REPORT ON TRAFFIC FLOW MODELING

IN 332

COMPUTATIONAL SCIENCE AND ENGINEERING

Submitted by:

Kishore Rathnavel

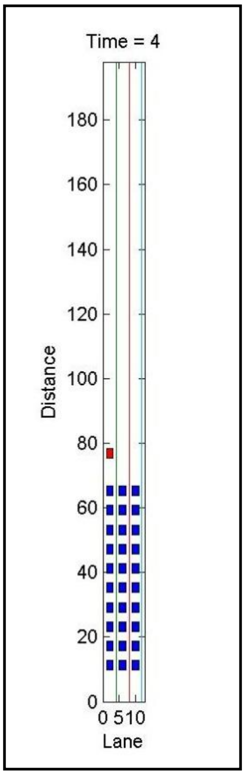
0800215

Problem Statement:

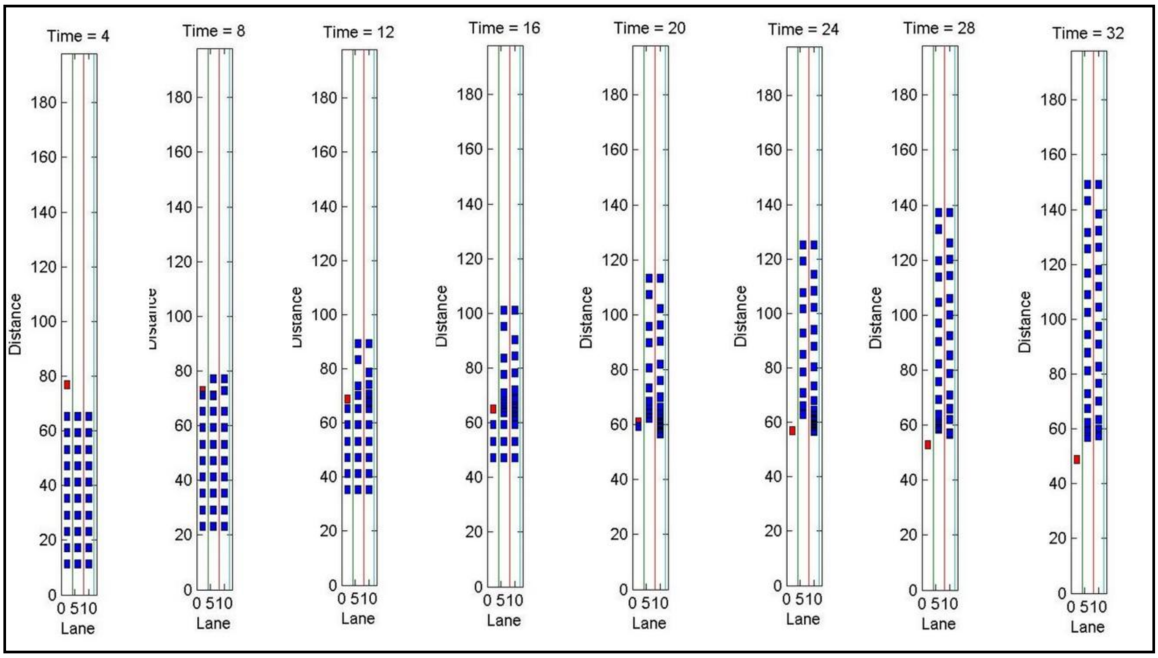
Consider traffic flow on the Visat-Gandhinagar Highway segment in front of the temporary campus of IIT Gandhinagar. On the basis of your assumptions with regard to typical traffic conditions on this highway, model the effects of the following scenarios on traffic flow on this road using TRAFLOW
The illegal sporadic traffic that often runs opposite in direction to the legal traffic direction within the same lane.

Assumptions:

- The Visat Gandhinagar Highway Segment consists of 3 lanes.
- The cars initially are in a orderly arrangement as shown alongside:
- There is only one illegal source of sporadic traffic. Here, this is indicated by the red blip.
- Each blip means a car. The red blip represents the car that is illegally moving in the opposite direction. The blue blips represent the cars that are moving in the proper direction.
- It has been assumed that the minimum distance between any 2 vehicles is supposed to be 2 meters.
- It has been assumed that the highest critical velocity is 3 metres per second which converts to 10.8 km/hr which is a reasonable assumption given that the traffic scenario seems pretty crowded.
- It has been assumed that the acceleration/deceleration occur at 0.5meters-per-second-square.
- The starting velocities of the cars are each 2 meters/second. This means that initially, they are allowed to accelerate and come to the critical velocity which is reflected in some of the velocity vs time graphs.
- The car is assumed to be 3 meters long and 2 meters wide.
- The lane is assumed to be 4 meters wide.



The traffic scenario during the course of the simulation varies as the images shows:



Notice the high concentration of traffic near the region of the illegally moving car and notice how the traffic causes the entire chain of cars to slow down. In the simulation, it can be clearly seen that a few cars some to a standstill during this shift of cars from the leftmost lane to the right lanes.

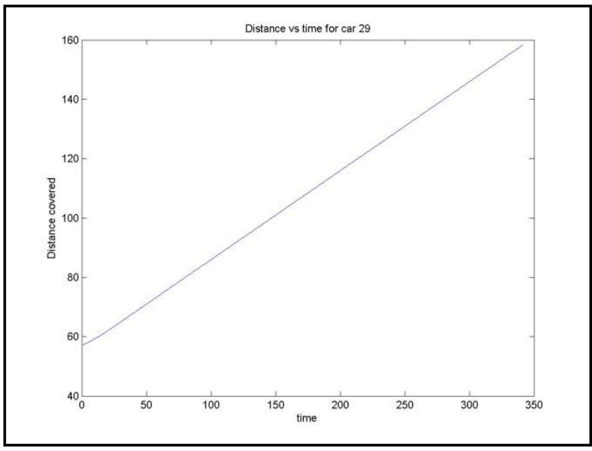
For sake of reference, the cars have been numbered from bottom to top, left to right as 1, 2, 3 … and so on. The position of the car in the Y direction and the velocity of the cars have been recorded.

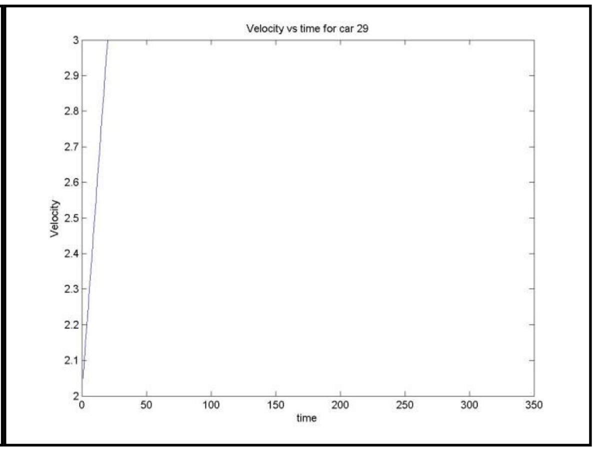
Let us observe the position vs time graph and the velocity vs time graph for a select few cars:

Please note that the rest of the graphs for all the cars can be obtained by running the MATLAB code attached at the end of this report.

CAR 29:

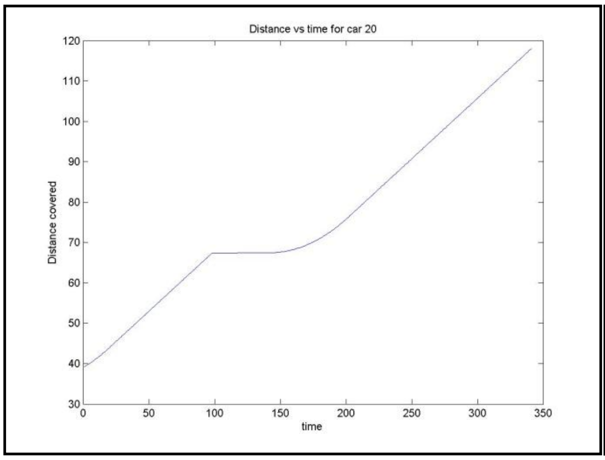
This is the car in the middle lane at the front most. It obviously faces no obstacles and this is reflected in the graph as a linear increment of its position.

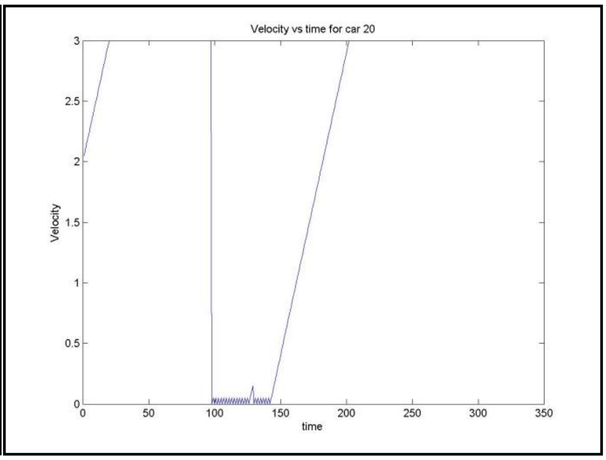




CAR 20:

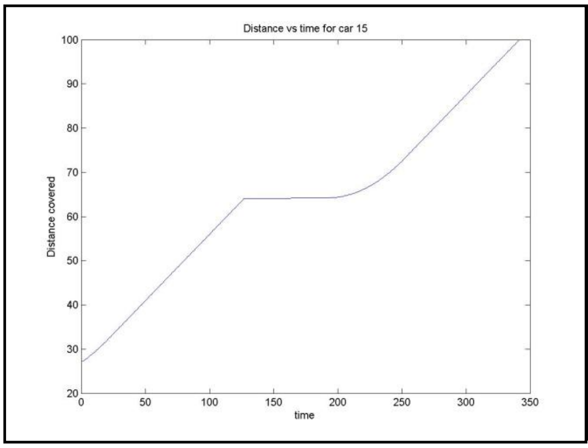
This car is the right car at the 4th row from the top. It has some time interval during which it comes to a standstill and then resumes again from there.

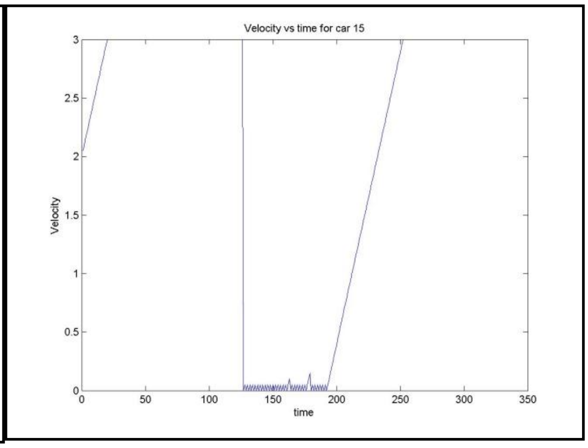




CAR 15:

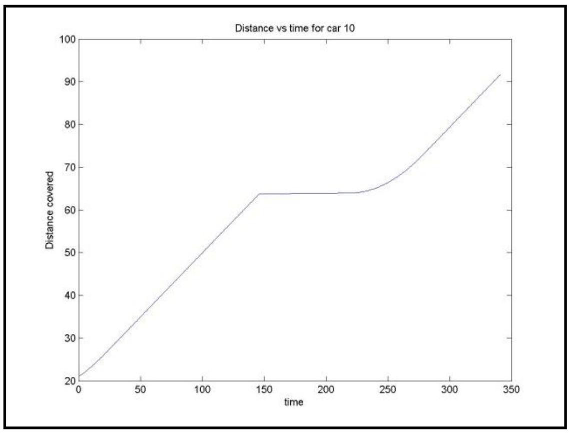
This car is much lower in the order compared to CAR 20 and hence, it has a higher stop time which can be easily seen in the graph.

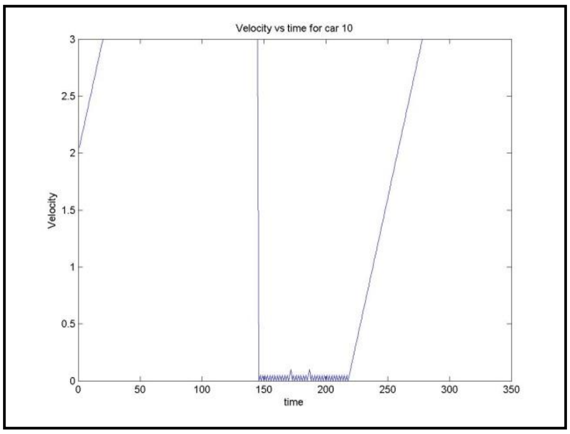




CAR 10:

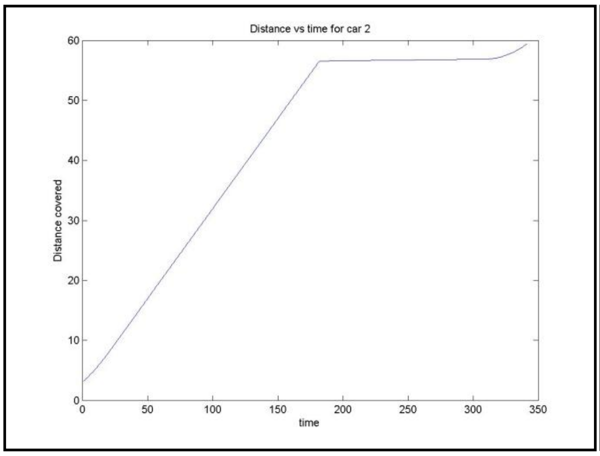
As we can see, the stop time of a car increases as we move to the back of the order. And the distance covered by the cars is obviously decreasing.

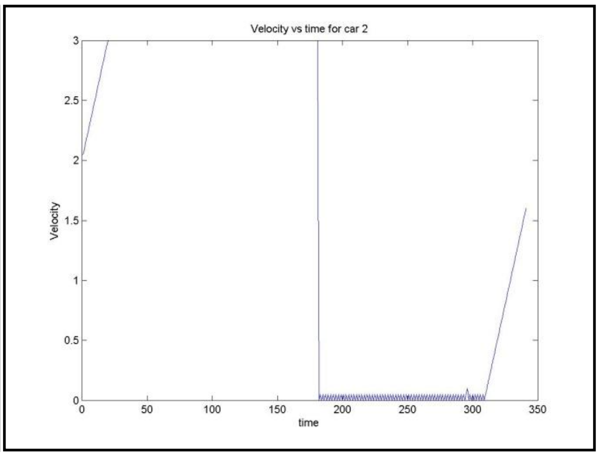




CAR 2:

This car is in the last row middle lane and it is observed for that it has a large time interval during which it is in a standstill condition.





Algorithm for performing this simulation:

1. Start
2. Iterate through the 30 cars on road
3. For car i, assume conditions for lane change and deceleration to be false
4. Update positions of the car i
5. If the car i is in the first lane, and if the illegal traffic is lesser than the minimum distance
   5.1 Change lane
   5.2 Decelerate
6. Find other cars in the present lane
   6.1. Find out the cars whose displacement is larger than car i
      6.1.1. Pick the lease displacement of the selected cars(this is the car that is immediately infront of the present car)
         6.1.1.1. Check if the distance between the selected car and car i is less than the minimum
            6.1.1.1.1. If yes,
               6.1.1.1.1.1. Change lane
               6.1.1.1.1.2. Decelerate
7. Save position and velocities of all the cars
8. Exit

MATLAB CODE:

```
clear all;
clc;

%% Properties of the road
RoadLength=200;
Width=12;
LaneWidth=4;
NumberofLanes=Width/LaneWidth;
UnitRoadLength=6;
RoadResolution=RoadLength/UnitRoadLength;

x=0:LaneWidth:Width;
y=0:UnitRoadLength:RoadLength;
[X Y] = meshgrid(x,y);
plot(X,Y);
daspect([1 1 1]);
hold on;

%% Properties of the car
CarWidth=2;
CarLength=3;
CarX=zeros(30,1);
CarY=zeros(30,1);

%% Initializing the car positions
for i=1:30
```

```
    if(mod(i,3)==1)
        CarX(i,1)=2;
    elseif(mod(i,3)==2)
        CarX(i,1)=6;
    else
        CarX(i,1)=10;
    end

    CarY(i,1)=3+6*floor((i-1)/3);
end

%% Displaying how the initial positions of the cars look
[numberofcars nothing]=size(CarX);%nothing = junk value
for i=1:numberofcars
    Xdata = [CarX(i)-
CarWidth/2,CarX(i)+CarWidth/2,CarX(i)+CarWidth/2,CarX(i)-CarWidth/2]';
    Ydata = [CarY(i)-CarLength/2,CarY(i)-
CarLength/2,CarY(i)+CarLength/2,CarY(i)+CarLength/2]';
    Zdata = ones(4,1);
    patch(Xdata,Ydata,Zdata,'g');
    daspect([1 1 1]);
    i
%     pause(0.5);
    hold on;
end

%% Assume initial velocities
CarXvel=zeros(30,1);%assume 0 velocity in the x direction
CarYvel=2*ones(30,1);%assume starting velocity to be 2meters/sec in y
direcion

%% Define a car coming in the reverse direction

ReverseCarX = 2;%assuming that the illegal traffic flows in lane1
ReverseCarY= 80;%assuming starting position of the illegal traffic
ReverseCarXvel=0;
ReverseCarYvel=-1;%the velocity of the illegal traffic is negative as it is
moving in the oppostite direction
accelerationFlag=1;%used to indicate if the velocity is to be increased or
decreased
min_Dist=2;%minimum distance between 2 vehicles taken to be 2 meters
acceleration=0.5;%assume that there is either an acceleration or
deceleration of 0.5meteres/sec squared
sze=(35-1)/0.1%size of the time frame
SaveCarXPos=zeros(30,sze);%save values of distance and velocity in the
following 3 matrices
SaveCarYPos=zeros(30,sze);
SaveCarYvel=zeros(30,sze);
count=1;
for t=1:0.1:35
    clf;

    x=0:LaneWidth:Width;
    y=0:UnitRoadLength:RoadLength;
    [X Y] = meshgrid(x,y);
    plot(X,Y);
    xlabel('Lane');
    ylabel('Distance');
    str=sprintf('Time = %d',t);
    title(str);
```

```
    axis([0 13 0 Inf]);
    daspect([1 1 1]);
    hold on;

    ReverseCarX=ReverseCarX+ReverseCarXvel*0.1;
    ReverseCarY=ReverseCarY+ReverseCarYvel*0.1;

    Xdata = [ReverseCarX-
CarWidth/2,ReverseCarX+CarWidth/2,ReverseCarX+CarWidth/2,ReverseCarX-
CarWidth/2]';
    Ydata = [ReverseCarY-CarLength/2,ReverseCarY-
CarLength/2,ReverseCarY+CarLength/2,ReverseCarY+CarLength/2]';
    Zdata = ones(4,1);

    patch(Xdata,Ydata,Zdata,'r');
    daspect([1 1 1]);

%     pause(0.5);
    hold on;

    for i=1:1:30

        accelerationFlag=1;%initially assume that the car should accelerate
away
        LaneChange=0;%assume that Lane change is not required
        CarX(i,1)=CarX(i,1)+CarXvel(i,1)*0.1;%modify the positions
        CarY(i,1)=CarY(i,1)+CarYvel(i,1)*0.1;



        if(CarX(i,1)==ReverseCarX)%This means that we are bothered only abt
the cars in the lane in which the reverse car is comming
            if(ReverseCarY-CarY(i,1)<=min_Dist)%check to avoid collision
                 LaneChange=1;%Change lane because the min distance
condition is violated
                accelerationFlag=-1;%also reduce speed
            end
        end



        PresentLaneCars=find(CarX==CarX(i,1));%find out cars in the present
lane
        PresentLaneCarsY=zeros(size(PresentLaneCars,1),1);
        for j=1:size(PresentLaneCars,1);
            PresentLaneCarsY(j,1)=CarY(PresentLaneCars(j),1);%find out Y
positions of cars in present lane
        end

         AboveLaneCars=find(PresentLaneCarsY>CarY(i,1));%we are interested
in cars which are ahead of the present car
         AboveCarsY=zeros(size(AboveLaneCars,1),1);
        for j=1:size(AboveLaneCars,1)
            AboveCarsY(j,1)=PresentLaneCarsY(AboveLaneCars(j));%find out Y
positions of cars in present lane
        end
         CompareCarY=min(AboveCarsY);%get the minimum Y postion. which is
essentially the car immediately infront of the present car.
         if(CompareCarY-CarY(i,1)<=min_Dist)%check if the distance between
the car immediately ahead is less than minimum distance
            if(CarYvel(i,1)>0);
```

```
            LaneChange=1;
            accelerationFlag=-1;%should the distance be less than min_Dist, 
decellerate
            end
        end

        %some gimmicks for showing blips moving on the screen
        Xdata = [CarX(i)-
CarWidth/2,CarX(i)+CarWidth/2,CarX(i)+CarWidth/2,CarX(i)-CarWidth/2]';
        Ydata = [CarY(i)-CarLength/2,CarY(i)-
CarLength/2,CarY(i)+CarLength/2,CarY(i)+CarLength/2]';
        Zdata = ones(4,1);
        patch(Xdata,Ydata,Zdata,'b');
        daspect([1 1 1]);
        hold on;

        CarYvel(i,1)=CarYvel(i,1)+accelerationFlag*acceleration*0.1;%using 
the v=u+at formula of kinematics

        if(CarYvel(i,1)<0)
            CarYvel(i,1)=0;%Critical low velocity
        end

        if(CarYvel(i,1)>3)%critical high velocity is 3m/s
            CarYvel(i,1)=3;
        end


        if LaneChange==1%execute this section if LaneChange has 
specifically been changed to 1
            if CarX(i,1)==2%car in lane 1 changes to lane 2
                CarX(i,1)=6;
            elseif CarX(i,1)==6%car in lane 2 changes to lane 3
                    CarX(i,1)=10;
            elseif CarX(i,1)==10% car in lane 3 changes to lane 2
                CarX(i,1)=6;
            end
            CarYvel(i,1)=0;
        end



    end
    SaveCarXPos(:,count)=CarX(:,1);%save the position profile of all the 
cars at the time t
    SaveCarYPos(:,count)=CarY(:,1);
    SaveCarYvel(:,count)=CarYvel(:,1);%save the velocity profile of all the 
cars at time t
    count=count+1;%increment the counter for the matrices that save the 
displacement and velocity profiles

    pause(0.1);

    if(mod(t,4)==0)
        str=sprintf('View at time = %d.jpg',t);
        saveas(gcf,str);
    end


end
```

```
%% Plotting distance and velocities


for i=1:30
    plot(SaveCarYPos(i,:)');
    xlabel('time');
    ylabel('Distance covered');
    str=sprintf('Distance vs time for car %d',i);
    title(str)
    str=sprintf('Distance vs time for car %d.jpeg',i);
    saveas(gcf,str);

    plot(SaveCarYvel(i,:)');
    xlabel('time');
    ylabel('Velocity');
    str=sprintf('Velocity vs time for car %d',i);
    title(str)
    str=sprintf('Velocity vs time for car %d.jpeg',i);
    saveas(gcf,str);
end
```