Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Project Report

Smart Driving Strategies

Luzian Hug & Colin Berner

Zürich
18.12.2016

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Smart Driving Strategies

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Berner	Colin
Hug	Luzian

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Bülach, 18.12.2016

Signature(s)
C. Berner

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

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.

Luzian Hug

Colin Berner

Table of Contents

1 Abstract 1
2 Individual contributions 1
3 Introduction and Motivations 1
4 Description of the Model 2
4.1 Ordinary Driver Modelling 2
4.2 Guide Car Modelling 3
5 Implementation 4
5.1 Slowing down a single car. 4
5.2 Many disruptions at different places in the line 5
5.3 Random starting speeds 5
6 Simulation Results and Discussion 5
6.1 Start and open road 5
6.2 Slowing down a single car 6
6.3 Many disruptions at different places in the line 10
6.4 Random starting speeds 12
7 Summary and Outlook 15
8 References 15
9 Appendix: MATLAB Code 16
9.1 Implementation of the IDM 16
9.2 Simulation with constant starting speed 18
9.3 Simulation with Random starting speed 20
9.4 Generation of Disturbance Matrix 21

1 Abstract

In this paper, a section of a single lane, interchange free highway is simulated and tested in its resistance to different disturbances in traffic flow. We introduce guide cars, which have an augmented perception of the road situation and know how to react to a disturbance. The guide cars look further ahead than a human driver would in order to detect a deceleration of the leading drivers, allowing them to brake earlier but less hard than a human would. The research aim of this project was to determine whether such guide cars have a viable impact in congestion mitigation. Such a guide car could be realized by using autonomous cars.

The human drivers were modelled using the intelligent driver model by Martin Treiber, whereas the guide cars are modelled by an extension of this model that allows them to communicate with the leading guide car. This model was then tested with respect to guide car frequency and disturbance type and frequency.

2 Individual contributions

Most of the simulation work was done in collaboration. Both team members actively contributed their ideas. The MATLAB code and this report were both written in Teamwork.

3 Introduction and Motivations

The human population of our planet is growing rapidly. In addition, people are getting wealthier, which means more and more people are able to afford cars. More cars mean more traffic, which needs to be accommodated on our road infrastructure. There is a large number of possible solutions in order to keep traffic flowing and mitigate traffic jams: Building more or larger roads or getting more people to use public transportation would be two possible solutions to the problem. However, one can see that these kinds of solutions only work to a finite extent.

In our eyes, a better idea would be to take a more fundamental approach: changing the way cars behave on the road. An often-made mistake is to brake too early or too hard when seeing the leading driver brake. The following driver will then brake even harder. This leads to a positive feedback effect, resulting that the last driver in a group suffers a much greater time loss than the driver who has originally caused the disturbance. Even a small disruption can cause a phantom traffic jam with no apparent root.

To teach all human drivers to change their driving behaviour would be a tedious and uneconomical undertaking. However, in recent times a possible way to solve this problem has emerged. Autonomous cars may very well become an ordinary occurrence on our roads. The idea in this project is that autonomous cars can be used to prevent such phantom traffic jams.

4 Description of the Model

4.1 Ordinary Driver Modelling

The decision on the Intelligent Driver Model (IDM) is founded on the paper "Delays, inaccuracies and anticipation in microscopic traffic models" by M. Treiber, A. Kesting, and D. Helbing [1]. The authors analyse a number of driving behaviour models of various simplicity. They concluded that "the destabilizing effects of finite reaction times can be compensated to a large extent by spatial and temporal anticipation such that the resulting stability and dynamics are similar to the case of the IDM with zero anticipation and reaction time" [1]. From this and the following text we conclude that the IDM is sufficient for our purposes.

The IDM is a second order non linear dynamical system. It is classified by the following two differential equations [2]:

$$\dot{x}_\alpha = v_\alpha \tag{1}$$

$$\dot{v}_\alpha = a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right] \tag{2}$$

All parameters and functions used are explained in the following.

- α corresponds to the numbering of the cars. The car with the lowest α is the first car.
- Acceleration a, the maximum acceleration a car can have.
- Maximum velocity v_0
- Acceleration exponent δ, usually set to 4
- Desired distance to the leading car $s^*(v_\alpha, \Delta v_\alpha)$
- Actual distance to the leading car $s_\alpha := x_{\alpha-1} - x_\alpha$ [1]
- Speed difference to the leading car $\Delta v_\alpha := v_\alpha - v_{\alpha-1}$

The desired distance to the leading car is defined as follows:

$$s^*(v_\alpha, \Delta v_\alpha) = s_0 + v_\alpha T + \frac{v_\alpha \Delta v_\alpha}{2\sqrt{ab}} \tag{3}$$

With parameters:

- Minimum distance to next car s_0
- Desired time headway T
- Comfortable braking deceleration b

---

[1] Dr. Treiber also includes a car length parameter in s_a. We did not include this parameter as it is not relevant to the simulation.

For even δ, $\dot{v}_a$ is always smaller or equal to a, fulfilling the condition that a is the maximum possible acceleration. As v_a approaches v_0, the term $\left(\frac{v_a}{v0}\right)^\delta$ approaches 1, setting the acceleration to 0. This ensures that v_a never exceeds v_0. The term $\left(\frac{s^*(v_a, \Delta v_a)}{s_a}\right)^2$ is responsible for keeping a distance to the leading car. It takes into account the minimum acceptable distance to the leading car, keeping a time headway and the velocity difference; corresponding to the first, second and third term in equation (3). The maximum possible braking deceleration is not as obvious, but it can be calculated setting $s_a = s_0, v_a = v_0$ and $v_{a-1} = 0$, which yields $-a\left(1 + \frac{v_0 T}{s_0} + \frac{v_0^2}{2\sqrt{ab}}\right)^2$.

By choosing the parameters v_0, s_0, T, a, and b one can recreate real driving behaviour sufficiently accurately. Dr. Treiber provides a suggestion for a reasonable parameter choice [2], which we adjusted to recreate certain situations better. The model has proved to be well suited for our simulation and we were able to produce the expected results.

4.2 Guide Car Modelling

The subject of our project is the introduction of guide cars, randomly distributed among the other drivers. To model the guide cars, we augmented equation (2), so that the car reacts better to a disturbance. A number of different models were tested, such as always keeping the distances to the leading and following cars equal or taking the velocities of the two leading cars into account. We decided on a slightly different approach, namely to look at the velocity of the next guide car. This is also more easily realizable than the other approaches, since one could easily equip the guide cars with a communication system. The alternative equation (2) for the guide cars is

$$\dot{v}_a = \begin{cases} a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2 - \frac{v_a - v_g}{c}\right], if\ s_a < d \\ a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha}\right)^2\right]\ otherwise \end{cases} \tag{4}$$

where v_g denotes the velocity of the next guide car, c is a constant for normalization and d is the so-called trigger distance. In open traffic, meaning the distances between cars are larger than d, the guide model is suboptimal, because the guide cars tend to brake unnecessarily, causing slow downs in traffic flow. For this reason, the trigger distance was introduced, returning the guide car to normal driving behaviour as soon as traffic opens up.

5 Implementation

The implementation and simulation of our traffic model was done entirely in MATLAB. The code is attached at the end of this document. The state space equations discussed above and special conditions (like disruptions) are written in the function idm_final.m. This function is then called by our simulation scripts and solved by MATLAB's ode45.

We exposed this model to different types of disruptions to see the influence of the guide cars. Different types are important to observe if the behaviour changes depending on the type.

By plotting the location of each car versus time we can observe their behaviour. We can also visualise the velocity, but this has proved to be less interesting.

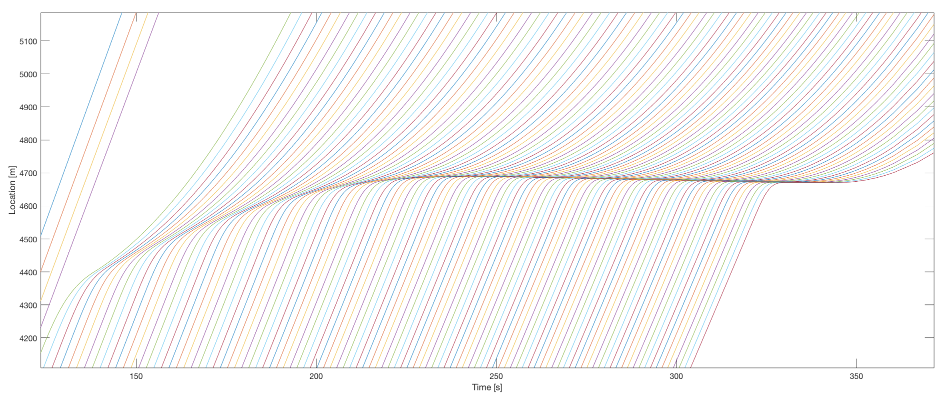

Figure 1: Each line represents one car. Example of a road without guide cars. It can be seen how a congestion can emerge when one car brakes.

5.1 Slowing down a single car.

For the first test we tell one car to slow down for a while and accelerate again to normal speed afterward. This is done by inserting a condition that checks whether the chosen car is in the declared section and sets his speed limit accordingly. We are then interested in how this interruption propagates through the road and how it affects cars far behind the one braking.

5.2 Many disruptions at different places in the line

For this simulation we first generate a random "disruption matrix" that contains the information on when which car will brake how hard and for how long. We then analyse this similarly to the first test by looking at the last car. Fewer but larger simulations of this type were run.

5.3 Random starting speeds

For the third test we randomize the starting speed of every car. This will also lead to congestion, as some cars have to brake immediately because the car in front can be slower.

6 Simulation Results and Discussion

6.1 Start and open road

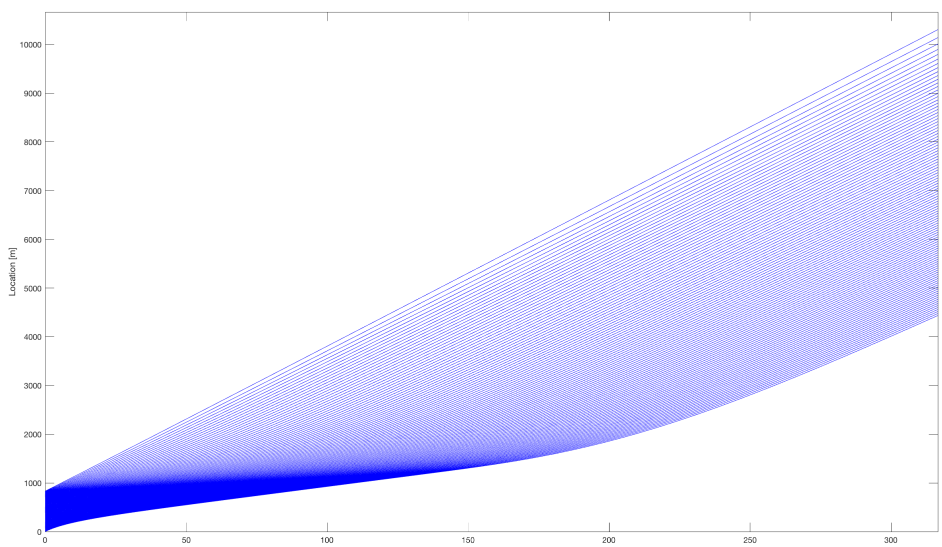

Figure 2: Start from close formation to open road driving without guide cars

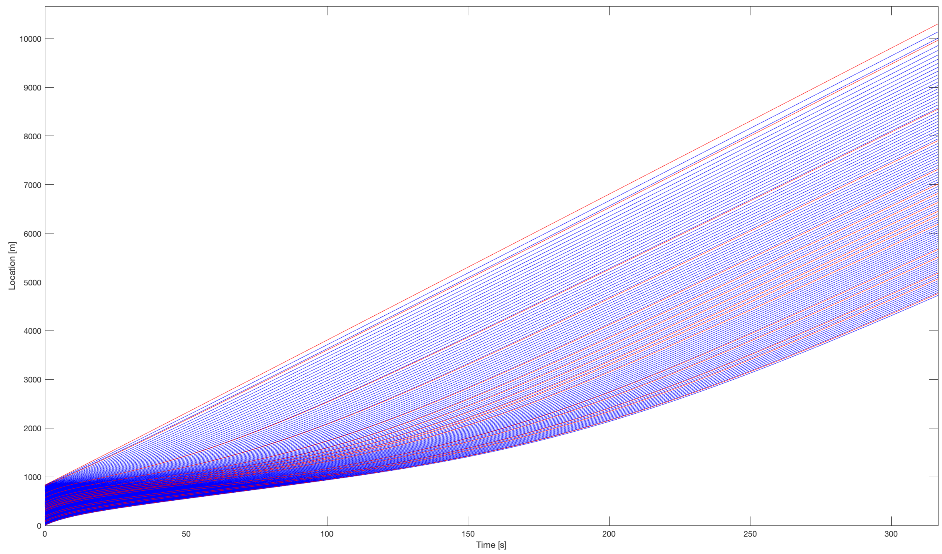

Figure 3: Start from close formation to open road driving with 15% guide cars (red)

When starting from close formation the cars can accelerate successively. When comparing the two Figures we can see that the set with guide cars can drive slightly more compact, but this is mostly because the guide cars will follow the car in front closer than normal cars. All in all we see very similar behaviour.

6.2 Slowing down a single car

When a car brakes more or less abruptly we can start to see how guide cars impact the road.

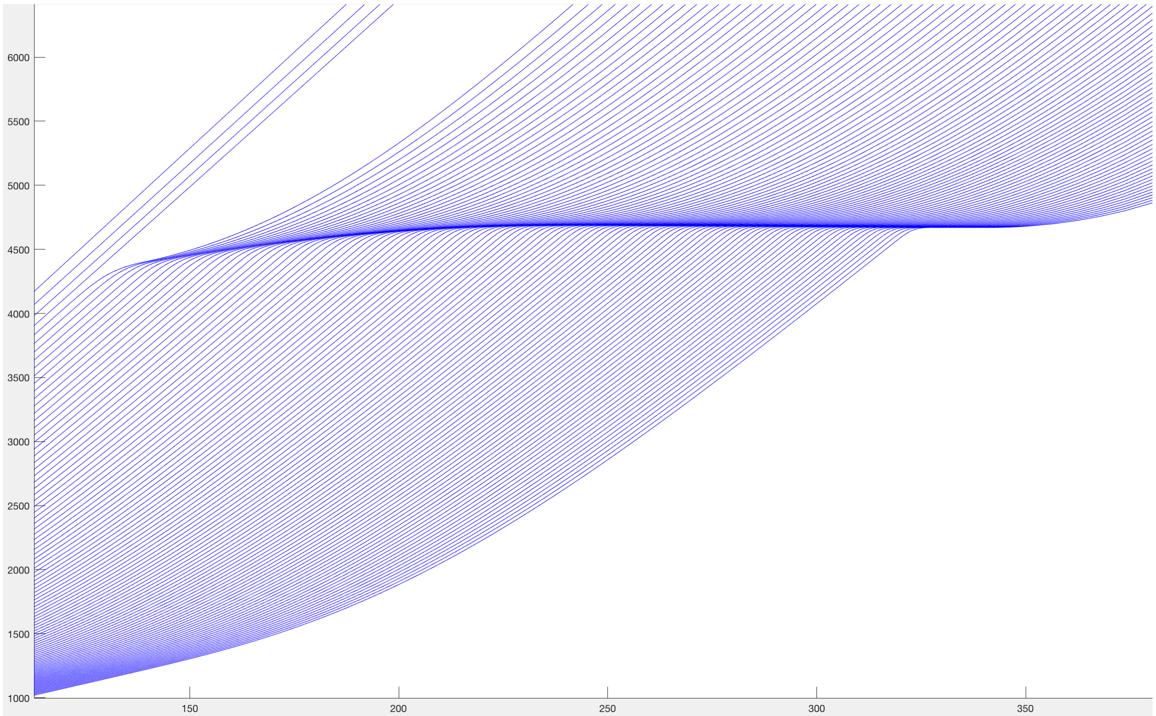

Figure 4: Situation without guide cars

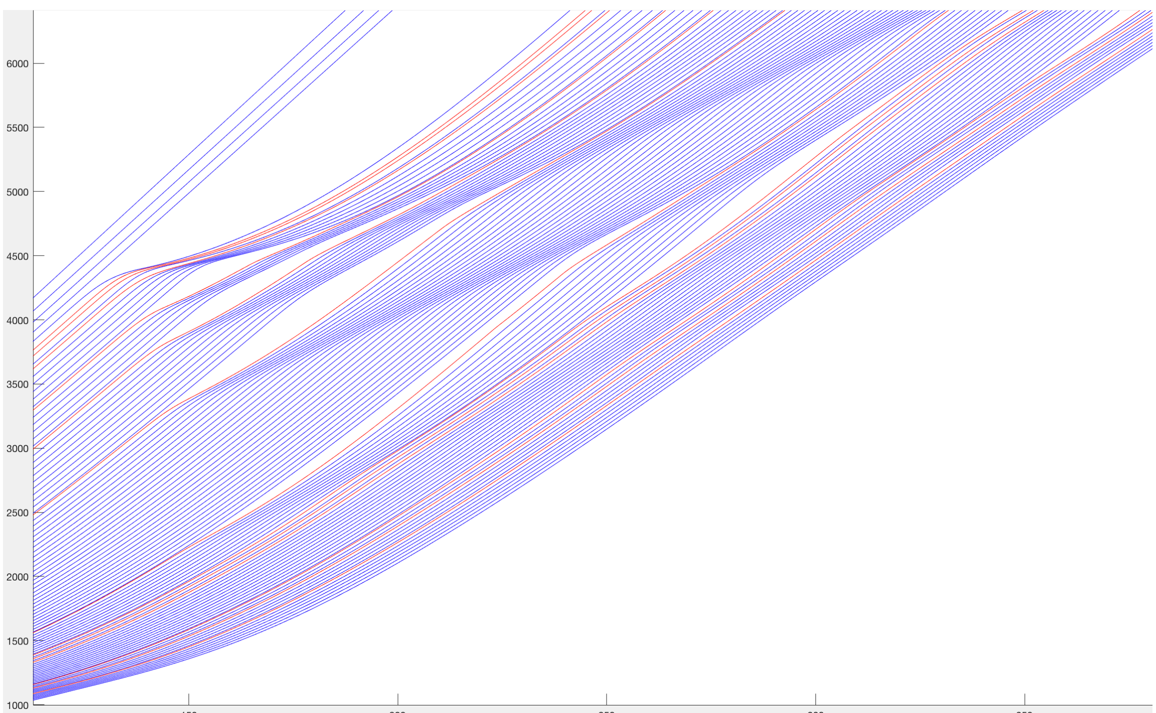

Figure 5: Situation with 13% guide cars (simulated with 10% probability for each car to be a guide car)

In Figure 4 we see that slowing down one car without any guide cars present causes a traffic jam so that posterior cars will come to a complete stop. This happens if the car "hits the brakes" hard enough. For only small changes in speed the disruption will dissolve. The important thing to notice is that the last simulated car is delayed much longer than the initial perpetrator of the congestion. The jam only gets worse the more cars follow.

In Figure 5 we added a percentage of guide cars. We immediately see that a traffic jam is almost completely mitigated. The last simulated car (100 cars behind the perpetrator) will barely be able to notice the interruption. It can also be seen how this comes to be. Some guide cars get caught and have to brake. The following guide cars will notice this sequentially through a change of velocity. This way they can brake earlier and therefore less hard because they access information much further ahead of them.

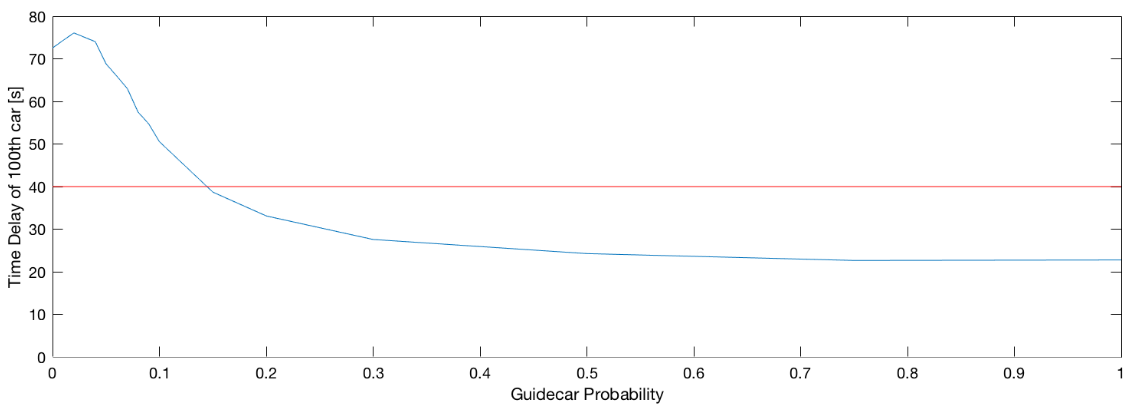

Figure 6: Time delay of last car versus guide car percentage (blue). Time delay of perpetrator as reference (red).

By measuring the time delay the last car experiences relative to an uncongested road we can rate how well the guide cars are doing. Every simulation was run 200 times in an attempt to eliminate the randomness of the guide car distribution. We cannot distribute them evenly because in a real world example we have to assume a more or less random distribution. A general trend is: More guide cars lead to less time delay and are therefore better. At about 15% the two lines intercept. This means that beyond that percentage the 100^{th} car experiences a shorter time delay than the perpetrator.

We cannot go as far as saying that this is the threshold we need for a traffic jam to be mitigated. Firstly this point is very dependant on the form of disruption. Secondly this happens at some lower percentages as well but further down the line (e.g. only for the 200^{th} car).

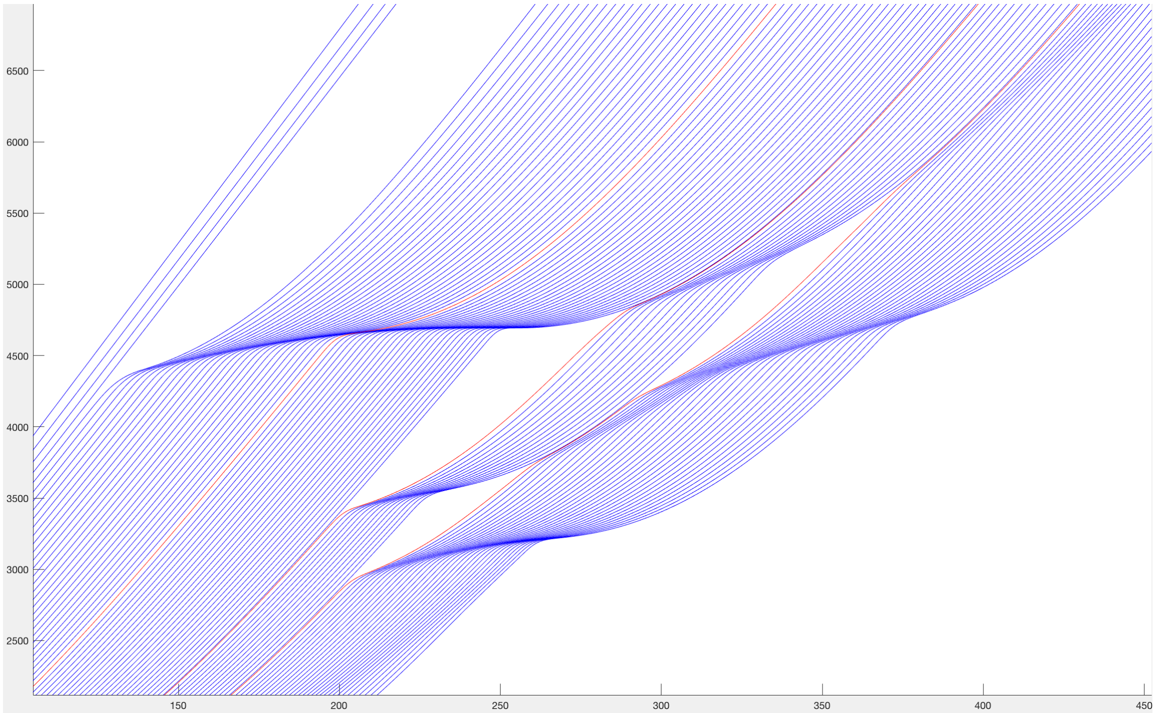

Figure 7: Situation with 3% guide cars (simulated with 2% probability for each car to be a guide car)

An exception to that general trend is the maximum at 2% instead of 0% as expected. If there are too few guide cars they will still brake early but there is no other guide car close enough to back that up so the congestion just gets delayed (in time), which does not help. This can be observed in Figure 7.

6.3 Many disruptions at different places in the line

To analyse this situation we chose a much larger simulation (1000 cars instead of 100) to see better what happens before the last car passes.

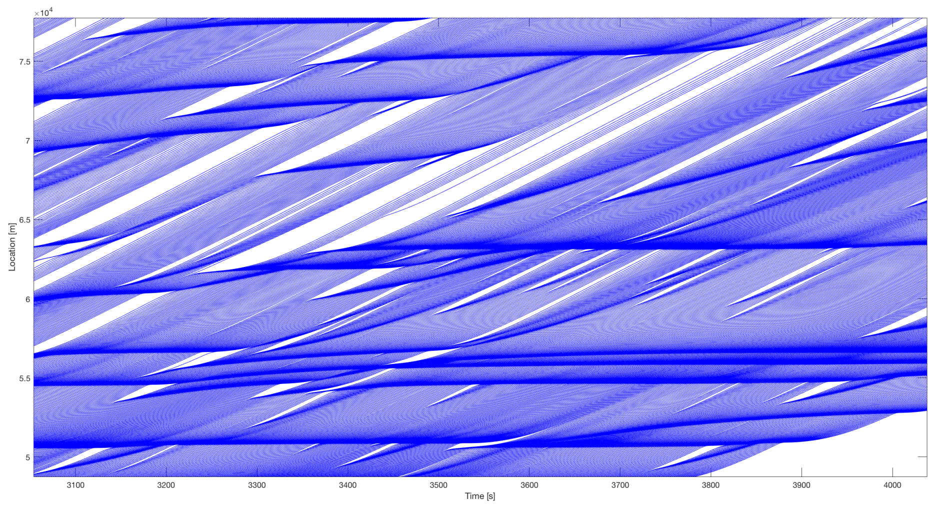

Figure 8: Section of Simulation without guide cars.

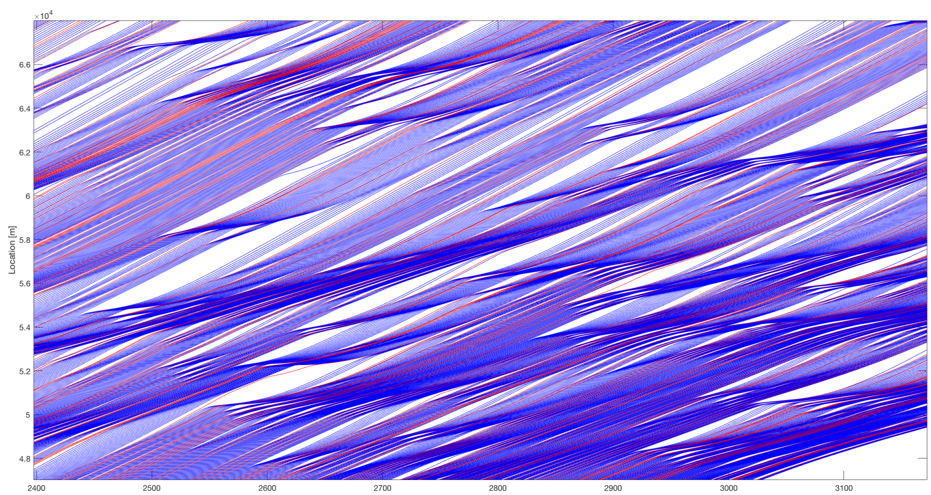

Figure 9: Section of Simulation with 15% guide cars.

In Figure 8 we can observe very similar patterns as in the previous experiment. Congestions propagate through the entire line and only dissolve when no car arrives there for a while due to previous congestions. Also like in the first experiment when we introduce guide cars they help resolving congestions with early intervention (Figure 9).

Due to very long simulation times (and therefore low sample size) we have to be careful with analysing this simulation quantitatively. In Figure 10 we can still see that the last car is delayed less the more guide cars we have. It arrives at the checkpoint 6 minutes earlier when there are 20% guide cars than when there are none.

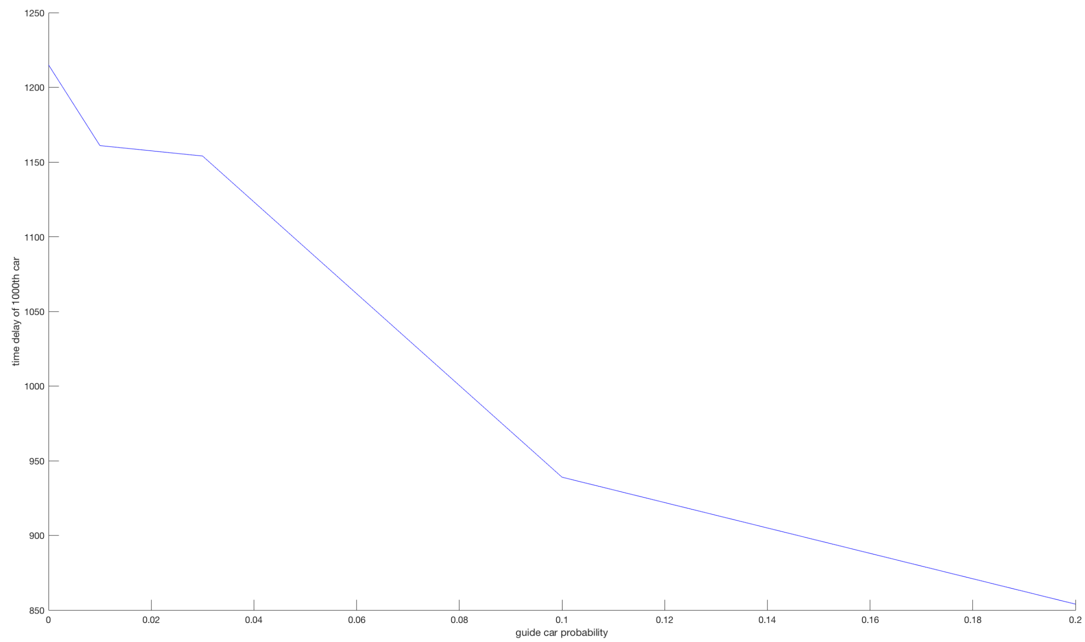

Figure 10: Time delay of last car versus guide car percentage. Please note the offset.

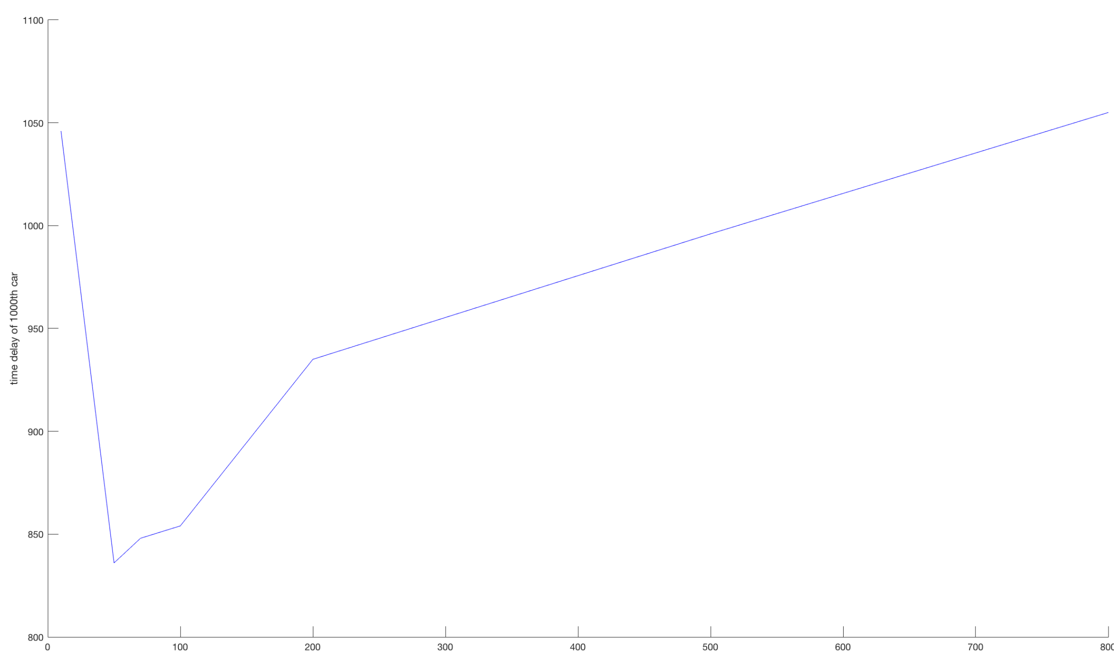

Figure 11: Time delay of last car versus guide car trigger distance [m]. Please note the offset.

For this simulation we also compared different guide car trigger distance values. If there is no car in front of the guide car it should not brake early. For this simulation we find an optimum at around 50 metres. This is different for other types of disruptions, which is why we settled for 100 metres in all the other simulations. What this result tells us is that the performance of guide cars is strongly dependant on the system parameters of the guide cars.

6.4 Random starting speeds

Instead of everyone starting with the same velocity like in the previous simulations we now give each car a different starting speed.

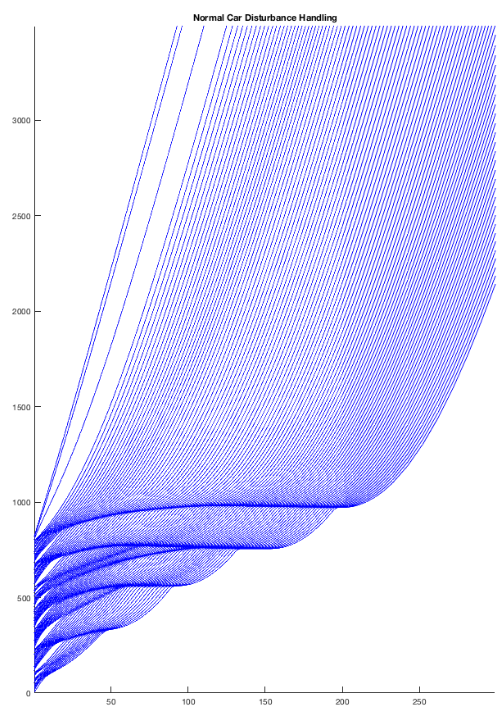

Figure 12: Comparison of all normal cars to all guide cars

For a first interesting result we can look at Figure 12. If there are no guide cars we can again observe traffic jams at different locations on the road. Guide cars handle this problem much better and are very efficient in resolving the congestion. However this is not at all a realistic scenario.

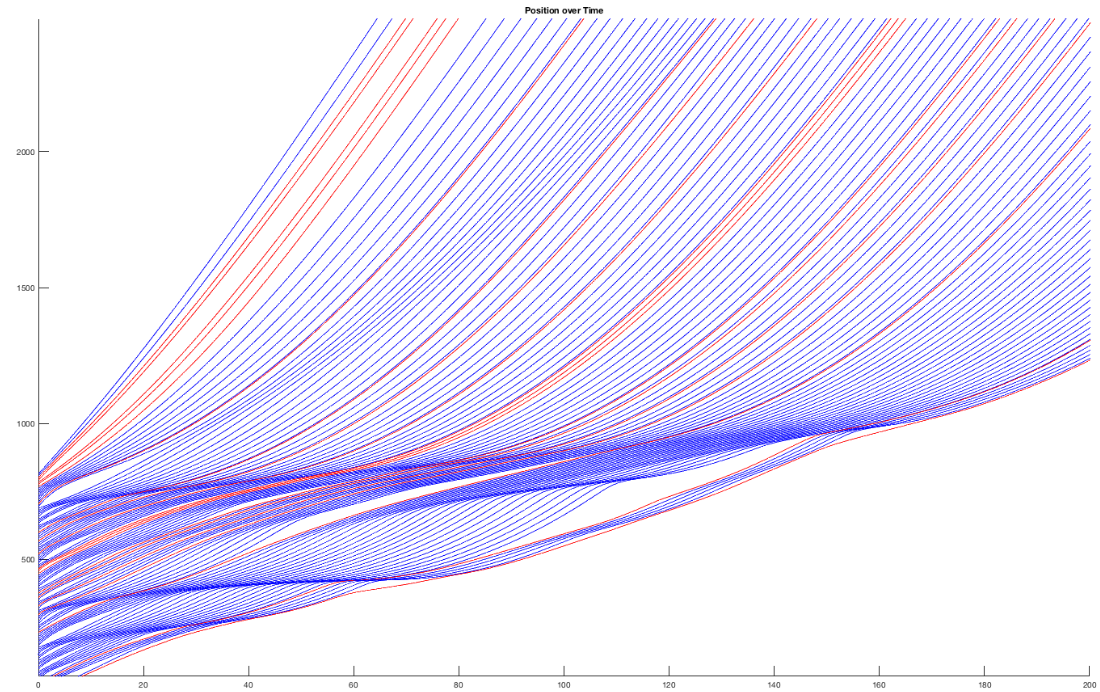

Figure 13: Random starting velocity with 15% guide cars.

Figure 13 shows how guide cars help mitigating traffic jams that occur because of this random speed distribution.

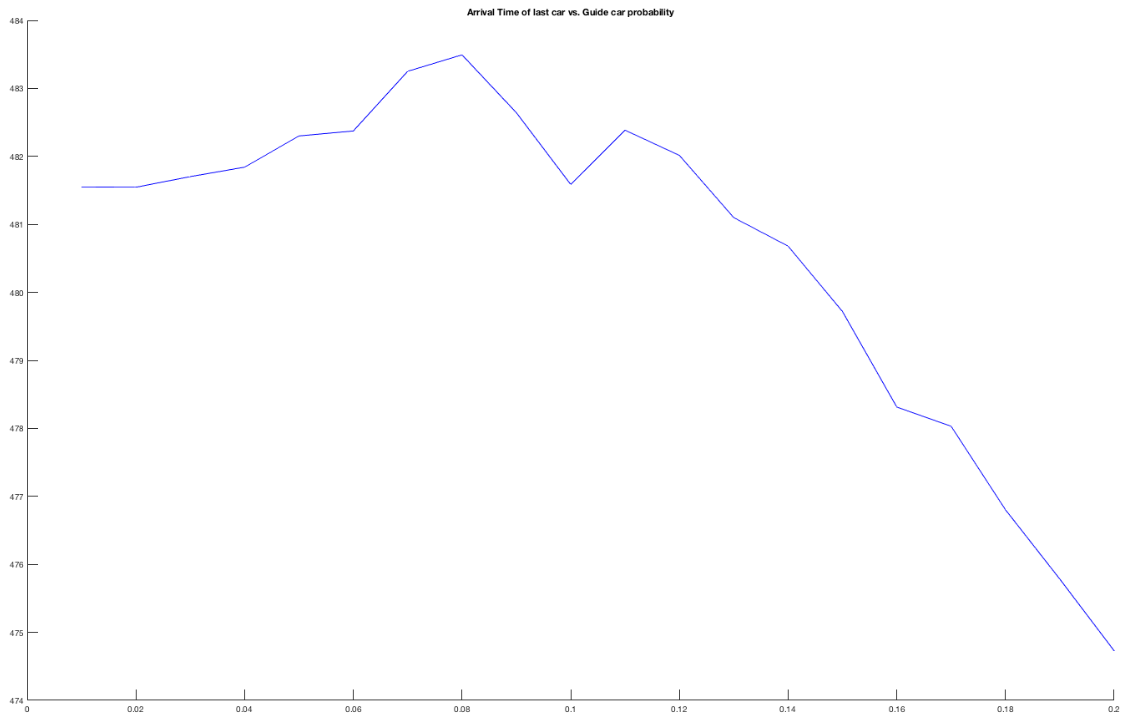

Figure 14: Arrival time of last car at checkpoint versus guide car percentage.

We again see that more guide cars help to resolve the congestion faster, which means that more guide cars are generally better. This simulation has a high variance despite being run 200 times for each sample. This means that the distribution of cars has a high impact on the result of the simulation.

Looking at the results of the three forms of disruptions we analysed, we can say that it is indeed possible to mitigate or help resolve traffic jams with the introduction of a guide car system. In all three simulations we find that in order to get significantly better results we optimally want a guide car percentage of 5 to 10 or even more, which is extremely difficult to achieve in a real world scenario today. However we think that in the near future it could very well be possible to implement such a system, as it does not need for any unknown technology. The guide cars don't need to be fully self driven, an enhanced cruise control and communication system is enough to bring that functionality.

Another result we found is that the results strongly depends on the exact guide car model. This is somewhat expected, but also has to be considered when talking about realisation of this idea.

7 Summary and Outlook

We wanted to find out if it was possible to prevent congestion and traffic jam by introducing a system of guide cars with different driving behaviour and decisions than normal human drivers. Our simulations suggested that this was indeed possible but requires for a decent amount of guide cars. But when this was met we could measure significant improvements in time delays and thus answer our research question positively.

One important thing to keep in mind is that these statements are based solely on simulations. The world can never be perfectly modelled. Some disparities in the results have to be expected when comparing simulation to real world, especially when modelling things like human decisions.

Another observation is the impact of the guide car model and parameters. We chose a model that we found to work well with the tested disruptions. But we are sure this could be more optimised and compared to different guide car models, which could be a topic for further research.

8 References

[1] M. Treiber, A. Kesting, D. Helbing, Delays, inaccuracies and anticipation in microscopic traffic models (2005)

[2] M. Treiber, A. Hennecke, D. Helbing, Congested traffic states in empirical observations and microscopic simulations (2000)

9 Appendix: MATLAB Code

9.1 Implementation of the IDM

Contents

- Parameter definitions
- Memory Allocation
- Application of different models


```
function result = idm_final(t,x,guideMap,disMatrix,triggerDis)
```

```
%IDM function used in simulate.m
```

Parameter definitions

```
%Simulation Parameters
L = 1000000; %Length of the highway in m
lcar = 0; %Length of the cars in m
%Model Parameters
%desired speed in free traffic
s0 = 0.5; %minimum distance to next car
T = 1; %desired time headway to vehicle in front
a = 0.3; %maximum acceleration of a car
b = 3; %comfortable braking deceleration
delta = 4; %exponent used in equation
sStar = @(va,dva,dva1) s0 + va*T + va*dva/2/sqrt(a*b); %influence of the following car
v0=30; %Max. speed
```

Memory Allocation

```
result = zeros(length(x),1);
Ncars = floor(length(x)/2);
```

Application of different models

```
for ii = 1:Ncars
    if ii > 1 %All but first car
        dva = (x(ii+Ncars) - x(ii-1 + Ncars));

        sa = x(ii-1) - x(ii) - lcar;

    else %First car
        dva = x(ii+Ncars);
        sa = L - x(ii);
        dva1 = 0;

    end

    %State Space Equation
    if ~guideMap(ii) %"Normal" cars
        result(ii) = x(ii+Ncars);
        result(ii+Ncars) = min(a*(1 - (x(ii+Ncars)/v0)^delta - (sStar(x(ii+Ncars),dva,dva1)
/sa)^2),a);

    else %Guide Cars
```

```
        result(ii) = x(ii+Ncars);
        %Determine preceding guide car
        k = ii-1;
        while k>0 && ~guideMap(k)
            k=k-1;
        end

        if k <= 0 %if it is the first guide car
            result(ii+Ncars) = min(a*(1 - (x(ii+Ncars)/v0)^delta - (sStar(x(ii+Ncars),dva,d
va1)/sa)^2),a);
        else
            result(ii+Ncars) = min(a*(1 - (x(ii+Ncars)/v0)^delta - (sStar(x(ii+Ncars),dva,d
va1)/sa)^2 - (sa<triggerDis)*(x(ii+Ncars)-x(k+Ncars))),a);%(x(ii+Ncars)>x(k+Ncars))*
        end

    end
    %Introducing a disturbance
    if disMatrix(ii,1) ~= 0
        disLoc = disMatrix(ii,1);
        disLength = disMatrix(ii,2);
        disV = disMatrix(ii,3);
        v02 = @(x) max(v0 - (x-disLoc)*(v0-disV)/400,disV);
        if x(ii) > disLoc && x(ii) < disLoc + disLength
            result(ii) = x(ii+Ncars);
            result(ii+Ncars) = min(a*(1 - (x(ii+Ncars)/v02(x(ii)))^delta - (sStar(x(ii+Ncar
s),dva,dva1)/sa)^2),a);
        end

    end
    if result(ii)<0
        result(ii) = 0;
    end

end
```

```
end
```

Published with MATLAB® R2016b

9.2 Simulation with constant starting speed

This was used for the simulations in sections 6.1 through 6.3

Contents

- Parameter definition
- Calculation
- Evaluation
- Plot


```
%Runs the simulation with a certain guide car frequency, randomly
%distributed.
function [res1,res2] = simulate3(freq,disMatrix)
```

```
%@param freq The chance of a car being a guide car
%@param dis 1:With disturbance 0:Without disturbance
%PRE: 0<=freq<=1
```

Parameter definition

```
%Simulation Parameters
Ttot = 10000; %Total simultaion time
Ncars = 1013;
```

Calculation

```
x0 = zeros(2*Ncars,1);

for ii = 1:Ncars
   x0(ii) = 8*(Ncars-ii); %Starting Position [m]
   x0(ii+Ncars) = 29; %Starting Velocity [m/s]
end

guideMap = (rand(Ncars,1) < freq); %randomly introduce guide cars

f = @(t,x) idm_final(t,x,guideMap,disMatrix,100); %map which of the cars are guide cars

[TOUT,YOUT] = ode45(f,[0 Ttot],x0);
```

Evaluation

```
%In this section we extract critical values for measuring the car
%throughput
[ycol, yrow] = size(YOUT);
measurement = zeros(3,3);
taken = zeros(3,3);
for ii = 1:ycol
     if YOUT(ii,5) > 160000 && taken(2,3) == 0
         measurement(2,3) = TOUT(ii);
         taken(2,3) = 1;
     end
     if YOUT(ii,1013) > 160000 && taken(3,3) == 0
         measurement(3,3) = TOUT(ii);
         taken(3,3) = 1;
     end
```

```
end

res1 = measurement(3,3);
res2 = measurement(2,3);
```

Plot

```
ColorMap = [guideMap zeros(Ncars,1) ~guideMap]; %color guide cars red, normal cars blue
set(gca, 'ColorOrder', ColorMap, 'NextPlot', 'replacechildren');
plot(TOUT,YOUT(:,1:Ncars));
title('Position over Time');
```

```
end
```

Published with MATLAB® R2016b

9.3 Simulation with Random starting speed

This was used for the simulations in section 6.4.

Contents

- Parameter definition
- Calculation
- Evaluation
- Plot


```
%Runs the simulation with a certain guide car frequency, randomly
%distributed.
function res = simulate_r(freq,disMatrix,startingSpeedMap,triggerDis)
```

```
%freq is the chance of a car being a guide car
```

Parameter definition

```
Ttot = 600; %Total simultaion time
Ncars = 103;
```

Calculation

```
x0 = zeros(2*Ncars,1);

for ii = 1:Ncars
   x0(ii) = 8*(Ncars-ii); %Starting Position [m]
   x0(ii+Ncars) = startingSpeedMap(ii); %Starting Velocity [m/s], random
end

guideMap = (rand(Ncars,1) < freq); %randomly introduce guide cars


f = @(t,x) idm_final(t,x,guideMap,disMatrix,triggerDis); %map which of the cars are guide c
ars


[TOUT,YOUT] = ode45(f,[0 Ttot],x0);
```

Evaluation

```
pos = 0;
ii = 0;
while pos < 8000
   ii = ii + 1;
   pos = YOUT(ii,Ncars);
end
res = TOUT(ii); %Arrival Time of last car at x = 8000m
```

Plot

```
ColorMap = [guideMap zeros(Ncars,1) ~guideMap]; %color guide cars red, normal cars blue
set(gca, 'ColorOrder', ColorMap, 'NextPlot', 'replacechildren');
```

```
plot(TOUT,YOUT(:,1:Ncars));
title('Position over Time')
```

```
end
```

Published with MATLAB® R2016b

9.4 Generation of Disturbance Matrix

```
function disMatrix = generateDis(pDis,Ncars)
%generates a random map for the disturbances to be introduced
disMatrix = zeros(Ncars,3);
for ii = 1:Ncars
    if rand(1) < pDis
       disMatrix(ii,1) = 40000 + rand(1)*80000; %Start of disturbance
       disMatrix(ii,2) = 100 + rand(1)*500; %Length of disturbance
       disMatrix(ii,3) = 5 + rand(1)*15; %Speed to which the disturbed car slows down
    end
end

end
```

Published with MATLAB® R2016b