Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

Parametric study of traffic flow in a two lane traffic model with lane changes

Ghosh Jyotishman & Varna Achinta

Zürich
16.12.2011

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of Originality

This sheet must be signed and enclosed with every piece of written work submitted at ETH.

I hereby declare that the written work I have submitted entitled

Parametric study of traffic flow in a two lane traffic model with lane changes

is original work which I alone have authored and which is written in my own words.*

Author(s)

Last name	First name
Varna H S	Achinta
Ghosh	Jyotishman

Supervising lecturer

Last name	First name
Balietti	Stefano
Donnay	Karsten

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (http://www.ethz.ch/students/exams/plagiarism_s_en.pdf). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zurich
Place and date

Signature

Agreement for free-download

We hereby agree to make our source code of this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Achinta Varna　　　　　　　　Jyotishman Ghosh

Table of Contents

Abstract

We have presented a two lane traffic model with symmetric lane changing. An optimal velocity model was used to simulate the two lane traffic flow. Lane changing is taken into account based on two criteria which is the safety and the incentive criteria. Using the above model , a detailed study of the effect of various parameters on the dymanics of traffic jam is made.

Increasing maximum velocity leads to increase in average density in the lane. Increasing the safety distance results in the decrease of lane change frequency. Presence of a rogue driver doesn’t globally effect the dynamics of traffic jam.

Individual contributions

1. Simulation of one and two lane models without lane changes, study of the effect of rogue driver, animation of the lane traffic flow were done by Jyotishman Ghosh.

2. Simulation of two lane model with lane changing, analysis of various cases and plotting their results were done by Achinta Varna.

Introduction and Motivations

All of us have faced traffic jams in our day to day life at some point of time. There are many theories as to why traffic jams occur. Some try to classify it using mathematical theories comparing it to a problem of fluid dynamics. Others have tried to reason them with the aid of economic theories. One way or the other, people are always interested in finding out the various factors that affect the natures of these jams.

Figure 1: An image showing a traffic jam caused during the rush hours in New Delhi, India

In this project we have tried to simulate ideal driver behaviour in a two lane car model. Using numerical simulation we try to find out how the traffic dynamics changes with changes in driving conditions. One real life example which we simulated is when one of the two lanes gets blocked due to an accident. This is a very common site on the Indian motorways which leads to causal of long traffic jams. Thus using this model we could get an insight into the various factors that are critical in controlling and prevention of such kind of jams and also to evaluate the time needed for the clearance of the jams. We also try to study the effect of a single 'rogue' driver on the whole system.

Description of the Model

2.1 Introduction

The two lane model which is used in our work is derived from [1]. The two lane traffic model considers two lanes of equal width present for a length of 400 meters. Cars are allowed to change lanes from left to right and vice versa. The two lanes have a predefined flow rate of the vehicles entering the lanes at x=0 meters. These flow rates are independent of each other. An optimal velocity model is used to determine the forward motion of the car. Finite Difference technique is used to discretize the differential equation. The lane changing takes place only when certain criteria are satisfied by the vehicles which depend on their position and velocities and also on the cars in front and back of them. These criteria are what which describe the behavior of the drivers. Major work is invested in determining the effect of different parameters on the density of traffic and the velocity of vehicles.

2.2 Optimal velocity model-Theory

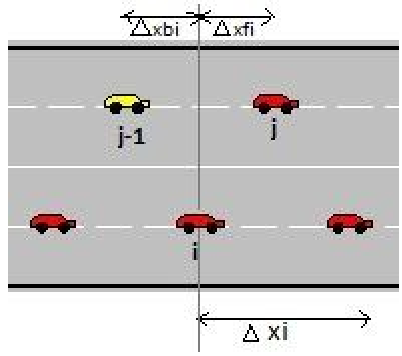
The motion of car at any given time instant can have two states. The first is the forward movement and the second is the sideways lane-changing action. The forward motion of car is governed by the optimal velocity model described in paper [1]. The model calculates the position of car using the following equations:

$$\frac{d^2 x_i}{dt^2} = a\left(V\left(\Delta x_i\right) - \frac{dx_i}{dt} \right)$$

$$V\left(\Delta x_i\right) = \frac{v_{\max}}{2}\left[\tanh\left(\Delta x_i - x_c\right) + \tanh\left(x_c\right)\right]$$

where x_i is the position of car at time t, $V\left(\Delta x_i\right)$ is the optimal velocity, Δx_i is the headway of car i at time t, a is the sensitivity, $v_{\max}$ is the maximal velocity and x_c is the safety distance.

Symmetric lane changing rules given in [1] are considered in the present study. The car changes its lane when both incentive and security criteria are valid. The following lane changing rules are implemented in the code:



- $v_i \succ 1.02 v_{i+1}; \Delta x_i \prec 4x_c$ - incentive criterion
  $\Delta x f_i \succ 2x_c; \Delta x b_i \succ x_c$ - security criterion

- $\Delta x_i \prec 2x_c$ - incentive criterion
  $\Delta x f_i \succ \Delta x; \Delta x b_i \succ x_c$ -security criterion

Car changes its lane when one of the two rules are satisfied.

Implementation

Finite difference technique is used to differentiate equation 1. The equation is discretized as follows:

$$\frac{x_i^{k+1} - 2x_i^{k} + x_i^{k-1}}{(\Delta t)^2} = aV(\Delta x_i) - a\left(\frac{x_i^{k+1} - x_i^{k}}{\Delta t}\right)$$

Hence, the position of car 'i' depends on its position in the previous 2 time steps and on position of car in the front (included in the optimal velocity calculation). The position of car 'i' at time step 'k+1' can be calculated by solving the above algebraic equation.

Velocity of the car is calculated using the formula:

$$v^{k+1}{}_i = \frac{x_i^{k+1} - x_i^{k}}{\Delta t}$$

The car at the front of the traffic is assumed to have maximum velocity (V_{max}).

An accident occurs at x=300m due to which cars slow down at that position. This is implemented in the code by reducing the velocity of cars (at x=300m) to zero.
New cars are added to lane 1 and lane 2 depending on the inflow rates at the two lanes. The position of the last car is checked and if its greater than the inflow, a new car is added to that lane. Hence, cars are continuously flowing in to the highway and get accumulated near the accident zone.

The lane changing rules are applied to both lane 1 and 2 using 'if' condition. The numbers of cars on lanes change when a car changes its lane. Hence, the dimension of position and velocity matrices are changed to accommodate a new car/remove an old car.

The density is calculated by considering a length of 10m. The number of cars present on the 10m strip is counted and the density at the midpoint of the 10m strip is calculated as:
Density= number of cars/10; (unit: number of cars per unit meter).

Simulation, Results and Discussion

Case 1

Simulation was carried out with X_c=4 m, a=1, V_{max}= 2 m/s and inflow lengths of 8m and 20m for lane 1 and lane 2 respectively. Since the inflow length is high for both lanes, localized jam appears near the accident zone (x= 300m). Figures 1 and 2 represent the densities on lane 1 and 2 respectively. The high density spots appear just behind the stopped car on lane 1. Due to high density near the stopped car on lane 1, cars try to change lane due to which there exists few high density spots on lane 2. The average density for lane 1 is higher than lane 2.

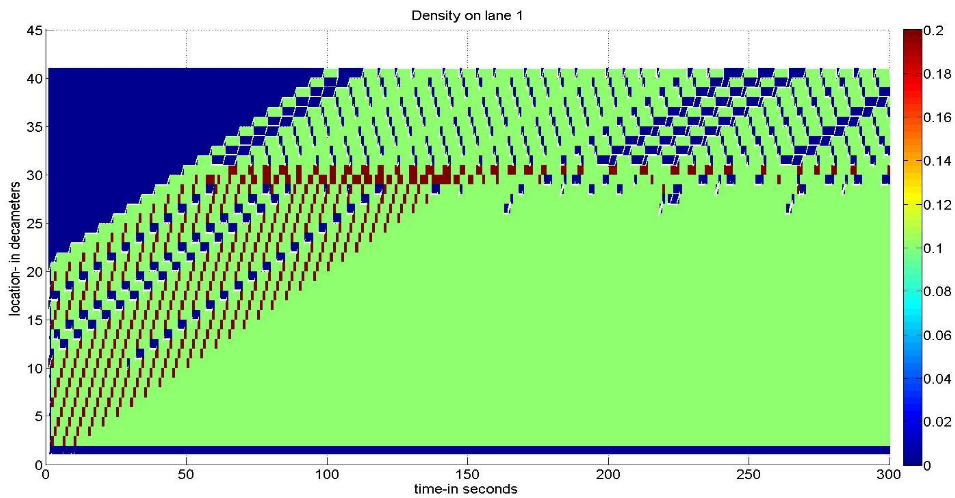


Fig 1: Density on lane 1 (case 1)

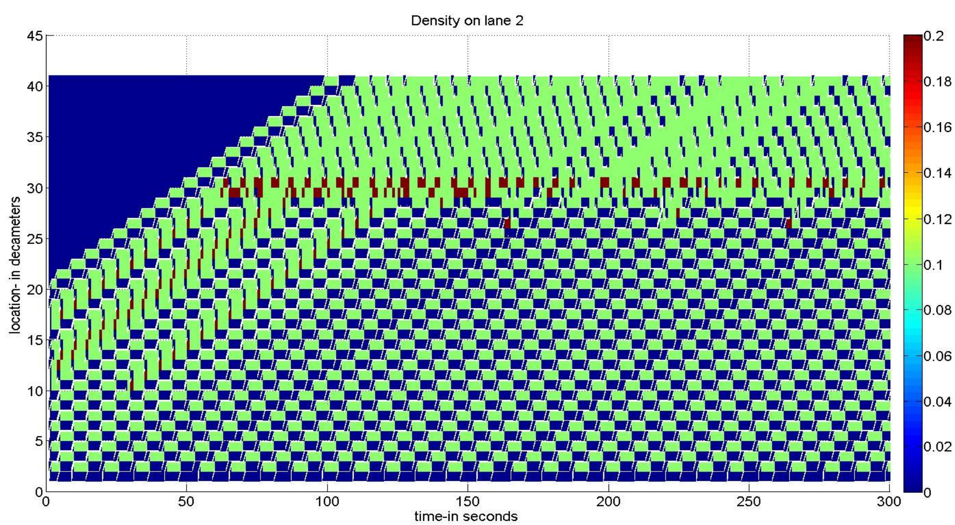


Fig 2: Density on lane 2 (case 1)

Figures 3 and 4 show the velocity of cars on lane 1 and 2 respectively. Lane 1 cars slow down near the stopped car and the slow velocity is region is confined to the accident zone. The velocity of cars on other regions of lane 1 is high(2 m/s) since the inflow length is high. The cars on lane 2 slow down suddenly when a car from lane 1 shifts to lane 2. Hence, the velocity of lane 2 cars fluctuates as seen in figure 4. The features observed in this model agrees with the results of paper [1].

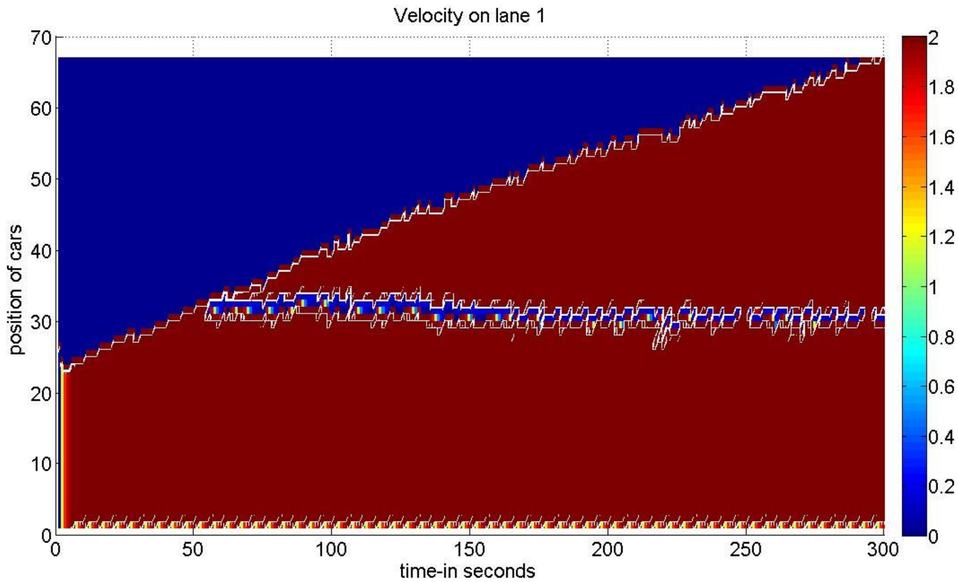


Fig 3: Velocity of cars on lane 1 (case 1)

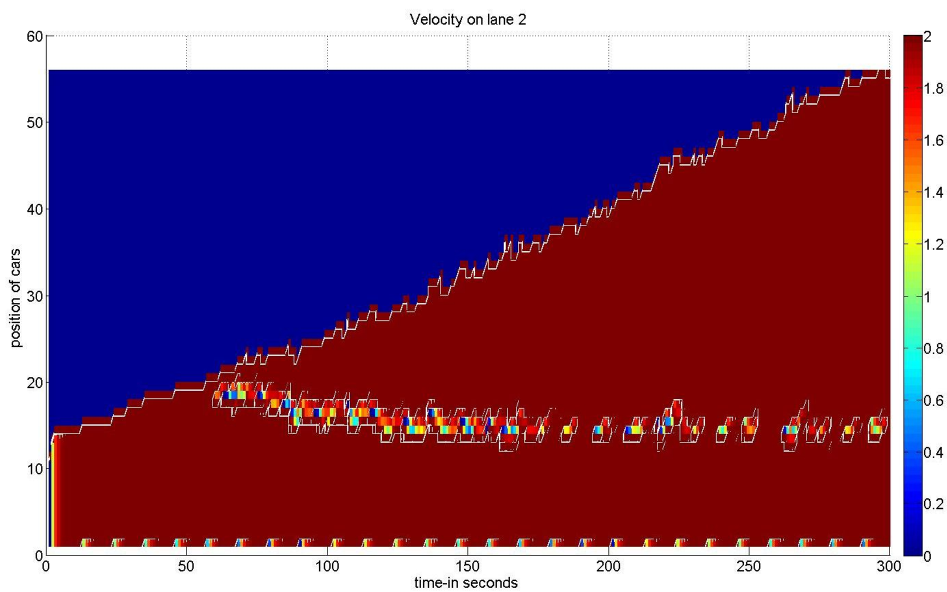


Fig 4: Velocity of cars on lane 2 (case 1)

Case 2

The inflow lengths were reduced to 4m and 8m for lane 1 and 2 respectively, while keeping other parameters constant. As seen in figure 5, high density region propagates behind the stopped car. The inflow of cars is high due to which the cars get accumulated near the accident zone. Hence, the high density region on lane 1 increases with time and extended traffic jam is observed. Corresponding to the traffic jam on lane 1, there exists a high density region on lane 2 as seen in figure 6. The length of high density region fluctuates with time and space due to lane changing of cars. Oscillatory jam is found for low inflow lengths. Paper [1] obtained similar results for low inflow lengths. The overall density of lane 1 is very high when compared to lane 2. On lane 1, the velocity of cars is very low near accident zone. The velocity of cars on lane 2 fluctuates due to lane changing as seen in figure 8. The extended jam is highly oscillatory when compared to localized jams. The results of this project matches with paper [1] and this comparison is a partial validation of our code.

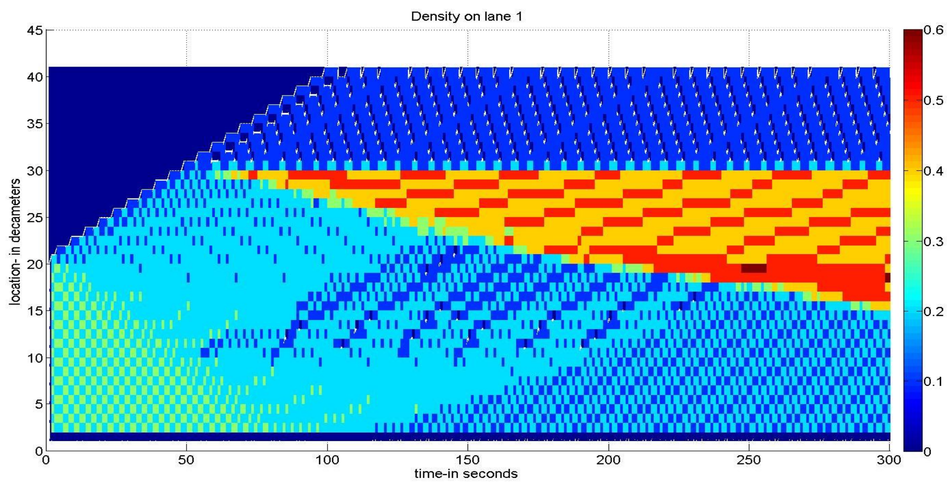


Fig 5: Density on lane 1 (case 2)

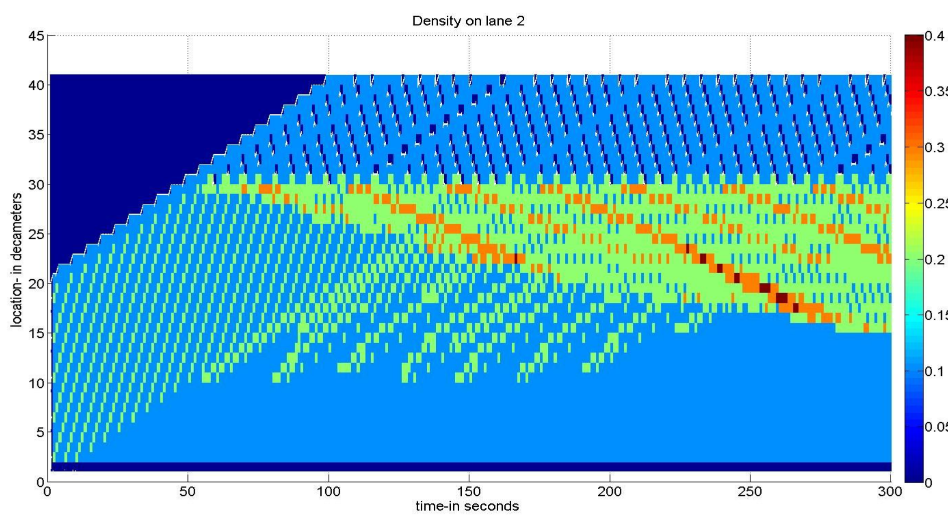


Fig 6: Density on lane 2 (case 2)

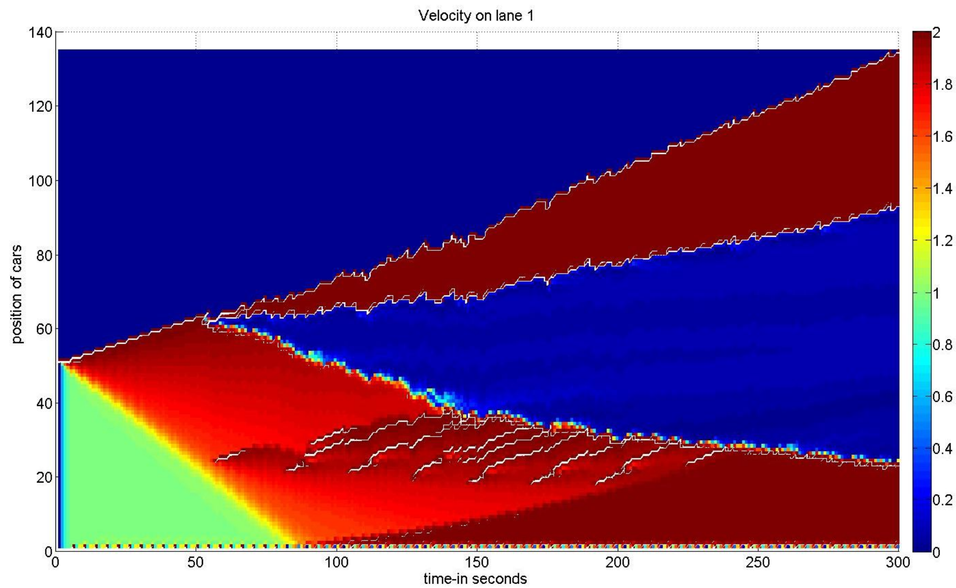


Fig 7: Velocity of cars on lane 1 (case 2)

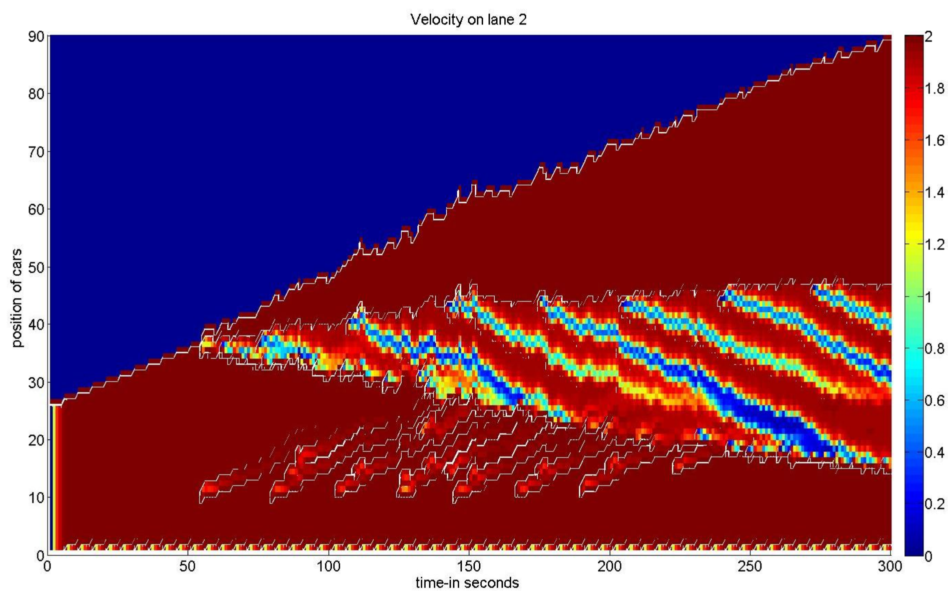


Fig 8: Velocity of cars on lane 2 (case 2)

Next, we do parametric study of extended jams. In the following cases, inflow lengths are 4m and 8m for lane 1 and 2 respectively. One parameter (a, V_{max}, X_c) is changed in each case and the results are compared with case 2.

Case 3

In this case, V_{max} was changed to 4 m/s. By comparing figures 5 and 6 with figures 9 and 10, we can conclude that the overall density of the lanes increase due to an increase in V_{max}. The high density region is less in this case when compared to case 2. The cars, speed up to V_{max} as soon as they get free space and hence the propagation of jam is less. The overall density of lane 2 is less than lane 1 as expected. The velocity of cars show similar trend as seen in case 2. Velocity of lane 2 cars fluctuate due to lane changing action.

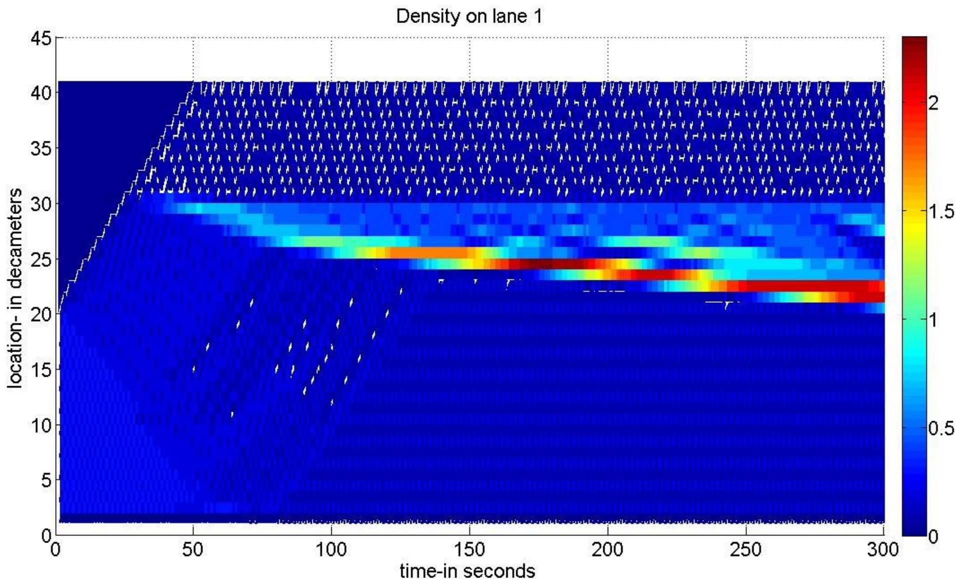


Fig 9: Density on lane 1 (case 3)

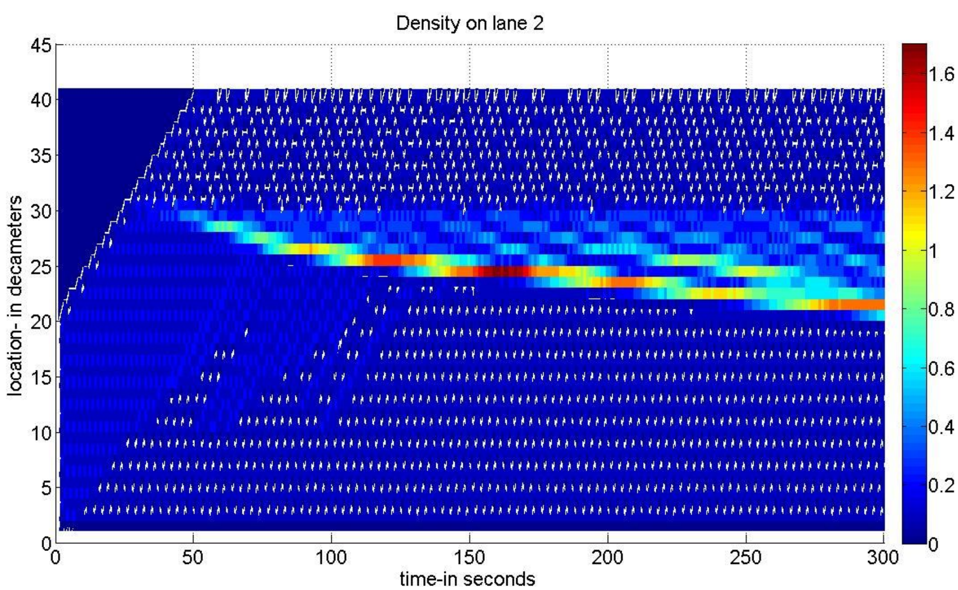


Fig 10: Density on lane 2 (case 3)

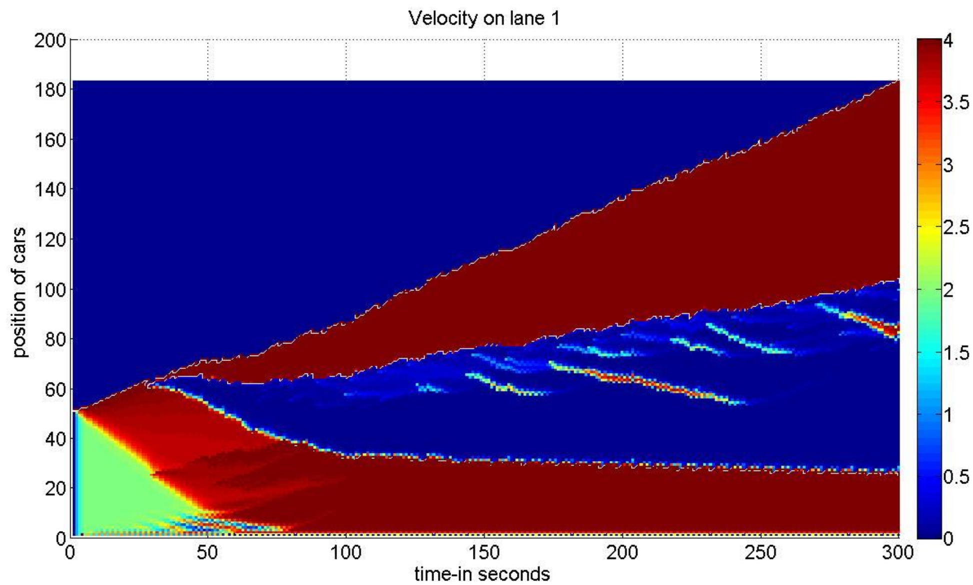


Fig 11: Velocity of cars on lane 1 (case 3)

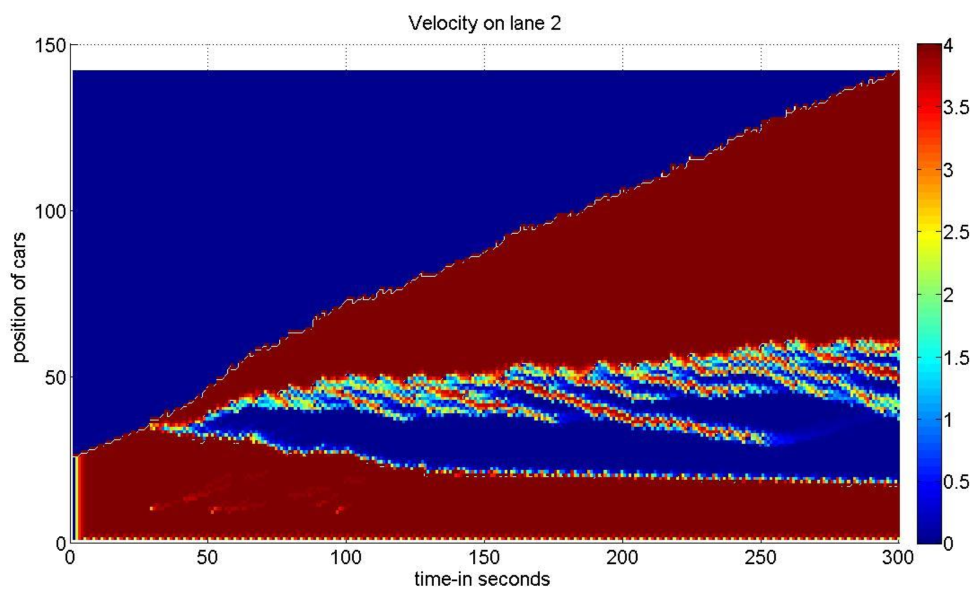


Fig 12: Velocity of cars on lane 2 (case 3)

Case 4

In this case, X_c was increased to 6m. Since the safety distance was increased, frequency of lane change reduces. One more observable phenomenon is that cars slow down even when the gap between its front cars is high. Due to this, high density region increases as seen in figures 13 and 14 respectively. When compared to figures 5 and 6, we observe that the peak density in this case is less (0.275 for lane 1 and 0.2 for lane 2) but the area of traffic jam is high. There is negligible jam in lane 2 due to the reduced frequency of lane change. The low velocity region on lane 1 is high in this case and the fluctuation of velocity of lane 2 cars is less.

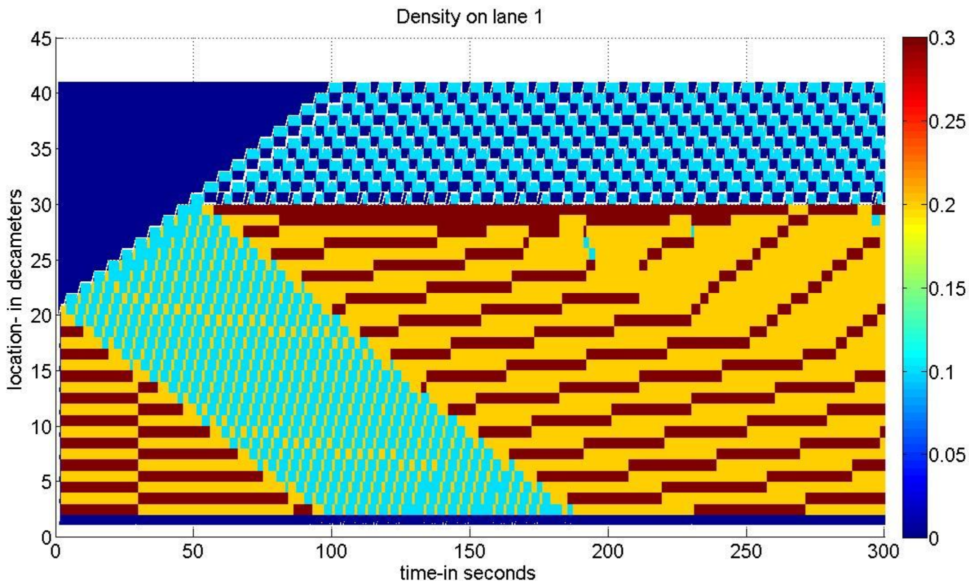


Fig 13: Density on lane 1 (case 4)

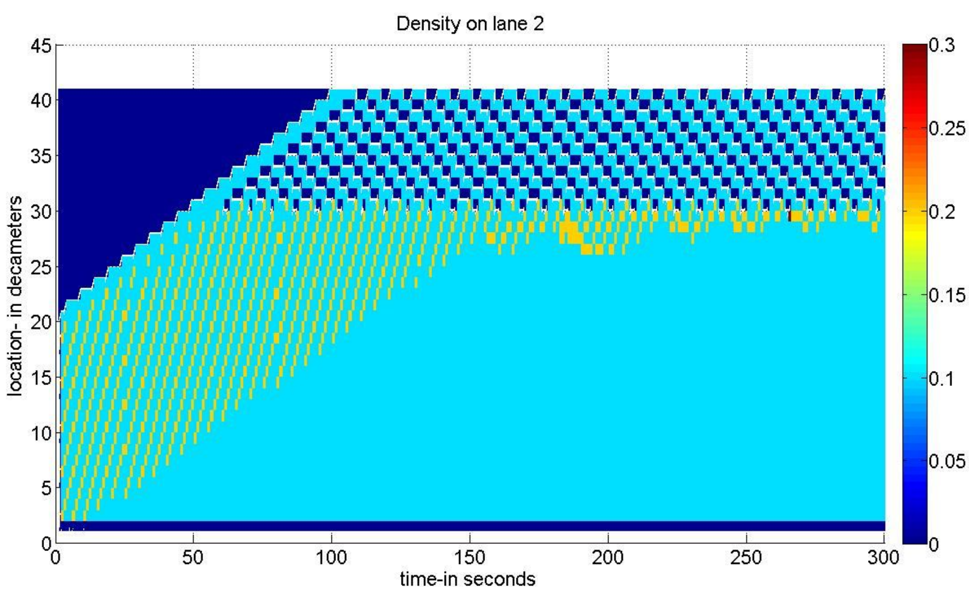


Fig 14: Density on lane 2 (case 4)

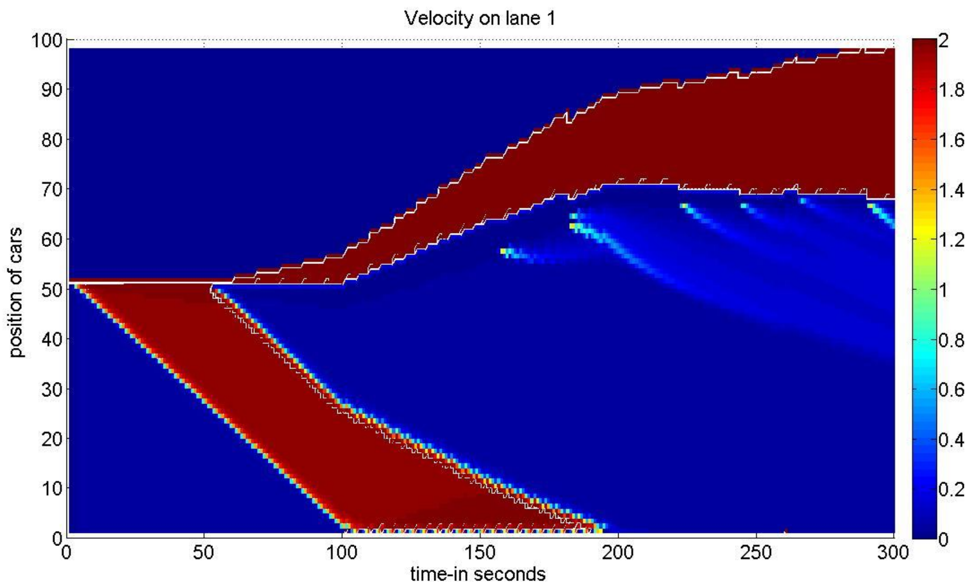


Fig 15: Velocity of cars on lane 1 (case 4)

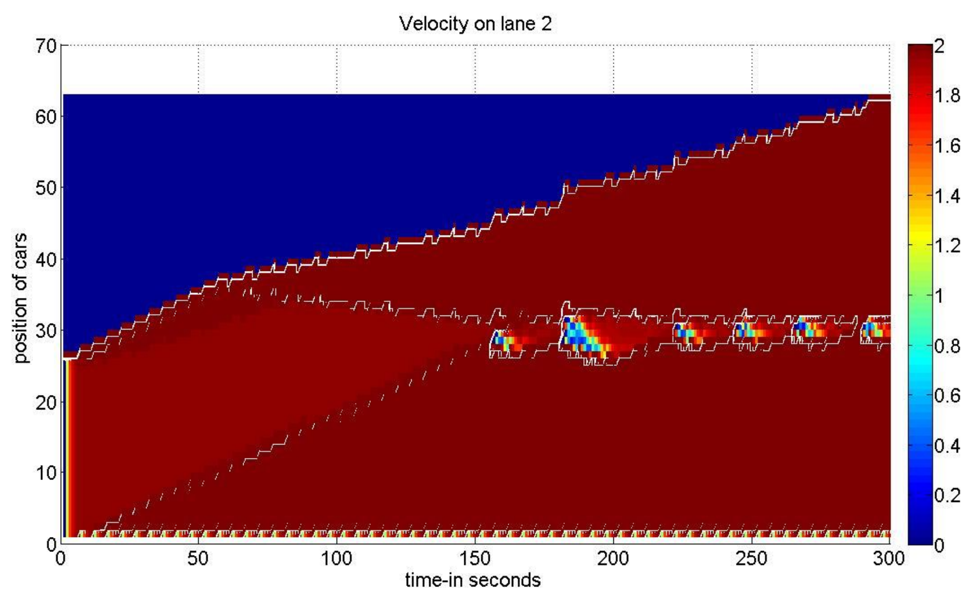


Fig 16: Velocity of cars on lane 2 (case 4)

Case 5

In this case, a is increased to 2. Since the sensitivity was increased, the fluctuation of velocity on lane 2 decreased as seen by comparing figures 20 and 8. There is minor difference between the density distribution in case 2 and case 5.

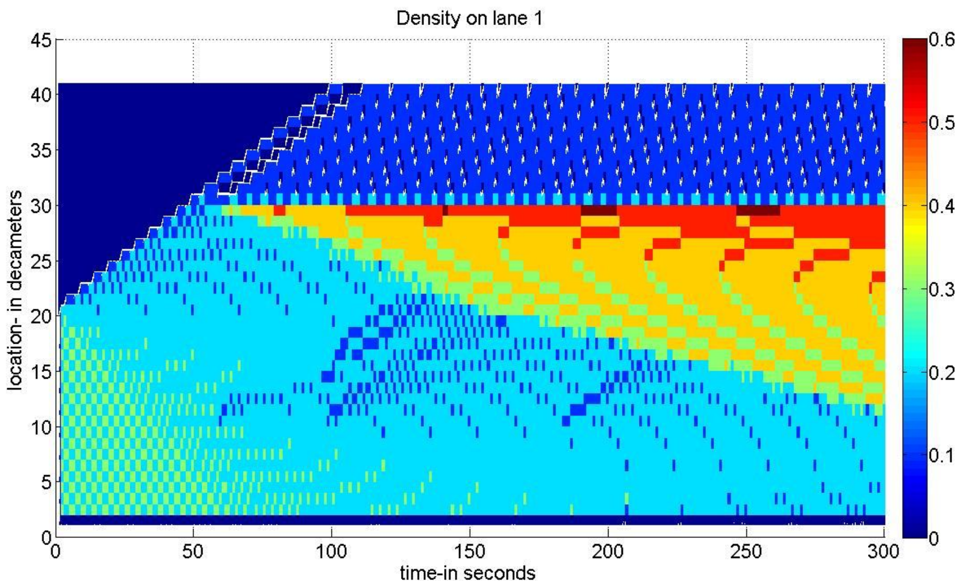


Fig 17: Density on lane 1 (case 5)

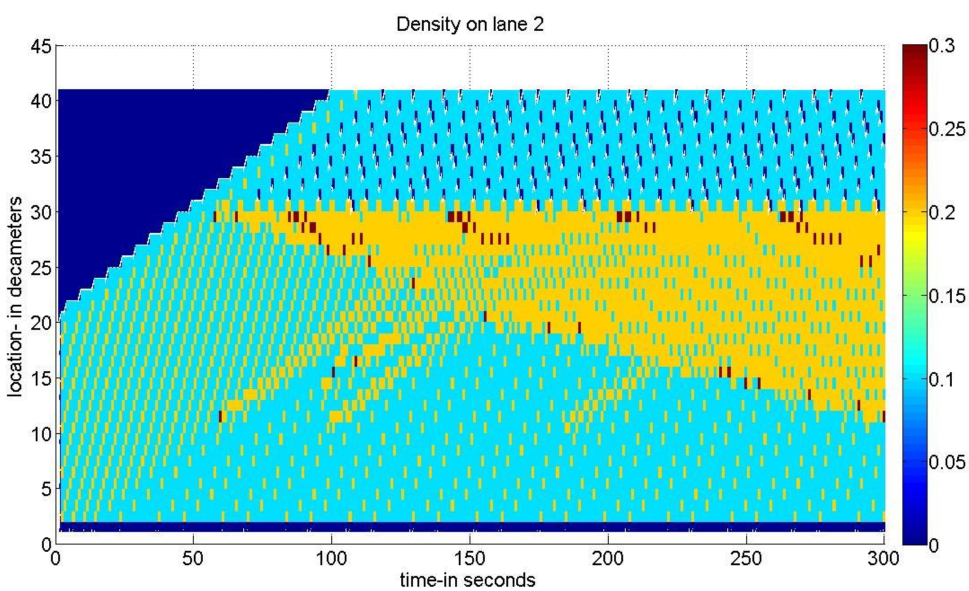


Fig 18: Density on lane 2 (case 5)

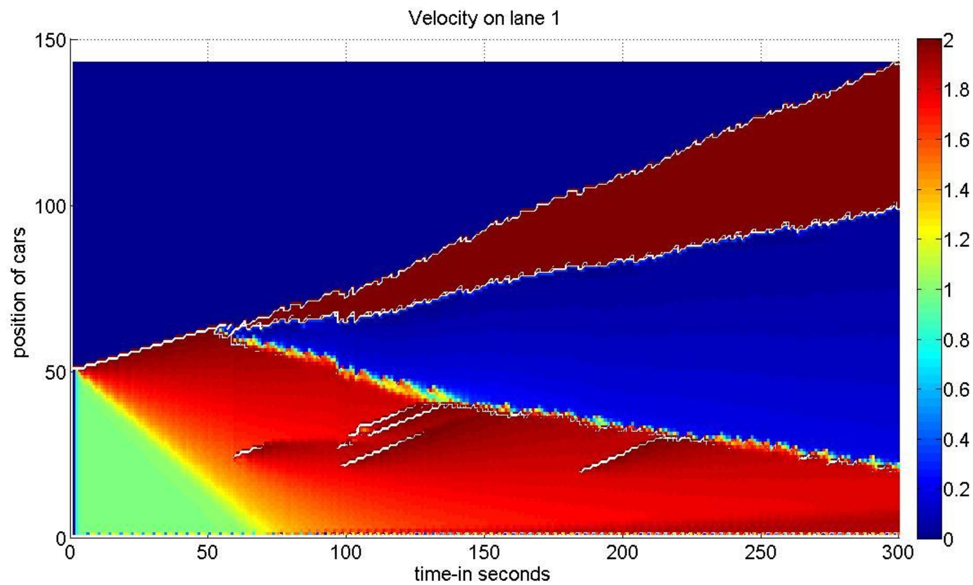


Fig 19: Velocity of cars on lane 1 (case 5)

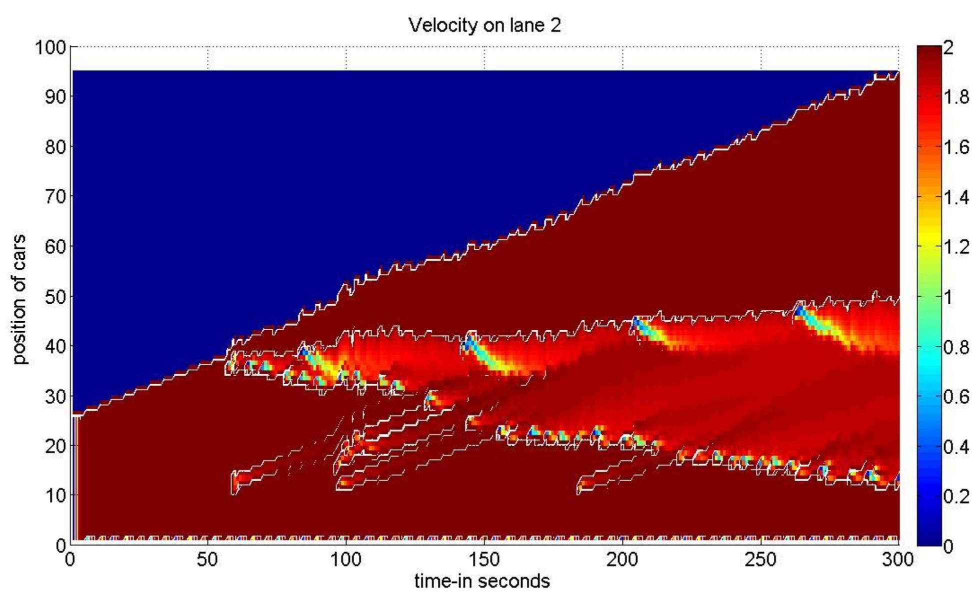


Fig 20: Velocity of cars on lane 1 (case 5)

Case 6

In case 6, cars slow down to a velocity of 0.2 m/s in the region x=250m to x=300m. Other parameters are the same as in case 2. As seen in figure 21, the high density region on lane 1 is high when compared to case 2. However, the velocity fluctuation of cars on lane 2 is less due the drop in the frequency of lane changing action. The cars do not stop, but slow down in a particular region. Hence, the peak density is less in this case and the cars do not change lanes frequently.

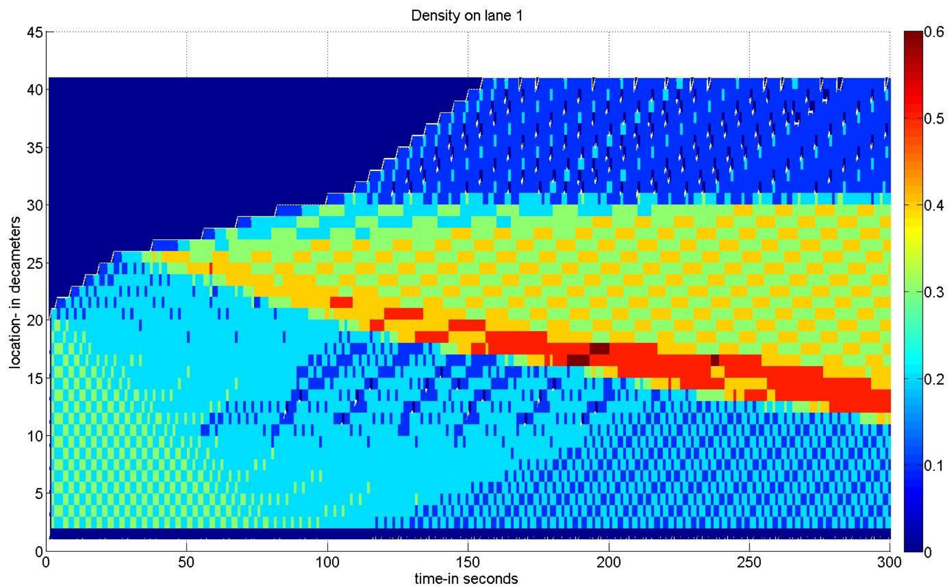


Fig 21: Density on lane 1 (case 6)

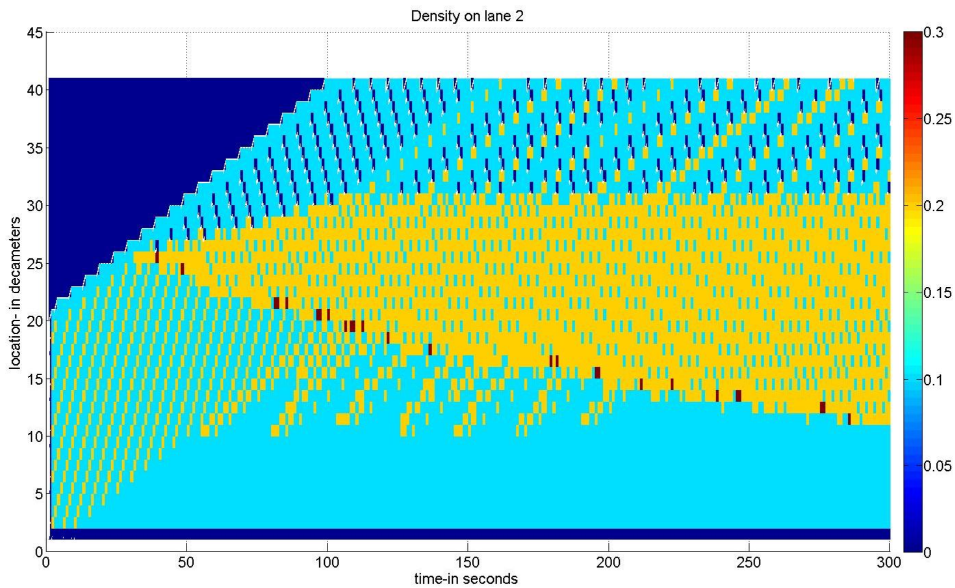


Fig 22: Density on lane 2 (case 6)

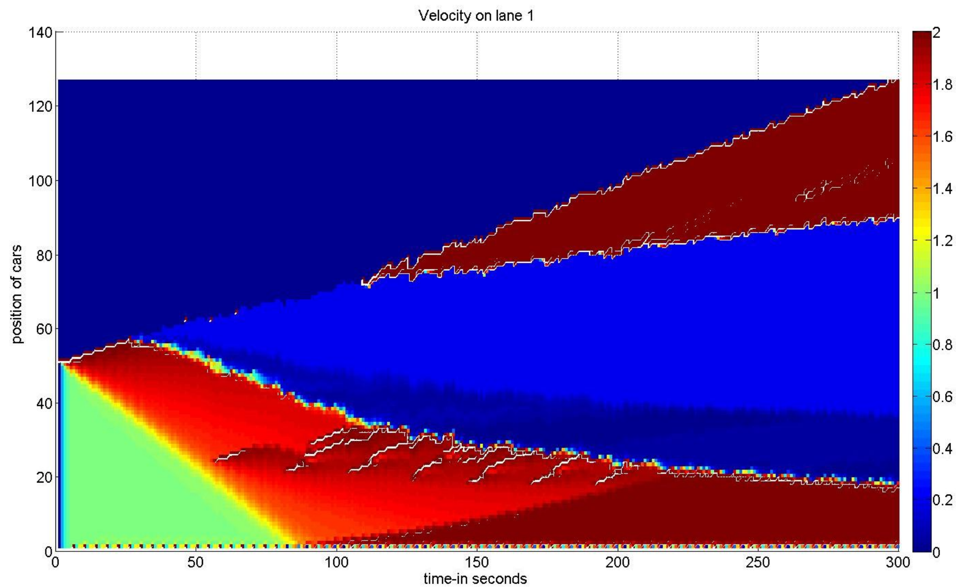


Fig 23: Velocity of cars on lane 1 (case 6)

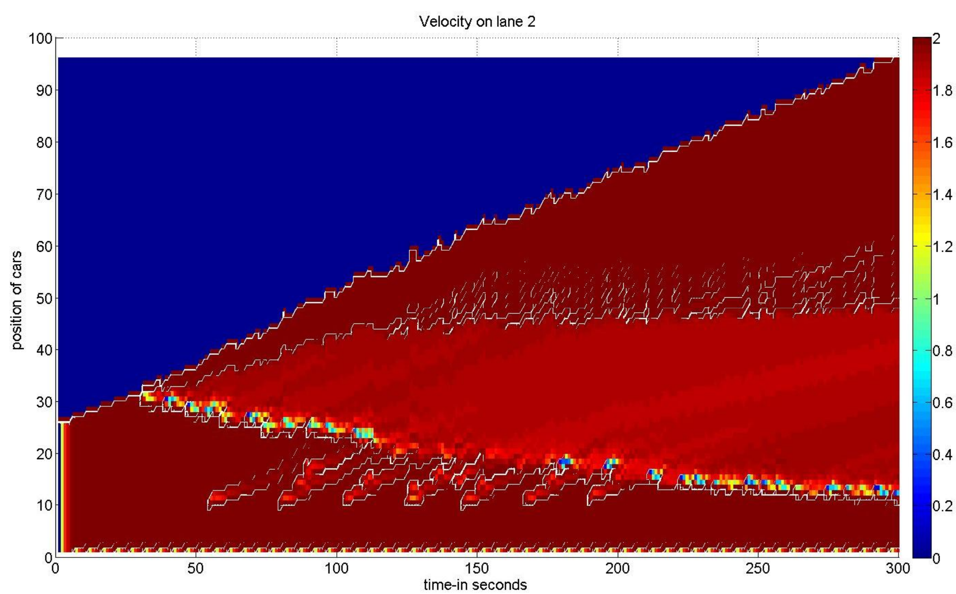


Fig 24: Velocity of cars on lane 2 (case 6)

Case 7

In this case, a car is stopped at x=300 m for 150 s and then it is allowed to move. Basically, the traffic jam is allowed to vanish. As seen in Figure 25, high density region on lane 1 increases for a certain time interval and then it decreases due to the clearance of road blockage. The density on lane 2 was less initially due to high lane-2 inflow length. Due to the lane changing action, densities of both the lanes become almost equal. The fluctuation of velocity of cars on lane 2 is reduced to the reduction in lane changing action.

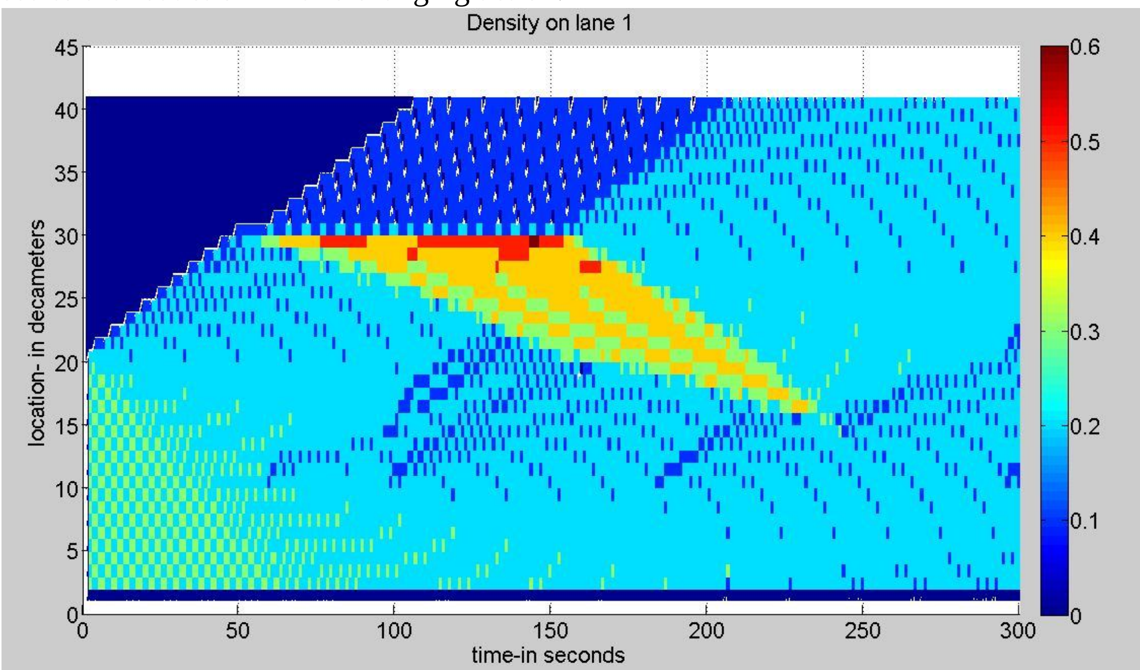


Fig 25: Density on lane 1 (case 7)

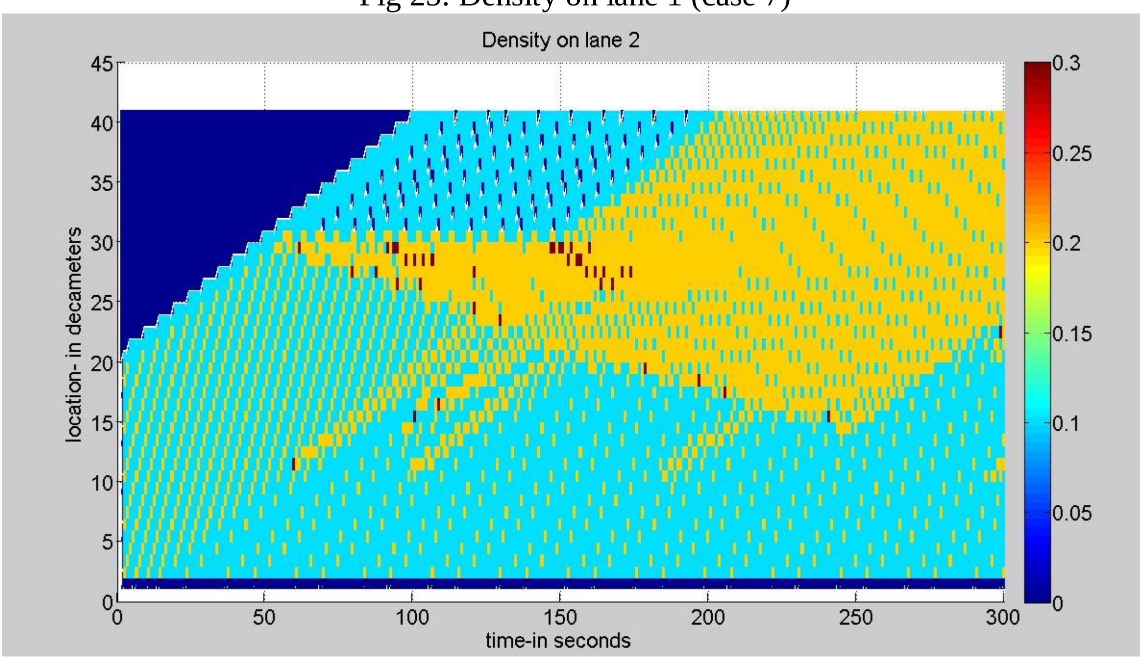


Fig 26: Density on lane 2 (case 7)

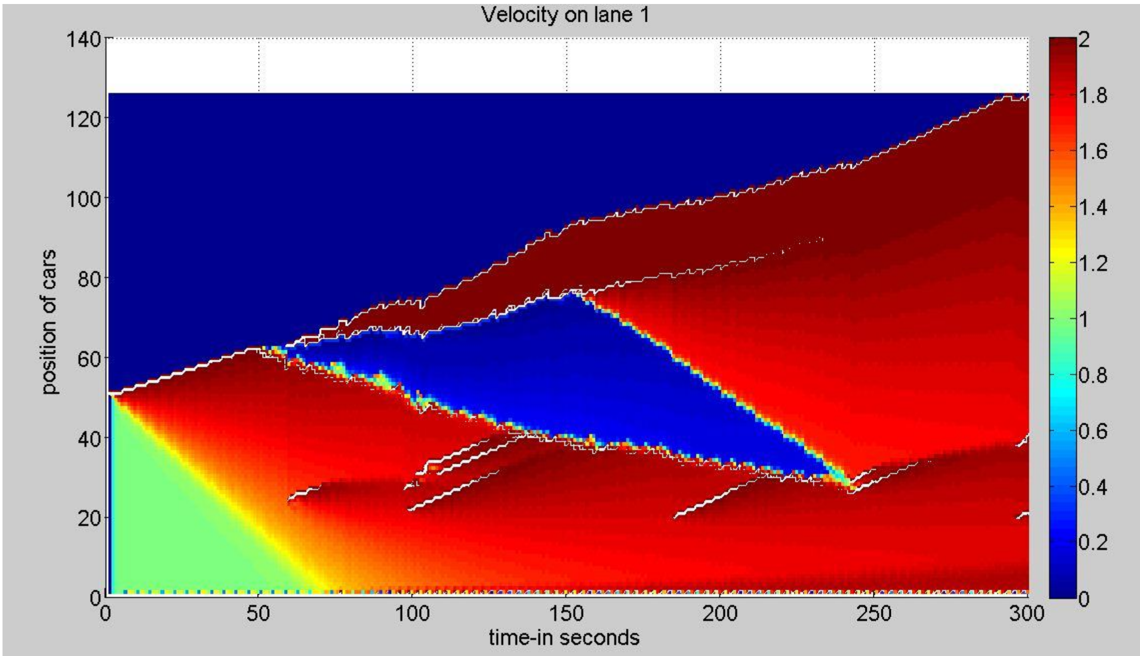


Fig 27: Velocity of cars on lane 1 (case 7)

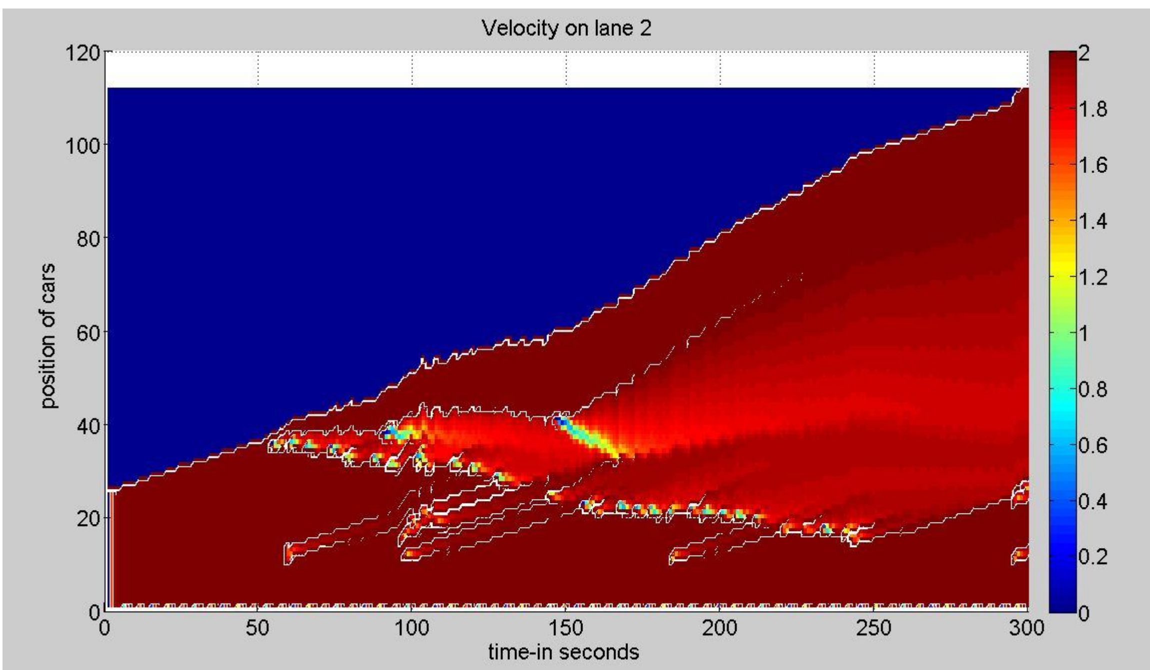


Fig 28: Velocity of cars on lane 2 (case 7)

Case 8

In this case, the effect of a single ‘rogue’ driver on the dynamics of traffic jam is studied. The parameters used in case 2 are used in this case. One driver was given a sensitivity value of 0.5 (a=0.5). By comparing figures 5, 6, 7 and 8 with figures 29, 30, 31 and 32, we observe that a single ‘rogue’ driver has negligible effect on traffic jam. We can observe that the length of the traffic jam and the peak densities are the same in cases 2 and 8. However, we can observe minor differences in the density and velocity plots. There are few spots of high density regions on lane 1 and 2 of case 8. This is caused by the erratic driving of the ‘rogue’ driver. Similarly, velocity plots show greater number of ‘low velocity’ regions due to the sudden lane-changing action of the ‘rogue’ driver.

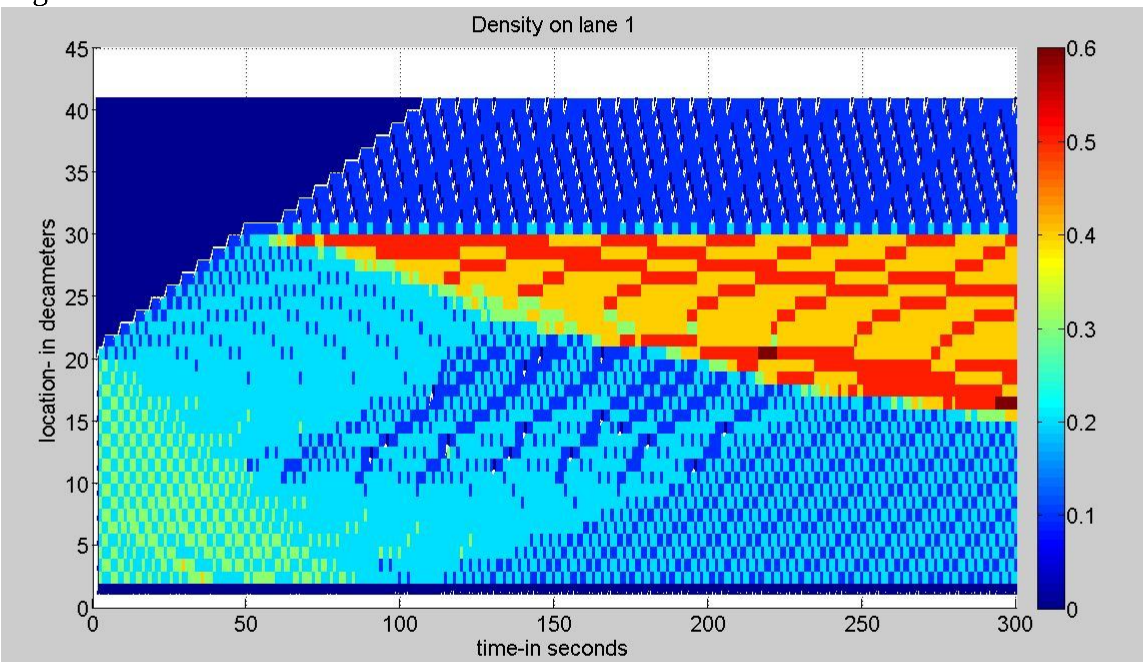


Fig 29: Density on lane 1 (case 8)

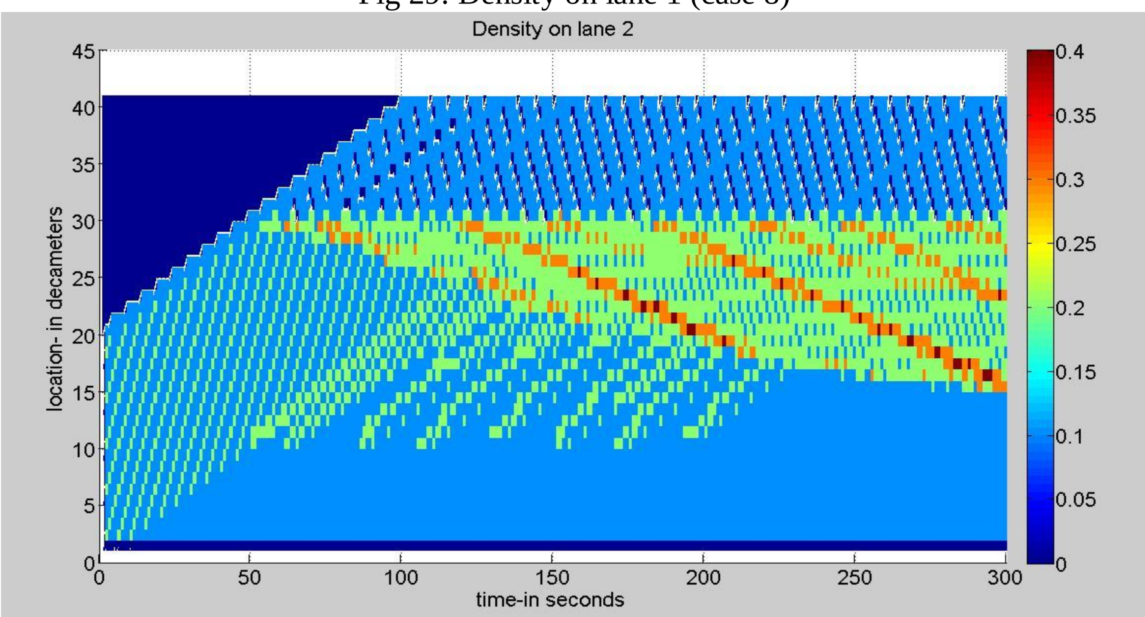


Fig 30: Density on lane 2 (case 8)

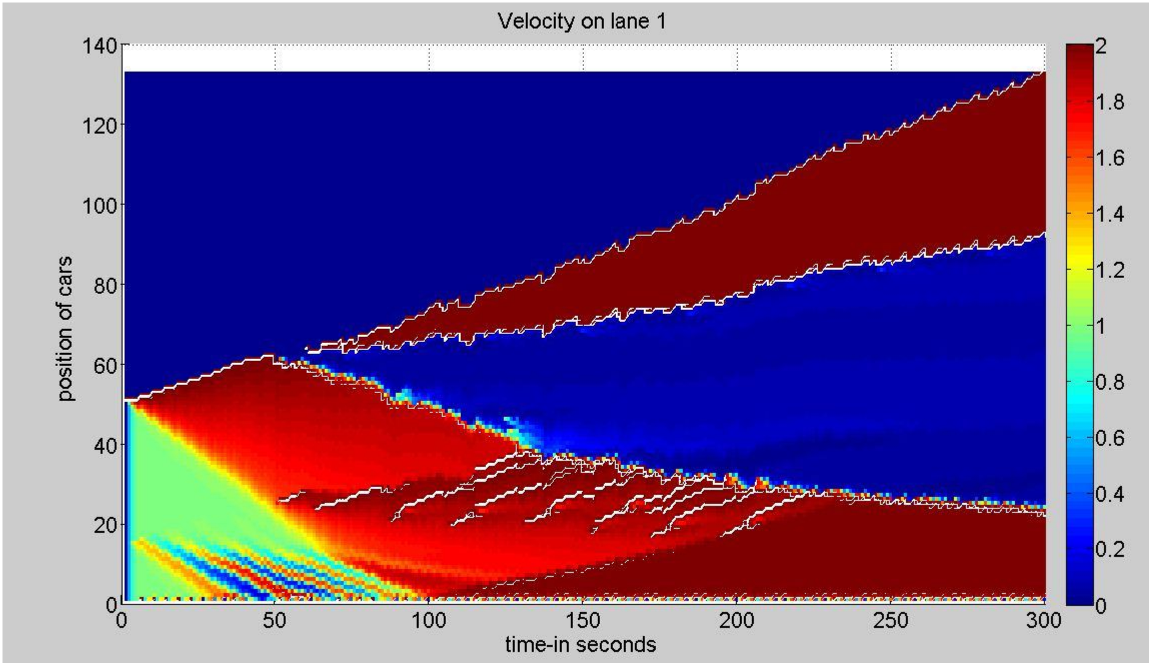


Fig 31: Velocity of cars on lane 1 (case 8)

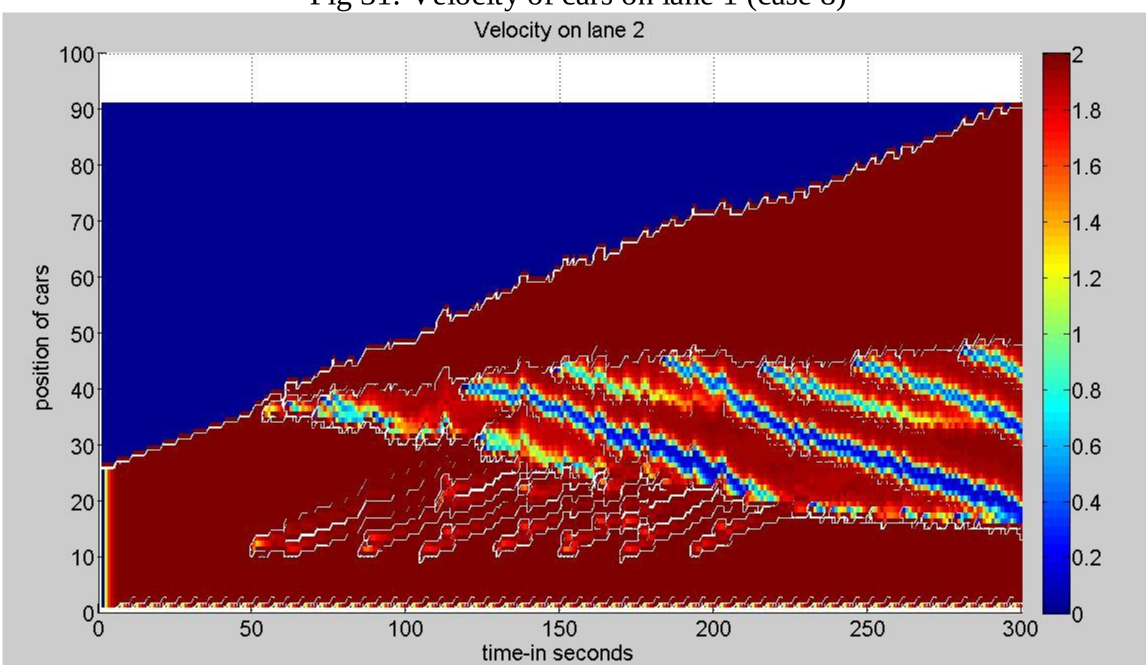


Fig 32: Velocity of cars on lane 2 (case 8)

Summary

Looking at the results and the animations obtained, we can say that the two lane traffic flow with lane changes model was able to simulate the real life situation in a very apt manner. We were able to answer most of the fundamental questions we had before starting this project. A detailed study of the influence of the various driver parameters on the traffic flow situation was enabled us to determine which are the crucial factors which affect the dynamics of traffic flow.

High inflow lengths lead to localized traffic jams and low inflow lengths result in extended jams. This was observed in paper [1] too. This comparison is a partial validation of our code since our model was able to replicate the results of [1].

Next, a parametric study was conducted to analyze the effect of different parameters on the dynamics of traffic jam. Increase in V_{max} increase overall density of both the lanes. Greater the value of safety distance (X_c), lower is the frequency of lane change. Hence, the fluctuation of velocity on lane 2 is reduced. Increasing the sensitivity (a) value has minor effect on the dynamics of traffic jam. When the cars slow down over a long stretch of road (case 6), propagation of traffic jam is severe. However, the frequency of lane-change and the fluctuation of velocity on lane 2 is reduced.

The effect of a drunken or a ‘rogue’ driver on the whole system was also studied and it seemed that a single inhomogeneous driver doesn’t cause too much difference in the behaviour of the whole system.

Regarding future scope of work, the driver model can be further extended by implementing features of HDM model [3] which might result into a more realistic driver behaviour.

References

[1] Shingo Kurata, T. N. (2002). Spatio Temporal dynamics of jams in two lane traffic flow with a blockage.

[2] S. Balietti, K. Donnay, (2011) Draw_car.m, Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB.

[3] Martin Treiber!, Arne Kesting, Dirk Helbing,(2005) Delays, inaccuracies and anticipation in microscopic traffic models

Appendix

Main driver file

```
% Traffic Dynamics
clear all
close all
clc

%% Defining constants
L=400; % length of 2 lane highway, L=400 m
x0=300; % Position of stopped car
vmax=2; % in m/s
Xc=4; % Safety distance
a=2; % Sensitivity
inflow1=4; % Inflow of cars on lane 1-distance between new car and next car,
in meters
inflow2=8; % Inflow of cars on lane 2

%% Control parameters
deltaT=.1; % timestep
n_time_steps=3000; % number of time steps
t=0:deltaT:n_time_steps*deltaT;

%to calculate density
delx=10;
x=0:delx:L;
nx=L/delx+1;
%Pre allocating
density1=zeros(nx,1);
density2=zeros(nx,1);

n1=L/2/inflow1+1; % number of cars in lane 1 (right lane)
n2=L/2/inflow2+1; % number of cars in lane 2 (left lane)

%% Initial conditions
%Initial position of cars
x1=0:inflow1:L/2;
x1=[x1' x1']; %lane 1 cars-Right lane
x2=0:inflow2:L/2; %lane 2 cars-Left lane
x2=[x2' x2'];

%Initial velocity of the cars
v1=zeros(n1,2); v1(n1,1:2)=vmax;
v2=zeros(n2,2); v2(n2,1:2)=vmax;

color1=zeros(n1,1); % For animation
color2=ones(n2,1);

timcount=0;
flowcount=0;

%% Time loop
for k=2:n_time_steps
```

```matlab
    count1=0; % no of cars which changed lane 1
    count2=0; % no of cars which changed lane 2

    timcount=timcount+1;

%% Optimal velocity model to lane 1 cars
    for i=1:n1-1
        delX=x1(i+1,k)-x1(i,k); % Headway of car i
        Vopt=vmax/2*(tanh(delX-Xc)+tanh(Xc)); % Optimal velocity
        x1(i,k+1)=(a*Vopt*deltaT^2  +  x1(i,k)*(2+a*deltaT)  -  x1(i,k-1))  /
(1+a*deltaT); %position
        v1(i,k+1)=(x1(i,k+1) - x1(i,k))/deltaT; %Velocity

        %Accident at x0=300m
        if((x0-Xc)<x1(i,k+1) && x1(i,k+1)<x0)
            x1(i,k+1)=x1(i,k);%Car stops, v1(i,k+1)=0
        end
    end

    %Accident at x0=300m(condition check for the 'n1' th car)
    if((x0-Xc)<x1(n1,k) && x1(n1,k)<x0)
        x1(n1,k+1)=x1(n1,k);%Car stops, v1(n1,k+1)=0
    else
        x1(n1,k+1)=x1(n1,k)+vmax*deltaT;
        v1(n1,k+1)=vmax;
    end

    %Inflow of cars
    if(x1(1,k+1)>inflow1)
       n1=n1+1;
       for j=n1:-1:2
           x1(j,k+1)=x1(j-1,k+1);
           v1(j,k+1)=v1(j-1,k+1);
           x1(j,k)=x1(j-1,k);  % initial condtions
           v1(j,k)=v1(j-1,k);
           color1(j,1)=color1(j-1,1);
       end
       x1(1,k+1)=0; %adding new car
       v1(1,k+1)=vmax;
       x1(1,k)=0;
       v1(1,k)=vmax;
       color(1,1)=0;
    end

%% Optimal velocity model to lane 2 cars
    for i=1:n2-1
        delX=x2(i+1,k)-x2(i,k); % Headway of car i
        Vopt=vmax/2*(tanh(delX-Xc)+tanh(Xc));
        x2(i,k+1)=(a*Vopt*deltaT^2  +  x2(i,k)*(2+a*deltaT)  -  x2(i,k-1))  /
(1+a*deltaT); %position
        v2(i,k+1)=(x2(i,k+1) - x2(i,k))/deltaT; %Velocity
    end

    x2(n2,k+1)=x2(n2,k)+vmax*deltaT;
    v2(n2,k+1)=vmax;

    %Inflow of cars
```

```
    if(x2(1,k+1)>inflow2)
       n2=n2+1;
       for j=n2:-1:2
           x2(j,k+1)=x2(j-1,k+1);
           v2(j,k+1)=v2(j-1,k+1);
           x2(j,k)=x2(j-1,k);
           v2(j,k)=v2(j-1,k);
           color2(j,1)=color2(j-1,1);
       end
       x2(1,k+1)=0; %adding new car
       v2(1,k+1)=vmax;
       x2(1,k)=0;
       v2(1,k)=vmax;
       color2(1,1)=1;
    end

%% lane changing for lane 1(right) cars
    for i=1:n1-1
     if(x1(i,k+1)>100)
        delX=x1(i+1,k+1)-x1(i,k+1); % Headway of car i

        [headway,ind]=min(abs(x2(:,k+1)-x1(i,k+1))); % Headway between car i
and the car in target lane

        flag1=0;

        if ((x2(ind,k+1)-x1(i,k+1))>0)
            delXf=headway; % Headway between car i and the front car in target
lane

            if(ind==1)
                delXb=10*Xc; %Assigning large value-because there is no car
behind
            else
                delXb=x1(i,k+1)-x2(ind-1,k+1); % Headway between car i and the
back car in target lane
            end
            flag1=1;
        else

            if(ind>=n2)
                delXf=10*Xc; %Assigning large value-because there is no car in
the front
            else
                delXf=x2(ind+1,k+1)-x1(i,k+1); % Headway between car i and the
front car in target lane
            end

            delXb=headway; % Headway between car i and the back car in target
lane

        end


        %Lane changing Rules
```

```
        if  ((v1(i,k+1)>1.02*v1(i+1,k+1)  &&  delX<4*Xc  &&  delXf>2*Xc  &&
delXb>Xc)|| (delX<2*Xc && delXf>delX && delXb>Xc))
            n2=n2+1;  % number of cars on lane 2(n2) is increased by 1 as
lane1 car enters lane2

            if(flag1==1)
                for j=(n2-1):-1:ind
                    x2(j+1,k+1)=x2(j,k+1);  %  Shifting  the  cars  forward-
assigning new positions on lane 2
                    v2(j+1,k+1)=v2(j,k+1);

                    x2(j+1,k)=x2(j,k);  %  this  is  done  as  the  difference
technique needs 2 previous time step values to calculate
                    v2(j+1,k)=v2(j,k);
                    color2(j+1,1)=color2(j,1);
                end

                x2(ind,k+1)=x1(i,k+1); % Assigning a new 'x2'matrix position
to lane 1 car which shifted to lane 2
                v2(ind,k+1)=v1(i,k+1);

                x2(ind,k)=x1(i,k);
                v2(ind,k)=v1(i,k);
                color2(ind,1)=color1(i,1);
            else
                for j=(n2-1):-1:(ind+1)
                    x2(j+1,k+1)=x2(j,k+1);
                    v2(j+1,k+1)=v2(j,k+1);

                    x2(j+1,k)=x2(j,k);
                    v2(j+1,k)=v2(j,k);
                    color2(j+1,1)=color2(j,1);
                end

                x2(ind+1,k+1)=x1(i,k+1);
                v2(ind+1,k+1)=v1(i,k+1);

                x2(ind+1,k)=x1(i,k);
                v2(ind+1,k)=v1(i,k);
                color2(ind+1,1)=color1(i,1);
            end


            for j=i:(n1-1)
                x1(j,k+1)=x1(j+1,k+1);   % Assigning new position to cars in
lane 1
                v1(j,k+1)=v1(j+1,k+1);

                x1(j,k)=x1(j+1,k);
                v1(j,k)=v1(j+1,k);
                color1(j,1)=color1(j+1,1);
            end

            x1(n1,k+1)=NaN;
            n1=n1-1; % Reducing the number of cars on lane 1 by 1.
        end
```

```
    end
    end

%% lane changing for lane 2(left) cars
    for i=1:n2-1
      if(x2(i,k+1)>100)
        delX=x2(i+1,k+1)-x2(i,k+1); % Headway of car i
        [headway,ind]=min(abs((x1(:,k+1)-x2(i,k+1)))); % Headway between car i
and the car in target lane

        flag2=0;

        if ((x1(ind,k+1)-x2(i,k+1))>0)
            delXf=headway; % Headway between car i and the front car in target
lane

            if(ind==1)
                delXb=10*Xc; %Assigning large value-because there is no car
behind
            else
                delXb=x2(i,k+1)-x1(ind-1,k+1); % Headway between car i and the
back car in target lane
            end

            flag2=1;
        else

            if(ind>=n1)
                delXf=10*Xc; %Assigning large value-because there is no car in
the front
            else
                delXf=x1(ind+1,k+1)-x2(i,k+1); % Headway between car i and the
front car in target lane
            end

            delXb=headway; % Headway between car i and the back car in target
lane

        end

        %Lane changing Rules
        if (v2(i,k+1)>1.02*v2(i+1,k+1) && delX<4*Xc && delXf>2*Xc && delXb>Xc
|| (delX<2*Xc && delXf>delX && delXb>Xc))
            n1=n1+1;

            if(flag2==1)
                for j=(n1-1):-1:ind
                    x1(j+1,k+1)=x1(j,k+1);
                    v1(j+1,k+1)=v1(j,k+1);

                    x1(j+1,k)=x1(j,k);
                    v1(j+1,k)=v1(j,k);
                    color1(j+1,1)=color1(j,1);
                end
```

```
                x1(ind,k+1)=x2(i,k+1);
                v1(ind,k+1)=v2(i,k+1);

                x1(ind,k)=x2(i,k);
                v1(ind,k)=v2(i,k);
                color1(ind,1)=color2(i,1);
            else
                for j=(n1-1):-1:(ind+1)
                    x1(j+1,k+1)=x1(j,k+1);
                    v1(j+1,k+1)=v1(j,k+1);

                    x1(j+1,k)=x1(j,k);
                    v1(j+1,k)=v1(j,k);
                    color1(j+1,1)=color1(j,1);
                end

                x1(ind+1,k+1)=x2(i,k+1);
                v1(ind+1,k+1)=v2(i,k+1);

                x1(ind+1,k)=x2(i,k);
                v1(ind+1,k)=v2(i,k);
                color1(ind+1,1)=color2(i,1);
            end

            for j=i:(n2-1)
                x2(j,k+1)=x2(j+1,k+1);
                v2(j,k+1)=v2(j+1,k+1);

                x2(j,k)=x2(j+1,k);
                v2(j,k)=v2(j+1,k);
                color2(j,1)=color2(j+1,1);
            end

            x2(n2,k+1)=NaN;
            n2=n2-1;
        end
      end
    end

%% Animation
    N=L;
    y1=0;
    y2=0.1;
        clf; hold on;
        plot(0:N, y1*(0:N), 'Color', [.75  .75 .75], 'LineWidth', 50)
        hold on;
        plot([0,N] ,[y2,y2] ,'Color', [.75 .75 .75], 'LineWidth', 50)
        hold on;
        plot([0,N] ,[y1,y1],'--w')
        hold on;
        plot([0,N] ,[y2,y2],'--w')
        hold on;
        plot([0,N] ,[0.05,0.05],'w','LineWidth', 1)
        hold on;
        plot([0,N] ,[-0.05,-0.05],'k','LineWidth', 2)
        hold on;
        plot([0,N] ,[0.15,0.15],'k','LineWidth', 2)
```

```
        xlim([0 N])
        ylim([-N/800 0.25+N/800])

        for i=1:n1
            draw_car(x1(i,k+1), y1, 4,1,color1(i));
        end
        for i=1:n2
            draw_car(x2(i,k+1), y2, 4,1,color2(i));
        end
        pause(.01)

%% Determining density
    for i=2:nx-1
        count_cars=0;
        for j=1:n1
            if((i*delx-delx/2)<x1(j,k+1) && x1(j,k+1)<(i*delx+delx/2))
                count_cars=count_cars+1;    %  counting  cars  present  on  the
region considered above
            end
        end
        density1(i,k+1)=count_cars/delx; % number of cars per unit decameter

        count_cars=0;
        for j=1:n2
            if((i*delx-delx/2)<x2(j,k+1) && x2(j,k+1)<(i*delx+delx/2))
                count_cars=count_cars+1;
            end
        end
        density2(i,k+1)=count_cars/delx;
    end

end  % End of TIME loop

%% Saving the data

save('position_1','x1','-ASCII');
save('position_2','x2','-ASCII');
save('velocity_1','v1','-ASCII');
save('velocity_2','v2','-ASCII');
save('Density_1','density1','-ASCII');
save('Density_2','density2','-ASCII');

%% Plots
figure; surf(density1(:,1:10:n_time_steps),'EdgeColor','none');
view(0,90); colorbar;
set(gca,'FontSize',16);
xlabel('time-in  seconds');  ylabel('location-  in  decameters');  title('Density
on lane 1');

figure; surf(density2(:,1:10:n_time_steps),'EdgeColor','none');
view(0,90); colorbar;
set(gca,'FontSize',16);
xlabel('time-in  seconds');  ylabel('location-  in  decameters');  title('Density
on lane 2');

figure; surf(v1(:,1:10:n_time_steps),'EdgeColor','none');
```

```
view(0,90); colorbar;
set(gca,'ZDir','rev','FontSize',16);  %to reverse the z-scale
xlabel('time-in seconds'); ylabel('position of cars'); title('Velocity on lane
1');

figure; surf(v2(:,1:10:n_time_steps),'EdgeColor','none');
view(0,90); colorbar;
set(gca,'ZDir','rev','FontSize',16);
xlabel('time-in seconds'); ylabel('position of cars'); title('Velocity on lane
2');
```

Function for drawing cars [2]

```
function draw_car(x0, y0, w, h,c);
% This function is drawing a car
% INPUT:
%   x0, y0: Central point of the car
%   w: Width of the car
%   h: Height of the car


% Hold the graphics
hold on


% Define the coordinates for the car chassi
chassi_x = [w/2 w/2 2*w/6 w/6 -2*w/6 -w/2 -w/2];
chassi_y = [0 h/2 h/2 h h h/2 0]*0.01;


% Define the coordinates for the wheels
angles = 0:0.1:(2*pi);
r = sqrt(w*w + h*h)/1000;
wheel_x = r*cos(angles)*100;
wheel_y = r*sin(angles);


% Draw the car!
if(c==1)
    patch(x0+chassi_x, y0+chassi_y, 'r')
end
if(c==0)
     patch(x0+chassi_x, y0+chassi_y, 'y')
end
if(c==2)
     patch(x0+chassi_x, y0+chassi_y, 'g')
end
patch(x0-w/4+wheel_x, y0+wheel_y, 'k')
patch(x0+w/4+wheel_x, y0+wheel_y, 'k')
```