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 Zwyssig

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.

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

Simulation of traffic jam at Gotthard tunnel

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

Author(s)

Last name	First name
Hayoz	Eric
Zwyssig	Janick

Supervising lecturer

Last name	First name
Kuhn	Tobias
Woolley	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 & Janick Zwyssig
Signature

*Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Print form

Contents

1 Abstract

Abstract zu Eric Hayoz & Janick Zwyssig, 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

Nino played the key role in developing our simulation. Patrick was in charge of the data-fetching. Urs was responsible for the visualizations. The analysis was preformed 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 an 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 restauration 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.

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.

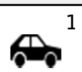
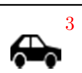
1. Acceleration: Each vehicle, that has not yet reached the maximum speed (which is 5), increases its speed by one unit.

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 cars speed is reduced to a value equivalent to the size of the gap.

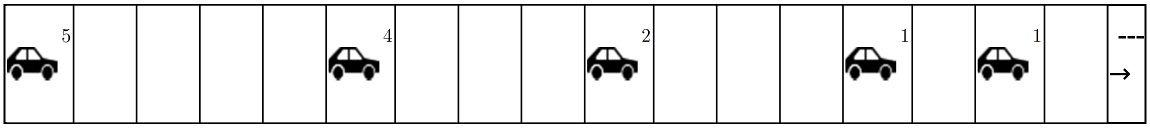
3. Lingering: Every vehicle reduces its speed by 1 with a probability p ($0 \leq p \leq 1$).

4. Moving forward: Every vehicle moves forward the number of cells equal to its speed.

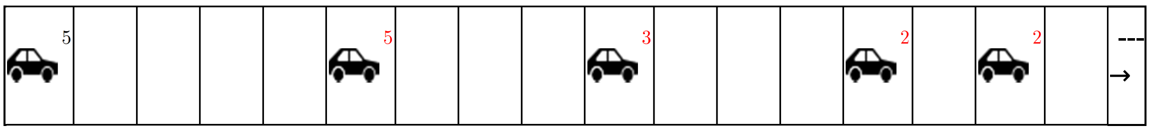
4.2 Example of an iteration

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)

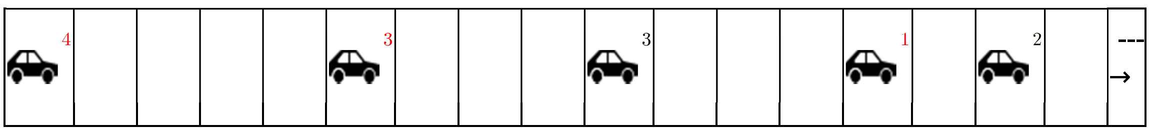
Random startup setting for time t



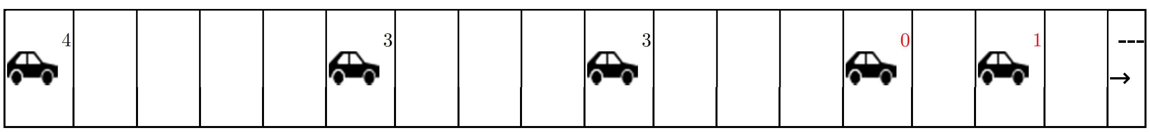
Step (1) – Acceleration, $v_{max} = 5$



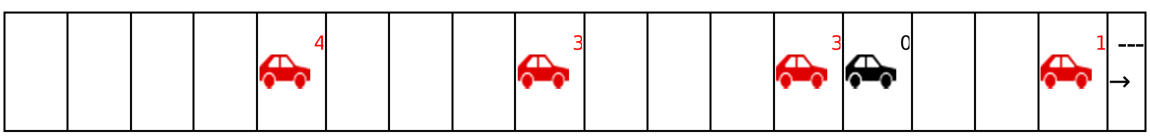
Step (2) – Deceleration



Step (3) – Lingering ($\rho = 1/3$, 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 vmax 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...120) to "Nagel-Schreckenberg-Speed" (0...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 einen 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</title>
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	Warte-zeit	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

In order to find the optimum parameters, we wrote a program which calculates specified values of a specified variable (moveProb and moveCorr) for a given number of times.

7 Summary and Outlook

8 Appendix

Code

Listing 1: Matlab-code

```
function [ error_tot ] = NaSch_Datasets_v1(dataset, moveProb, smallChanges, redLight_act, isAnimated, moveCorr)

% values
% NaSch_Datasets_v1(1,.525,0,0,0,.055): perfectly symmetric
% NaSch_Datasets_v1(2,.525,0,0,0,.05):  less error

%% parameters
% INPUT:
%   dataset: choose one of 4 datasets, {1,2,3,4} (don't use 3,4: evaluation)
%   moveProb: the probability for a car to move forwards, 0..1
%   smallChanges: do small changes for congestion measuring, 0..1, 0 is off
%   redLight_act: enable/disable red light (0 off, 1 on)
%   isAnimated: start program with animation, boolean 1=true 0=false
% OUTPUT:
%   error_tot:  total
%
% EXAMPLE:
%   NaSch_1C_Stats_v1(2, .5, 0 , 1, 0)

% parameters for comparison model - reality
lC = 4.5;       % length of each cell (average length of cars => Skoda Octavia)
lR = 19000;     % length in Reality (Erstfeld - Goeschenen, 13min)
N = round(lR / lC); % 4222 cells
[I, S, T] = Datasets(dataset);   % length = 960
nIter = length(I)*180;   % number of iterations; datasets: #cars/180s => we want 1 iteration / s
q = 0;          % running variable for reading outSet

% set parameter values
conv = 1000/lC;  % "convert", #cells that matches 1km
cC = N-22;       % "change cell", where cars have to change the lane
vmax = 5;        % maximal velocity of the cars (vmax = 5 = 100 km/h)
L = 15;          % length of lane where cars can change (in front of cC)
vmax_L = 3;      % maximal velocity in L
a = 0.3;         % min probabilty that car changes lane at cC - L
```

```
b = 0.7;          % max probability that car changes lane at cC
laneChange = .1; % the probability for a car to change the lane, 0..1
dropCounter = 2; % dropCounter: #seconds a car can pass the redlight, redlight is active
                  % after 2km congestion. Use '1' to turn off redlight. Do NOT
                  % use odd numbers.
redLight = 0;     % automatic redLight, do NOT turn on

% use quadratic increments for the probabilty between a and b (p = k*x^2+d)
k = (a - b)/(L*L-2*cC*L);            % for x=cC-L is p = a
d = b - cC*cC*(a - b)/(L*L-2*cC*L); % for x=cC   is p = b

% set statistical variables
vSum = 0;         % sum of speeds
nCars = 0;        % #cars on road

% define variables in a block (2 x bL)
bL = 50;          % block length:   length of a block // 225m
bD = 0;           % block density:  density of cars in a block, 0..1
bV = 0;           % block velocity: average velocity in a block, 0..vmax
bC = 0;           % block counter:  counts number of blocks (congestion), 0..(N % bL)
bE = 0;           % empty counter:  counts number of blocks (no congestion), {0,1,2}

% congestion length in each round
if mod(dataset,2)
    divider = 72;
else
    divider = 36;
end
congLength = zeros(1,nIter/divider); % 2400s = 40min are 1/72 of 2 days
congLength_prev = 0;                 % 2400s = 40min are 1/36 of 1 day
congPlot = zeros(1,divider);         % final values for plotting
currentCongestion = 0;
cong_r_prev = 0;                     % real cong length
u = 1;
z = 0;
INFLOWMATRIX = zeros(1,nIter);
first = 1;                           % congestion optimization
opt_act = 0;
congStart = 0;

% COUNTER
inflowCounter = 0;
outflowCounter = 0;

% define road (-1 = no car; 0..vmax = car, value represents velocity)
```

```
X = -ones(2,N);


%%%%%%%%%%%%%%%%%% average inflow (every 2h(48) 1/2h(24) %%%%
%%{
if mod(dataset,2)
    inflow = zeros(1,24);
    for i = 0:23
        inflow(1,i+1) = sum(I(1,1+i*40:40+i*40)) / 40;
    end
else
    inflow = zeros(1,48);
    for i = 0:47
        inflow(1,i+1) = sum(I(1,1+i*10:10+i*10)) / 10;
    end
end
p=1;
%}

%%%%%%%%%%%%%%%%%% filter inflow %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rateI_prev = I(1,1)/(2*180);
```