LA-UR 97-3531
Approved for public release; distribution is unlimited

TRANSIMS traffic flow characteristics

Kai Nagel, Paula Stretz, Martin Pieck, Shannon Leckey, Rick Donnelly,

Christopher L. Barrett

TRANSIMS traffic flow characteristics

Kai Nagel[†], Paula Stretz[†], Martin Pieck[†], Shannon Leckey[†], Rick Donnelly[†,*], Christopher L. Barrett[†]

[†] Los Alamos National Laboratory, Mail Stop M997, Los Alamos NM 87545, USA, (505) 665-0921 phone, (505) 982-0565 fax, kai@lanl.gov

[*] Parsons Brinckerhoff, Inc., 5801 Osuna Road NE # 220, Albuquerque NM 87109, USA

PREPRINT March 19, 1999

Abstract

Knowledge of fundamental traffic flow characteristics of traffic simulation models is an essential requirement when using these models for the planning, design, and operation of transportation systems. In this paper we discuss the following: a description of how features relevant to traffic flow are currently under implementation in the TRANSIMS microsimulation, a proposition for standardized traffic flow tests for traffic simulation models, and the results of these tests for two different versions of the TRANSIMS microsimulation.

keywords: traffic simulation, traffic flow, intersections

I. INTRODUCTION

One could probably reach agreement that the traffic flow behavior of traffic simulation models should be well documented. Yet, in practice, this turns out to be somewhat difficult. Many traffic simulation models are under continuous development, and the traffic flow dynamics documented in a certain publication is often a "snapshot", valid at the time of writing, but no longer the true state of the model.

It thus makes sense to agree on a certain set of tests for traffic flow dynamics which should be run and documented together with "real" results. In this paper, we propose a (probably incomplete) suite of traffic flow measurements. Also, some of the results in this paper are arguably unrefined with respect to reality. Yet, as we stated above, we are continuously working on improvements, and this publication represents both a snapshot of where we currently stand and an argument for a standardized traffic flow test suite for simulation models. We hope that this publication will both open the way for a constructive dialogue on which standardized traffic flow tests should be run for traffic simulation models, and which of the features of our traffic simulation models may need improvement.

This paper starts with a general section on validation and calibration (Sec. 2), followed by a high-level description of the TRANSIMS microsimulation approach (Sec. 3). Sec. 4 is a fairly technical description of the actual implementation. Sec. 5 contains a description of the test cases that we ran for this paper and presents the simulation results. Sec. 6 contains an example of parameter sensitivity testing for the case of a yield sign, followed by a short section outlining differences in the logic when the simulation was used for the so-called Dallas case study (Sec. 7). The paper is concluded by a discussion section and a summary.

II. VALIDATION, CALIBRATION, ETC.

Prerequisite of any simulation model to be used is a certain amount of confidence in its output. The process of building confidence depends on human nature and is sometimes hard to explain. Yet, an organized process towards model acceptance would help. Such an acceptance process may be composed of the following four elements [1]:

- **Verification** – have the hypothesized behavioral rules been implemented correctly?
- **Validation** – do the hypothesized behavioral rules produce correct emergent behavior, such as correct fundamental diagrams? Note that this does not specify a quantitative procedure; plausibility, consistency with theory and experience, and documentation of emergent behavior are the important elements here.
- **Calibration** – have the model parameters been optimized to (possibly site specific) settings? This requires a decision on a data set and a decision on an objective function that can *quantify* the closeness of the simulation to the data set.
- **Accreditation** – Given a question, is the model indeed powerful enough to help with it?

Note that this process is not uni-directional. For example, if one cannot calibrate a model very well for a given scenario and a given objective function, one will go back and change the microscopic rules and then have to go through verification and validation again.

Also, a formally correct verification process can be shown to be mathematically hard or computationally impossible except in very simple situations (see, e.g., Chapters 14 - 16 in [2]). Intuitively, the problem is that seemingly unrelated parts of the implementation can interact in complicated ways, and to exhaustively test all combinations is impossible. For that reason, both practitioners and theoreticians suggest that one needs to allocate resources intelligently between verification and validation.

Sometimes, the word "validation" is also used when a simulation model, *after* calibration to a scenario and data set A, is run under another scenario to test its predictive performance. Since this represents in principle the same procedure – run the simulation model against a scenario without further adjustment in the process – we do not see a problem in the use of the word validation in both cases.

Next, one needs to decide on which networks to run the above studies. The following seem to be useful:

- **Building block cases** such as "traffic in a loop" or "traffic through yield sign". The chapters of the Highway Capacity Manual [3], despite being under discussion, seem to be a good starting point here. Maybe these cases will not be very useful for calibration since "clean" data on these cases is difficult if not impossible to obtain. Yet, these cases would certainly allow plausibility check of a simulation model, and comparison to other simulation models.

- **Complicated test cases**, which test a variety of behavior such as merging or traffic signals, in a larger *context* (i.e. when interacting). It would be nicest to have test cases from the real world, together with real data. – These test cases would best be made electronically available.

Of course, models have always been validated and calibrated, e.g. [4–6]. For fluid-dynamical models, calibration can be formalized [7,8]. Yet, we would like to stress that there are two diverging tendencies here:

- Models which are simple (i.e. have few parameters) are easy to be formally calibrated in the sense that one can adjust the parameters so that some objective function is minimized. Yet, the model may be too simple to indeed reflect the "meaning" of the data.[1]

- Models which have many parameters are in principle capable of representing a much wider variety of dynamics. Yet, they are difficult for formal calibration because the degrees of freedom are too large. Here, the intuition of the developer is important, who prescribes the simplifications, usually by making the problem more homogeneous than it is (for example prescribing that drivers only fall into few behavioral classes). – Microscopic models fall into this category.

Ref. [9] nicely illustrates the problem: The authors indeed decide on an objective function (match the two parameters of a two-fluid model description of the real world traffic); yet the procedure is trial and error in the sense that the authors themselves decide on which aspects of NETSIM they believe to be important.

This indicates, consistent with our own experience, that formal calibration (in the sense of a formal procedure as opposed to trial-and-error) of *microscopic* models is currently very hard to achieve. This, in addition to the generally valid argument that calibration does not protect one against having the wrong model, implies to us that on the "validation" level, comparable and meaningful test suites should be constructed, and that the model behavior in these test suites should be publicly documented. This effort should be geared towards *understanding* the strength and weaknesses of a/the participating model (as opposed to deciding which is the "best" model).

In this paper, we want to concentrate on the "validation" part in the above sense in conjunction with "building block" test cases. We mean that as a first important step; in the future, we would like to be able to say something like "the simulations in this study are based on driving rules with their emergent behavior documented in the appendix", which would recognize the fact that the rules may have changed since the last "major" publication. This does not preclude that we will attempt to construct more realistic test scenarios in the future.

[1] Bluntly, one can always fit a straight line to a data cloud.

III. THE TRANSIMS MICROSIMULATION APPROACH

When designing a traffic microsimulation model, the first idea might be to measure all aspects of human driving and put them in algorithmic form into the computer. Unfortunately, such attempts cause many problems. The first is a data collection problem, because one can certainly not measure "all" aspects of human driving and is thus faced with the double sided problem that the necessary data collection process is extremely costly and still selective. Second, what if the emergent flow properties of such a model are clearly wrong, for example producing an hourly flow rate that is much too high?

For that reason, the TRANSIMS (TRansportation ANalysis and SIMulation System [10]) microsimulation starts with a *minimal* approach. A minimal set of driving rules is used to simulate traffic, and this set of rules is only extended when it becomes clear that a certain important aspect of traffic flow behavior cannot be modeled with the current rule set.[2] Besides the conceptual clarity, this also has the advantage that it is usually computationally fast – minimal models have few rules and thus run fast on computers.

The last paragraph leaves open what the "important aspects" are. In our view, this can only be decided in the proper context, i.e. when the question or problem area of application is known. The questions that TRANSIMS is currently designed for are transportation *planning* questions. These questions have traditionally been approached using traffic assignment models based on link performance functions (link capacity functions). Link performance functions are known to be dynamically wrong in the congested regime [11]; they simply do not model queue build-up when demand is higher than capacity.

The most important result of a transportation microsimulation in that context should be the *delays*, since they dominate travel times, and also hinder discharge of the transportation system, thus leading to grid-lock. Delays are caused by congestion, and congestion is caused by demand being higher than capacity. This implies that the first thing the TRANSIMS traffic microsimulation has to get right are capacity constraints (and possibly their variance). Capacity constraints are caused by a variety of effects:

- Undisturbed roadways such as freeways have capacity constraints given by the maximum of the flow-density diagram.
- Typical arterials have their capacity constraints given by traffic lights.
- In the case of unprotected turning movements (yield, stop, ramps, unprotected left, etc.), the capacity constraints are given as a function of opposing traffic flows. For example, the number of vehicles making an unprotected left turn depends on the oncoming traffic.

Building a simulation which can be adjusted against all these diagrams seems a hopeless task given the enormous amount of degrees of freedom. The TRANSIMS approach for that reason has been to *generate* the correct behavior from a few much more basic parameters. The correct behavior with respect to the above criteria can essentially be obtained by adjusting

[2] Note, though, that it is certainly desirable to have *reasonable* microscopic rules.

two parameters: (i) The value of a certain asymmetric noise parameter in the acceleration determines maximum flow on freeways and through traffic lights; (ii) the value of the gap acceptance determines flow for unprotected movements.

It needs to be emphasized again that these remarks are only valid in our context: There are many questions for which the models need to have a higher fidelity, and then more details, higher resolution, etc. may need to be added (e.g. [12,13]).

There is sometimes debate whether the model we thus obtain is truly "microscopic". We use the term "microscopic" with respect to the *resolution* of the model, i.e. a model is microscopic as soon as it allows the identification of individual particles (here cars). The proposed area of *application*, though, is where traditionally more macroscopic models have been used [11,14–16].

IV. RULES OF THE MODEL

A. Single lane uni-directional traffic

Our traffic simulation is based on a cellular automata technique, i.e., a road is composed of cells, and each cell can either be empty, or occupied by exactly one vehicle [17,18], see Fig. 1 (a). Since movement has to be from one cell to another cell, velocities have to be integer numbers between 0 and v_{max}, where the unit of velocity is [cells per time-step]. It turns out that reasonable values are [18,19]:

- length of a box $= 1/\rho_{jam} = 7.5$ m (ρ_{jam} = density of vehicles in a jam).
- time step = 1 sec
- maximum velocity = 5 boxes per time step $= 5 \cdot 7.5$ m/sec = 135km/h ≈ 85mph

For other conditions, such as higher or lower speed limits, this can be adapted.

Note that this approach implies a *coarse graining* of the spatial and temporal resolution and therefore of the velocities. A vehicle which has a speed of, say, 4 in this model stands for a vehicle which has a speed anywhere between $3.5 \cdot 7.5$ meters/sec ≈ 95 km/h (59 mph) and $4.49999 \cdot 7.5$ meters/sec ≈ 121 km/h (75 mph).

Vehicles move only in one direction. For an arbitrary configuration (velocity and position), one update of the traffic system consists of two steps: a velocity update step consisting of three consecutive rules, and a movement step according to the result of the velocity update. The whole update is performed simultaneously for all vehicles. The complete configuration at time step t is stored and the configuration at time step $t+1$ is computed from that "old" information. Computationally we calculate in time step t (with the three rules) the new velocity of each car and write this newly calculated velocity in the same site without moving the car (velocity update). After that we move all cars according to their newly calculated velocity (movement update).

1. (velocity update)

   For all particles i simultaneously, do the following:

   ```
   IF ( $v_i \geq gap_i$ )
   ```

```
        $v_i := \begin{cases} gap_i - 1 & \text{with probability } p_{noise} \text{ if possible}^3 \\ gap_i & \text{else} \end{cases}$ (close following/braking)
    ELSE IF ( $v_i < v_{max}$ )
        $v_i := \begin{cases} v_i & \text{with probability } p_{noise} \\ v_i + 1 & \text{else} \end{cases}$ (acceleration)
    ELSE (i.e. ( $v_i = v_{max}$ AND $v_i < gap_i$ )
        $v_i := \begin{cases} v_{max} - 1 & \text{with probability } p_{noise} \\ v_{max} & \text{else} \end{cases}$ (free driving)
    ENDIF
2. (movement update)
    Move all particles $i$ to $x_i(t+1) = x_i(t) + v_i$.
```

The index i denotes the position (an integer number) of a vehicle, $v(i)$ its current velocity, v_{max} its maximum speed, $gap(i)$ the number of empty cells ahead, and p_{noise} is a randomization parameter.

The first velocity rule represents noisy car following or braking. If the vehicle ahead is too close, the vehicle itself attempts to adjusts its velocity such that it would, in the next time-step, reach a position just behind where the vehicle ahead is at the moment. Yet, with probability p_{noise}, the vehicle is a bit slower than this.

The second velocity rule represents noisy acceleration. Essentially, the acceleration is linear (i.e. independent from current speed), but with probability p_{noise}, no acceleration happens in the current time step (maybe as a result of switching gears etc.). Instead of an acceleration sequence of $0 \to 1 \to 2 \to 3 \to \ldots$, a possible acceleration sequence can now be $0 \to 0 \to 1 \to 2 \to 2 \to 2 \to 3 \to \ldots$.

The last velocity rule represents free driving. Instead of remaining always at the same speed, such vehicles fluctuate between v_{max} (with probability $1 - p_{noise}$) and $v_{max} - 1$ (with probability p_{noise}). Note that a vehicle which is set to $v_{max} - 1$ will go through the acceleration step next time, thus in the next time step either staying at $v_{max} - 1$ with probability p_{noise} or getting back to v_{max}. Note that the resulting average speed of a freely driving vehicle is thus $v_{max} - p_{noise}$.

In terms of a microscopic foundation, the model is composed of the following elements:

- If a vehicle does not have enough space ahead, speed is proportional to space headway, which implies constant time headway (Pipes' theory, [20]).

- If there is enough space ahead for the given velocity, the vehicle accelerates linearly either to maximum speed or until the space headway becomes too small for further acceleration. A more realistic acceleration would probably be proportional to $1/v$, where v is the speed. This would be computationally more burdensome; nevertheless, studies about the effect are under way. The effect on the principal traffic dynamics seem to be minimal [21].

- Deceleration is instantaneous within one time step. If one wants to constrain the model to realistic deceleration values, one needs to look at the velocity of the vehicle ahead. This is again computationally more burdensome, and the precise difference when changing this element alone with respect to the traffic dynamics is unclear [22] although it is clear that throughput will go up [23].

- On top of these rules, we add a fairly large amount of random noise in the velocity decision.

Somewhat shorter, the model enforces constant time headway for close following and for braking, but acceleration is "delayed". This puts the model into a large class of dynamically similar models which use "time delayed" constant time headway, e.g. of the type $a(t) \propto V[\Delta x(t)] - v(t)$ or $v(t+\tau) \propto V[\Delta x(t)]$, where a is the acceleration, v is the vehicle's velocity, Δx is the space headway, $V(\Delta x)$ is a desired speed function, and τ is a time delay. It is certainly arguable that this does not catch all aspects of traffic; yet, all of these models are remarkably robust with respect to their traffic dynamics behavior, both in microscopic [22,24,25] and in fluid-dynamical [26,27] implementations.

B. Lane changing for passing

For multi-lane traffic, the model consists of parallel single lane models with additional rules for lane changing. Here we describe the two lane model which can be modified to any kind of multi lane model. Lane changing is modeled by an additional update step, which is added before the velocity update. The new sequence of steps is presented below. Steps two and three are the same in the single lane model and they are executed separately for each lane.

1. **Lane changing decision**
2. **Velocity update**
3. **Vehicle movement**

According to this lane changing rule set the vehicles are only moving sideways during the lane changing step; forwards movement is done in the vehicle movement step. One should, though, look at the combined effect of the lane changing and vehicle movement, and then vehicles will usually have moved sideways *and* forwards. The decision to change lane is implemented as strictly parallel update, i.e. each vehicle is making its decision based upon the configuration at the beginning of the update.

- **Lane changing decision for passing**

```
– IF neighboring position $x_o(i)$ in other lane is vacant
    * THEN Calculate:
        · $gap(i)$ Gap Forward in Current Lane,
        · $gap_o(i)$ Gap Forward in Other Lane,
        · $gap_b(i)$ Gap Backward in Other Lane,
        · IF ($gap(i) < v(i)$ AND $gap_o(i) > gap(i)$ )
          – THEN $weight1 = 1$
          – ELSE $weight1 = 0$
        · $weight2 = v(i) - gap_o(i)$
        · $weight3 = v_{max} - gap_b(i)$.
    * **IF** ( $weight1 > weight2$ ) AND ( $weight1 > weight3$ )[4]
        · **THEN** mark vehicle for lane change[5]
```

The rules are working in the following way (see Fig. 1 (b)): First we look at the neighboring position in the target lane. If this cell is vacant, we calculate the gap forward in the current lane (*gap*), the gap forward in the target lane (gap_o), and the gap backward in the target lane (gap_b). With these results we calculate the *weight*1 to *weight*3 described above. Finally if the weight comparisons render true the car will change to the new lane. After executing the lane changing decision we calculate the new velocity for all cars and move them according to this velocity.

This lane changing implementation follows a usual structure [29,30]:

- Reason to change lanes? (Slow car ahead? Need to make turn later (see below)?)
- If yes: Target lane empty? (Definition of "empty" depends on "urgency")
- If yes: change lanes except for stochastic noise

Lane change implementations using this framework are remarkably robust in their dynamic behavior [29–32]. This allows us, for example, not to look at other vehicle's velocities: The forward condition for the target lane, $gap_o \geq v$, is consistent with the condition $v \leq gap$ for the car following; the backward condition for the target lane, $gap_b \geq v_{max}$ is simply a worst case scenario which nevertheless does not perform, in the analysis of the emergent properties, any worse than a condition which depends on the velocity of the other car (compare, e.g., [33] with [34]).

For three or more lanes, a simultaneous implementation of the lane changing decision can lead to collisions. For example, in a three-lane road two vehicles on the left and right lane could decide to go to the same spot in the middle lane. From an algorithmic point of view, this is possible because the lane changing decision is based on the configuration on time t; but it is also an entirely realistic situation.[6] To avoid collision we only allow lane changes in a certain direction in each time step:

[4] Weights are used because of extensibility towards "lane changing for plan following". See below.

[5] In the current version, the lane change is actually still rejected with a probability of 0.01 even when all the rules are fulfilled. This is in order to break the following artifact or variations of it: Assume one lane is completely occupied and one is completely empty. The above rule set will result in these vehicles just changing back and forth between the lanes—the vehicles will never get smeared out across the lanes. See Ref. [28] for more details.

[6] In a deeper sense, the problem is caused by the fact that the underlying decision making dynamics has a time scale which is smaller than the time resolution of the simulation. The simulation thus must resolve the conflict by other means [35].

- **IF** the time step is even

  THEN start procedure *lane changing decision* to the *left* for cars on the middle and then on the right lane

- **IF** the time step is odd

  THEN start procedure *lane changing decision* to the *right* side for cars on the middle and then on the left lane

Thus, left lane changes occur only on even time steps, right lane changes occur only on odd time steps. This behavior is collision free.

C. Lane changing for plan following

Vehicles in TRANSIMS follow route plans, i.e. they know ahead of time the sequence of links they intend to follow. This means that, when they approach an intersection, they need to get into the correct lanes in order to make the intended turn. For example, a vehicle which intends, according to its route plan, to make a left turn at the next intersection needs to get into one of the lanes which actually allow a left turn.

This is achieved in TRANSIMS by supplementing the basic lane changing rules with a bias towards the intended lanes. This bias increases with increasing urgency, i.e. with decreasing distance to the intersection. Technically, this is achieved by adding another weight to the acceptance conditions for lane changing:

- **IF** $(weight1 + weight4 > weight2)$ AND $(weight1 + weight4 > weight3)$

  THEN change lane

$weight4$ is calculated according to

$$weight4 = \max\left[\frac{d^* - d}{v_{max}}, 0\right]$$

for lane changes in the desired direction as long as the vehicle is not in one of the correct lanes, cf. Fig. 1 (c). d is the remaining distance to the intersection, d^* is a parameter; both are given in the unit of "cells". d^* is currently set to 70 cells, i.e. approx. 500 m or 1/3 of a mile, throughout the simulation. In consequence, $weight4$ increases from zero to $d^*/v_{max} = 14$ during the approach to the intersection. If $weight4 = 0$, then it does not influence lane changing decision. $weight4 = 1$ has the same effect as a slower vehicle ahead on the same lane. Further increases of $weight4$ more and more override the security criterions that the forward and the backward gap on the destination lane need to be large enough. $weight4 > v_{max}$ lets the vehicle make the lane change even if only the neighboring cell on the destination lane is free.

Once a vehicle is in one of the "correct" lanes within 70 cells (525 m) of the intersection, it is only allowed to change lanes if the target lane is also "correct". For movements that are allowed on multiple lanes through the intersection, this leads to equal usage of these lanes. This algorithm is *not* capable of leaving a single "correct" lane temporarily when encountering, say, a stopped bus on the same lane.

D. Unprotected turning movements

A necessary element of traffic simulations are unprotected turning movements. By this we mean that that for the movement the driver intends to make, some other lanes have priority. Examples are stop signs, yield signs, on-ramps, unprotected left turns.

The general modeling principle for this in TRANSIMS is based on a gap acceptance in the opposing (in TRANSIMS sometimes called "interfering") lanes, see Fig. 1 (d). Opposing lanes are the lanes which have priority; for example, for a stop-controlled left turn onto a major road this would be all lanes coming from the left plus the leftmost lane coming from the right. In order to accept the turn, there has to be a sufficient gap in each of these lanes.

Note that "gap divided by the velocity of the oncoming vehicle" is the oncoming vehicle's time headway, so the dynamics of this follows the Highway Capacity Manual [3]. If one wants a time headway on an opposing lane of at least 3 seconds, then a vehicle with a velocity of 4 cells/second would have to be at least 12 cells away from the intersection.

The current TRANSIMS microsimulation uses a gap acceptance (gap between intersection and nearest car to the intersection which is approaching) of 3 times the oncoming vehicle's velocity, i.e. when the gap on each opposing lane is larger than or equal to the first vehicle on that lane, the move is accepted. For example, if the oncoming vehicle has a speed of 3, at least 9 empty cells have to be between the oncoming vehicle and the intersection. A special case is if the oncoming vehicle has the velocity zero, in which case no gap is necessary.

E. Signalized intersections

In TRANSIMS, we distinguish between signalized intersections and unsignalized intersections. In signalized intersections, the priorities are changing in time and regulated by signals. In unsignalized intersections, the priorities are fixed.

When a simulated vehicle approaches a *signalized* intersection, the algorithm first decides if, according to its current speed, it potentially wants to leave the link, i.e. its current speed (in cells per update) is larger than or equal to the remaining number of cells on the link.[7] If a vehicle wants to leave the link, the algorithm checks the "traffic control", which determines if the vehicle can leave the link. If it encounters a red light, it can *not* leave the link and no further action is taken. If it encounters a protected (green arrow) or caution (yellow) signal, the vehicle is allowed to enter the intersection. If it encounters a permitted signal (green, for example permitted left turn against oncoming traffic), the vehicle checks all opposing flows for a gap that is larger or equal to 3 times the oncoming vehicle's velocity (see Subsec. IV D above).

If the movement into the intersection is accepted, the vehicle is moved into an "intersection queue"; there is one queue for each incoming lane. This queue models vehicle behavior inside an intersection. The vehicle gets a "time stamp", before which it is not allowed to leave the intersection; this time stamp is representative of the duration of the movement through the intersection. The intersection queues have finite capacity; once they are full, no

[7] Vehicles may accelerate or slow down before they actually reach the intersection. See below.

more vehicles are accepted and the vehicles start to queue up on the link. This models the finite vehicle storing capacity of an intersection.

Once a vehicle is ready to leave the intersection, it moves to the first cell on the destination link if available. The speed of the vehicle is not changed when it is in the intersection queue so it exits on the destination link in the first cell with the same velocity that it had when it entered the queue.

Note that vehicles turning against opposing traffic make their decision to accept the turn when they *enter* the intersection queue, not when they leave it. This can have the effect that a vehicle enters the intersection queue when there is no oncoming traffic, but, because of other vehicles ahead of it in the same queue, cannot make its turn immediately. Yet, since the turn was already accepted, it will be executed as soon as all vehicles ahead in the same queue have cleared the queue and a cell on the destination link is available. The turn can occur during oncoming traffic. So in some sense vehicles will go "through" each other. Yet, note that on average the result is still correct. The approach described above will not let more vehicles through the intersection than a gap acceptance calculated when *leaving* the intersection queue. The above logic was chosen for simplification purposes since unsignalized intersections (see below) do not have queues and thus *need* to make their acceptance decisions when entering the intersection.

F. Unsignalized intersections

Unsignalized intersections in TRANSIMS have no internal queues, i.e. vehicles go right through them.[8] Also, vehicles leaving an unsignalized intersection go down the destination link as far as prescribed by their velocity, not just into the first cell as in the signalized intersections. Apart from these two differences, unsignalized intersections are similar to signalized ones.

When a simulated vehicle approaches an unsignalized intersection, the algorithm first decides if, according to its current speed, it potentially wants to leave the link, i.e. its current speed (in cells per update) is larger than or equal to the remaining number of sites on the link. If a vehicle wants to leave the link, the algorithm checks the "traffic control", which determines if the vehicle can leave the link. Currently occuring traffic controls are: no control, yield, and stop.

If a "no control" is encountered, the vehicle is moved to its destination cell without any further checks. For example, if a vehicle has a velocity of 5 cells per update and 2 more cells to go on its link, then it attempts to go 3 cells into the destination link. If that cell is already reserved (either by another "reservation" or by a real vehicle), then the next closer cell is attempted, etc., until the algorithm either finds an empty cell or returns that the destination lane is full. "No control" is usually used for the major directions, i.e. for the lanes which have priority.

[8] Again, technically the vehicles only reserve cells on the destination links. The actual move through the intersection happens later and can also be postponed if after the velocity update the vehicle actually does *not* make it to the intersection.

If a yield sign is encountered, the vehicle checks the gap on all opposing lanes. According to the same rules as above, on all opposing lanes the gap needs to be larger or equal three times the first vehicle's speed on that lane. If the movement is accepted, the destination cell is selected according to the same rules as with the "no control" case.

If it encounters a stop sign, the vehicle is brought to a stop. Only when the vehicle has a velocity of zero for at least one time step on the last cell of the link is it allowed to continue. If the result of the regular velocity update indeed accelerates the vehicle,[9] then it attempts to go through the intersection. On all opposing lanes the gap, according to the same rules as above, needs to be larger or equal to three times the first vehicle's speed on that lane. If the movement is accepted, a vehicle coming from a stop sign will always go to the first cell on the destination link (if empty) and will have a velocity of one.

G. Parking locations

In the current TRANSIMS microsimulation, vehicular trips start and end at parking locations. Each link in the microsimulation, except for freeway ramps, freeway links, and some "virtual" links such as centroid connectors, has at least one parking location. Parking locations thus represent the aggregated parking options on that link. Parking locations have rules about how vehicles enter and exit the simulation:

- Each vehicle in TRANSIMS has a complete route plan, together with a starting time. At the starting time, the vehicle is added to a queue of vehicles that want to leave the same parking location. When the vehicle is the first one in the queue, it attempts to enter the link. The acceptance logic is in spirit similar to the logic of the unsignalized intersections, i.e. vehicles check the available gap and make their decision based on that. Parking accessory logic is not the focus of the current paper, and since that logic may change in TRANSIMS in the near future and we also expect no influence on the results presented here, we omit further technical details.

- A vehicle that has reached its destination parking location according to its plan will leave the microsimulation.

H. Parallel logic

TRANSIMS is designed to run on parallel computers, such as coupled workstations, desktop multi-processors, or supercomputers. The parallelization approach used for the microsimulation is a geographical distribution, i.e. different geographical parts of the simulated area are computed on different CPUs.

The current TRANSIMS microsimulation has these boundaries always in the middle of links. This is done in order to keep the complexity of the parallel computing logic as far away as possible from the complexity of the intersection logic.

[9] I.e. there is a probability of $1 - p_{noise}$ that the vehicle will not accelerate in the given time step.

Information needs to be exchanged at the boundaries several times per update in order to keep the dynamics consistent. For example, if a vehicle changes lanes and ends up close in front of another one, that other one is probably forced to brake. Now, if the lane changing vehicle is on one CPU and the following one on another, one needs to communicate the lane change. This will be called "Update boundaries" in the following section.

I. Complete scheduling

For a complete transportation microsimulation, we need to specify when movements are accepted, and also how conflicts are resolved. For example, vehicles simultaneously attempting to change lanes into the middle lane represent such a conflict. Another conflict is two vehicles from two different links competing for the same site on the destination link.

The complete update of the current TRANSIMS microsimulation is as follows. Assume that the state at time t is the result of the last update. Let $t1$, $t2$, etc. be intermediate partial time steps.

1. Vehicles which are ready to leave intersection queues from signalized intersections reserve cells on outgoing lanes. They only attempt to reserve the first cell on the link; their velocity is the same as it was when they entered the intersection. When the cell is occupied (either by another "reservation" or by a vehicle), then the vehicle cannot leave the intersection. Note that there can be a conflict between different queues for the same destination cell. The current solution in TRANSIMS is that queues are served on a first come first served basis in some arbitrarily defined way, i.e. a queue which happens to be treated earlier in the microsimulation has a slightly higher chance of unloading its vehicles. — Result: t_1 information.

2. Vehicles change Lanes. Use information from time t_1 to calculate situation at time t_2.

3. Exit from Parking. Results in t_3 information.

4. Exchange boundary information for parallel computing.

5. Non-signalized intersections reserve sites on target lanes. Note that there can be a conflict of two incoming links competing for the same destination cell. The current solution in TRANSIMS is that links are served on a first come first served basis, i.e. a link which happens to be treated earlier in the microsimulation has a slightly higher chance of unloading its vehicles. Note that this conflict only happens between minor links. Major links never compete for the same outgoing link except when there is a network coding error; and for the competition between major and minor links, the major link always wins because of the opposing lanes conditions.[10] Result: t_4 information.

[10] Note that the situation slightly different when the speed of the vehicle on the major link is zero – see below.

6. Calculate speeds and do movements. If a vehicle scheduled for an intersection does not go through the intersection as a result of the velocity update, the reservation is cancelled. Vehicles which go through unsignalized intersections have p set to zero, i.e. if it turns out that the result of the velocity update indeed brings them into the intersection, they need to go to the site on the destination lane which was reserved earlier. Result: $t_5 = t + 1$ information.

7. Exchange boundary information and migrate vehicles for parallel computing.

V. TOWARDS A STANDARDIZED FLOW TEST SUITE FOR SIMULATION MODELS

In order to control the effect of driving rules, TRANSIMS provides controlled tests for traffic flow behavior. These tests are simplified situations where elements of the microsimulation can be tested in isolation. This test suite uses the standard microsimulation code in the same way it is used for full-scale regional simulations, and it also uses the same input and output facilities: The test network is currently defined via a table in an Oracle data base, in the same format as the Dallas/Fort Worth network is kept. Input of vehicles is, following individual vehicle's plans, via parking locations, the same way vehicles enter regional simulations.[11] Output is collected on certain parts of the network on a second-by-second basis, the same way it can be collected for regional microsimulations. The collected output is then post-processed to obtain the aggregated results presented in this paper.

The test cases we look at in this paper are the following (see also Fig. 1 (e)):

- One-lane traffic, in order to see if car following behavior generates reasonable fundamental diagrams.
- Three-lane traffic, in order to see if the addition of passing lane changing behavior still generates reasonable fundamental diagrams, and in order to look at lane usage.
- Stop sign, yield sign, and left turns against oncoming traffic, in order to see it the logic for non-signalized intersections generates acceptable flow rates.
- A signalized intersection, in order to see of we obtain reasonable flow rates, and in order to check lane changing behavior for plan following purposes.

A. Measured quantities

We look at three minute averages of the following quantities:

[11] Route plans are simply necessary to be consistent with the way the simulation is normally used; for the test cases we use very few types of generic route plans (like "enter the microsimulation and keep on driving in a circle indefinitely") and replicate them with different starting times to fulfill our needs. This is not much different from departure rates.

- **Flow, Volume.** Flow q is defined as usual by:

  $$q = \frac{N}{T} \qquad [vehicles/hour]$$

  N is the number of cars which pass a certain site at a time period T.

- **Density.** Density is in principle easily defined, $\rho = N/L$, where N is the number of vehicles on a piece of roadway of length L. Yet, given current sensor technology, this is not easy to achieve since one would need a sensor which counts, say once a second, cars on a predefined stretch of length L of the roadway. For that reason, empirical papers sometimes resort to occupancy, which is the fraction of time a given sensor has been occupied by a vehicle. Currently TRANSIMS measures density according to its original definition, i.e., once a time step, we count the number of vehicles on a stretch of roadway of $L = 5$ sites $= 5 \times 7.5$ m $= 37.5$ m.[12] We add these counts for $k = 180$ measurement events and then divide the resulting number by L and by k:

  $$\rho = \frac{N}{k * L}$$

  The result can be scaled to convenient units, for example "vehicles per km".

  Note that this way of computing density averages the counts over a length of 37.5 m, which is longer than most traffic detectors. The effect of this should be systematically studied.

- **Space Mean Speed, Travel Velocity.** It is well known that one can measure velocity either analogous to our flow definition (Time Mean Speed, Spot Speed) or analogous to our density definition (space mean speed, travel velocity). Under non-stationary conditions, the measurements give different results, since, for example, the first definition never counts vehicles with velocity zero. Time mean speed is easier for field measurements; space mean speed is easier to interpret since it is equal to the travel velocity and it is also the velocity which needs to be used in the fundamental relationship between flow, density, and velocity, $q = \rho \cdot v$. Since in a simulation model both are similarly easy to measure, we measure the more meaningful travel velocity. Once a time step, we sum up the individual velocities of all vehicles on a stretch of roadway of $L = 5$ sites $= 5 \times 7.5$ m $= 37.5$ m. We add these sums for $k = 180$ measurement events and then divide the resulting number by N and by k, where N is the same number as obtained during the density measurement above:

  $$v = \frac{\sum v}{k * N}$$

[12] The "magical" number of $L = 5$ sites is equal to the maximum velocity of $v_{max} = 5$ sites/update. This ensures that each vehicle is counted at least once.

- **Lane usage.** Lane usage of a particular lane is the number of cars on this lane divided by the number of cars on all lanes. It can be computed as:

  $$f_i = \frac{\rho_i}{\sum_{j=1}^{n} \rho_j \cdot n} ,$$

  where i is the lane we look at and n is the number of lanes.

B. Test networks

Essentially two test networks are used: a circle of 1 000 sites = 0.75 km in various configurations, and a simple signalized intersection. Most of the tests are run on the circle networks. The circle can have one, two, or three lanes. In all tests, the circle is slowly loaded with traffic via a parking location at site $x = 1$ (where the unit of x is "cells"). Velocity, flow, and density are measured on $486 \leq x \leq 490$, thus generating the fundamental diagrams for one-lane, two-lane, and three-lane traffic. Since the circle gets slowly loaded, the complete fundamental diagram is generated during one run.

For testing yield signs and stop signs, an incoming lane is added on the right side of traffic at $x = 501$. The characteristics of the incoming traffic are measured by a detector on the last 5 sites of the incoming lane. The incoming lane is operated at maximum flow, i.e. with as many vehicles as possible entering. The incoming vehicles are removed at $x = 900$ via a parking accessory. The result of this measurement is typically a diagram showing the flow of incoming vehicles on the y-axis versus the flow on the circle on the x-axis.

For testing left turns against oncoming traffic, an opposing lane is added so that it ends at $x = 500$. The traffic control here is again a "yield" logic; the difference from before is that vehicles only *traverse* the opposing traffic, they do not join it.

Last, a three-lane intersection approach is used. The left lane makes a left turn, the middle lane goes straight, the right lane makes a right turn. Incoming vehicles have plans about their intended movement at the intersection and attempt to reach the corresponding lane. The intersection has signals with 1 minute green phase and 1 minute red phase. The typical output from this run is the flow of vehicles which go through the intersection, and the number of vehicles which cannot make their intended turn because they did not reach their lane.

C. Results

The results are shown in Figs. 2 to 5.

- Single lane traffic (Fig. 2a) has a realistic value of maximum flow (= capacity), but one may argue that it is at a somewhat low density. The problem here is that we do not include slow vehicles; introducing slow vehicles into a single lane closed circle simulation just means that all fast vehicles bunch up behind them, which does not result in a very useful fundamental diagram. In terms of the "building block" philosophy, we prefer to run the single lane test with identical vehicles.

- Our lane changing rules do neither change maximum flow per lane nor the density (per lane) at maximum flow. That need not be the case, [31]. Again, the density at maximum flow seems a bit low. This changes considerably when one introduces slower vehicles: The free flow part of the curve then bends more to the right and the maximum is at higher densities [32]. Also, there are measurements in Germany where traffic *with* trucks reaches maximum flow at approx. 20–22 veh/km/lane [36], so without more specific data this discussion seems pointless. – We think that the curve without slow vehicles is "cleaner" and thus facilitates comparison between models; in reality, the problem is more complicated anyway.

  Also, we generate equal lane usage between the lanes, as should be expected for a symmetric lane changing model (in the absence of on-ramps).

- The flow through a traffic signal that is 50% green should be at half the value of the maximum single lane flow, i.e. at 1000 veh/hour, which is what we find (Fig. 4).

- The curves for traffic through stop and yield signs follows the general form of the curve of the Highway Capacity Manual [3]. We added the HCM curves for comparison only. In general, we find that a yield sign, when there is no traffic on the major road, generates the same traffic as if there were no sign at all, which should be expected the way the simulation is set up. (It is a bit lower than for the "circle" before because the speed limit is lower here.) The stop sign generates a much lower flow in the same situation, because the explicit stop decreases capacity.

  From there, the curves for "traffic into" the major road decrease roughly linearly to zero when the flow on the major road reaches capacity. The curve for traffic across a single lane road looks similar to its "traffic into" counterpart, which is to be expected because the number of opposing lanes is one in both cases. The curve for traffic across a two lane road provides roughly half the flow of traffic across a single lane road.

  For densities above capacity on the major road, all curves bend "back on themselves". If the major road is congested, the speed there is zero, and the gap acceptance criterion "accept if $gap \geq 3 \cdot v_{oncoming}$" is always fulfilled, even for $gap = 0$. Nevertheless, for "traffic into", very little traffic makes it through the yield or the stop sign. The reason is that in TRANSIMS, vehicles on the major road that may go through the intersection "reserve" the first cell at the beginning of the next link, thus blocking this link for vehicles from the minor link even if the gap acceptance rule would allow the movement. For "traffic across", this restriction does not exist, and many vehicles make it through the intersection, probably many more than is realistic. – Note that the HCM does not provide any information in the congested regime.

VI. YIELD SIGN BEHAVIOR

All runs for this paper were first done with an experimental code and then repeated with the TRANSIMS production code; all results shown so far were obtained from the TRANSIMS production code. The disadvantage of an experimental code is that actual implementation in the production version may still introduce changes in the results due to small discrepancies.[13] The advantage of an experimental code is that turnover (compile times, complexity of code, etc.) is much better than with a production version. We used that advantage to test many different rules. In the following, we want to present a small subset of tests.

All results presented in this section refer to the situation of a 1-lane minor street merging into a 1-lane major street, with the intersection control being a yield sign. Fig. 6 (a) shows what happens if the "reservation" rule from the TRANSIMS production code is no longer used. Clearly, if vehicles from the major road do reserve cells on the outgoing link only if they are actually going there, many more vehicles from the minor lane can make the turn, effectively leading to an "alternating" vehicle pattern. This may be desirable in some situations.

Figs. 6 (b) shows what happens when one then changes "accept when $gap \geq 3 v_{oncoming}$" to "accept when $gap > 3 v_{oncoming}$". This seems like a negligible difference in the rules; yet, the results are quite different in the congested regime. Whereas in the first, many vehicles are able to get into the congested major road, in the second, only few of them make it. The difference is easiest explained by looking at a vehicle of speed zero on the major road just in front of the merge point, with space for a vehicle downstream of the merge point. With the first rule, a vehicle at the yield sign will accept the move and move in front of the vehicle on the major road, in the second case, it will not. Both scenarios seem to be plausible to us; only systematic measurements can probably resolve which one is better for a simulation model. – Also note that the rule in (b) generates similar flows as the TRANSIMS production version.

Fig. 6 (b), (c) and (d) show the result of different speed limits (same speed limit for both streets). A high average free speed of approx. 130 km/h ($\approx$ 80 mph, generated by $v_{max} = 5$), maybe a freeway merge, generates a flow of approx. 2000 veh/hour/lane in the incoming lane when there is no traffic on the major road (Fig. 6 (c)). From there, maximum incoming flow decreases continuously. Lower average free speeds of approx. 75 km/h (50 mph, Fig. 6 (b)) and 50 km/h (30 mph, Fig. 6 (d)) generate lower maximum incoming flows and are generally closer to the Highway Capacity Manual curve. Yet, it should be clear that, contrary to the HCM, the "minor" flow is also a function of the speed limit and not only of the gap acceptance (the gap acceptance is the same in all three simulations).

A last series of experiments shows the effect of different values for the gap acceptance. Figs. 6 (e) and (f) show "accept when $gap > v_{oncoming}$ and $gap > v_{max}$". Clearly, more vehicles are accepted, leading to a higher flow of turning vehicles as a function of the flow on the major road. Note that the flow via the yield sign is never higher than 1800 minus the flow on the major road. This reflects the fact that the major road cannot have a higher flow than 1800 veh/h/lane (free speed approx 50 mph); traffic through the yield sign can thus at most fill the major road to capacity. This explains why the acceptance of much smaller gaps do not produce a stronger difference. The situation is clearly different for unprotected turns *across* instead of *into* traffic, as can be seen for the left turns in the next section.

[13] This explains the differences to the TRB preprint version of this paper, which contained results from the experimental code.

VII. COMPARISON TO CASE STUDY LOGIC

The gap acceptance logic presented here and used in the March 1998 TRANSIMS microsimulation is different from the logic used in the "Dallas/Fort Worth Case Study" [37,38]. The logic during that case study was: "Accept an unprotected movement if in all opposing lanes the gap is larger than $v_{max} = 5$." This means that at low density on the major road, more turns were accepted, whereas at high density on the major road, less turns were accepted – with the extreme case that no turns were possible against oncoming traffic of speed zero.

Fig. 7 compares the results for the current gap-acceptance logic and the one used in the case study for the case where the major road is a 3-lane road. Note that the results for the turns *into* other traffic are not that much different whereas the result for the turns *across* other traffic yields much higher uncongested and much lower congested flows with the case study logic. This is due to the fact that for turns *into* other traffic, there is a capacity constraint of the form that the joint flows from the major and the incoming road cannot exceed capacity of the major road, see last section. Such a constraint obviously does not exist for turns *across* the major road.

VIII. SHORT DISCUSSION

We presented test of what we believe are "building blocks" of microsimulation models. Further "building blocks", not included here, are probably freeway ramps with merge lanes, and freeway weaving sections. We plan to include these tests into future versions.

As pointed out earlier, we believe that "clean" real world measurements of the "building block" situations are hard to obtain. Thus, one may consider them primarily useful for comparing simulations with each other and with theory; nevertheless, we think that one can judge from the results at least if the simulation is "in the right ballpark". It would certainly be desirable in the future to also have test suites for more complex situations. – For the same reason, we did not make any attempt to get "better" results than the ones presented here: we know that the results change in more complex scenarios, and it is therefore unclear if a change "to the better" in the test cases may not be a change "to the worse" with respect to reality.

Also, we would shortly like to point out again that "verification" of simulation models, i.e. the question if an actual code corresponds to a (possibly incomplete) specification in a paper, is in practice a difficult question. An alternative approach would be to try to find a suite that decides if we are macroscopically convincing without the need to go through testing the rules on an individual scale. Arguing about the microscopic rules could then be left to a small group of specialists; the end user could just look at the test suite results and judge in a matter of minutes if the simulation has faults that would seriously affect the analysis of their problem.

Last, all these problems imply to us that one should expect that simulation models will undergo continuous improvements, and it seems more realistic to us to expect "test suites" to be run at regular intervals instead of expecting that parts of simulation models can be validated and calibrated "once and for all" at certain stages and then never be touched again. In consequence, we would like to shift the argument from a discussion whether a model is “correct or not” to the discussion about which tests should be run to enable the user to make that decision, and how these tests can be made comparable between different simulation models.

IX. SUMMARY AND CONCLUSION

In transportation simulation models for larger scale questions such as planning, the flow characteristics of the traffic dynamics are in some sense more important than the microscopic driving dynamics of the vehicles itself. This becomes especially true since a “complete” representation of human driving is impossible anyway, both due to knowledge constraints and due to computational constraints. Yet, calibrating a traffic simulation model against all types of desired behavior (for example against all HCM curves and values mentioned in this paper) seems a hopeless task given the high degrees of freedom.

TRANSIMS thus attempts to generate plausible emergent macroscopic behavior from *simplified* microscopic rules. This paper described the more important aspects of these rules as currently implemented or under implementation in TRANSIMS. Before we implement rules in the TRANSIMS production version, we usually try to run systematic studies with more experimental versions. The results of the traffic flow behavior from that study were presented. Also, we showed the effects of some changes in the rules for the example of a yield sign. Finally, some comparisons were made between the logic currently under implementation and the logic used for the Dallas/Fort Worth case study.

One problem with microscopic approaches is that, in spite of all diligence, subtle differences between design and actual implementation can make a significant difference in the emergent outcome. For that reason, this paper should also be seen as an argument for a standardized traffic flow test suite for simulation models. We propose that simulation models, when used for studies, should first run these tests to demonstrate the dynamics of their emergent macroscopic flow behavior. We think that the combination of results presented in Figs. 2 to 5 are a good test set, although extensions may be necessary in the future (e.g. merge lanes, weaving, etc.). We will attempt to provide future TRANSIMS results also with updated versions of the results of the traffic flow tests.

ACKNOWLEDGMENTS

This work has been performed at Los Alamos National Laboratory, operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36.

REFERENCES

[1] M. Van Aerde, personal communication.

[2] J. Van Leeuwen, editor. *Formal models and semantics*, volume B of *Handbook of Theoretical Computer Science*. Elsevier and MIT Press, 1990.

[3] Transportation Research Board. *Highway Capacity Manual*. Special Report No. 209. National Research Council, Washington, D.C., 3rd edition, 1994.

[4] M.J. Cassidy and J. Han. Validation and evaluation of freeway simulation models. final report. Technical Report FHWA/CA/Purdue-RR-95-1, Purdue University, School of Civil Engineering, West Lafayette IN 47907, USA, 1995.

[5] H.S. Mahmassani, J.C. Williams, and R. Herman. Performance of urban traffic networks. In N.H. Gartner and N.H.M. Wilson, editors, *Transportation and Traffic Theory*, page 1. Elsevier Science Publishing Co., Inc., 1987.

[6] M. Ponzlet and P. Wagner. Validation of a CA–model for traffic simulation of the Northrhine-Westphalia motorway network. In *The 24th European Transport Forum, Proceedings*, volume P404-1, 1996.

[7] M. Cremer and M. Papageorgiou. Parameter identification for a traffic flow model. *Automatica*, 17(6):837–843, 1981.

[8] M. Cremer and H. Schütt. A comprehensive concept for simultaneous state observation, parameter estimation, and incident detection. In *Proceedings of the 11th Int. Symposium on Transportation and Traffic Theory*, Yokohama, Japan, 1990.

[9] R.W. Denney, J.C. Williams, S.C.S. Bhat, and S.A. Ardekani. Calibrating NETSIM for a CBD using the two fluid model. In *Large Urban Systems. Proceedings of the Advanced Traffic Management Conference*. Federal Highway Administration, 400 7th Street SW, Washington DC, USA, 1993.

[10] TRANSIMS, TRansportation ANalysis and SIMulation System, Los Alamos National Laboratory, Los Alamos, U.S.A. See `transims.tsasa.lanl.gov`.

[11] Michael Patriksson. *The Traffic Assignment Problem: Models and Methods*. Topics in Transportation. VSP, VSP, P.O. Box 346, 3700 AH Zeist, The Netherlands, 1994.

[12] R. Wiedemann. Simulation des Straßenverkehrsflusses. Schriftenreihe Heft 8, Institute for Transportation Science, University of Karlsruhe, Karlsruhe, Germany, 1994.

[13] M. Van Aerde, B. Hellinga, M. Baker, and H. Rakha. INTEGRATION: An overview of traffic simulation features. 1996. A paper accepted for presentation at the 1996 Transportation Research Board Annual meeting.

[14] G.L. Chang, H.S. Mahmassani, and R. Herman. A macroparticle traffic simulation model to investigate peak-period commuter decision dynamics. *Transportation Research Record*, 1005:107–120, 1985.

[15] T. Schwerdtfeger. *Makroskopisches Simulationsmodell für Schnellstraßennetze mit Berücksichtigung von Einzelfahrzeugen (DYNEMO)*. PhD thesis, University of Karsruhe, 1987.

[16] R. Herman and I. Prigogine. A two-fluid approach to town traffic. *Science*, 204:148–151, 1979.

[17] K. Nagel. Freeway traffic, cellular automata, and some (self-organizing) criticality. In R.A. de Groot and J. Nadrchal, editors, *Physics Computing '92*, page 419, Prague, 1992. World Scientific.

[18] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France*, 2:2221, 1992.

[19] C.L. Barrett, S. Eubank, K. Nagel, J. Riordan, and M. Wolinsky. Issues in the representation of traffic using multi-resolution cellular automata. Los Alamos Unclassified Report (LA-UR) 95-2658, Los Alamos National Laboratory, Los Alamos, NM, U.S.A., see `www.lanl.gov`, 1995.

[20] A.D. May. *Traffic flow fundamentals.* Prentice Hall, Englewood Cliffs, NJ, 1990.

[21] K. Nagel and M. Paczuski. Emergent traffic jams. *Physical Review E*, 51:2909, 1995.

[22] S. Krauß, P. Wagner, and C. Gawron. Metastable states in a microscopic model of traffic. *Physical Review E*, 55(5):5597–5602, 1997.

[23] C.L. Barrett, M. Wolinsky, and M.W. Olesen. Emergent local control properties in particle hopping traffic simulations. In D.E. Wolf, M. Schreckenberg, and A. Bachem, editors, *Traffic and granular flow*, pages 169–173. World Scientific, Singapore, 1996.

[24] K. Nagel. Particle hopping models and traffic flow theory. *Physical Review E*, 53(5):4655, 1996.

[25] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2):1035, 1995.

[26] R.D. Kühne and R. Beckschulte. Non-linearity stochastics of unstable traffic flow. In C.F. Daganzo, editor, *Proc. 12th Int. Symposium on Theory of Traffic Flow and Transportation*, page 367. Elsevier, Amsterdam, The Netherlands, 1993.

[27] B.S. Kerner and P. Konhäuser. Structure and parameters of clusters in traffic flow. *Physical Review E*, 50(1):54, 1994.

[28] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour. Two lane traffic simulations using cellular automata. *Physica A*, 231(4):534–550, 1996.

[29] U. Sparmann. *Spurwechselvorgänge auf zweispurigen BAB–Richtungsfahrbahnen.* Number 263 in Forschung Straßenbau und Straßenverkehrstechnik. Bundesminister für Verkehr, Bonn–Bad Godesberg, Germany, 1978.

[30] P. G. Gipps. A model for the structure of lane-changing decisions. *Transportation Research B*, 20B(5):403–414, 1986.

[31] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour. Two lane traffic simulations using cellular automata. *Physica A*, 231:534, 1996.

[32] K. Nagel, D.E. Wolf, P. Wagner, and P.M. Simon. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2):1425–1437, 1998.

[33] P. Wagner, K. Nagel, and D.E. Wolf. Realistic multi-lane traffic rules for cellular automata. *Physica A*, 234:687, 1997.

[34] P. Wagner. Traffic simulations using cellular automata: Comparison with reality. In D E Wolf, M.Schreckenberg, and A.Bachem, editors, *Traffic and Granular Flow*, Singapore, 1996. World Scientific.

[35] C.L. Barrett. Personal communication.

[36] R. Wiedemann. Beschreibung des Staus. In H. Keller, editor, *Beiträge zur Theorie des Straßenverkehrs*. Forschungsgesellschaft für Straßen- und Verkehrswesen, Köln, Germany, 1995.

[37] R.J. Beckman et al. TRANSIMS–Release 1.0 – The Dallas-Fort Worth case study. Los Alamos Unclassified Report (LA-UR) 97-4502, Los Alamos National Laboratory, see `transims.tsasa.lanl.gov`, 1997.

[38] K. Nagel and C.L. Barrett. Using microsimulation feedback for trip adaptation for realistic traffic in Dallas. *International Journal of Modern Physics C*, 8(3):505–526, 1997.

FIGURES

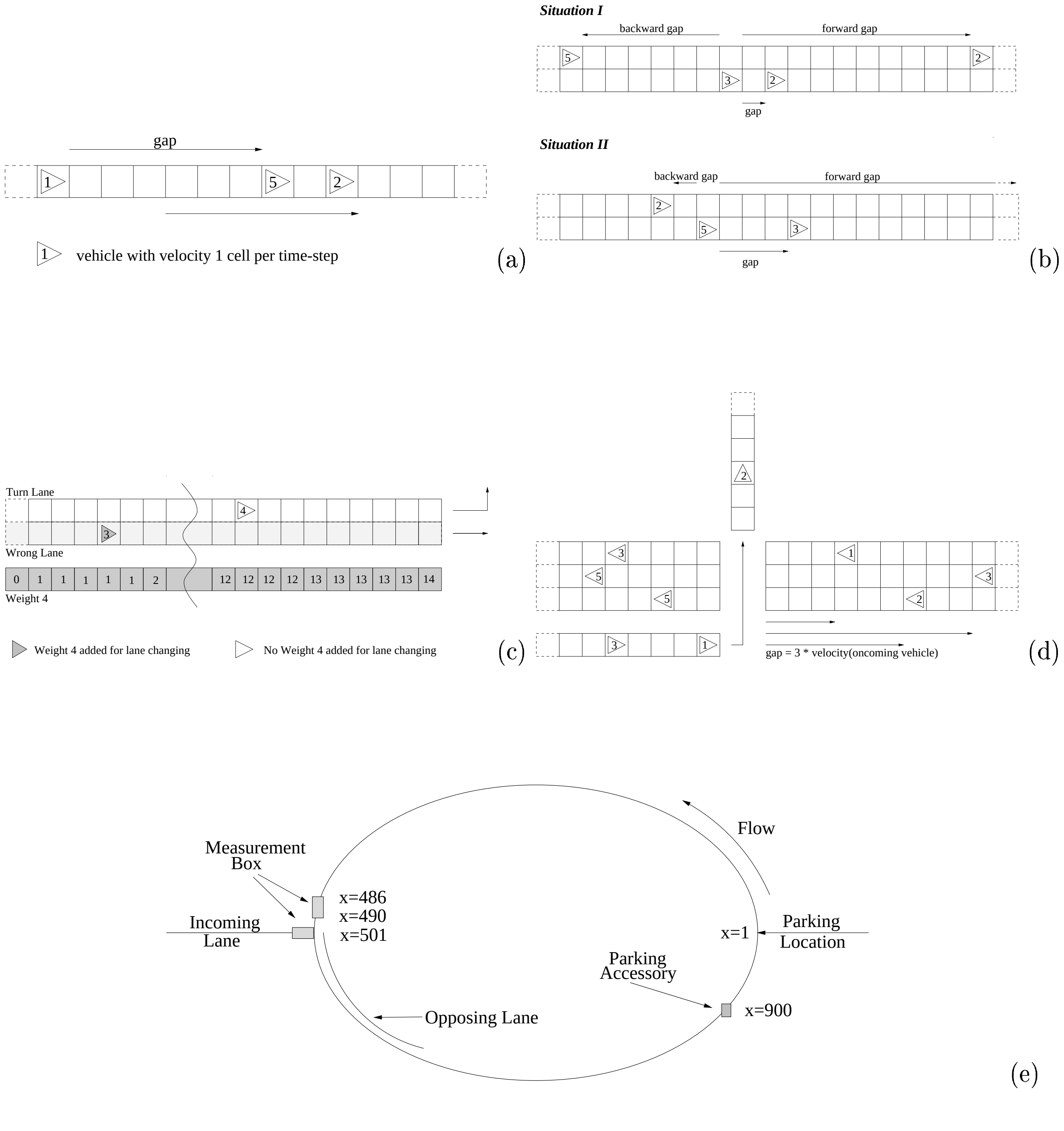

gap
1
5
2
1 vehicle with velocity 1 cell per time-step
(a)
Situation I
backward gap
forward gap
5
2
3
2
gap
Situation II
backward gap
forward gap
2
5
3
gap
(b)
Turn Lane
4
3
Wrong Lane
0 1 1 1 1 1 2 12 12 12 12 13 13 13 13 13 14
Weight 4
Weight 4 added for lane changing
No Weight 4 added for lane changing
(c)
2
3
5
5
3
1
1
3
2
gap = 3 * velocity(oncoming vehicle)
(d)
Flow
Measurement
Box
x=486
x=490
x=501
Incoming
Lane
x=1
Parking
Location
Parking
Accessory
x=900
Opposing Lane
(e)

FIG. 1. *(a)* Definition of *gap* and examples for one-lane update rules. Traffic is moving to the right. The leftmost vehicle accelerates to velocity 2 with probability 0.8 and stays at velocity 1 with probability 0.2. The middle vehicle slows down to velocity 1 with probability 0.8 and to velocity 0 with probability 0.2. The right most vehicle accelerates to velocity 3 with probability 0.8 and stays at velocity 2 with probability 0.2. Velocities are in "cells per time step". All vehicles are moved according to their velocities at a later phase of the update. *(b)* Illustration of lane changing rules. Traffic is moving to the right; only lane changes to the left are considered. Situation I: The leftmost vehicle on the bottom lane will change to the left because (i) the forward gap on its own lane, 1, is smaller than its velocity, 3; (ii) the forward gap in the other lane, 10, is larger than the gap on its own lane, 1; (iii) the forward gap in the target lane is large enough: $weight2 = v - gap_o = 3 - 10 = -7 < 1 = weight1$; (iv) the backward gap is large enough: $weight3 = v_{max} - gap_b = 5 - 6 = -1 < 1 = weight1$. Situation II: The second vehicle from the right on the right lane will not accept a lane change because the gap backwards on the target lane is not sufficient. *(c)* Value of $weight4$ when in wrong lane during the approach to the intersection. *(d)* Example of a left turn against oncoming traffic. The turn is accepted because on all three oncoming lanes, the gap is larger or equal to three times the first oncoming vehicle's velocity. *(e)* Test networks.

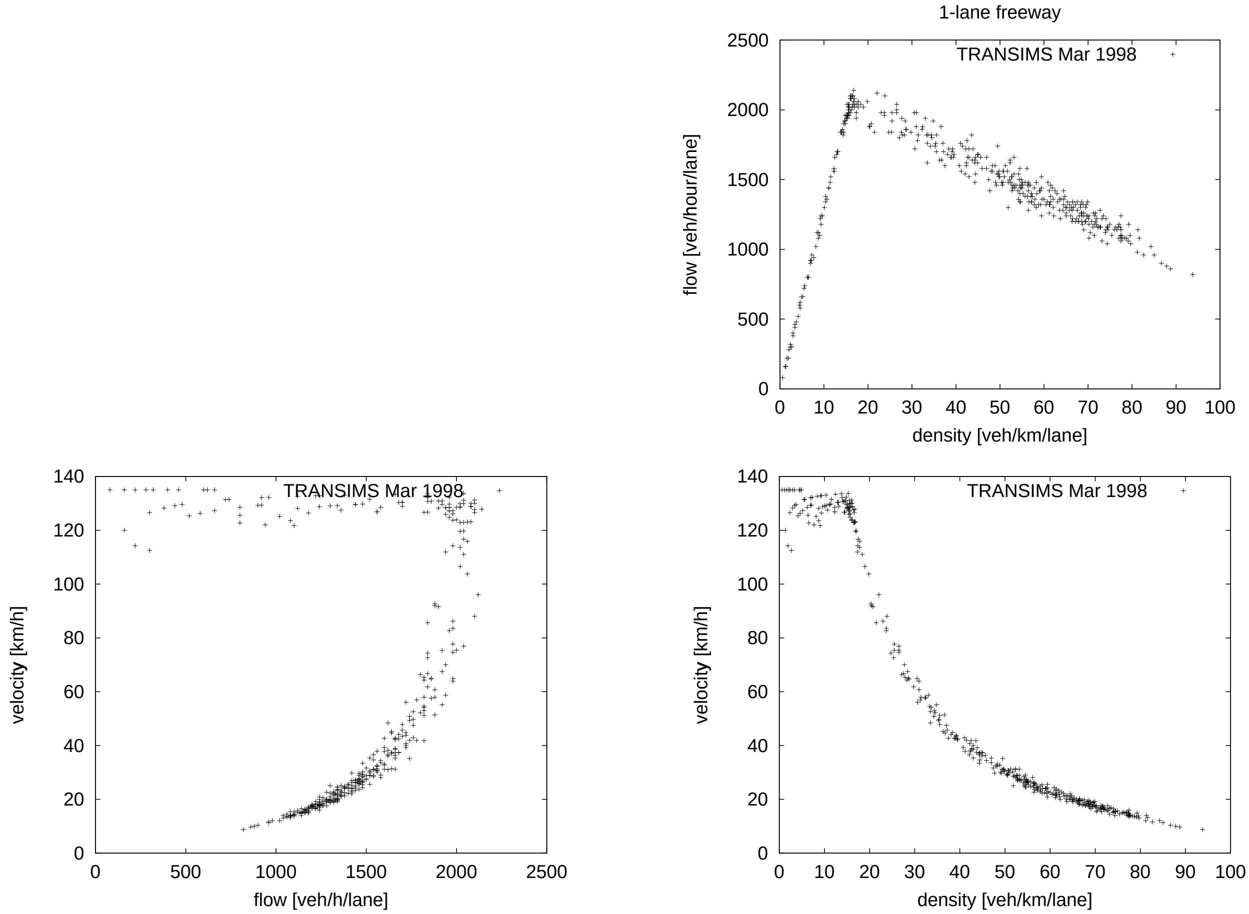


FIG. 2. One-lane traffic: Flow vs. density, travel velocity vs. flow, and travel velocity vs. density.

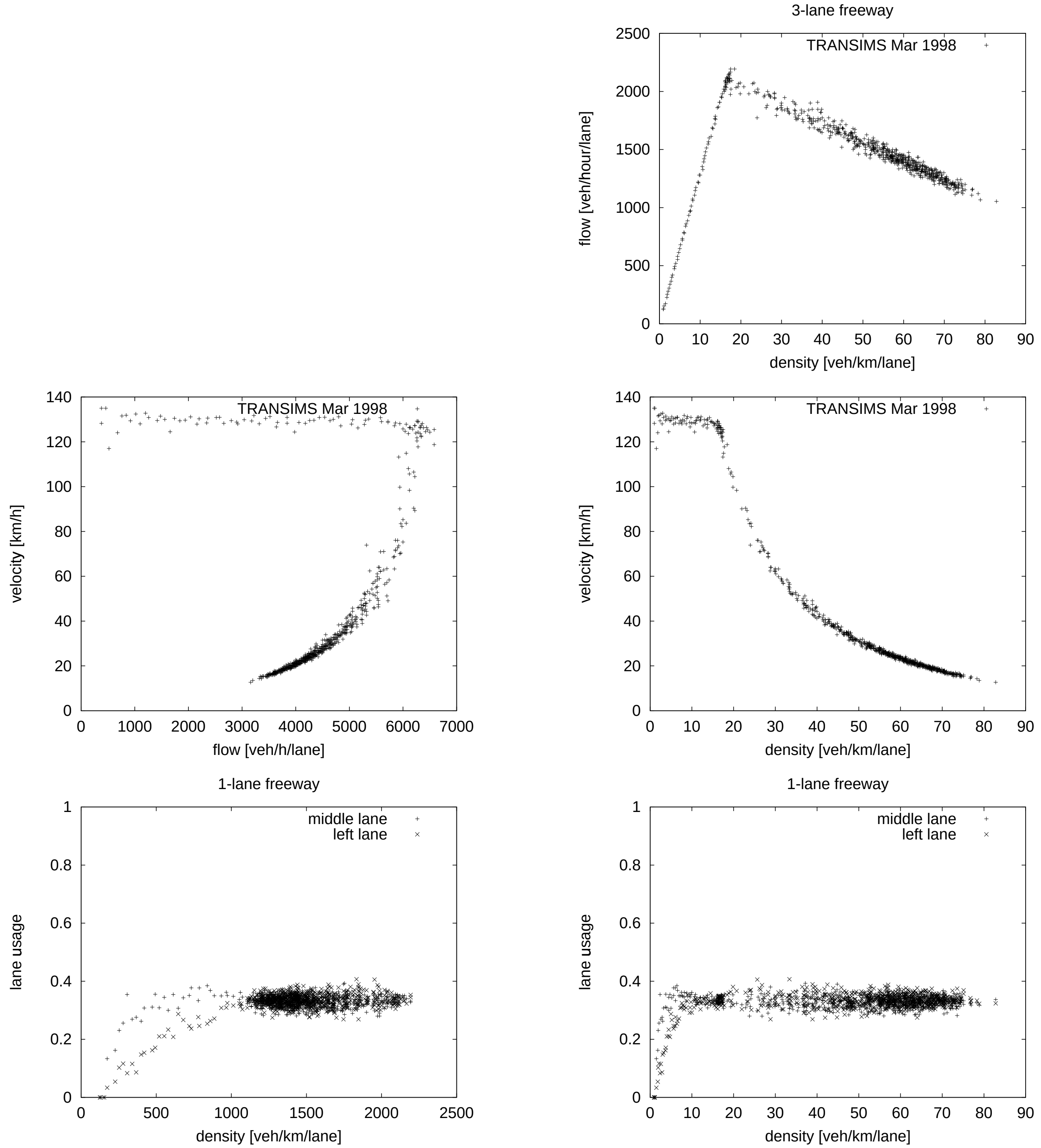


FIG. 3. Three-lane circle: Flow vs. density, travel velocity vs. flow, travel velocity vs. density, lane usage vs. flow, and land usage vs. density. The asymmetry in the lane usage at low densities is due to the fact that the parking locations start filling in vehicles on the right lane, and they only move to the left when traffic on the right lane becomes dense.

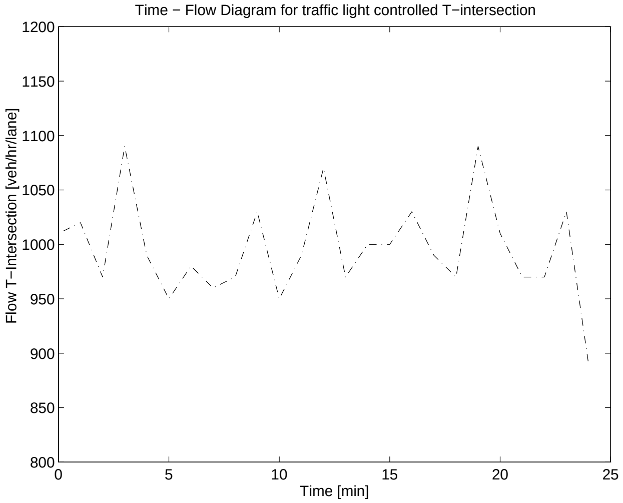


FIG. 4. Number of vehicles going through the intersection and number of vehicles "off plan" (= 0) per green phase, re-scaled to hourly flow rates per lane.

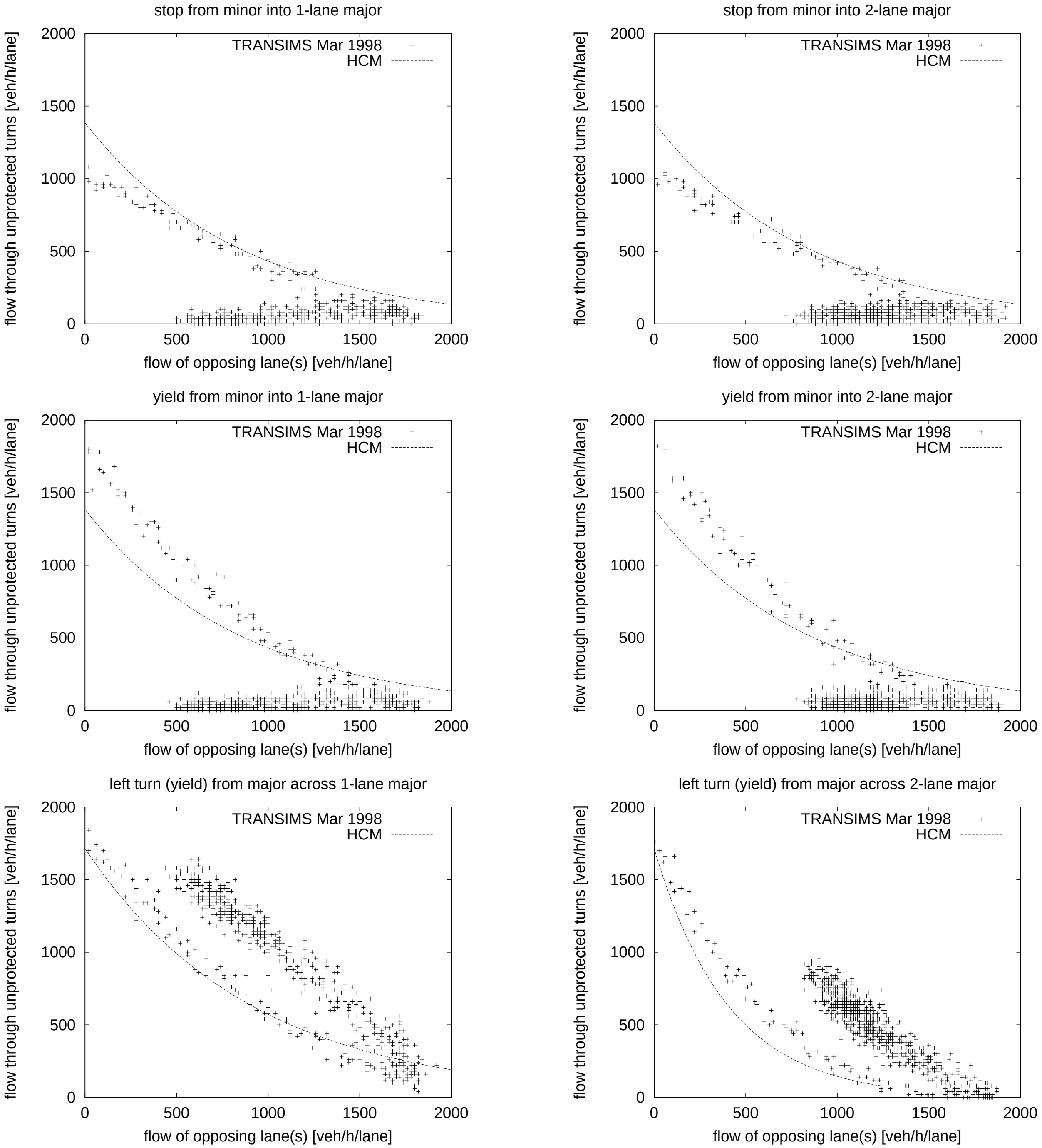
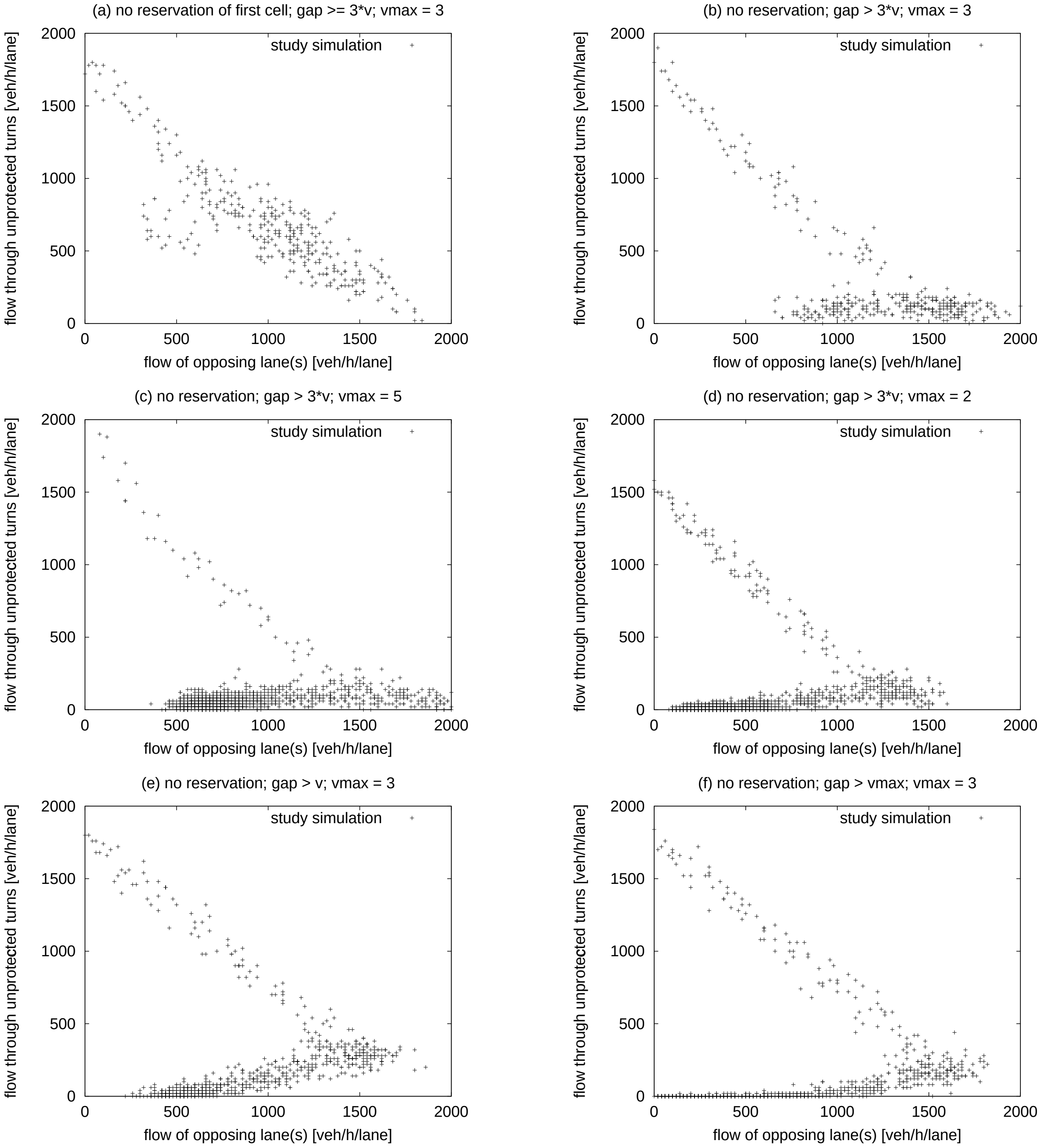


FIG. 5. Flow through stop sign, yield sign, and unprotected left turn. Left column: Major road ("circle") has one lane. Right column: Major road ("circle") has two lanes. Solid line: Highway Capacity Manual [3]. $v_{max} = 3$, gap acceptance rule is "accept if $gap \geq 3 \cdot v_{oncoming}$, and if first site on target lane available". Note that for "left turn across two lanes" (bottom right) the opposing volume is the sum of both lanes, i.e. twice the value shown on the x-axis.


(a) no reservation of first cell; gap >= 3*v; vmax = 3

(b) no reservation; gap > 3*v; vmax = 3

(c) no reservation; gap > 3*v; vmax = 5

(d) no reservation; gap > 3*v; vmax = 2

(e) no reservation; gap > v; vmax = 3

(f) no reservation; gap > vmax; vmax = 3

FIG. 6. Comparison between different rules for the case of a 1-lane minor road controlled by a yield sign merging into a 1-lane major road. *(a)* Same as Fig. 5 (i.e. $v_{max} = 3$ and "accept if $gap \geq 3 \cdot v_{oncoming}$"), except that traffic on major road does not reserve the first cell on the outgoing link, thus giving traffic from the yield sign more opportunities. Note that this seemingly small difference has big consequences in the congested regime. *(b)* Same as (a) except that acceptance rule now "accept if $gap > 3 \cdot v_{oncoming}$". *(c)* Same as (b) except that $v_{max} = 5$. *(d)* Same as (b) except that $v_{max} = 2$. *(e)* Same as (b) except that acceptance rule now "accept if $gap > v_{oncoming}$. *(f)* Same as (b) except that acceptance rule now "accept if $gap > v_{max}$".

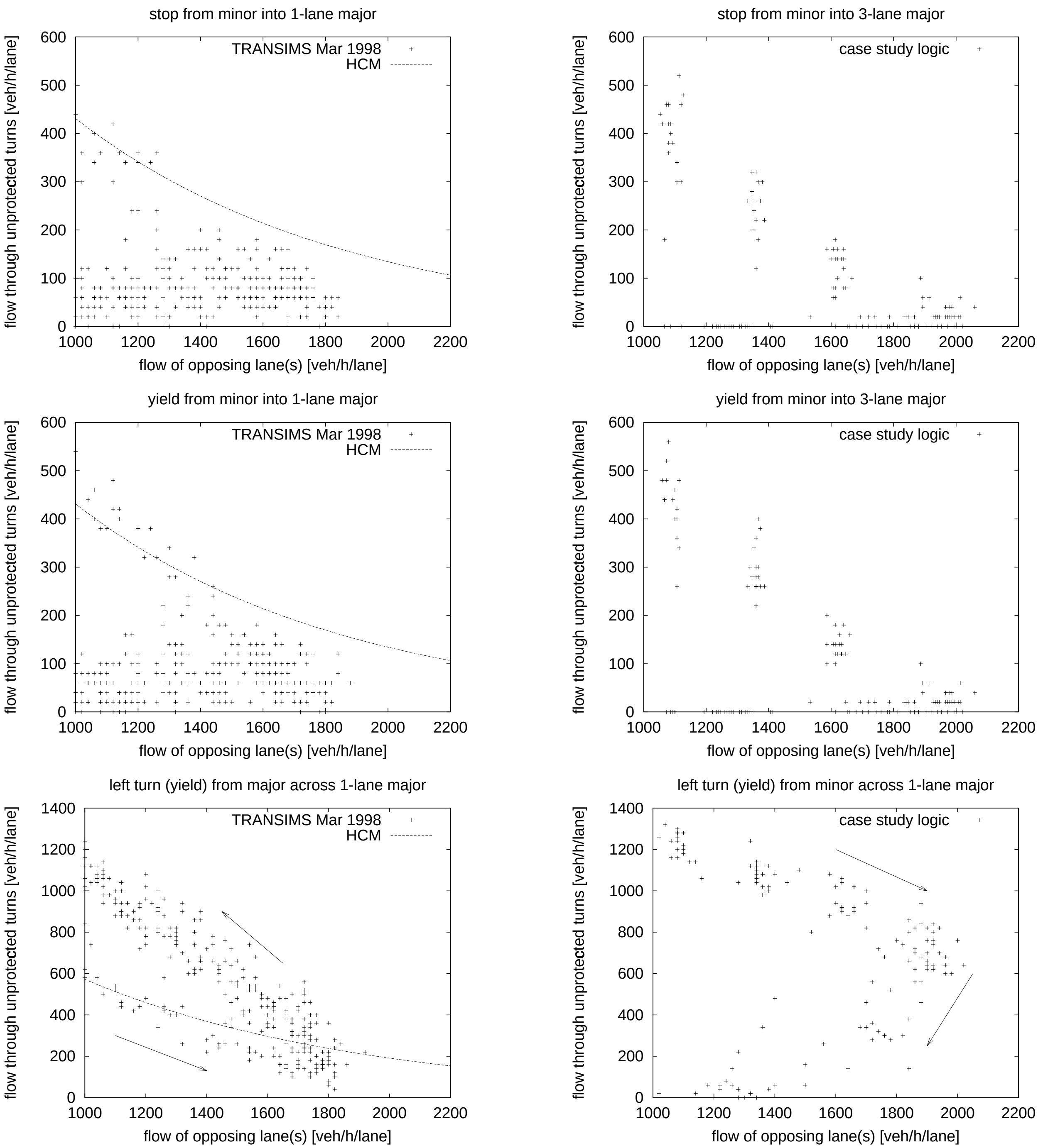


FIG. 7. Comparison between the March 1998 TRANSIMS microsimulation gap acceptance logic and the one used in the case study. Flow through stop sign, yield sign, and unprotected left turn into/across traffic on major road. Left column: March 1998 TRANSIMS microsimulation. Right column: case study TRANSIMS microsimulation. The arrows in the left turn case indicate the direction of increasing congestion. – The results are not strictly comparable because (i) the simulations in the right column were run with a maximum speed of $v_{max} = 5$ cells/update (135 km/h) vs. $v_{max} = 3$ cells/update (81 km/h) in the left column (mostly noticeable in the lower maximum flow on the major road); and (ii) the stop and yield cases on the right describe flow into a 3-lane road vs. flow into a 1-lane raod in the left column. Note that the results for the turns *into* other traffic ("stop" and "yield") are not that much different between the two whereas the result for the turns *across* other traffic ("left turn") leads to much higher flows in the uncongested and lower flow in the congested regime with the case study logic.