ON-LINE AND OFF-LINE SIMULATION OF LARGE MOTORWAY NETWORKS

Toorenburg, Jacob A.C. van*, **Kijk in de Vegte, N.,**
Transpute bv, The Netherlands
Bédorf, J.,
Leiden Observatory & partialy Transpute bv, The Netherlands

ABSTRACT

This contribution describes a new, data-driven approach to traffic flow simulation. The approach allows coping with large networks and realistic traffic situations at comparatively little costs for network preparation and model calibration. Results obtained with respect to performance, flexibility and speed are presented and the principles underlying the model are pointed out. The paper concludes with the advantages and limitations of the approach.

Keywords: *Simulation, Motorway-network, Data-driven, Low-cost*

INTRODUCTION

The focus of this paper is the data-driven simulation model **FlowSimulator** [1]. It is a cellular model for motorway sections, stretched up to handle motorway junctions as well. It iterates a simple traffic state equation over cells and over time. The prime difference with likewise models is the extent to which traffic first-principles are exchanged for field measurement-data. As a result, the model is intrinsically simpler and easier to handle than similar models that simulate traffic flow on motorway networks. It came as a bonus that much of the otherwise tedious calibration work now appeared to enter the model implicitly via the input data.

First some results, attained with the model, will be presented in order to give the reader some idea of the capabilities of the model. Then the principles, used to construct the model will be pointed out, followed by a discussion of the strong and weak points in the approach. Finally, we address current R&D to further develop the model.

DISCUSSION OF RESULTS OBTAINED

The model network is shown in fig. 1. It consists of the complete motorway network of the Netherlands.

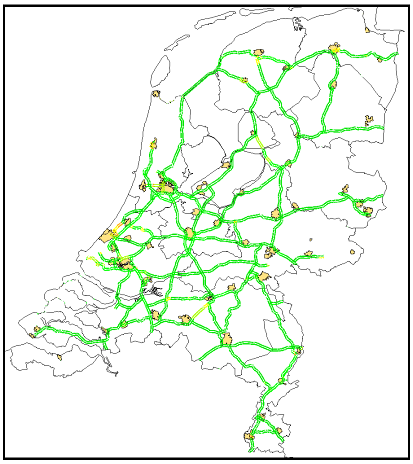

Fig. 1: The simulated network

A snapshot for a busy moment in a busy region, taken from a simulation representing the average working day, after calibration, is given in fig. 2. Presented is the speed on the network for that timeslice, in color.

Fig. 2: Arbitrary snapshot of part of the network, showing simulated speeds taken from a during a simulation

The snapshot of fig. 3, also showing speeds on the network, is taken from field data. It allows the reader to form an impression of the resemblance. Please be aware, however, that day to day differences are substantial and that the purpose of this graph is to give an impression only.

Fig. 3: Snapshot of traffic conditions, measured in the field on a particular day on a comparable moment

The network is split into 100m-sections and is simulated with a time step of 3 seconds. Earlier versions used 200m/6s, but this led to inability to model certain junctions accurately. Simulation of one day took - before

---

Jaap van Toorenburg – Transpute bv, Muurhuizen 159, 3811EG Amersfoort, the Netherlands, +31-334618100, jvt@transpute.nl

parallelization, see further - about a quarter of an hour. Model-input has been prepared for the average working day, for the vacation working day, the Saturday and the Sunday. The model proved able to mimic how the daily queues evolve very realistically. In a way this is not too surprising, as current measurement data served as the basis for the model-input and obtaining similarity with the field was the aim of the subsequent calibration process. The model also, however, proved to be able to answer many what-if questions as long as these do not reflect situations too far from the current one. Think of estimating the effect of modest changes in local or global traffic demands or the impact of changing certain bottle-neck capacities. The model has been extensively used over the last years to optimize the scheduling of major reconstruction works in the Netherlands (which road works should be scheduled subsequently, which ones simultaneously instead? Can it be done over day, should it be done overnight, in the weekend or during holidays? etc.) [2]. Also it turned out to be an efficient tool to check rerouting plans, incident management scripts, road-capacity improvements and the like. FlowSimulator has evolved to a mature off-line instrument in this field. But how about on-line use?

On-line simulation

Encouraged by the success of the approach, a next step in development was undertaken: to bring the model on-line. Why not simulate traffic some time ahead, using the actual traffic situation as starting point? The Dutch motorway network offers particular favorable conditions for such an experiment as the larger part of it is equipped with on-line loop detectors at short intervals. Two limitations had to be overcome, however. The first was simulation time, the second was how to start the model on the fly, using the current traffic situation as launcher. To cut down simulation time, the computing-intensive part of the code was ported to a PC-graphic card [3]. This part of the work was carried out by Bédorf, leaning on the experience of Leiden and Amsterdam astronomers with the numerical simulation of star clusters. A careful reconsideration of tasks led to a version for parallel computing that cut down simulation time to less than one minute. Moreover, multiple scenarios can now be started in parallel on the same platform, thus not increasing the time to availability of results. The on-line version is now implemented in two different shells to test its predictive performance. The first one is a fully automated one-hour-ahead simulator that is re-launched every five minutes. First taking in the loop detector speeds and flows, then simulating, then refreshing its map of predicted speeds, see fig. 4. The second is a decision support system for intervention handling. It is targeted at the operator room to evaluate various interventions quickly during the period that an incident disturbs the network. It simulates a ‘do-nothing’-option and three others that can be specified on the fly simultaneously, and presents results in the form of a map-images within one minute.

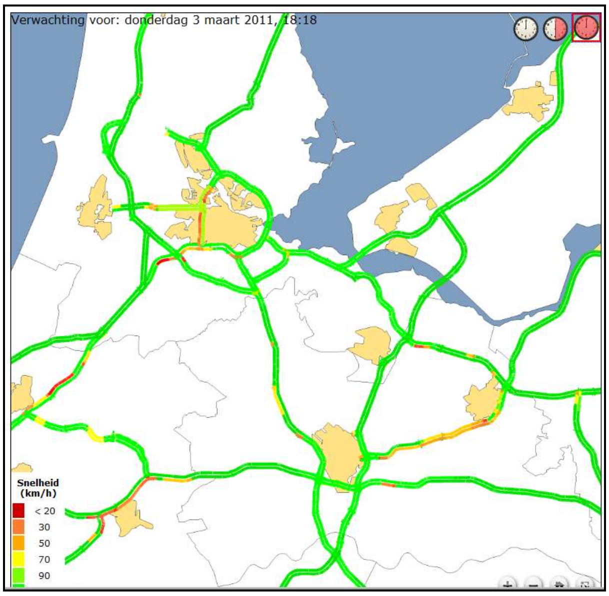

Fig. 4: **One-hour ahead prediction that the testapplication 'uurvooruit' made at thursday 3 march 2011, 17:18 hr for 18:18 hr, showing predicted speeds on the network**

FLOWSIMULATOR PRINCIPLES & ESSENTIALS

The kernel of the model is a cellular automaton that evaluates a fixed state equation between density, speed and volume, complemented by conservation of vehicles. In this respect it is not unique at all. The FlowSimulator approach is special, however, in the way the kernel is controlled:

- it considers a conceptual quantity called ‘traffic demand’ that represents the amount of traffic that wishes to make use of a certain link on a certain time; would there be no congestion on the network, it would coincide with the momentary traffic volume, if there is congestion, it will differ;
- traffic demand on a link is estimated from measured traffic volume time-profile data of that link only;
- the network is calibrated inside-out;
- the nodes of the network are tolerant to conflicting inputs, which means that inflow does not need to equalize outflow;
- in its present form FlowSimulator does not make use of an OD-matrix.

It can be informative to explain how these peculiarities came about. The reason for the first three lies in the difficulties that we met while calibrating large networks in a logical way. If we tuned the network-inputs such that the queue at one bottle-neck evolved correctly, this calibration was ruined when attention was turned to the next. Please note that, typically, networks have a handful of bottle-necks that dominate the course of flow. At these locations apparently the most outspoken excess of road capacity is revealed. If one would know the extent of this excess, one could predict the queue that the bottle-neck will generate. This is the idea behind demand-estimation. The calibration of the model will be half-done already if only the excesses at the important bottle-necks could be reproduced in the model. This brought up the idea to try calibration inside-out. Calibration inside-out starts with the known bottle-necks and then works itself outward to the less interesting parts of the network. It consists of assigning a virtual quantity ‘traffic demand’ to each link. Traffic demand reflects the amount of traffic that “wants to be there” as a function of time, see discussion in the next paragraph. This step is the equivalent of the traffic assignment step in an OD-model. Also field assessments of bottle-neck capacities, where available, are used to tune network capacities at those locations. Traffic demands are worked out linkwise until the whole network is done, the entry-links likewise. When simulation is started, the flows on the entry links (corresponding to their estimated traffic demands) will fill the network - and the nodes will propagate these flows - in such a way that at the bottle-necks the precalculated excesses will build up. Upstream queues will form in a realistic manner because the FlowSimulator kernel will make them obey the state equation wherever they appear. Please note that this very shortcut of replacing assigned flows by measurement-based traffic demands could only be taken thanks to the existence and measurability of the real network. This is why we call it data-driven.

Now we get to the 4th mentioned peculiarity. This one allows the network to be inconsistent in the conservation of flow *at nodes*. This adaptation was also introduced not to ruin one part of the network while doing the other. In its most basic form, it says that if two succeeding links have estimated traffic demands of 4000 and 4400 veh/hr respectively, may be caused by two non-exact traffic counters, may be by other calibration inaccuracies, we will not bother about the conflict and just believe both to be true, thus let the node make eleven vehicles flow out for every ten flowing in. Smuggling away inconsistencies in this way keeps the basic calibration intact and blocks further propagation of errors throughout the network.

Estimating traffic demand

How traffic demand is estimated from the measured traffic volume-profile is clarified by fig. 5. The thick line is the traffic volume as measured at a bottle-neck. The thin line is traffic demand. It is a hypothetical construct that represents the traffic volume had there been no capacity restraints in the network. It is constructed by seeking the best reproduction of the measured profile by building it up from three primitives. Those primitives are depicted in fig. 6 and might be interpreted as the natural time-lines for home-to-work, work-to-home and other traffic. They are mathematically computed, however, by a technique resembling Singular Value Decomposition, applied to a large set of measured volume-profiles, recorded in uncongested areas. As far as we could check, these curves are highly universal throughout the country and did not change noticeably over time.

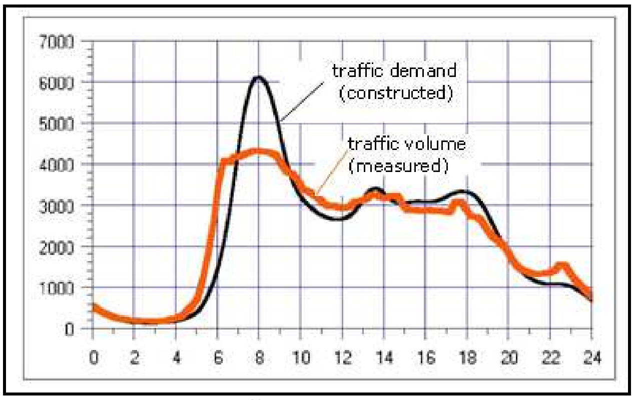

Fig. 5: **Measured traffic volume at a bottle-neck location (thick line) and its reconstructed hypothetical traffic demand profile (thin line); the last one is input for FlowSimulator**

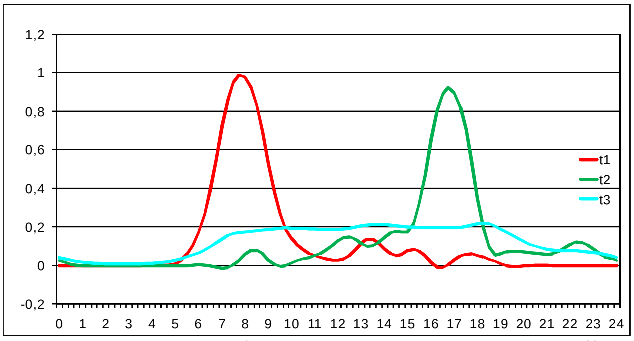

Fig. 6: **The three uniform primitives used to build up traffic demand profiles as in fig. 4 at all locations; the profiles are determined mathematically, but appeared to converge to very natural timelines for home-to-work, work-to-home and ‘other’ motives respectively.**

Reconstruction of a demand profile at an undisturbed location is a matter of calculating the best possible fit. In the case of a disturbed measured volume-profile, as in fig. 4, there is no penalty on discrepancy during the peak hour time block, its only the time-block’s total that participates in the fit. Further discussion of this interesting topic is not appropriate here, unfortunately. It is remarkable though that a pure mathematical technique comes up with a beautiful representation of the three main motives for making a trip. At this place, however, the notion that the technique leads to a plausible estimate of traffic demand should suffice. We like to emphasize again at this point that traffic demand for a link thus is computed using the measured traffic volume profile of that link only. In FlowSimulator, traffic demand is entered as a quarterly time series per link. Traffic demand is calculated for all links where traffic volume profile measurements are available. Known bottle-necks get extra attention in order to avoid casual mistakes, as these points will dominate the simulation later.

Filling out the network

Starting from the known bottle-necks and other links where the traffic demand could be well established from measurements, the other links traffic-demands are calculated and/or estimated by inter- and extrapolation. For uninteresting parts of the network it is of little interest how this is done. At places where the demand-capacity ratio is substantial, a more careful treatment is required. A static OD-model might be used to guide this work, but a process of manually controlling and correcting precalculated interpolations mostly does as well. Further automating this process makes part of the current improvement program.

Split factors at diverging nodes, backward queue propagation at converging nodes

The ratio with which traffic splits at diverging nodes is the same as the ratio between the traffic demand functions of the outgoing links. As these functions vary over time, split factors will vary accordingly.

At converging nodes it is sometimes necessary to explicitly specify the ratio with which the outflow to the subsequent link is assigned over the incoming ones in order to reproduce the bifurcation of queues as observed in the field. Again, this ratio can be obtained by inspecting traffic volumes during queueing conditions in the field.

STRENGTHS AND WEAKNESSES

In broad view, what can be considered as valuable in the FlowSimulator approach and what are the limitations? We always experienced the ease with which a ‘current situation' reproducing tool could be set up as very valuable. The amount of questions that can be answered, once such a tool is available, is astonishing. However, the limitations also are very severe and are met quickly once questions are asked that go one step further. For instance, FlowSimulator as described has no intrinsic equilibrium seeking mechanism. Thus, if a link capacity is halved for example, traffic just runs up against the new constraint and will not redistribute itself. Draconic queue formation will occur. For an unexpected situation, this will be close to the truth for the first half hour or so. The repelling effect of anticipation by drivers will make the model prediction worthless quickly thereafter, however. The extra traffic introduced by new build-up areas is another issue of which it is self-evident that it falls outside the scope of the model, Such expectations should be evaluated externally, where after they can be applied to the model-input, however.

For off-line applications, the scope of the approach can thus be extended somewhat by doing such things manually. Seeking equilibrium, for example, can be done by trial and error in a few iterations by hand. For such purposes we developed tools that speed up the work of imposing manually assigned changes over many links simultaneously. Redirecting the flow of a closed motorway, simulating the effect traffic information, finding the approximate new network equilibrium after a long-stay construction zone is introduced are all cases that can be worked out this way. It takes some work but it is very feasible and cases like these have in fact formed the main area of application over the last few years. It can be done with relative easy because simulation time is very short, thus results are available almost instantly. It is not elegant, however, and we did try to weave the use of OD information and some traffic assignment techniques into the data-driven approach. It looks as if too much connections with the classical model approach have been cut through, however, as up to now these efforts inevitably led to conflicting amalgams.

For the on-line model we are experimenting with ways to detect and measure the anticipation of drivers to severe obstructions automatically by inspecting the deviation from expected levels. The idea is 'if we cannot predict them, we can at least measure them'. We tried monitoring split fractions at junctions for example and taking over the new value if there is a large deviation or a sudden change. We did the same for link volume levels. Using these adaptations improves the medium-term prediction substantially as the repelling effect of obstructions is taken over from the field. The adaptations also introduced instabilities, however, as the same algorithm works all over the network, so spontaneous fluctuations are sometimes misinterpreted as change in traffic demand. Nevertheless, we expect to realize a number of improvements along this line.

CONCLUSION

FlowSimulator has proven itself to be a cost-effective tool for working out many questions in traffic- and road-management. This proven record is not yet the case for the on-line setting, however. Yet performance is impressive and results are very promising indeed.

For both on- and off-line settings the inability to re-assign flows automatically according to changed network conditions, according to various equilibrium assumptions, according to known OD-relationships, forms a severe limitation.

REFERENCES

[1] Toorenburg J.A.C. van (2003). Flowsimulator, description of the model (in Dutch). Transpute/Rijkswaterstaat-AVV, Gouda.
[2] Nijenhuis, T.A., Elbers J.A.C.M., Kijk in de Vegte N. (2006-2010). Large roadworks 2007/…/-2011 ('Groot onderhoud 2007/../2011', in Dutch). Transpute/Rijkswaterstaat-VCNL. Amersfoort.
[3] NVIDIA (2011). GPU computing and CUDA, USA.