



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Project Report

**Simulation of traffic jam
at Gotthard tunnel**

Eric Hayoz, Janick Zwysig

Zurich
December 2014

Agreement for free-download

We hereby agree to make our source code for 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.

Eric Hayoz



Janick Zwyssig





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

Predicting Traffic Jam at Gotthard

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

Author(s)

Last name

Hayoz
Zwyssig

First name

Eric
Janick

Supervising lecturer

Last name

Kuhn
Woolley

First name

Tobias
Olivia

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, 12. December 2014

Place and date

Eric Hayoz

Signature

Janick Zwyssig

Contents

1	Abstract	5
2	Individual contributions	5
3	Introduction	5
3.1	Motivation	5
4	Description of the Model	6
4.1	The Nagel-Schreckenberg model	6
4.2	Example of an iteration	6
4.3	Situation	7
5	Implementation	8
5.1	Correlation between reality and model	8
5.2	Inflow	8
5.3	Average speed	9
5.4	Lane Change	9
5.5	Red Light	9
5.6	Congestion Length	9
5.7	Processing the datasets delivered by ASTRA	10
6	Simulation Results and Discussion	10
6.1	Preparing dataset – curve fitting	11
6.2	train model	12
7	Summary and Outlook	12
8	Appendix	12

1 Abstract

Abstract zu Eric Hayoz & Janick Zwysig, Traffic Jam in front of the Gotthard-Strassentunnel, 2014: Diese Arbeit handelt von einem Verkehrsmodell, das den Stau vor dem Gotthard-Strassentunnel in Erstfeld/UR, beschreibt. In einem ersten Punkt wird erläutert wie die Strassenverhältnisse

2 Individual contributions

We divided the code work as follows (but at the end we both were involved in every part):

Eric Hayoz:

- Implementation NaSch-Modell for 2 cars
- Handling Dataset with inFlow and Speed
- Correlation Reality - Model
- Creating optiFinder

Janick Zwysig:

- Implementation NaSch-Modell for n cars
- Implementation laneChange
- Implementation redLight
- Layout Street with Cars

The written part was performed together.

3 Introduction

Nowadays, traffic jams belongs to every day life. Because Switzerland is an important transit way to Southern Europe for traffic, the Gotthard tunnel becomes to an unavoidable bottleneck. The tunnel stretches out over 15 Kilometers from Göschenen, Kanton Uri to Airolo in the canton of Ticino. Over and over again it comes to long traffic jams, at peak times up to 20 Kilometers. Responsible for that are on the one hand holidays in Switzerland or neighbor countries and on the other hand the lane reduction from two to one lane. Switzerland has been discussing for many years about this issue and building a second tunnel. The purpose of the second tunnel

should be built for security reasons and not in order to increase the traffic flow. This is due to an election 20 years ago, called the Alpenschutzinitiative. In near future there will be a restoration for the tunnel. That's why the discussion about a second tunnel will go on again.

The authors have the goal to investigate the Gotthard tunnel on the north side in Göschenen. That's why a model has been created. By means of the number of cars passing through Erstfeld (a village 19 Kilometers distance from the tunnel), the length of congestion can be measured. The Nagel-Schreckenberg model will be employed for the simulation.

3.1 Motivation

As one of the authors is born in the canton of Uri he knows about the traffic issues at the Gotthard tunnel. This explains the personal motivation to investigate the phenomenon of traffic congestions. The other author is often on the way by car, he has already wasted a lot of time in traffic congestions and thus has also a burning interest in efficient solutions to handle them.

4 Description of the Model




4.1 The Nagel-Schreckenberg model

This model was created in the 90s by the German physicists Kai Nagel and Michael Schreckenberg to simulate traffic flow on highways. It explained the so called "Phantom-Congestion" which occurs when drivers linger by not accelerating even if they have the opportunity to do so.

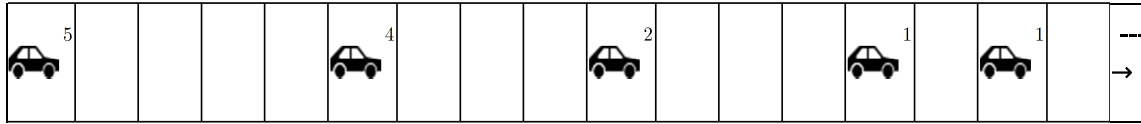
The model is a cellular automata based on four very simple rules, which are applied in sequence. One complete loop is called an iteration.

1. Acceleration: Each vehicle, that has not yet reached the maximum speed (which is 5), increases its speed by one unit.
2. Deceleration: If the size of the gap (in cells) between two vehicles is less than the speed of the following car (in speed units), this car's speed is reduced to a value equivalent to the size of the gap.
3. Linging: Every vehicle reduces its speed by 1 with a probability p ($0 \leq p \leq 1$).
4. Moving forward: Every vehicle moves forward the number of cells equal to its speed.

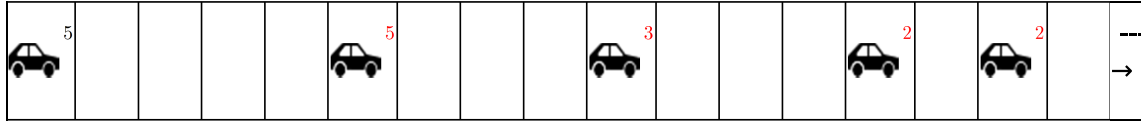
4.2 Example of an iteration

Symbol	Meaning
	A vehicle with a speed of 1 (27 km/h)
	A vehicle that has accelerated or decelerated to a speed of 3 (81 km/h)
	A vehicle has moved with a speed of 2 (going 2 cells forward)

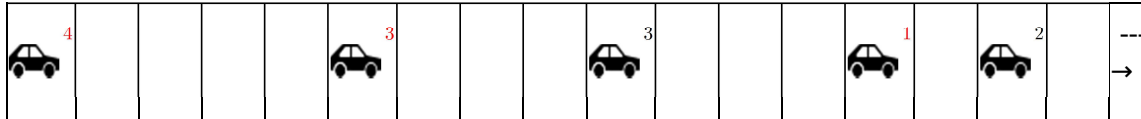
Random startup setting for time t



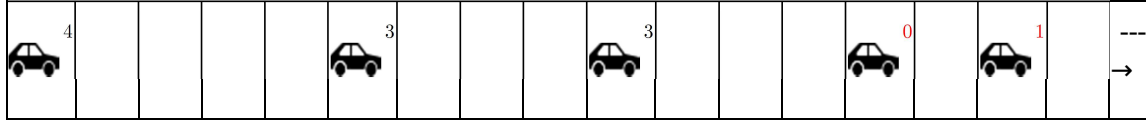
Step (1) – Acceleration, $v_{max} = 5$



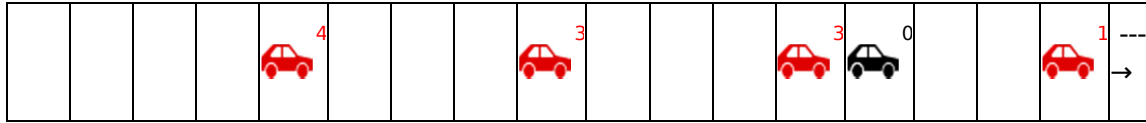
Step (2) – Deceleration



Step (3) – Linging ($\rho = 1/3$, affecting the two leading cars)



Step (4) – Moving (Setting for the time $t + 1$)



4.3 Situation

The distance between Erstfeld and the Gotthard tunnel is relevant for the model. The highway to the tunnel is two-laned and will be reduced to one lane in front of the tunnel. As on the figure «NR» shown, the authors resigned to depicted any exits. The reason for this is first to reduce complexity of the model and second to neglect irrelevant factors. In case of congestion there is a red light that regulates the traffic flow, that is also took in account in the model.

5 Implementation

5.1 Correlation between reality and model

Item	Reality	Model
Time	1s	1 iteration
Maximum speed	120 km/h (we defined v_{max} as 100 km/h to get even values)	5 (only 6 levels)
Length of an average car (Skoda Octavia)	4.5 m	1 cell
Distance Erstfeld - Göschenen	19'000 m	4222 cells (19'000 m / 4.5 m, rounded)

Variable name	Meaning	Values
moveProb	Probability for a car to move forward	0...1
moveCorr	Modify moveProb for a given time (in order to improve congestion length prediction)	-1...1
laneChange	Probability for a car to change its lane	0...1
-	Value of laneChange as a function of distance to changeCell (for locations where cars have to change from left to right)	Constants in equation
redLight	Illuminate a dropcounter redlight in front of tunnel (it turns on if congestion length > 2 km and redLight is set to 1)	0, 1
dropCounter	Length of a period when redlight is ON	0...Inf
-	Maximum speed just in front of tunnel	0...5

5.2 Inflow

The inflow is given in $\#cars / 180$ s. For every iteration in our model, the cars are spawned in Erstfeld (starting position) on the two lanes with a total probability $p = \#cars/180$. As there are normally slightly more vehicles on the right lane, we multiply p for the right lane with a factor of 1.1 and the p for the left lane with a factor of 0.9.

5.3 Average speed

First we convert the average speed from km/h ($0 \dots 120$) to “Nagel-Schreckenberg-Speed” ($0 \dots 5$). If the speed is more than 5, we round down to 5. Example: Average Speed is 94 km/h. Converting into Nagel-Schreckenberg-Speed results in 4.7. Our model would in this case spawn 70% of the cars with a speed of 5, and 30% with a speed of 4.

5.4 Lane Change

Weil Autos in der Realität oft die Spur wechseln, beispielsweise zum Überholen, soll auch in diesem Modell ein Spurwechsel implementiert werden. Dabei wurde berücksichtigt, dass Autos öfter von der linken zur rechten Spur wechseln als umgekehrt. Damit soll bewerkstelligt werden, dass auf der rechten Seite etwas mehr Autos fahren als auf der linken. Ungefähr 60 Meter vor der Spurreduktion wechseln die Autos mit einer erhöhten Wahrscheinlichkeit nach rechts.

5.5 Red Light

Das Rotlicht wird nur bei Stau eingeschaltet. Es regelt ca. 40 Meter vor der Spurreduktion die Anzahl Fahrzeuge, die durch den Tunnel fahren dürfen. Im Modell wurde es ebenso ca. 40 Meter vor dem Tunnel errichtet. Es wird nach 2 Kilometer Stau eingeschaltet und lässt jede Sekunde ein Fahrzeug durch.

5.6 Congestion Length

The length of a congestion is given in Kilometers. First we calculate the length of the congestion in our simulation and then multiply the length (unit: cells) by the length of a cell (4.5m) to enable a comparison between the virtual (calculated) congestion with the real (measured) congestion.

Measuring the congestion length in our model: To obtain a stable output, we divide the highway into blocks with a length of 50 cells. If a block fulfils the two following criteria, it is declared a congestion block: high density and low average speed. We count the congestion blocks starting in Göschenen heading northbound to Erstfeld, and we only accept one uncongested block between any two blocks, otherwise we consider the congestion as terminated at the point where two or more blocks do not fulfill the congestion criteria anymore.

5.7 Processing the datasets delivered by ASTRA

The attached extract shows among other things that we have information about the inflow values and average speeds in 3 minutes intervals. We also have the length of the congestion at specific times (random times and course measurements). These are the three values that we used.

Datum	Fahrzeuge pro 3Min	Durchschnittsgeschwindigkeit	Fahrzeuge pro 3Min Vergleichstage Fr 25.7.2014 - Sa 26.7.2014	Durchschnittsgeschwindigkeit Vergleichstage Fr 25.7.2014 - Sa 26.7.2014	Fahrzeuge pro 3Min Vergleichstage Fr 1.8.2014 - Sa 2.8.2014	Durchschnittsgeschwindigkeit Vergleichstage Fr 1.8.2014 - Sa 2.8.2014
2014-07-18 00:00-00:03	30	114	22	110	33	110
2014-07-18 00:03-00:06	22	116	15	98	37	107
2014-07-18 00:06-00:09	22	115	25	116	42	115

Figure 1: Inflow values with average speeds

Datum	Uhrzeit	zwischen AS	Name	und AS	Name	Stau-km	Wartezeit	Verkehrs-information	Empfehlung
02.08.2014	05:10	39	Wassen	40	Göschenen			stockender Verkehr	keine
02.08.2014	05:30	39	Wassen	40	Göschenen	2	20	Stau, Verkehrsüberlastung	keine
02.08.2014	06:11	39	Wassen	40	Göschenen	3	30	Stau, Verkehrsüberlastung, Einfahrt Gö gesperrt.	keine

Figure 2: Congestion data

6 Simulation Results and Discussion

The reference data for congestion measurement is provided by ASTRA (Bundesamt für Strassen). There are four datasets:

- Dataset 1: measurement of 2014 July, 18. - 19.
- Dataset 2: measurement of 2014 August, first

- Dataset 3: measurement of 2014 July, 25. - 26.
- Dataset 4: measurement of 2014 August, second

In a first step the model will be trained by dataset 1 and 2. The authors wrote a program which calculates specified values of a specified variable (moveProb and moveCorr) for a given number of times. With other words, it finds the optimal parameter values that minimizes the error of measured congestion. In a second step, the datasets 3 and 4 are used to evaluate the previous trained model.

6.1 Preparing dataset – curve fitting

The timestamps for the measurement are irregular. In order to compare the congestion from the model to the reference data, the breakpoints will be interpolated. The resulting curve can be used to generate regular timestamps for measurement. The authors have decided to go for a linear interpolation. That means that every point will be connected by a straight line with its neighbour, see on the figures below. A linear interpolation is reasonable, because the congestion length between two breakpoints can't change very quickly.

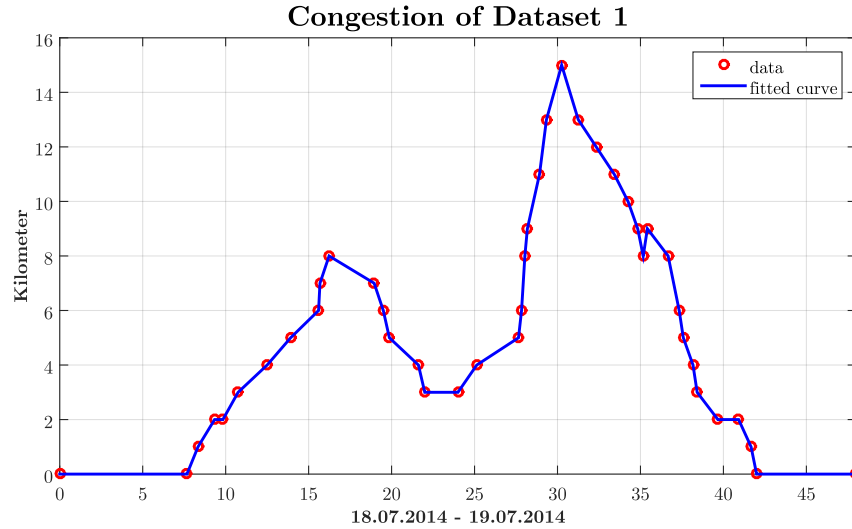


Figure 3: Dataset 1

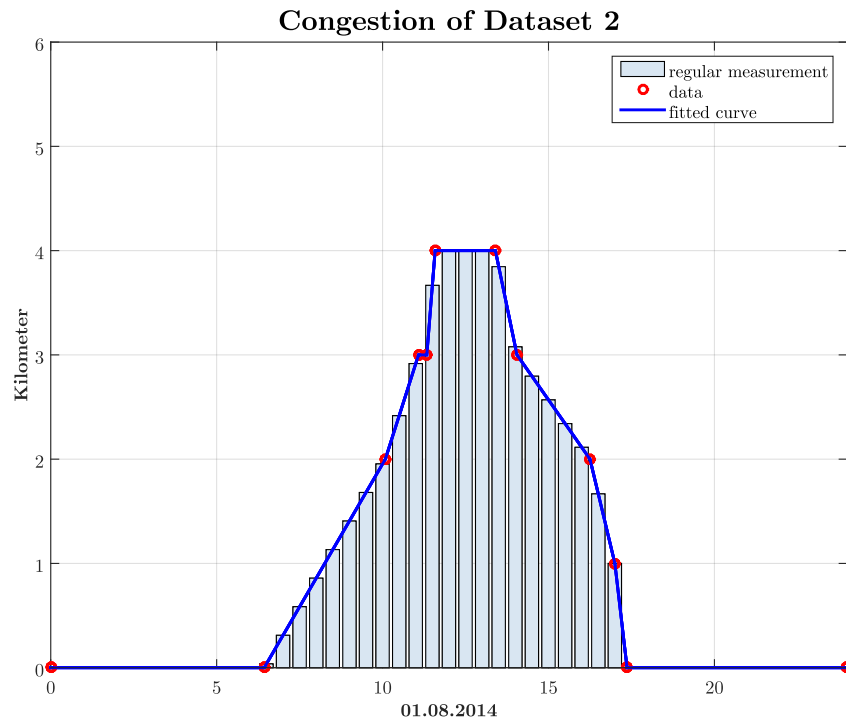


Figure 4: Dataset 2, added regular measurements for comparison

6.2 train model

7 Summary and Outlook

8 Appendix

Code

Listing 1: Matlab-code

```

1 function [error] = NaSch_Datasets_v1(dataset, moveProb, redLight_act,
   isAnimated, moveCorr)
2
3 % values
4 % NaSch_Datasets_v1(1,.525,0,0,0,.055): perfectly symmetric
5 % NaSch_Datasets_v1(2,.525,0,0,0,.05): less error

```

```

6
7 %% parameters
8 % INPUT:
9 %   dataset: choose one of 4 datasets, {1,2,3,4} (don't use 3,4:
      evaluation)
10 %   moveProb: the probability for a car to move forwards, 0..1
11 %   redLight_act: enable/disable red light (0 off, 1 on)
12 %   isAnimated: start program with animation, boolean 1=true 0=false
13 % OUTPUT:
14 %   error_tot: total
15 %
16 % EXAMPLE:
17 %   NaSch_lC_Stats_v1(2, .5, 0 , 1, 0)
18
19 % parameters for comparison model - reality
20 lC = 4.5; % length of each cell (average length of cars
      => Skoda Octavia)
21 lR = 19000; % length in Reality (Erstfeld - Goeschenen, 13
      min)
22 N = round(lR / lC); % 4222 cells
23 measureInterval = 30; % measure every #min
24 [I, S] = Datasets(dataset);
25 nIter = length(I)*180; % number of iterations; datasets: #cars/180s =>
      we want 1 iteration / s
26 q = 0; % running variable for reading outSet
27
28 % set parameter values
29 conv = 1000/lC; % "convert", #cells that matches 1km
30 cC = N-22; % "change cell", where cars have to change the lane
31 vmax = 5; % maximal velocity of the cars (vmax = 5 = 100 km/h)
32 L = 11; % length of lane where cars can change (in front of
      cC)
33 vmax_L = 3; % maximal velocity in L
34 a = 0.2; % min probability that car changes lane at cC - L
35 b = 0.6; % max probability that car changes lane at cC
36 laneChange = .1; % the probability for a car to change the lane, 0..1
37 dropCounter = 2; % dropCounter: #seconds a car can pass the redlight,
      redlight is active
38 % after 2km congestion. Use '1' to turn off redlight.
      Do NOT
39 % use odd numbers.
40 redLight = 0; % automatic redLight, do NOT turn on
41 inflowCounter = 0; % count cars
42
43 % use quadratic increments for the probability between a and b (p = k*x
      ^2+d)
44 k = (a - b)/(L*L-2*cC*L); % for x=cC-L is p = a
45 d = b - cC*cC*(a - b)/(L*L-2*cC*L); % for x=cC is p = b
46
47 % set statistical variables

```

```

48 vSum = 0;           % sum of speeds
49 nCars = 0;          % #cars on road
50
51 % define variables in a block (2 x bL)
52 bL = 50;            % block length:   length of a block // 225m
53 bD = 0;             % block density:   density of cars in a block, 0..1
54 bV = 0;             % block velocity:  average velocity in a block, 0..vmax
55 bC = 0;             % block counter:   counts number of blocks (congestion),
    0..(N % bL)
56 bE = 0;             % empty counter:   counts number of blocks (no
    congestion), {0,1,2}
57
58 % congestion length in each round
59 if mod(dataset,2)
60     divider = 48*60 / measureInterval;
61 else
62     divider = 24*60 / measureInterval;
63 end
64 congLength = zeros(1,nIter/divider);
65 congPlot = zeros(1,divider);
66 currentCongestion = 0;
67 first = 1;           % congestion optimization
68 opt_act = 0;
69 congStart = 0;
70
71 % define road (-1 = no car; 0..vmax = car, value represents velocity)
72 X = -ones(2,N);
73
74 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% average inflow (every 2h(48) 1/2h(24) %%%
75 %{
76 if mod(dataset,2)
77     inflow = zeros(1,24);
78     for i = 0:23
79         inflow(1,i+1) = sum(I(1,1+i*40:40+i*40)) / 40;
80     end
81 else
82     inflow = zeros(1,48);
83     for i = 0:47
84         inflow(1,i+1) = sum(I(1,1+i*10:10+i*10)) / 10;
85     end
86 end
87 p=1;
88 %}
89
90 %% main loop, iterating the time variable, t
91 for t = 1:nIter
92     %% NaSch and laneChange
93     % cars change lane with given probability laneChange
94     for i = 1:cC-L
95         % left to right --> probabiltly laneChange

```

```
96         if X(1,i) ~= -1 && X(2,i) == -1 && rand < laneChange
97             X(2,i) = X(1,i);
98             X(1,i) = -1;
99         end
100         % right to left --> probabiltiy 0.95*laneChange
```
