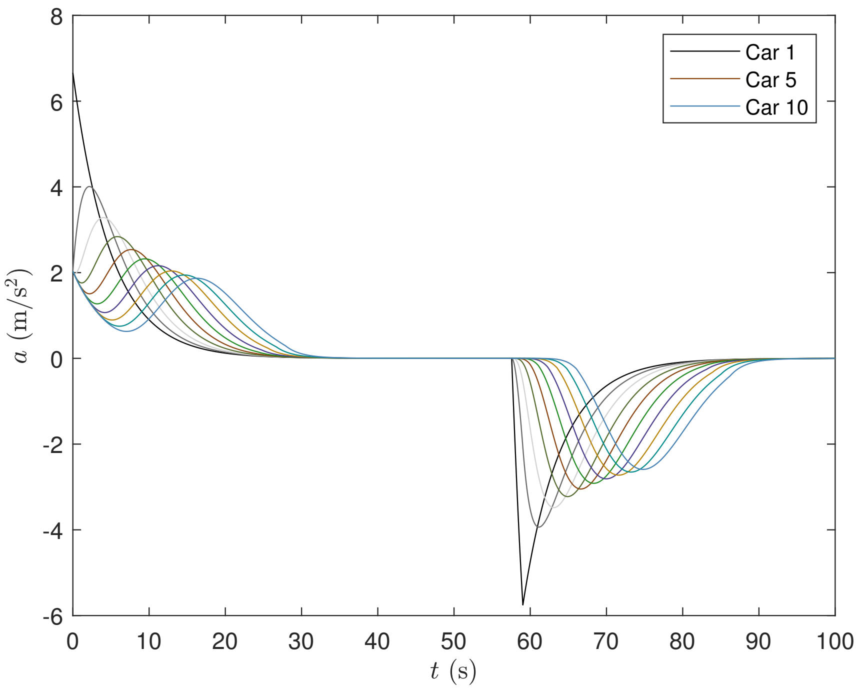

Figure 6: Acceleration versus time graph

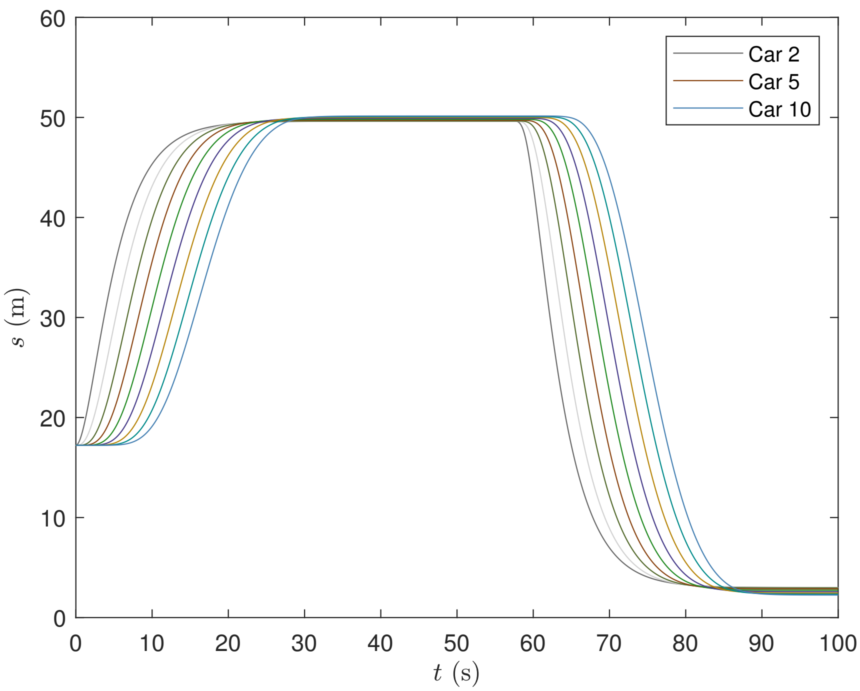

Figure 7: Gap versus time graph

For the gap versus time graph, we do not plot the gap of car 1 since the gap is the difference between the car position x_1 and x_{position} which is not of interest.

4.1.2 Pitfalls

There is one pitfall with how we update the first car's acceleration. In section 3.2, we choose $v_l = v_\alpha$ and $s_\alpha = x_{\text{destination}} - x_\alpha$ where $\alpha = 1$. The problem with this solution is that it simulates a car that is permanently stuck at $x_{\text{destination}}$ but with the same velocity as the first car. When the first car approaches $x_{\text{destination}}$, the model predicts that it is safe to move past $x_{\text{destination}}$ since the simulated car has non-zero velocity and so, it will move out of the way. This does not happen, so the car move past $x_{\text{destination}}$. Due to how v_{opt} is computed, $v_{\text{opt}} = 0$ once the first car is past $x_{\text{destination}}$. In other words, the first car does eventually slow down and stop. It is better to state that $x_{\text{destination}}$ is when the first car will begin to slow down and eventually stop after some distance.

Also, observe that acceleration is sometime unrealistic especially for the first car. At $t = 0$, the first car's acceleration is 6.66 $\mathrm{m/s^2}$ and the first car's deceleration is 5.7525 $\mathrm{m/s^2}$. The unrealistic acceleration occurred at the start of the simulation as the first car attempts to reach the optimal velocity as fast as it can. When the first car goes past $x_{\text{destination}}$, the car immediately decelerates to slow down. This results in unrealistic accelerations which manifests itself in simulations involving bottlenecks and lane changes.

4.2 Obstacle

4.2.1 Implementation

In this section, we will implement an obstacle that obstructs a group of cars traveling along a one-lane road. After some period of time, we will remove the obstacle which allow the cars to move again. This simulates a variety of scenarios such as railroad crossings, debris blocking a road, or even a chicken crossing the road.

We follow the same implementation as in section (4.1) for the cars and road. To implement an obstacle, we create a virtual stationary vehicle with position $x = 1200$, velocity $v = 0$, and acceleration $a = 0$. We never update this vehicle and so, the vehicle is always stationary. We initialize this vehicle at $t = 30$ and remove it at $t = 75$.

4.2.2 Data

We present multiple graphs of position, velocity, acceleration, and gap. We also present graph about density and flow rate. Density is computed as

$$\rho(x, t_0) = \frac{\Delta N}{\Delta x} \tag{8}$$

where ΔN is the number of cars in the interval $[x, x + \Delta x]$ at a time t_0. This measures how densely packed the car are. Flow rate is computed as

$$Q(x_0, t) = \frac{\Delta N}{\Delta t} \tag{9}$$

where ΔN is the number of cars that passed the position x_0 in the interval $[t, t + \Delta t]$. This measures how many cars are passing though x_0 in the time from t to $t + \Delta t$.

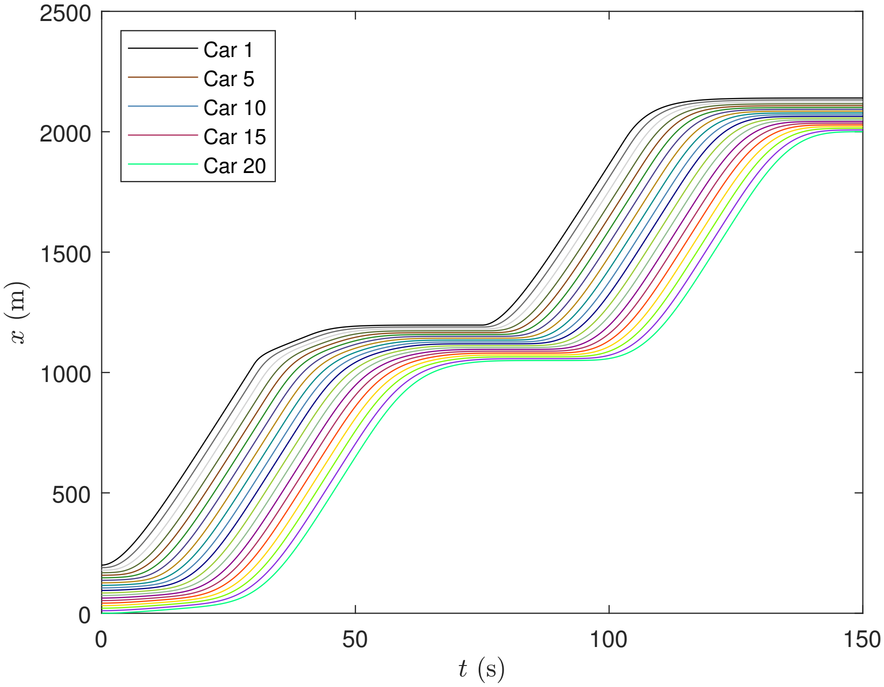

Figure 8: Position versus time graph.

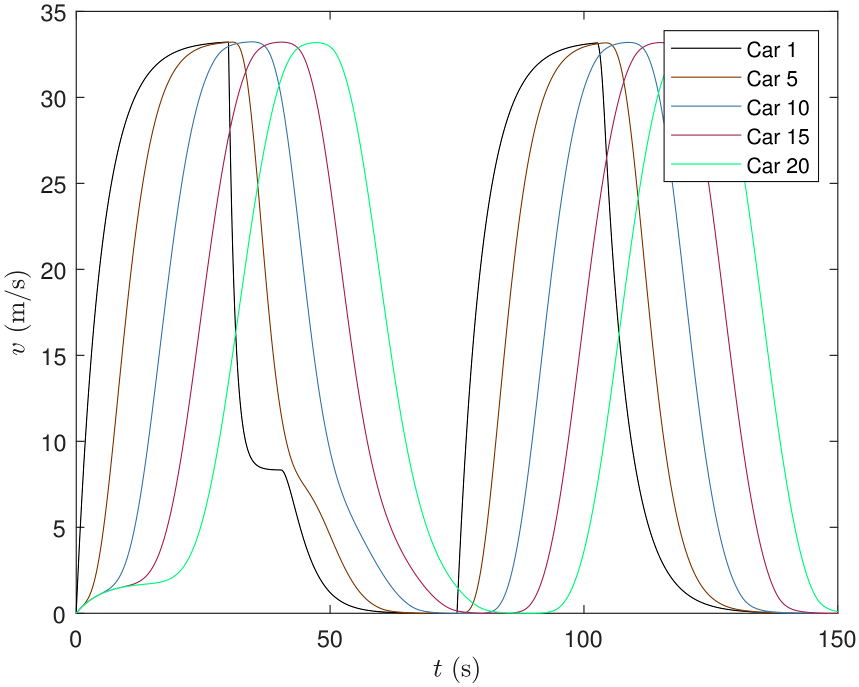

Figure 9: Velocity versus time graph.

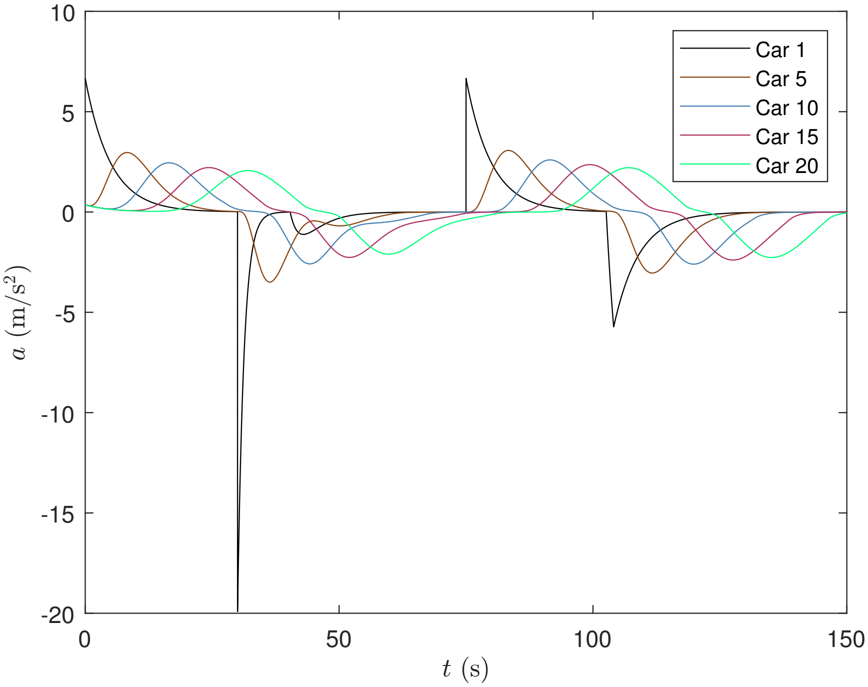

Figure 10: Acceleration versus time graph.

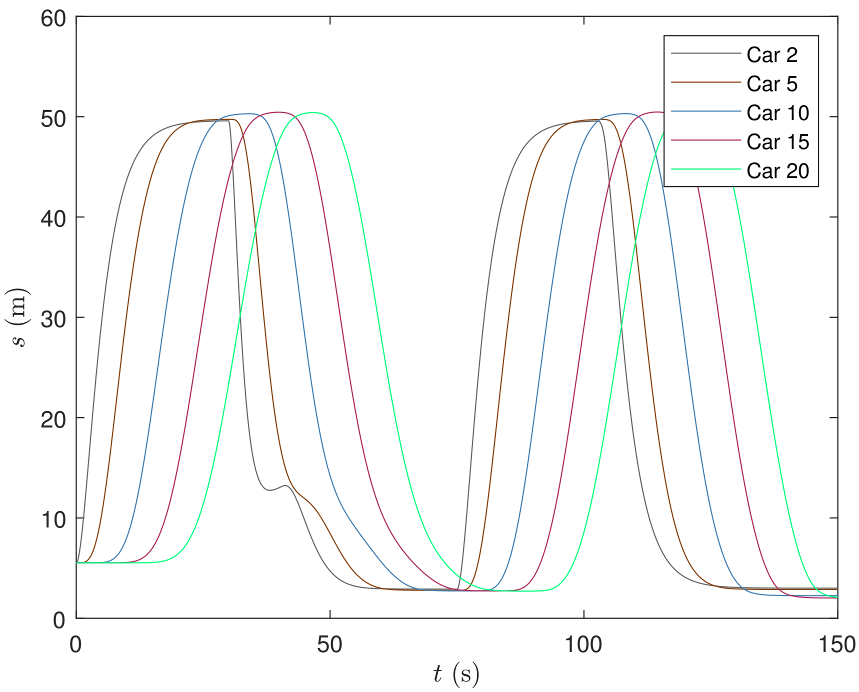

Figure 11: Gap versus time graph.

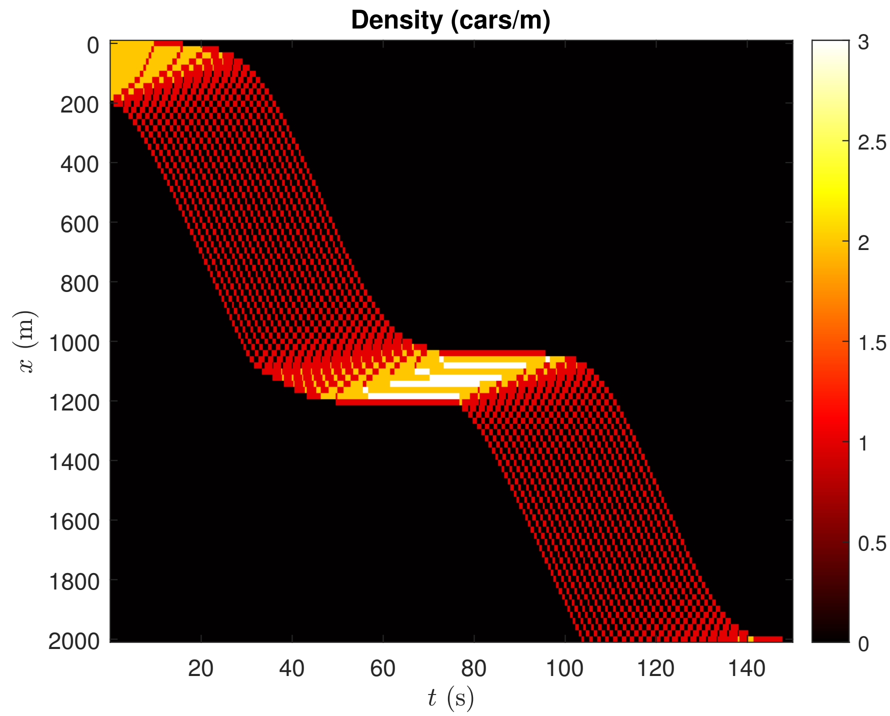

Figure 12: Density versus time (s) and position (m).

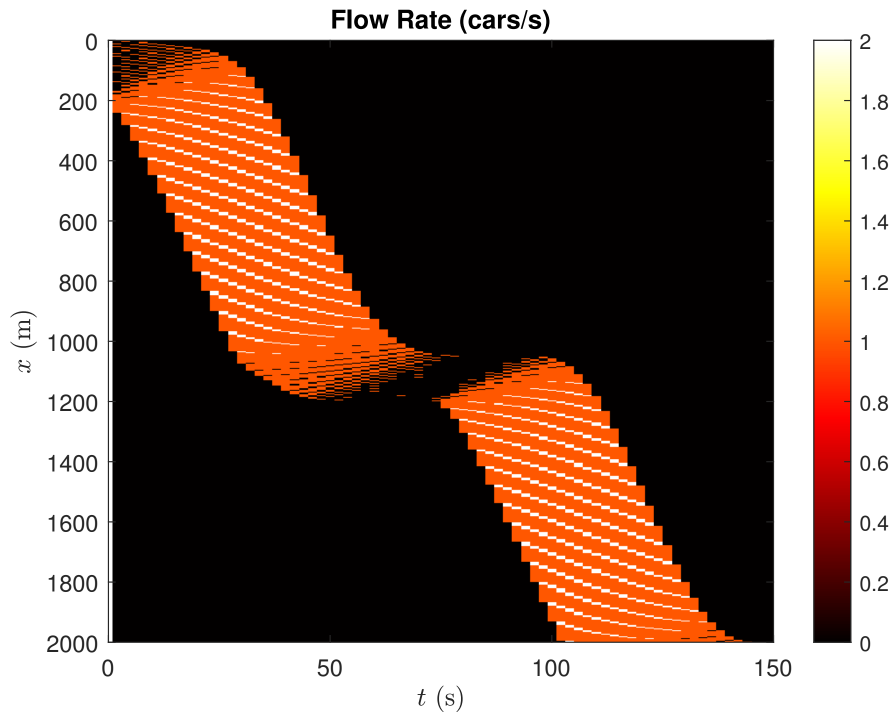

Figure 13: Flow Rate time (s) and position (m).

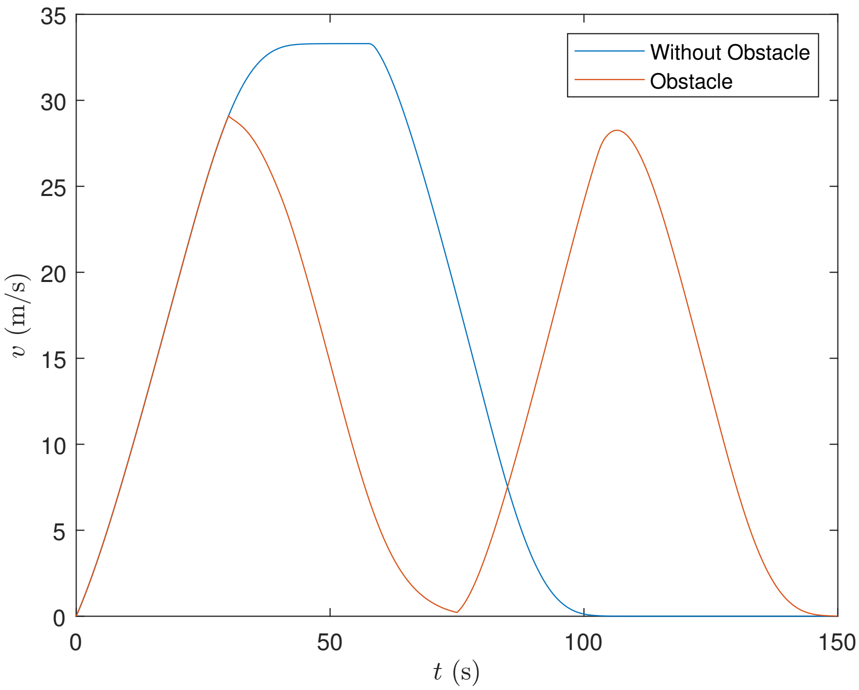

Figure 14: Mean velocity of traffic with and without an obstacle.

4.2.3 Analysis

We first examine the position, density, and flow rate of the cars. From figure (8), at $t = 30$, the first car begins to slow down due to the obstacle. Subsequent cars begin to slow down. Around $t = 50$, the first car's position is $x \approx 1192$. This results in the density of the cars increasing to around 3 cars in a 20 m interval as shown in figure (12). Furthermore, since the cars are unable to move due to the obstacle, the flow rate is 0 as shown in figure (13). The mean velocity of all the cars is the same for traffic with and without the obstacle till roughly $t = 30$. Looking at figure (14), the maximum mean velocity is 33.3 m/s for homogeneous traffic. Meanwhile, the maximum mean velocity is 29.0946 m/s for the traffic with the obstacle. The cars never achieve a maximum mean velocity of 33.3 m/s because as cars approach their maximum velocity, the first car's position is $x = x_{\text{destination}}$. So, the first car begins to slow down. This results in the mean velocity decreasing to 0.

We now examine each car's acceleration. From figure (10), the first car decelerates at $t = 30$ which is when the obstacle first appears. The other cars gradually decelerate following this. Roughly 6 seconds later, the fifth car reaches its maximum deceleration of -3.4932 m/s^2. Roughly 8 seconds later, the tenth car reaches its maximum deceleration of -2.5861 m/s^2. Roughly 8 seconds later, the fifteenth car reaches its maximum deceleration of -2.2633 m/s^2. Roughly 8 seconds later, the last car reaches its maximum deceleration of -2.0998 m/s^2. This phenomena is known as a stop-and-go wave [TK13]. The effects of the first car stopping is not felt for the last car until much later. In this case, it took the wave to travel 30 seconds to travel to the last car. Furthermore, the intensity of the braking lessens as the wave propagates from the first car to the last car. This suggests that given a sufficient number of cars and sufficient gaps between the cars, the wave would eventually die out.

4.3 Lane Changes

4.3.1 Implementation

Lane changes are dealt with on a microscopic level, meaning that at we decided whether or not to change lanes at every timestep for every vehicle. In our model, we will only consider changing to adjacent lanes in a single time step. So, at time t, vehicle number α has three choices: go left, go right, or stay in the same lane.

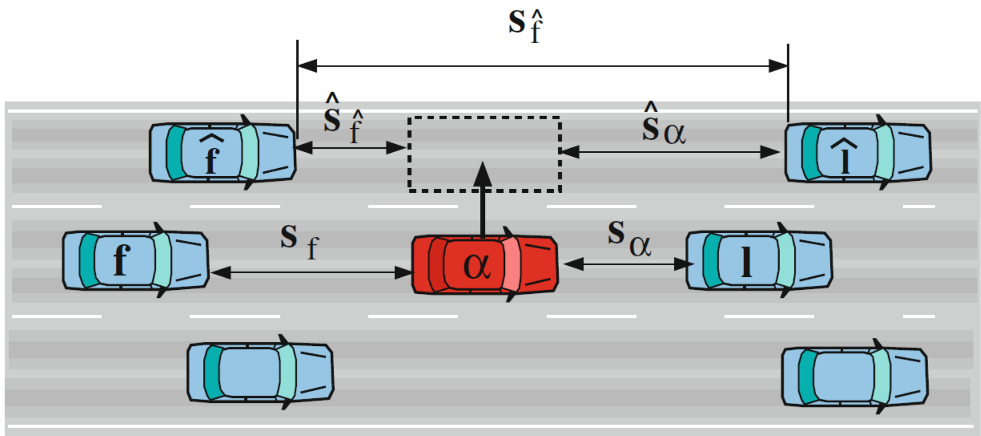

Figure 15: An example of the notation we are using for lane changes. l denotes the leading car and f denotes the following car. A hat refers to quantities after the lane change. From page 242 in [[TK13]].

When a vehicle decides to change lanes, it must do two things: check that the lane change is safe and check that there is an incentive to change lanes.

Checking to see if a lane change is safe is straightforward. All the vehicle must do is ensure that $\hat{s}_{\hat{f}}$ is smaller than a predefined safe gap. This minimum gap is dependent on factors like the velocities of the vehicles and the reaction time of the drivers. Considering this, we can define a minimum safe gap as

$$s_{\text{safe}}(v_{\hat{f}}, v_\alpha) = v_{\text{opt}}^{-1}\left[v_{\hat{f}} - \tau b_{\text{safe}} + \tau\gamma(v_{\hat{f}} - v_\alpha)\right] \tag{10}$$

where b_{safe} is the limit for a safe deceleration and τ is the adaptation time.

The inverse of the optimal velocity function, is not defined for negative velocity values, so we extend its domain and get

$$v_{\text{opt}}^{-1}(v) = \begin{cases} s_0 & v \leq 0, \\ s_0 + Tv & v > 0. \end{cases} \tag{11}$$

To check the incentive, we need to consider the gap in the current lane, the gap in the lane we want to switch to, and the velocities of the leading cars in both lanes. We only want to switch lanes if there is either a large gap in the other lane or if the leading car in the other lane is faster. Therefore, we can define a gap, s_{adv}, that encapsulates the benefit of staying in the same lane.

To prevent erratic lane changes, we also introduce an advantage threshold, Δa, such that the vehicle will only change lanes if potential gain in acceleration is greater than Δa. Left lanes are also usually designated as passing lanes, so right lanes are slower by convention. So, we introduce a_{bias} which lets our model prefer the left lane over the right one.

Combining all of the factors above gives us a measure of the opportunity cost of changing lanes:

$$s_{\mathrm{adv}} = s_{\alpha} + v_{\mathrm{opt}}^{-1} \left[\tau(\Delta a + a_{\mathrm{bias}} + \gamma(v_l - v_{\hat{l}})) \right]. \tag{12}$$

So, a vehicle can only change lanes if $\hat{s}_{\hat{f}} > s_{\mathrm{safe}}$ and $\hat{s}_{\alpha} > s_{\mathrm{adv}}$.

Typical lane changing parameters in a highway are given in the table below.

Parameter	Value
b_{safe}, limit for safe deceleration	2 m/s^2
Δa, changing threshold	0.1 m/s^2
a_{bias}, keep left directive	0.3 m/s^2 in right lane, -0.3 in left lane

Figure 16: Typical parameters for highway traffic for the lane changing algorithm. From page 244 in [TK13]

When adding the lane changing algorithm to our existing code, we must ensure that the order of the cars are tracked during every timestep because a_{mic} and the lane changing algorithm use the state of the leading and following cars. If one car overtakes another, the array of cars will go out of order. We alleviated this by reordering our array of cars at the end of each timestep.

Algorithm 2 Simplified algorithm for FDVM with lane changes

Require: Initial state variables for each car at $t = 0$.
Require: carArr, an array of cars.

```
for i = 1 :numsteps do
    for j = length(carArr):−1:1 do
        State variables of jth car ← Update jth car by a timestep.
        New lane of jth car ← carArr(j).changeLane()
    end for
    sort(carArr)
end for
```

4.3.2 Pitfall

The major issue with the lane changing algorithm comes from the fact that we can only think one timestep ahead and look at one lane to the side. This means that our algorithm can't make tactical decisions like switching into a slow lane and then switching again to a fast lane. Because we extrapolate the gaps and velocities at the current timestep, we are effectively only thinking one timestep ahead too. So, lane changes are often short-sighted. This manifests in what we call *lane oscillations* where a car will switch to a different lane and then switch back very quickly.

4.4 Multi-lane Bottleneck

An example of a multi-lane bottleneck is construction on the side of a highway. We can examine what happens when two lanes of a three-lane highway are closed for maintenance.

4.4.1 Implementation

To implement the closure of two lanes, we placed two virtual stationary vehicles. One was at lane one with a length of $l = 1100$ and the front at $x = 2000$. The other vehicle was in lane two and had a length of $l = 1000$ with the front also at $x = 2000$. So, lane three is left empty.

4.4.2 Data

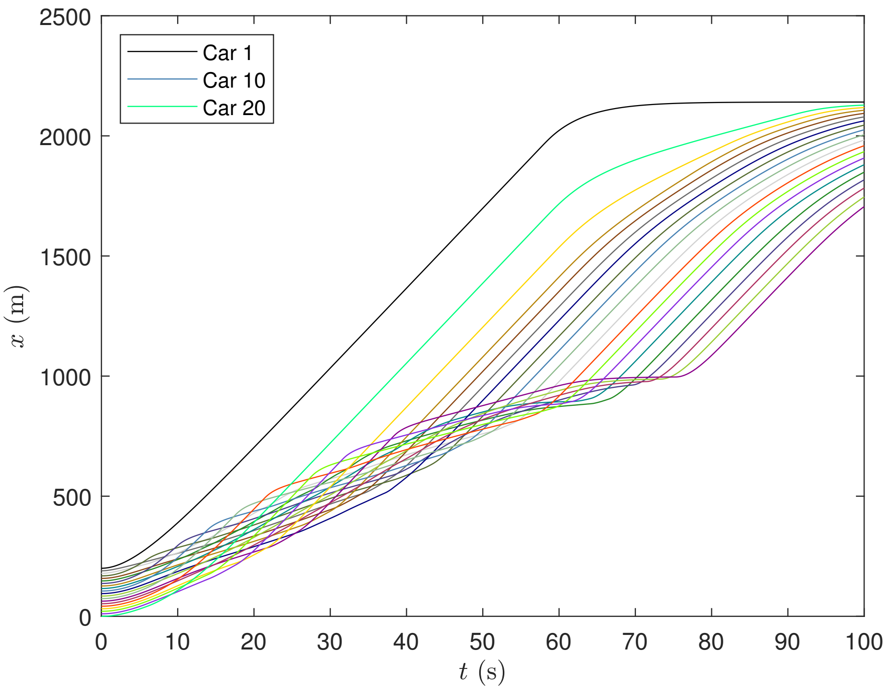

Figure 17: Position versus time graph. Each line represents a single vehicle, and vehicles in all lanes are shown. Car 1 started in lane 3.

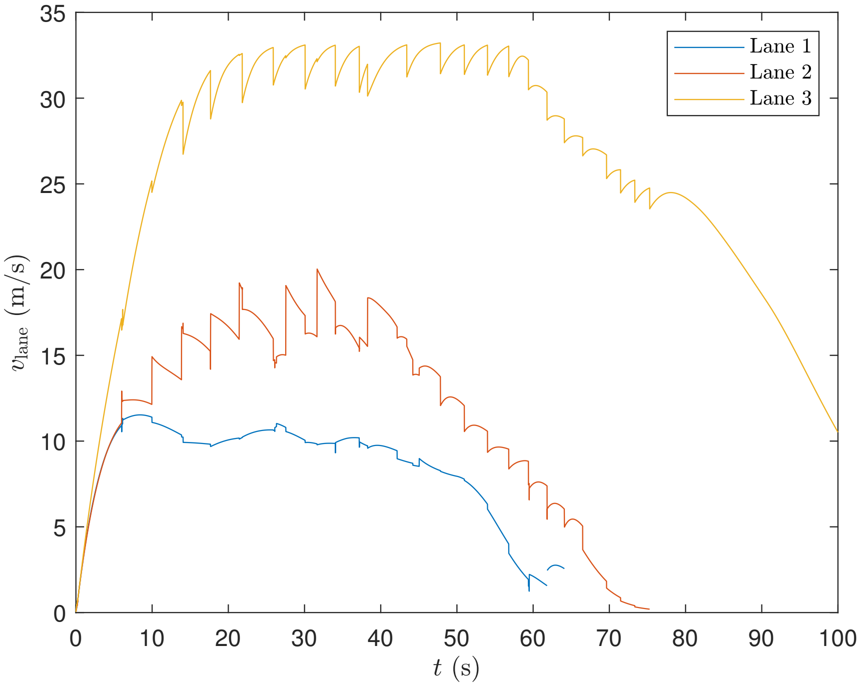

Figure 18: Mean speed of vehicles in each lane

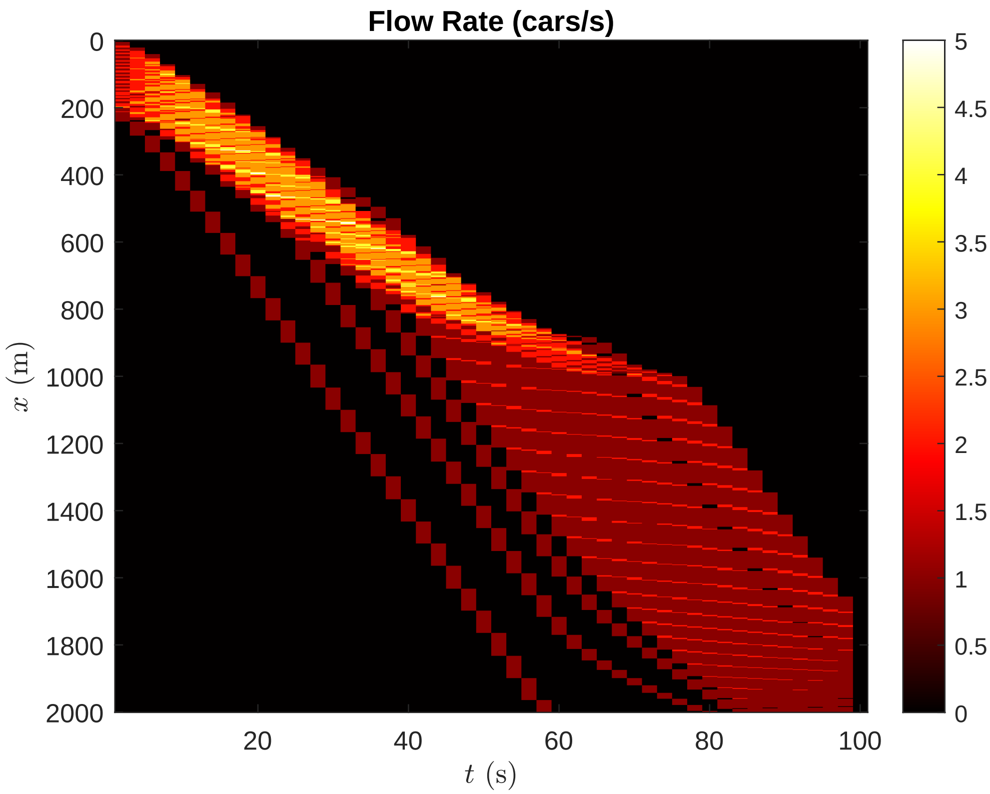

Figure 19: Flow rate versus time (s) and position (m)

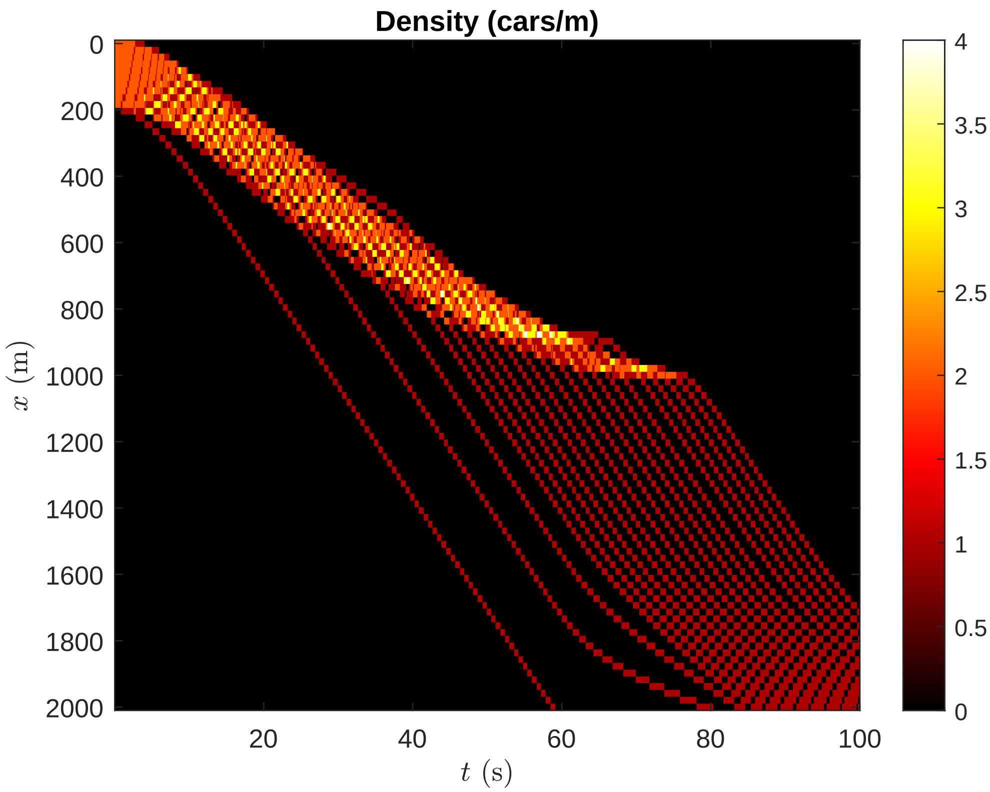

Figure 20: Density versus time (s) and position (m)

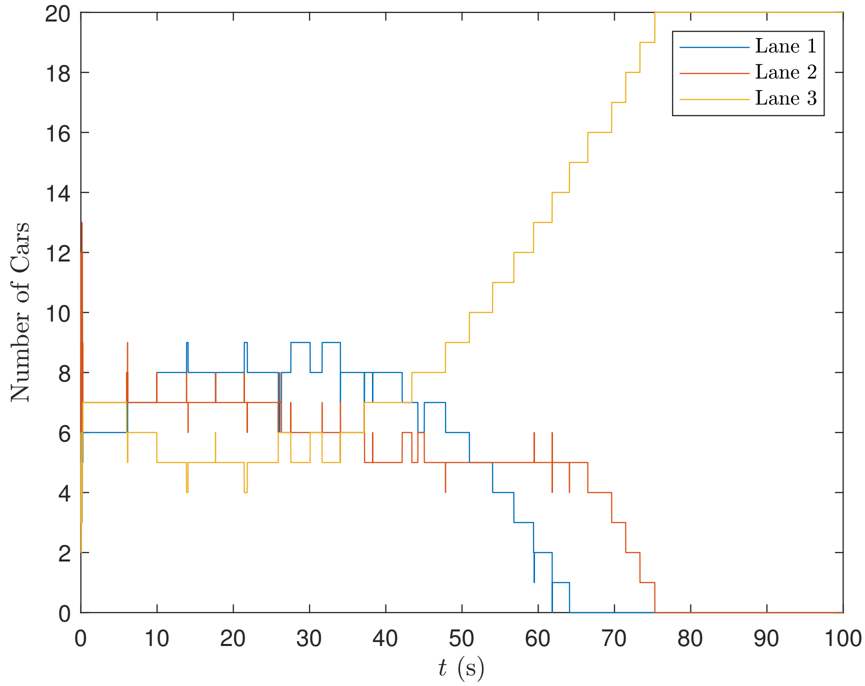

Figure 21: Number of cars per lane

4.4.3 Analysis

In the position versus time graph (17), most of the cars appear to slow down before the roadblock at $x = 900$ except for car one. This is caused by the cars in the front slowing down and changing to the empty lane in anticipation of the roadblock. Since car one is the first car, it is able to avoid the traffic caused by lane changes farther back.

From the mean speed graph (18), the mean speed of cars in lane three are much higher than those in lanes one and two, as expected. There are a lot of sudden dips in v_{lane} for lane three and these dips correspond sudden peaks in lane two. This is caused by cars changing lane from lane two to three, hence cars in lane two accelerate and cars in lane three decelerate.

Looking at the flow heatmap in figure (19), we observe a flow rate that is much higher for x values before the roadblock. After the roadblock, the flow spreads out in space and the steepness increases, indicating faster moving cars and a lower density.

The heatmap of density in figure (20) shows the same overall picture as figure (19).

Figure (21) confirms that all cars eventually end up in lane three. There is a sharp decrease of cars in lane one at $t = 50$ and a decrease of cars in lane two at $t = 70$. We can also see *lane oscillations* as discussed in (4.3.2).

5 Conclusion

In this paper, we have successfully implemented a microscopic car following model and lane changing algorithm and used it to analyze several common scenarios. The model we are using has some pitfalls, such as unrealistically large accelerations. However, it does help us understand the creation and propagation of stop-and-go waves in single lane traffic. Similarly, our lane changing algorithm can only look ahead one timestep and look across one lane. Despite this, it still illustrates traffic flow bottlenecks when lanes in a highway are closed.

Our paper is only the tip of the iceberg. The car following model is rather powerful and can be used to study many other traffic phenomena. Future iterations of this project could explore cars entering a priority road and the flow of traffic in a roundabout.

List of Symbols and Constants

x	position [see sec. 2.1].
v	velocity [see sec. 2.1].
a	acceleration [see sec. 2.1].
t	time [see sec. 2.1].
s	gap [see sec. 2.1].
v_0	desired speed (typically 33.3 m/s) [see ch. 3.1]
s_0	minimum distance gap (typically 3 m) [see sec. 3.1]
T	time gap (typically 1.4 s) [see sec. 3.1]
τ	adaptation time (typically 5 s^{-1}) [sec sec. 3.1].
γ	speed difference sensitivity (typically 0.6 s^{-1}) [sec sec. 3.1].
ρ	density [sec sec. 4.2.1].
Q	flow rate [see sec. 4.2.1].
b_{safe}	limit for safe deceleration (typically 2 m/s^2) [see sec. 4.3.1].
Δa	changing threshold (typically 0.1 m/s^2) [see sec. 4.3.1].
a_{bias}	keep left directive [see sec. 4.3.1].

References

[Ste11] David E. Stewart. *Dynamics with Inequalities*. SIAM, 2011. ISBN: 978-1-611970-70-8.

[TK13] Martin Treiber and Arne Kesting. *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin Heidelberg: Springer, 2013. ISBN: 978-3-642-32459-8.

[van+15] Femke van Wageningen-Kessels et al. "Genealogy of traffic flow models". In: *EURO Journal on Transportation and Logistics* 4 (2015), pp. 445–473. DOI: `https://doi.org/10.1007/s13676-014-0045-5`.