NISS

Failure Detection and Diagnosis in Micro-Simulation Traffic Models

Baohong Wan, Nagui Rouphail, and
Jerome Sacks

Technical Report Number 154
July 2005

National Institute of Statistical Sciences
19 T. W. Alexander Drive
PO Box 14006
Research Triangle Park, NC 27709-4006
www.niss.org

Failure Detection and Diagnosis in Micro-Simulation Traffic Models

Baohong Wan
Martin/Alexiou/Bryson, PLLC

Nagui Rouphail
Institute of Transportation Research and Education
North Carolina State University

Jerome Sacks
National Institute of Statistical Sciences

July 25, 2005

ABSTRACT

Effective and feasible procedures for validating microscopic, stochastic traffic simulation models are in short supply. Exercising such micro-simulators many times on specific (real) networks may lead to the occurrence of traffic gridlock (or simulation failures) on some or all replications. While the lack of failures may not assure validity of the simulator for predicting performance, the occurrence of failures can provide clues for identifying deficiencies of the simulation model and invite strategies for model improvement.

We define failure as a severe malfunction in one or more traffic links on the network where vehicles are unable to discharge for an unusually long period. Such malfunctions can be detected through the use of link-based time traces of vehicle trips. Identifying locations where malfunctions arise requires further spatial analyses. A procedure for identifying "whether", "when" and "where" failures occur is described. The simulator CORSIM serves as the test-bed simulator for the proposed methodology but the procedure is applicable to any comparable microscopic model; real-world traffic networks are simulated as case studies.

Possible root causes of detected failures are: (1) flaws in the simulator behavioral algorithms, (2) improper calibration of inputs, and (3) capacity problems related to the specific network under study (for example, when heavy traffic demand projections are simulated). Strategies to identify "why" failures occur are used in the case studies.

The results indicate that the proposed failure detection and diagnosis process is an effective (and essential) way to explore the validity of traffic simulators and find improvements where needed.

KEYWORDS
Traffic micro-simulation, traffic anomaly detection, simulation failure, model validation

1. INTRODUCTION

The reliable use of traffic simulations, a key tool in transportation engineering analyses, is often limited by a lack of methods for the assessment of their validity (***1***). While strategies now exist (***2***) to treat comparatively simple deterministic models, as those found in the Highway Capacity Manual (***3***), adequately implemented methodology is lacking for treatment of micro-simulators in current practice.

A sign of possible simulator difficulty is the appearance of unrealistic traffic congestion (even gridlock) in runs set to represent traffic conditions on a real network (e.g.**,** ***4, 5***). The occurrence of such situations, called *simulation failures,* can compromise the validity of the simulation. The issue addressed is how to effectively determine whether, when, where and why failures occur and thereby provide a tool for assessing the validity of simulators and mitigating their shortcomings.

Discovering simulator flaws that lead to failures requires exercising the simulator many times especially if failures occur with small probability. Finding outlier values in a performance measure defined on the simulator runs is a traditional method for discovery but may be inadequate: failure may occur toward the end of a simulation run with only a minor impact on a performance measure and additional methods are still necessary to identify locations and causes of the failures. Visual observation of animations is a powerful discovery tool but is far too laborious (even if feasible) in dealing with many runs for a complex network. A more automatic process is needed.

The approach taken is to scan key traffic features at specific times and locations, analyze the temporal/spatial patterns that arise and identify *anomalies.* The methods described in Section 3 examine the time traces of link trips for each link and identify when, if ever, failure occurs on a link. Links with very small numbers of trips are disregarded. Section 3 also considers identifying the locations in the network where failures first appear. Section 4 treats these possible causes of failures: (a) flaws in the simulator behavioral algorithms, (b) incorrect calibration or specification of inputs, and (c) capacity problems related to the specific network under study (which may appear when heavy traffic demand projections are simulated).

Inaccurate calibration, inappropriate defaults or over-tuning may hide flaws; flaws may prevent accurate calibration. This confounding of causes of types (a) and (b) above is an obstacle to assessing validity and finding simulator flaws, if present (***1***). The results of our approach in several examples indicate how flaws may be found despite efforts to tune key parameters to remove failures. The approach provides a way to both assessing validity and revealing why a simulator may be inaccurate.

Gridlock detection in communications networks and incident detection in transportation networks have similarities to the flaw detection issues addressed here and have points of contact with the procedures we use. Most communication networks are able to collect passive measurements of traffic at routers and switches. Different analysis approaches have emerged to support the detection process using a variety of tools, such as sample statistics (e.g., ***6***), time-series analysis (e.g., *7*), and wavelet analysis (***8***). Automated Incident Detection (AID) systems use traffic data to assess the likelihood of incidents by applying specialized algorithms to the data. The family of incident detection algorithms includes pattern recognition (e.g., the California Algorithms, ***9***), traffic model algorithms such as the McMaster Algorithm (***10***), traditional statistical fitting tests, time series, etc., and artificial intelligence methods (e.g., ***11***).
The methods we propose here are simpler, tailored to the specific issues of failure, and usable without large-scale data acquisition.

The key conclusions: (1) the methods proposed are reasonably effective automatic ways for identifying failures; (2) finding the causes is less direct and employs a blend of straightforward regression techniques and visualization; (3) adjusting default settings of important inputs may not lead to a complete cure – simulator flaws are thereby exposed. The methods we describe can be employed in model development by exercising simulators using a real network with realistic enough inputs designed to produce congested conditions, creating a process of probing the simulator for weaknesses under stress.

2. TEST-BED

The principal test-bed we use is the traffic simulator TSIS/CORSIM 5.1 (***12***). We will also show examples for earlier versions of CORSIM namely, versions 4.3.2 and 5.0, in order to make several cogent points as well as showing the suitability of the methods for different simulators. The network used is a surface street (sub-) network of the city of Chicago, Illinois connecting a major freeway with the central business district (FIGURE 1).

The network contains 56 surface street segments, two freeway segments, 24 signalized intersections, and 8 un-signalized intersections. About 13,500 vehicles entered the study network during one-hour data collections in the evening and morning peak periods. Congestion was observed more frequently during the evening peak period than in the morning.

Each one-hour simulation period (one for the evening, one for the morning) was divided into 12 five-minute time intervals. Since traffic signals operated on a common cycle length of 75 seconds, the simulation outputs in each time interval were aggregated over a period of four signal cycles.

Signal timing, network geometry, demand and turning movements were obtained from the Chicago Department of Transportation and from field (video) observations. Readily measured traffic parameters, such as free-flow-speed, were input based on field observations (***13***). Initially, default settings are used for all other immeasurable (i.e., too expensive or hard to measure/estimate in the field) traffic behavior parameters. Some of these parameters are later tuned to mitigate failures observed in the initial case studies; the details of this process are described more fully in Section 4.

The parameters we focus on for tuning are queue discharge headway (QDH), start up lost time (SLT), probability of vehicle becoming the first (second) vehicle in spillback (SbPr1, SbPr2), left-turn lagging probability if left-turner is within 2 (4) seconds after start of NO GO interval (LLPr1, LLPr2), probability of lead left-turn vehicle jumping across all oncoming lanes at beginning of green phase (LJPr). Other behavioral parameters could have been considered; the ones we selected seemed most closely associated with traffic breakdowns observed in animations. Note that QDH is often deemed measurable in the field. But since it is link specific in CORSIM, it will become too expensive to measure QDH in the more than one hundred links of the test-bed network, particularly in cases that are quite complex (such as shared through/left lanes) .

The performance measure we use is System Queue Time (SQT) which accumulates delay on links whether vehicles discharge or not. It is tempting to believe that failures and large values of SQT are synonymous but we will see that is not necessarily so.

The distinctions among the CORSIM versions 4.3.2, 5.0 and 5.1 are numerous. Among the most relevant are these:

- From 4.3.2 to 5.0 changes were made to modeling stop sign and yield sign control; the probability of joining spillback was extended to left turning vehicles.

- From 5.0 to 5.1 the logic for joining spillback was changed and default values were changed.

These changes lead to differently performing simulators as will be seen below.

3. DETECTION OF FAILURES

To develop our method we select as a key indicator the number of link-trips per unit time. Other commonly available simulator outputs could have been used but link-trips are simple, appropriate and effective for our purposes. By definition, link-trips in a unit time interval is the number of vehicles *discharged* from the link during the time interval. Denote this by ***LTO*** (link-trips-out). The number of link-trips-in (***LTI***) is the number of vehicles *entering* the subject link during the same time interval. For time interval ***r*** and link ***(i,j)*** we use $\boldsymbol{LTO}^{r}_{(i,j)}, \boldsymbol{LTI}^{r}_{(i,j)}, \boldsymbol{LC}^{r}_{(i,j)}$ to denote, respectively, link-trips-out, link-trips-in and link (vehicle) content at the end of the time interval (of course $\boldsymbol{LC}^{r+1}_{(i,j)} = \boldsymbol{LC}^{r}_{(i,j)} + \boldsymbol{LTI}^{r+1}_{(i,j)} - \boldsymbol{LTO}^{r+1}_{(i,j)}$). In CORSIM, as in other simulators, ***LTO*** and ***LTI*** are readily collected for each link and each time interval.

If there are minor fluctuations in traffic demand, the stochastic simulation model should be in statistical equilibrium and the number of vehicles entering a link should approximate the number of vehicles exiting from it. If, however, a simulation failure occurs, causing ***LTO*** to drop, the link will fill up and ***LTI*** will fall to 0 as well. Spillback will then form and traffic on upstream links will be affected. Until the initial vehicle blockage dissipates, vehicles entering the bottleneck areas will exacerbate the situation. These considerations lead to the formal definition: a link ***(i, j)*** is blocked starting at time interval ***r*** if

$$\begin{aligned} &(a)\ \boldsymbol{LTO}^{t}_{(i,j)} = 0;\ \text{all}\ t \geq r \\ &(b)\ \boldsymbol{LC}^{r}_{(i,j)} > 0 \end{aligned} \qquad (1)$$

In Eqn (1) (a) implies that recovery does not take place subsequent to time ***r*** while (b) is necessary to avoid treating empty links as being blocked.

A simulation failure will typically result in spillback over a large portion of the network with blockages at several locations. The most critical links are usually those that fail earliest. Links having the earliest failure time are named *first-failing links*, and the associated nodes (downstream end of first-failing links) are referred to as *first-failing nodes*. There may be multiple first-failing links and nodes in a given replication, especially if the unit time intervals are lengthy. If the simulator is run *n* times (replicating the same inputs) we record the first-failing frequencies for each link ***(i, j)*** and node ***k***:

$$\begin{aligned} \boldsymbol{FFL}_{(i,j)} &= \sum_{p=1}^{n} \boldsymbol{FFL}^{p}_{(i,j)} \\ \boldsymbol{FFN}_{k} &= \sum_{p=1}^{n} \boldsymbol{FFN}^{p}_{k} \end{aligned} \qquad (2)$$

($\boldsymbol{FFL}^{p}_{(i,j)}$ and $\boldsymbol{FFN}^{p}_{k}$ indicate absence or presence of first-failure and are either 0 or 1).

Example 1. CORSIM 5.1; P.M. Peak; Defaults

With this setting 100 runs (the choice of how many runs to make is discussed below) were made, 38 of which manifested failures that is, runs where at least one link is blocked starting at some ***r*** $\leq$ 12 (Eqn (1)). This is a surprisingly large number of simulation failures considering that the real network exhibited no spillback during the one-hour period examined, though video records

revealed substantial congestion on various parts of the network. An interesting and somewhat unexpected pattern emerges from a histogram of System Queue Time (SQT) () gathered from the simulator output: there are 13 outliers (those with SQT > 350) and while very large SQTs are typically associated with failure, many failures occur at moderate values of SQT. This undercuts the utility of equating outlier runs to simulation failures.

Also interesting is that some runs had no failures but very large SQT, indicating that complete blockage had yet to occur on some links by the end of the simulation period. A refinement of Eqn (1) in which failures are found if there are major declines in output, perhaps not reaching 0, can be implemented but we forego doing so.

Recording and plotting the frequencies of first-failing links and first-failing nodes (Eqn (2)) provide valuable spatial information. TABLE 1 records those links and nodes which appear a large number of times and FIGURE 3(a) translates this information into a plot that clearly shows the location of problematic behavior.

Example 2. CORSIM 5.1; Morning Peak; Defaults

Here only 3 of the 100 simulation runs contained failures. The pattern of failures is exhibited in FIGURE 3(b), similar to what was done for FIGURE 3(a) but the failures appear in a different area than in the evening.

To exhibit the utility of the procedure for comparing different simulators we utilized the same network and information with other, earlier, versions of CORSIM.

Example 3. CORSIM 4.3.2; Evening Peak; Defaults

Here the number of failures among the 100 runs was 13 and a spatial analysis similar to that leading to FIGURE 3 is exhibited in Figure 4(a)

Example 4. CORSIM 5.0; Evening Peak; Defaults

In this instance failures occurred in 32 of the replicate runs and Figure 4(b) exhibits the spatial characteristics. Though the frequencies of the two spatial distributions in Figure 4 differ, there is a concentration on the Franklin and Orleans corridors. Comparing with FIGURE 3(a) we see that with CORSIM 5.1 there is some considerable shift away from the Franklin corridor. These differences reflect changes in default values from simulator to simulator and in the behavioral algorithms. In any case the depicted results show the varying behavior across simulators, indicate where difficulties arise and carry us to the next step of determining the causes and potential changes.

4. CAUSES OF FAILURE

Examining the animations connected with failure runs (which should be viewed starting at least one time interval before the first failure interval) can be instructive for identifying simulator behavior (or misbehavior) leading to failure. In Examples 3 and 4 northbound left-turn vehicles on Franklin Street joined downstream queues on Grand at the end of green time thereby blocking the intersection and creating spillback that propagated throughout the area. In Example 1 left turn southbound vehicles on Orleans at Illinois were obstructed by heavy opposing through traffic, causing the left turn bay to fill up and blockages to form. The link between Grand and Illinois on Orleans is focused upon below; we label it as *L1* (FIGURE 1).

The animations thus cast suspicion on how left-turn laggers and spillback are treated by the simulator. Changes that were made to get to version 5.1 had been directed towards these issues and while difficulties along Franklin Street were cured, problems on Orleans were not.

Questions raised by this set of circumstances are whether the default values for inputs accurately represent real conditions, can be tuned to reasonable values leading to removal of failures, or whether the simulator has intrinsic flaws that require algorithmic changes. Responding to the first two questions we select inputs that seem to be of concern, determine ranges of plausible values for them and then choose values for the inputs that reduce, possibly to 0, the frequency of failures.

Choosing a final, good set of inputs requires evaluating the simulator output over the ranges of the inputs. Doing so directly, or by trial and error, is problematic (even if feasible) because many input combinations must be evaluated and it may take many replications to evaluate each such combination. Instead we proceed by performing a computer (simulator) experiment as follows:

- Choose a set of combinations of selected inputs (*design* the simulator experiment)
- For each combination obtain 100 simulator replicate runs (100 is selected somewhat arbitrarily; see comment in Section 6)
- Fit a *response surface* model to the output (= proportion of failure runs) as a function of the inputs
- *Optimize* the response surface model over the feasible region of inputs to produce a small value for the output
- Run the simulator 100 times at the optimized setting to *confirm* the predicted value obtained by optimizing the response surface approximation

In fact, we proceeded in stages by first singling out the key local variables QDH and SLT to examine their effect and then followed up by considering the other behavioral inputs (Spillback, Left-turn Lagging, Left-turn Jumping). Both QDH and SLT are local (can be specified for each link) while the others are global (same for all links).

Experiment 1: CORSIM 5.1; Evening; Defaults except for QDH and SLT on the link *L1* (Orleans between Grand and Illinois).

The default value for QDH is 1.8 sec with allowable range [0, 3.5] but, in the network studied, the possible values are realistically covered in {1.6, 1.8, 2.0}. The SLT default is 2.0 sec with allowable range [0, 4.0] but with realistic values in {1.7, 1.8, 1.9, 2.0, 2.1}.

Rather than carry through the process described above it is simpler here and more direct to exercise the simulator at each of the 15 possible combinations of QDH and SLT. The results obtained are in Table 2. Except for statistical fluctuations it is clear (and unsurprising) that failures decline as QDH and SLT are reduced. Drastic reduction from the default values is unlikely to hold in reality (and indeed doesn't as shown in Lin, ***14***) but we nonetheless make some reduction to assess whether such tuning has an effect. Accordingly we chose QDH = 1.7 and SLT = 1.8. The predicted value for failure probability can be obtained by linear interpolation of the values 0.36 and 0.25 at (QDH, SLT) = (1.6, 1.8) and (1.8, 1.8) (or by more sophisticated methods). An additional 100 runs at these new values (1.7, 1.8) of (QDH, SLT) and the default values of the other 5 inputs produced a failure rate of 0.32, a value consistent with that predicted. The failure rate of ~0.30 is still quite large and we then move on to the next experiment.

Experiment 2: CORSIM 5.1; Evening; Defaults for QDH and SLT except on the link *L1* where QDH=1.7, SLT=1.8, defaults for all other parameters except SbPr1, SbPr2, LLPr1, LLPr2, LJPr, the spillback and left-turn parameters.

Design

The range of values for the five parameters to be varied is in Table 3. Knowledge of the particular network suggested that the default for LLPr1 may be too low and that for LJPr too high for the conditions during the peak period. Since covering all combinations of the values in Table 3 requires a prohibitive number of runs a selection of combinations is made by use of a Latin Hypercube Design (***15***) of size 30, far fewer than the 3750 possible combinations. Such designs have good space filling properties and have found extensive use in computer experiments. The 30-point design is given in TABLE 3 along with observed proportions of failure from 100 runs at each design point (the column labeled simulation).

Response Surface Fit

There are many possible methods for fitting a function of the five variables to the observed data in Table 3. We opt for a simple approach here and use a backward stepwise regression allowing quadratic terms. The result is the linear regression equation

$$FPr = -0.929 + 1.56\ SbPr1 + 0.312\ SbPr2 - 0.515\ LJPr \qquad (3)$$

Neither quadratics nor interactions carry much weight; an R^2 of .85 for Eqn (3) assures a good quality of fit. Used on the 30 inputs in Table 3, Eqn (3) produces the column labeled "regression predictions" in Table 3.

Optimization

Using the fitted function of Eqn (3) to optimize leads to lower boundary values 0.70 for SbPr1 and 0.20 for SbPr2 and the upper boundary value 0.38 for LJPr. The simplicity of the linear relationship makes optimization a trivial matter. The predicted value from Eqn (3) for FPr at these new parameter values (LLPr1, LLPr2 set at defaults) is 0.03, a great improvement over the original 0.38.

Confirmation

Experiment 3. CORSIM 5.1; Evening; (SbPr1, SbPr2, LJPr) = (0.70, 0.20, 0.38); (QDH, SLT) = (1.8, 2.0) except on link *L1* where (QDH, SLT) = (1.7, 1.8).

Again 100 runs of CORSIM were made for this tuned network producing 8 failure runs, slightly higher but within statistical variation of the 0.03 failure rate predicted by the regression fit of Eqn (3). The histogram plot corresponding to Figure 3 is given in Figure 5. The region of difficulty (Orleans, Grand, Illinois) remains the same; the tuning cannot eliminate the difficulty that CORSIM has in clearing obstacles to left-turns against heavy oncoming traffic. Interestingly, a histogram SQT for these 100 simulations exhibits no extreme values even though failures do occur on important links of the network.

Because the tuned values are at the extreme end of the realistic intervals for the parameters we might expect even worse behavior if these parameters were more accurately imputed. For example, LJPr is likely to be far less than 0.38 in which case the failure rates would increase significantly. Recall that LJPr represents the probability that a lead left turn will actually enter the intersection *prior* to the opposing through traffic in a permissive phase. Our observations of the Chicago network indicate that such behavior is rare (limited sampling at two

key intersections on the Chicago network indicate values closer to 0.10). The behavior also is unlikely to be "global", but should be highly dependent on the particular geometry of the intersection which may encourage/discourage this behavior. What is clear however, is that without having a large fraction of such "jumpers", the CORSIM simulation model is likely to yield failures, as evident from the surface response model

It is worth noting that using default values for QDH, SLT everywhere or using the values QDH=1.7 and SLT=1.8 everywhere lead to similar numbers of failures (eight) in the same region. Even the use of a 25% drop in the size of acceptable gaps (amply justified by the observed aggressive driver behavior on this network) does not eliminate failures altogether. Many of these parameters clearly have only a small effect on failures of the type we find though they do affect performance such as SQT.

5. "STRESSING" THE SIMULATOR

By increasing the demand a simulator can be forced to collapse. As one approaches critical values for demand the stress on the simulator can reveal flaws that may not have been apparent under less than capacity conditions.

Example 5. CORSIM 5.1; Morning; Parameters of tuned network of Experiment 3.
No failures were found in 100 runs of CORSIM. This is not surprising since only 3 failures were found using all parameters set at defaults (Example 2). Though the 3 failures found exhibited some flaws the tuning would appear to have removed the problem. We proceeded to then experiment by increasing demand and retaining the same turning percentages as measured in the field. Demand increases (uniformly at each entry node of the network) of 5% and 10% produced no failure runs but an increase of 15% produced a failure rate of 0.40 with a spatial distribution of first-failing intersections given in Figure 6. Viewing animations of the failed runs revealed the problem:

- Traffic congestion on the link *L2* (in Figure 1) of Erie from LaSalle to Clark (one-way traffic) caused spillback to the upstream intersection of Erie at LaSalle.
- The problem link *L2* was controlled by a stop sign at Erie and Clark that gave priority to the one-way southbound traffic on Clark
- Under higher demand on Clark long queues formed on *L2* and led to spillback and blockage upstream.

Further analyses were not pursued but the conclusion here is that, by stressing the simulator, potential difficulties (for example, caused by anticipated growth in demand over time) may be revealed. [Note: Recognition of the problem at Erie and Clark led the Chicago Department of Transportation to replace the stop control by signal control.]

6. COMMENTS AND CONCLUSIONS

6.1 False Alarms

The notion of "false alarm" where a failure is found but the simulator is "innocent" and the failure is a result of an unusual juxtaposition of unfavorable random occurrences would seem to be relevant in our analyses. When there are small flows on a link it may be that ***LTO*** is 0 purely by chance. To ward off uninformative analyses in such situations we exclude from consideration those links with small hourly flows, say less than 96. A link with that flow would, under the assumption that flows are Poisson, have Probability[***LTO*** =0] = e^{-8} = .00034 in a single 5 minute

time interval and an expected number of (random) failures in 100 replications of 12 time intervals = 1200 • 0.00034 = 0.41. Random false alarms for more active links drop dramatically (for flows of 120 vehicles per hour the expected number of random failures is 0.06). The actual number of false alarms for a given link would be much smaller since link-trips-out must remain at 0 until the end of simulation.

6.2 Number of Replications

Our choice of 100 as the number of replications in all the analyses is somewhat arbitrary. One consideration is that with 100 replications the estimate of failure rate is a binomial proportion with a 95% confidence interval of less than ± 0.10. Thus, in Example 1, we have 95% confidence that the actual failure rate is between 0.28 and 0.48 and that failure is not due to chance alone. Moreover, with 100 replications we would be almost (95%) certain to detect the presence of failures if their actual rate of occurrence is as small as 0.06. We could have used fewer replicates with smaller confidence or more replications to gain sharper estimates. Eventually what matters is the discovery (or lack) of flaws; formal methods for selecting the number of replications to accomplish the task are not yet developed.

The use of 100 replications in the analyses leading to Eqn (3) (the fitting of the response surface) is again arbitrary but driven by the need to acquire some precision about the output response to the given inputs. Plausibly, one can arrange the 3000 runs in a different way by covering more (than 30) input combinations with fewer replications. Again this is an issue without clear guidance for resolution and our selection of 100 is intuitively based.

6.3 Response Surface

We used a least squares stepwise regression to obtain the surface of Eqn (3). More complex methods may be needed in other applications and there are a slew of available techniques. One approach could be to utilize the methods deployed in Bayarri et al (***16***) to provide a sophisticated approximation of the mean response (failure rate) to the set of inputs. The examples in this paper are amenable to the simpler approach taken in Section 4.

6.4 Other Inputs

We also considered the addition of gap acceptance parameters to those used in the examples above. These are global parameters (cannot be changed only on local links) and while they may mitigate the problem they would, at the same time, produce inaccurate inputs for the network as a whole. For example, when an opposing link exhibits severe congestion left-turn gap acceptance may become more skewed to very low values, but such values are inappropriate for less congested conditions elsewhere in the network or even at other time intervals on the same link. We probed this issue for the situation described in Example 1 and Experiment 2 and reduced the gap acceptance values by factors of 0.9 and 0.75 for the setting of Experiment 2. The results were a reduction in failures to 7 and 3 respectively but the failures persisted in the same region: the flaw is not easy to overcome.

6.5 Validity

It is tempting to conclude that the presence of failures after optimum tuning establishes the invalidity of the simulator. But that may be premature since it is quite possible, despite the appearance of failures, that the simulator predicts specific relevant performance with accuracy. On the other hand if, after tuning, the failure rate is 0 we cannot assert validity of the simulator.

One reason: the simulator may predict specific performance inaccurately even if failures are absent. This occurs in Example 5 where failures are reduced to 0 but average stopped time per vehicles stopped (STVS) on eastbound (one-way) Ohio at LaSalle is 3.3 sec compared to 15.4 observed in the field (***13***) and on southbound LaSalle at Ohio the CORSIM STVS is 6.8 sec compared to 27.8 observed in the field. These discrepancies are extreme and point out that tuning can produce unrealistic parameter inputs and unrealistic predictions of performance.

Our methods are aimed to uncover flaws and while they assist a process for validation, they cannot substitute for the more encompassing approach described in Bayarri et al. (***16***), where computer model discrepancies and calibration are analyzed together to produce effective assessments of validity. As noted earlier, this encompassing approach has yet to be implemented for complex stochastic simulators, and is an open area for study.

7. SUMMARY

In this paper we have demonstrated a failure detection method to identify the occurrence of simulation failures and their causes. This capability can guide the process of micro-simulation model calibration and validation and have significant value to model developers and users alike though there remains the need for implementation of a comprehensive approach to calibration/tuning and validation.

8. ACKNOWLEDGEMENTS

This research was sponsored by National Science Foundation Grant DMS-0073952 to the National Institute of Statistical Sciences.

9. REFERENCES

1. Bayarri, M., J. Berger, G. Molina, N. Rouphail and J. Sacks, "Assessing Uncertainties in Traffic Simulation: Key Component in Model Calibration and Validation," *Transportation Research Record 1876*, TRB, Washington D.C., pp.32-40. 2004.
2. Paulo, R., J. Lin, N. Rouphail and J. Sacks, "Calibrating and Validating Deterministic Traffic Models: Application to the HCM Control Delay at Signalized Intersections", *presented at the 84th Annual Meeting of TRB* Washington, DC, January 2005.
3. Highway Capacity Manual, TRB 2000, National Research Council, Washington, DC.
4. Park, B., N. Rouphail and J. Sacks, "Assessment of Stochastic Signal Optimization Methods using Microsimulation." *Journal of Transportation Research Board*, Washington, D.C., No. 1748, pp. 40-45. 2001.
5. Rouphail, N., B. Park and J. Sacks, "Direct Signal Timing Optimization." *Proceedings, XI Pan-American Conference in Traffic and Transportation Engineering*, Gramado, Brazil, November 19-23, pp. 195-206. November 2000.
6. Feather. F, Siewiorek, D., and Maxion, R., "Fault detection in an ethernet network using anomaly signature matching." *Proceedings of ACM SIGCOMM '93*, San Francisco, CA, September 1993.
7. Brutlag, J. D. "Aberrant behavior detection in time series for network monitoring." *Proceeding in 14th Systems Administration Conference (LISA 2000)*, December 2000.
8. Barford, P., Kline, J., Plonka, D., and Ron, A., "A signal analysis of network traffic anomaly." *Proceeding of ACM SIGCOMM Internet Measurement Workshop,* Marseilles, France. November 2002.

9. Payne, H. J., Helfenbein, E. D. & Knobel, H. C., "Development and testing of incident-detection algorithms: Vol. 2, research methodology and detailed results." *Report FHWA-RD-76-12,* Federal Highway Administration, Washington, D.C. 1976.
10. Masters, P. H., J.K. Lam, and K. Wang, "Incident Detection Algorithms for COMPASS – An Advanced Traffic Management System," *Proc. Vehicle Navigation and Information System Conf.*, Dearborn, MI, pp. 295-310. 1991.
11. Ritchie, S.G., R. L. Cheu, and W. W. Wrecker, "Freeway Incident Detection Using Artificial Neural networks," *Engineering Foundation Conf*., Ventura, CA, 1992.
12. ITT Industries, Systems Division. *TSIS User's Guide. Version 5.1*. March 2002.
13. Sacks, J., N. Rouphail, B. Park and V. Thakuriah, "Statistically Based Validation of Computer Simulation Models in Traffic Operations and Management", *Journal of Transportation and Statistics*, Vol 5 (1), pp. 1-15. 2002.
14. Lin, J., "Assessing the Value of Model Calibration for Signalized Intersections". M.S. Thesis, North Carolina State University. August 2004.
15. T. Kollig and A. Keller. Efficient Multidimensional Sampling. *Computer Graphics Forum*, 21(3):557--563, September 2002.
16. M.J. Bayarri, James O. Berger, David Higdon, Marc C. Kennedy, A. Kottas, Rui Paulo, Jerome Sacks, James A. Cafeo, James Cavendish, C.H. Lin, J. Tu. *A Framework for Validation of Computer Models.* NISS Technical Report #128. October 2002.

LIST OF TABLES

TABLE 1 First-Failing Links and Nodes in CORSIM 5.1 Simulation

First-Failing Locations		Frequency of First Failures	Percentage of Total Failures
First-Failing Links	Southbound Orleans St at Grand Ave	19	50%
	Southbound Orleans St at Illinois St	14	37%
	Eastbound Grand Ave at Orleans St	13	34%
	Northbound Kingsbury St at Illinois St	9	24%
	Southbound Kingsbury St at Illinois St	8	21%
First-Failing Nodes	Orleans St at Grand Ave	25	66%
	Orleans St at Illinois St	18	47%
	Grand Ave at Kingsbury St	9	24%

TABLE 2 Simulation Results Varying QDH and SLT; 100 Runs at Each Combination

QDH	SLT	Proportion of Failures
1.6	1.7	0.270
1.6	1.8	0.360
1.6	1.9	0.360
1.6	2.0	0.400
1.6	2.1	0.430
1.8	1.7	0.190
1.8	1.8	0.250
1.8	1.9	0.430
1.8	2.0	0.360
1.8	2.1	0.560
2.0	1.7	0.250
2.0	1.8	0.320
2.0	1.9	0.450
2.0	2.0	0.490
2.0	2.1	0.500

TABLE 3 Computer Experiment: Design and Results

Observation	SbPr1[1]	SbPr2[2]	LlPr1[3]	LlPr2[4]	LjPr[5]	Proportion of Failures	
						Observed in Simulation	Predicted by Regression
1	0.70	0.40	0.80	0.45	0.32	0.07	0.12
2	0.75	0.50	0.50	0.45	0.00	0.46	0.39
3	0.90	0.20	0.90	0.05	0.32	0.37	0.37
4	0.85	0.60	0.70	0.45	0.38	0.43	0.39
5	0.75	0.30	0.70	0.15	0.16	0.23	0.25
6	0.70	0.30	0.50	0.25	0.32	0.09	0.09
7	0.70	0.50	0.80	0.15	0.38	0.14	0.12
8	0.80	0.20	0.80	0.25	0.08	0.34	0.34
9	0.75	0.60	0.60	0.25	0.16	0.39	0.34
10	0.70	0.20	0.70	0.35	0.24	0.09	0.10
11	0.75	0.20	0.80	0.25	0.08	0.27	0.26
12	0.90	0.40	0.90	0.05	0.00	0.54	0.60
13	0.75	0.40	0.90	0.25	0.24	0.16	0.24
14	0.80	0.50	0.70	0.05	0.16	0.37	0.39
15	0.90	0.60	0.80	0.35	0.16	0.60	0.58
16	0.85	0.30	0.50	0.15	0.24	0.36	0.36
17	0.80	0.20	0.60	0.45	0.00	0.28	0.38
18	0.80	0.40	0.60	0.35	0.38	0.15	0.24
19	0.85	0.20	0.60	0.15	0.38	0.24	0.26
20	0.70	0.30	0.70	0.05	0.08	0.25	0.21
21	0.85	0.30	0.50	0.35	0.00	0.58	0.49
22	0.90	0.40	0.80	0.15	0.00	0.64	0.60
23	0.90	0.60	0.60	0.45	0.32	0.44	0.49
24	0.80	0.60	0.90	0.35	0.08	0.40	0.46
25	0.80	0.50	0.60	0.05	0.24	0.35	0.35
26	0.90	0.50	0.50	0.35	0.24	0.45	0.50
27	0.85	0.50	0.90	0.25	0.38	0.42	0.35
28	0.75	0.40	0.70	0.05	0.32	0.36	0.20
29	0.70	0.60	0.50	0.15	0.08	0.24	0.31
30	0.85	0.30	0.90	0.45	0.16	0.47	0.40

Note:
1. CORSIM default 0.80, possible value (0.70, 0.75, 0.80, 0.85, 0.90) assumed;
2. CORSIM default 0.40, possible value (0.20, 0.30, 0.40, 0.50, 0.60) assumed;
3. CORSIM default 0.50, possible value (0.50, 0.60, 0.70, 0.80, 0.90) assumed;
4. CORSIM default 0.15, possible value (0.05, 0.15, 0.25, 0.35, 0.45) assumed;
5. CORSIM default 0.38, possible value (0, 0.08, 0.16, 0.24, 0.32, 0.38) assumed.

LIST OF FIGURES

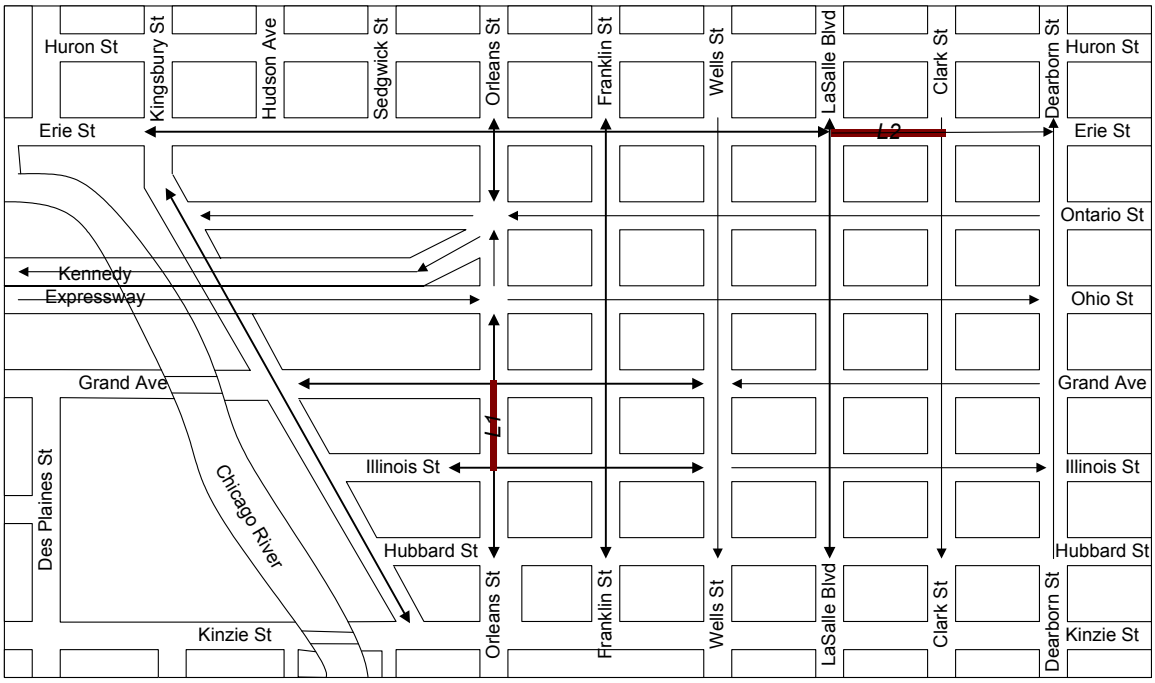

FIGURE 1 Test-bed Network in Chicago.

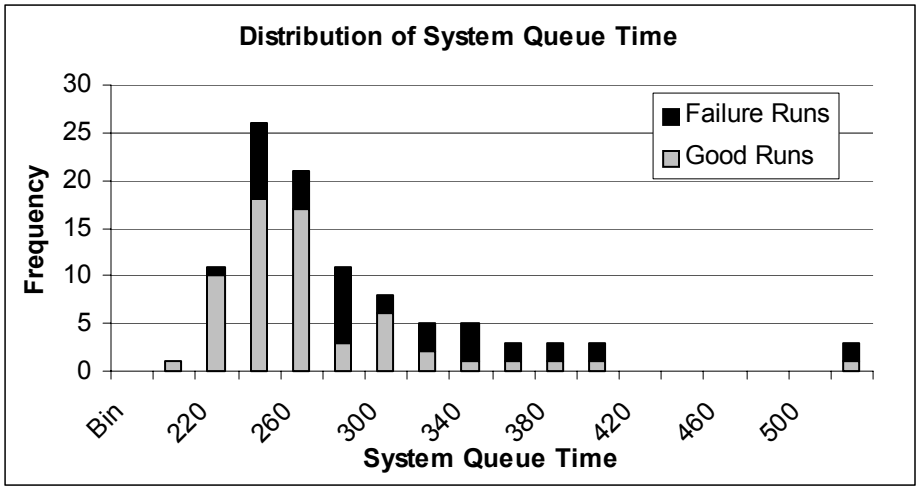

FIGURE 2 Distribution of System Queue Time in CORSIM 5.1 Simulation.

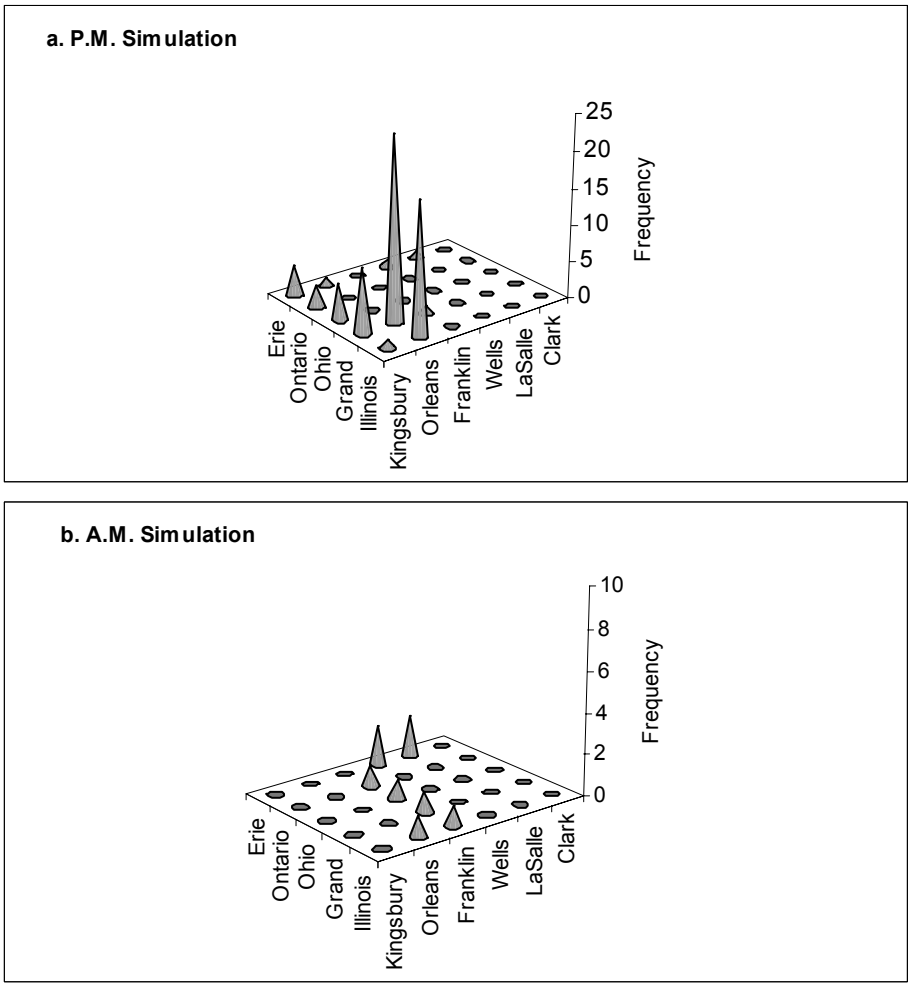

FIGURE 3 **Spatial Distribution of First-Failing Intersections in CORSIM 5.1 (a) P.M. and (b) A.M. Simulations.**

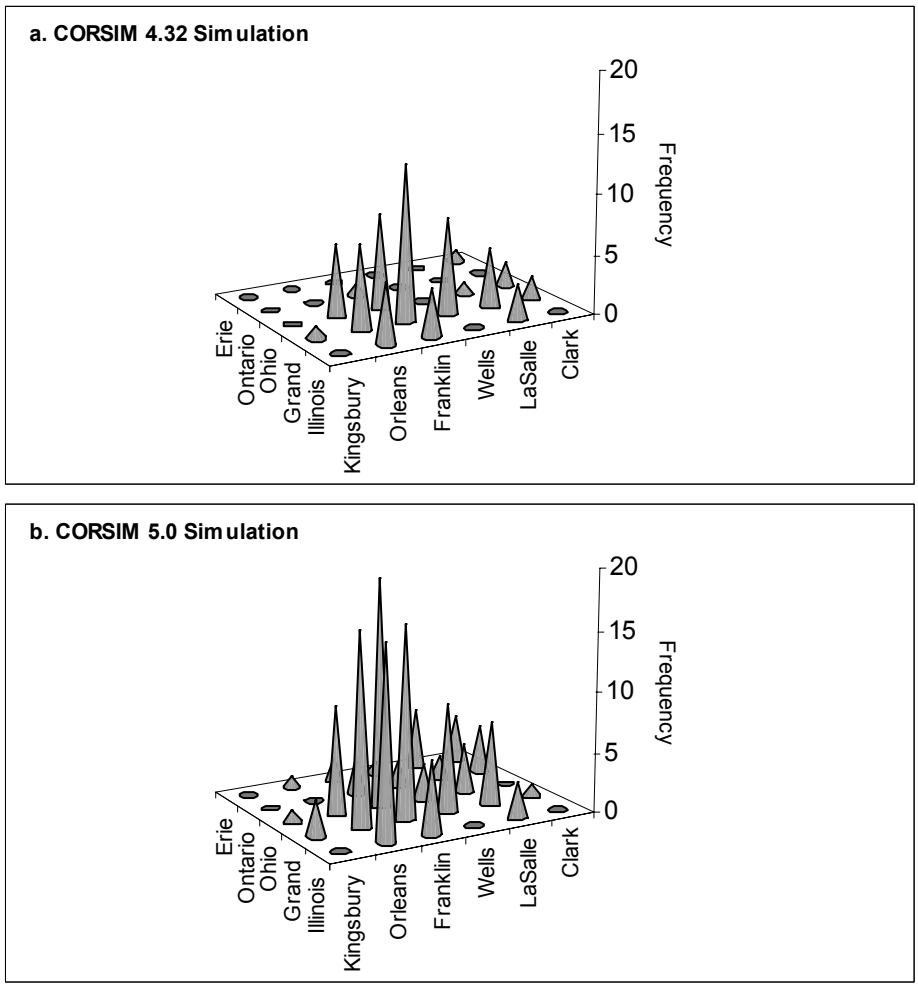

FIGURE 4 **Spatial Distribution of First-Failing Intersections in (a) CORSIM 4.32 and (b) CORSIM 5.0 Simulations.**

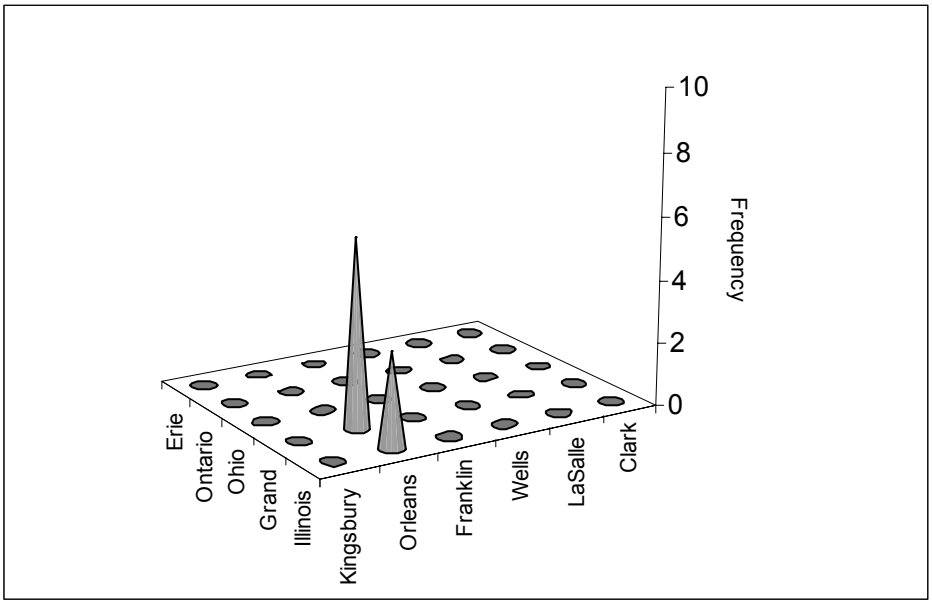

FIGURE 5 Spatial Distribution of First-Failing Intersections with Tuned Traffic Parameter Values.

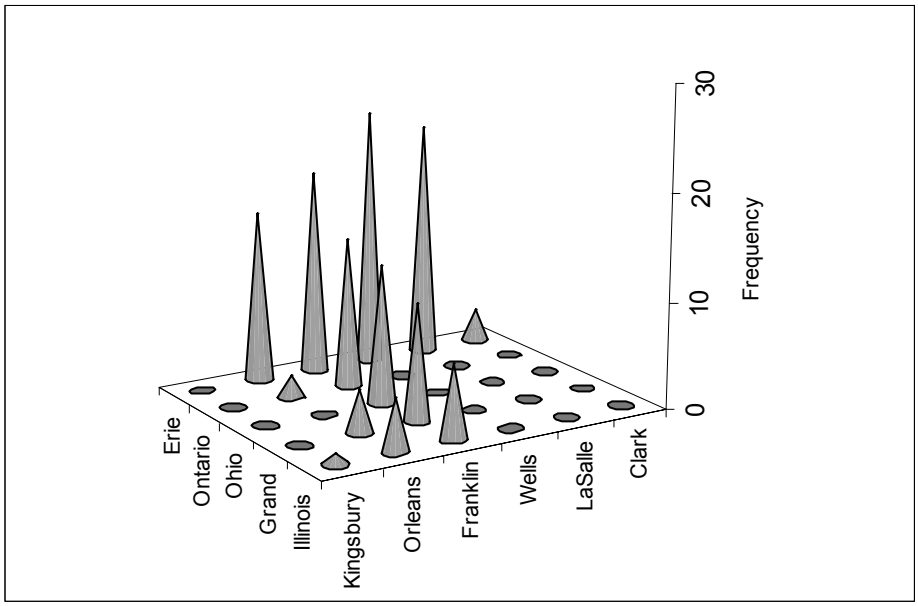

FIGURE 6 Spatial Distribution of First-Failing Intersections; Increased (15%) Traffic Demand in A.M.