Microscopic Freeway Simulation with Automatic Calibration

Iisakki Kosonen
Helsinki University of Technology
P.O.box 2100, 02015 TKK, Finland
iisakki.kosonen@tkk.fi

***Abstract* — The microscopic simulation is getting increasingly common in traffic planning and research because of the detailed analysis it can provide. The drawback of this development is that the calibration and validation of such a detailed simulation model can be very tedious.**

This paper summarizes the research on automatic calibration of a high-fidelity micro-simulation (HUTSIM) at the Helsinki University of Technology (TKK). In this research we used ramp operation as the case study.

The automatic calibration of a detailed model requires a systematic approach. A key issue is the error-function, which provides a numeric value to the distance between simulated and measured results. Here we define the distance as combination of three distributions namely the speed distribution, gap distribution and lane distribution.

We developed an automated environment that handles all the necessary operations. The system organizes the files, executes the simulations, evaluates the error and generates new parameter combinations. For searching of the parameter space we used a genetic algorithm (GA).

The overall results of the research were good demonstrating the potential of using automatic processes in both calibration and validation of simulation models.

I. Introduction

At the Helsinki University of Technology microscopic traffic simulation model (HUTSIM) has been developed over ten years [7]. Recently the object-oriented and rule-based simulation model has been extended from the urban traffic simulation towards simulation of freeway traffic. Modeling of freeway traffic involves more inherent fluctuations and variation of the vehicle dynamics, thus increasing the model complexity.

Several calibration studies have been carried out based on field data from various countries. The ordinary calibration studies can handle only very limited number parameters. While the number of variables and parameters increases, also the number of possible parameter values is exploding too large to handle without automation.

II. The Simulation Model

The HUTSIM urban traffic simulation system is an object-oriented micro-simulation system with high-fidelity modeling of the vehicle interactions. The simulation system includes detail modeling of the street infrastructure and lane organization. All the static objects of the system are attached to each other with connections that assign their operation within the simulation system. All the properties and connections of the simulation objects are defined by a separate editor-program.

The vehicle objects are generated during the simulation and updated typically 10-20 times per second. The high frequency updating is taking place in random order considering all objects as parallel. This approach allows detailed modeling of the interactions between any types of objects within the simulation system.

A central object of the system is the vehicle. The vehicle object takes care of the interactions with other objects namely other vehicles, pedestrians, street and traffic controllers.

The HUTSIM vehicle dynamics is based on discrete speed levels and rule set that handles the transitions from one speed level to another. For each update cycle, the vehicle can keep the same speed level, or increase/decrease the speed by one step. Since only three choices exist, the rule base can be simple. Due to the high update frequency, any realistic speed curve can be approximated with the rule-based control.

The rule set consists of few rules from which the latter ones always overrule the previous ones. The space in front of the vehicle is divided to three zones namely free zone, stable zone and forbidden zone. The borders of these zones depend on vehicle's speed and its speed difference to the vehicle/obstacle ahead. The default rule is to maintain the current speed level. The second rule allows speed increment if current speed is less than desired speed and certain time has elapsed from last increment. The third rule forbids speed increment if vehicle is inside the stable zone. The fourth rule forces to speed decrement if vehicle is in a forbidden zone.

These few rules provide a consistent and stable car-following and obstacle yielding behavior in all traffic situations.

III. High-fidelity freeway simulation

The basic HUTSIM urban traffic model was enhanced for freeway simulation in co-operation with the Royal Institute of Technology [6]. The need of new features was observed soon. The freeway simulation called for more inherent randomness and variation in modeling of the traffic flow. Also new types of interaction were required in freeway operations. Especially the ramp operation is more complex than the non-signalized yielding in the urban environment. The new features increased the number of variables and distributions in the simulation system and therefore also more efficient approach was needed in the calibration research.

One of the new features implemented to the HUTSIM freeway model was modeling of the driver attitude. The driver attitude parameter [0..99] is set upon generation of each vehicle. During the simulation, the aggressive type of driver (>50) tends to choose the high end of the desired speed distributions (depending on speed limit of the freeway section) and the low end of the gap distributions (e.g. car-following gap, lane change gaps).

The basic urban traffic model used a fixed time headway parameters for the car-following distance (the stable zone). In the freeway model the fixed time gap value was replaced with a distribution. Each vehicle picks up its own car-following gap from the distribution. Also the desired speed is picked up from a distribution each time vehicle enters a new speed limit zone.

Another freeway related feature is modeling of the driver awareness. It is assumed that while no stimulus exist in certain time, the driver gets to a more relaxed state of mind. In this situation the driver observation and vehicle speed is not updated, but only the vehicle position advanced. For example if the driver observation is updated only once per second, then any stimulus that happens is delayed randomly between 0.1-1.0 seconds. This variable delay is causing inherent and random fluctuation to the traffic flow.

The ramp operation is fairly complex in the freeway simulation model. Since stopping is not allowed in the freeway, the vehicles must find a gap to go in any situation. To allow smooth ramp operation, the ramp area was divided into three regimes. The approaching area allows vehicles to adapt their speed and to start looking for a suitable gap. The actual entry is done within the ramp area when suitable gap is found. If a suitable gap was not found then vehicle enters final entry area, which forces the vehicle to perform the merging anyway.

The ramp operation takes place in four phases. Firstly the vehicle in the ramp has to choose a target vehicle to adapt its own speed level. Secondly the vehicle has to select the gap either in front or behind of its target vehicle. Thirdly the vehicle starts changing its speed in order get into right position for the selected gap. Finally the vehicle changes the lane as soon as it gets into a proper position so that the minimum gap requirements are fulfilled both behind and in front of the merging vehicle. At the beginning of the ramp the normal lane change gap is requirement is used, but towards the end of the ramp the gap requirement is decreasing towards minimum in order to force all the vehicles to enter the main road.

If a vehicle enters into a small gap then it has to start widening the gap by setting its own speed one step below the speed of the vehicle in front. The vehicles behind may also have to do the same, which is causing a shockwave that moves backwards. The slowing speed on the basic lane also increases the overtaking pressure of the vehicles behind, which tend to start overtaking the ramp area.

The off-ramp operation is similar to on-ramp operation if the vehicle is in the overtaking lane. However, there is usually more space to get into the right lane in time. The weaving sections consist of complex interactions as vehicles enter and exit at same time while also voluntary lane changes can exist.

IV. The calibration system overview

A framework for automated simulation environment (HUTMAT) was created first. The MATLAB-software was used for implementing the necessary functions of the system. MATLAB is also controlling the overall operation launching the simulation runs and other tasks. In the second stage the genetic algorithm was applied for generating the parameter combinations. The basic version GA was later enhanced with more advanced features and with a window-based user-interface.

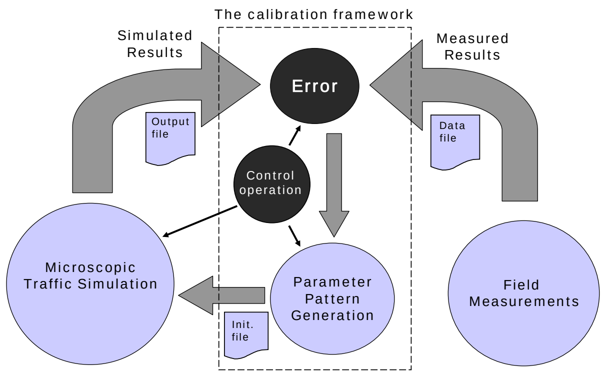

Fig. 1. The automated calibration environment

The HUTMAT-simulation environment is shown in figure 1. The value of each parameter is set in the init-file before each simulation run. After execution of a simulation run, the output file is analyzed by a tool program that computes the error-value. Based on the error-value, another tool program generates a new pattern of parameter values and submits them for next round of simulation.

The most critical component in the system is the error-function, since it determines the distance of the simulated results from the expected (measured) ones. One approach is to simply use detector counts as error the error-function [8].

We defined the error-function in more detailed manner as composition of three components namely the speed distribution, the time headway distribution and the lane distribution. Since the test area involves two lanes there are actually two distributions for each lane. The distributions are expressed as histograms. They show the amount of vehicles that lay in the same magnitude of the quantity.

Because the total amount of vehicles varies by lanes and by time intervals, the distributions are normed by dividing each column height by the total amount of vehicles per lane. The comparison of each distribution is done according to the following formula (1).

$$\varepsilon_{m,n} = \sqrt{\frac{1}{2}\sum_i \left(\frac{f_i}{\sum_j f_j} - \frac{g_i}{\sum_j g_j}\right)^2} \tag{1}$$

f is the simulated distribution

g is the measured (reference) distribution

$\varepsilon_{m,n}$ is the error of m-th detector

This approach gives a single value that describes the distance of the distributions. For the time headway distribution and speed distributions the error is computed alike. The lane distribution is obtained in a little more complex way. The investigated file is divided into time slices of 15 minutes. Then the proportion of vehicles in a lane in respect to the total flow is calculated for each time slice.

As the components of the total error-function are time headway, speed and lane distribution for each lane separately, we should combine the partial errors obtained from every distinct distribution. In this work the total error-function is chosen to be the average of the components. Thus there would be six components of which the two derived from the lane distributions are same.

V. THE CASE STUDY

Ramp operation was chosen as case study for calibration process, since it is the most critical operation in freeway modeling. The ramp operation of involves a lot of complex interactions to be modeled and calibrated.

The test site is located in Sweden, in the north of Stockholm, and the measurements were done during two days. The ramp is located north of Stockholm and the traffic is directed towards the south. The length of ramp is approximately 300 meters and the length of junction area 150 meters.

The test site provided all the data that was needed to compare the simulation with the reality. The available field data included actual speed, direction, net and gross time gaps, vehicle type and vehicle length. The ramp and the detectors used configuration is described in figure 2.

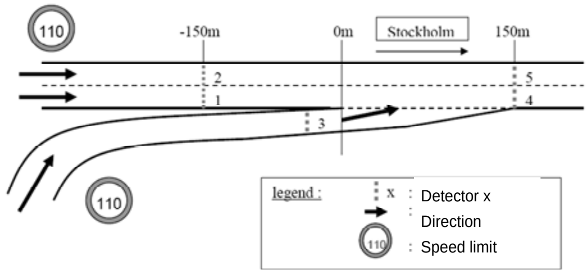

Fig. 2. The test site near Stockholm

We chose set of 20 most essential parameters for the calibration process. These parameters include the car-following gap distributions, lane change gap parameters, ramp speed parameters etc. Minimum and maximum boundary values were set to each chosen parameter. An initialization file was used to feed all the parameters values and simulation settings to the simulation program.

TABLE I

SELECTED PARAMETERS FOR THE RAMP CALIBRATION

No	Type of parameter	Number	Unit
1	Car-following non-stability factor	1	% of cycles
2	Voluntary lane changing sensitivity values	2	0.0 – 1.0
3	Voluntary lane changing gap requirements	4	% of sec.
4	Forced lane changing gap requirements	4	% of sec.
5	Ramp speed deviation	1	km/h
6	Overtaking speed increase	1	km/h
7	Overtaking speed loss threshold	1	km/h
8	Mean value of the car-following gap distribution	1/lane	sec.
9	Variance of the car-following gap distribution	1/lane	
10	Mean value of the desired speed distribution	1/lane	km/h
11	Variance of the desired speed distribution	1/lane	

In table I The 'car-following non-stability factor' is defining the percentage of updates, where a random speed fluctuation takes place. The 'voluntary lane changing sensitivity' parameters (2) control the decisions to pursue lane change. The 'lane changing gap' parameters (4) define the minimum required gap in front, behind, left side and right side (as proportion of the car-following gap). The 'ramp speed deviation' sets the speed difference while looking for a gap in merging. The 'overtaking speed increase' is used to temporarily increase the desired speed level during the overtaking. The 'overtaking speed loss threshold' parameter defines how much the speed can reduce from the desired speed, before the vehicle decides to overtake. The car-following gap and desired speed are defined as distributions set by mean value and variance for each distribution.

VI. The Calibration Algorithm

So far we have defined the traffic simulator, the calibration framework, the error-function and the test application (ramp operation). The missing part is the algorithm needed for generating the new parameter combinations.

In this particular application the number of selected parameters was limited to 20. Generally speaking the number of such parameters can easily be much higher. Therefore we needed an algorithm that could handle large search spaces as well as small ones. Also we have to keep in mind that the properties of the target function cannot be expressed in mathematical form.

Based on these requirements, we chose a genetic algorithm (GA). GA is a search algorithm that uses a generations of populations [4]. An individual in the population represents one combination of parameters. From each generation the fitness of every individual is evaluated with the error-function. Only the best individuals are selected to next generation. After evaluation of the whole population and new generation is produced.

The operation of GA is based on few primitives namely the crossover, the mutation and the selection of the fittest. For how to implement these operations, there are plenty of choices. The first choice is how to code the parameters into the “genes”. We used a floating point representation, where each parameter combination is represented as a vector with all of its components between zero and one.

In the first release of the calibration system, a basic version genetic algorithm used. The second version included methods to improve the GA performance. One way to improve the performance is to use smaller population sizes, since each individual requires one simulation run. Small populations are used by so-called micro-genetic algorithm [5]. The micro-GA is executed several times using different initial populations [2]. Other features to boost the performance involve fitness sharing and guided crossover. The fitness sharing is used to allocate the individuals in optimal regions of the fitness landscape [3]. The guided crossover is used for fine tuning of the parameter values with gradient-like abilities [9].

VII. Results

We used the HUTMAT-system to run automatically hundreds of simulations. Each simulation lasted three hours of simulation time. The results of the simulations are shown in *figure 3*. The average error of the population started immediatedly to converge towards the zero. After ten generations, the average error was no more decreasing significantly. The lowest error was reached in the 20th generation. The simulation number 822 gave the error value of 0.0459.

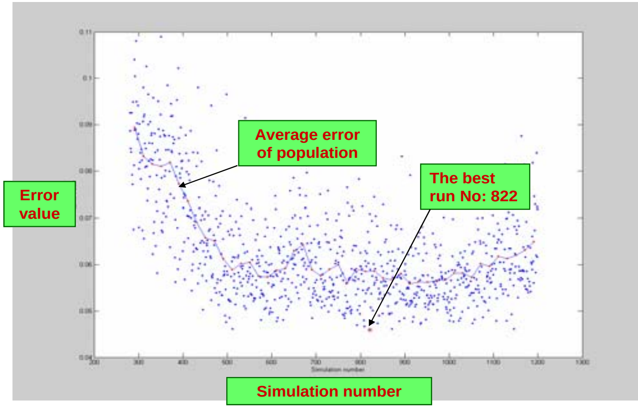

Fig 3. The values of the error as function of simulation number. The solid line shows the mean value of each population.

Figure 4 demonstrates the results of the best simulation run by visualizing all the components of the error function. According to *figure 4*, the gap distributions seem to have the right shape, which is similar to the generalized gamma-function.

As to the speed distributions the speeds should be faster on the overtaking lane than on the basic lane. The simulation model produced a slightly less speed difference than required. The shapes of the simulated distributions are correct, but the variances differ a bit from the measured ones.

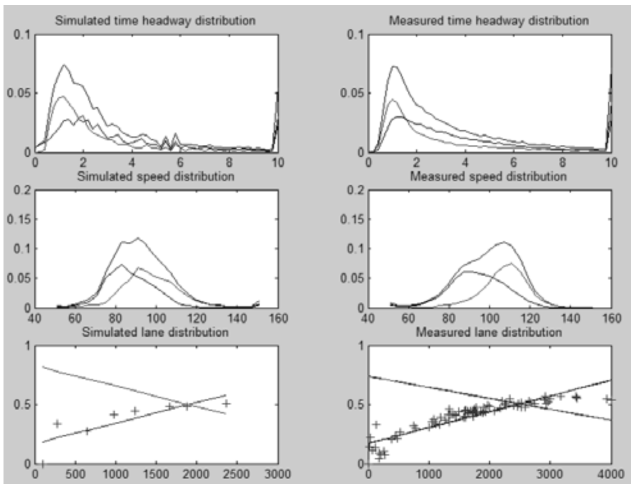

Fig 4. Visualizing the components of the error-function. The results of the best simulation run (822) compared against the field measurements.

Finally the third component of the error-function. In *figure 4* the lane distribution is shown as a function of the flow rate. The simulated lane distribution is fairly good compared to measurements. Both simulated and measured distributions slightly deviate from the linear model in a similar way.

VIII. Conclusion

The automated calibration environment has turned out to be a useful tool in the traffic simulation research. The productivity of model development and calibration is increased enormously since any changes on the model or parameter values can be immediately evaluated against the field measurements.

Developing and setting up an automated calibration environment is a big effort. Therefore it is important that we implemented the basic framework in such a generic way that test settings can be changed easily. With the proposed system it is easy to choose any set of parameters to be calibrated.

In general the proposed calibration system is suitable for situations where the simulation parameters cannot be directly measured, but the values has to be estimated indirectly based on measurements from the field. The present calibration system is limited to the use of detector data only, but the number of detectors is not limited.

The generation of parameter combinations was established as a separate unit that can be implemented with any algorithm if necessary. The use of genetic algorithm turned out to be a good choice especially when boosted with several options in reproduction, crossover and mutations of the GA.

This HUTMAT-project was part of a larger framework project called DigiTraffic, which is focused on traffic modeling and ITS. Simulation research is one important part of the project. The further research involves using of the calibration system with other traffic environments, too.

Also in real-time simulation several parameters need to be tuned automatically. For example the vehicle speeds and turning movements need to be tuned in very short time interval. The further research will show if the proposed system can be applied to real-time simulation as well.

Even more detailed modeling than micro-simulation is called nanoscopic simulation. The nanoscopic simulation involves very detailed modeling of the driver perception, decision making and errors. Because of the increasing number of unknown parameters, an automatic calibration tool is even more important.

Acknowledgment

The author would like to acknowledge Mr Daniel Aalto, Mr Loris Chabredier and Mr Remi David for their work in the development of the calibration system.

References

[1] L. Chabredier and R. David, “Programming book for HUTMAT”. Helsinki University of Technology, Laboratory of Transportation Engineering, Espoo April 2003.

[2] C. A. Coello and G. T. Pulido, “Multiobjective Optimization using a Micro-Genetic Algorithm”. In SECTOR Lee et al., editor, Proceeding of the Genetic and Evolutionary Computation Conference, San Francisco, California, USA. Morgan Kaufmann Publishers, 2001.

[3] D. E. Goldberg and J. Richardson, “Genetic algorithms with sharing for multimodal function optimization”. In J.J. Grefenstette, editor, Proceeding of the Second International Conference on Genetic Algorithms and their Application, Hillsdale, New Jersey, USA. Lawrence Erlbaum Associates, 1987

[4] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”. Addison-Wesley, Reading, Massachusset, 1989.

[5] D. E. Goldberg, “Sizing Populations for Serial and Parallel Genetic Algorithms”. In J. David Schaffer, editor, Proceeding of the Third International Conference on Genetic Algorithms, San Mateo, California, USA. Morgan Kaufmann Publishers, 1989.

[6] F. Davidsson, I. Kosonen, and A. Gutowski, “TPMA-Traffic Performance on Major Arterials”. Royal Institute of Technology. Centre for Traffic Simulation Research. Stockholm 2002.

[7] I. Kosonen, “HUTSIM - Urban Traffic Simulation and Control Model: Principles and Applications”. Doctoral thesis. Helsinki University of Technologie. Transportation Engineering Publications 100. Espoo 1999

[8] T. Ma, B. Abdulhai, “Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters”. Transportation Research Record No.1800. Transportation Research Board, Washington D.C. 2002. pp.6-15.

[9] K. Rasheed, Guided crossover: “A new operator genetic algorithm based optimization”. In Proceedings of the Congress on Evolutionary Computation, 1999.