21 Road Intersection Model in SUMO

Daniel Krajzewicz , Jakob Erdmann
German Aerospace Center, Berlin, Germany

21.1 Abstract

Besides basic models for longitudinal and lateral movement, a traffic simulation needs also models and algorithms for right-of-way rules. This publication describes how passing an intersection is modeled within SUMO, including a description of an earlier and the currently used model.

Keywords: Road Traffic Simulation, Intersection Model.

21.2 Introduction

SUMO [1][2] is an open source road traffic simulation package developed at the Institute of Transportation Systems at the German Aerospace Center. Being a microscopic road traffic simulation – each vehicle is modeled explicitly – it uses a car-following and a lane changing model for computation of a vehicle's longitudinal and lateral movement, respectively. The models used in SUMO were initially described in [3]. For simulation of real-life networks, further models are necessary. This paper describes the current implementation of the intersection control model used in SUMO. An earlier model which is also described was tied to a fixed simulation time step length of 1 second. The current model was implemented in order to enable variable simulation step lengths.

The rest of this document is structured as following: At first, the original and the currently used model for intersection control are described. Then, the model used for simulating the speed determination of a vehicle at an intersection without the right of way is presented.

21.3 Intersection Model

Generally, road networks are represented as graphs in SUMO. An intersection ("node") consists of incoming and outgoing edges, where an "edge" represents a road with one or more lanes. Each lane has a unique id which is derived from the edge id and a number representing the lane index starting with 0 at the rightmost lane. Within an intersection, lie so called "internal lanes" which connect the incoming lanes with the outgoing lanes. Vehicle movements across an intersection proceed along these internal lanes just as they would on regular lanes.

A lane may have more than one successor lanes. The connectivity among lanes is defined with "links". In older versions of SUMO, before the introduction of internal lanes, there was a link between an incoming lane and an outgoing lane. Since the introduction of internal lanes there is a link between an incoming lane and an internal lane (called an "entry link" and another link between the internal lane and an outgoing lane (called an "exit link") as shown in Figure 21-1. The entry links of an intersection are numbered from 0 to n. Since there is exactly one exit link for each entry link, the link index uniquely defines a connection across the intersection from an incoming lane to an outgoing lane. The link indices are computed using the following scheme: first, the incoming edges are sorted in clockwise fashion. Then, the lanes, starting at the top-most are traversed. The links outgoing from a lane are then iterated, starting with the right-most destination, relative to the incoming edge. These link indices are used when discussing right of way computations.

At most intersections, vehicles wait at the stop line at the border of the intersection until they may cross conflicting streams of traffic. However, on some types of intersections, left-turning vehicles are allowed to wait in the middle of the intersection. This is modelled in SUMO by splitting internal lanes at the halting position and introducing an "internal intersection" that lies within the original intersection. Vehicles using these internal lanes always pass the entry link to the intersection and then wait at the internal intersection instead. The right-of-way computation for internal intersections follows the same principles as that of regular intersections.

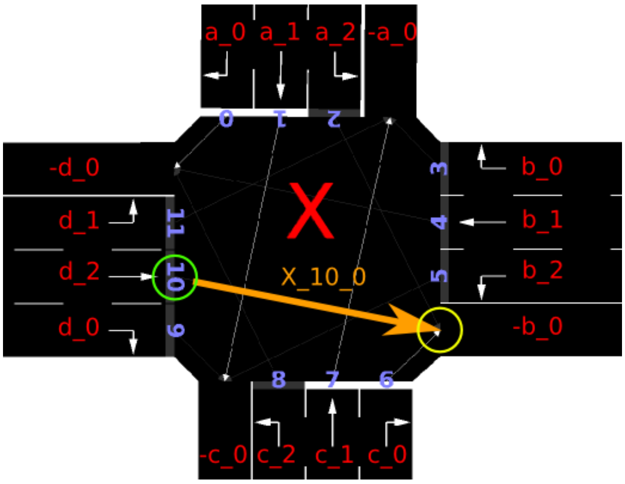

Figure 21-1: Intersection model terminology in SUMO. The intersection has id *X* and features the incoming roads *a,b,c,d* and the outgoing edges, *-a,-b,-c,-d* The connection from incoming lane *d_2* to outgoing lane *–b_0* crosses *X* on the internal lane *X_10_0*. The entry link with index 10 is circled in green. The exit link is circled in yellow.

21.3.1 Earlier Model

The right-of-way model that was implemented in the initial release of SUMO is a strong simplification of real world behaviour. When approaching an intersection a vehicle at first set the information about its approach to the intersection. After this has been done for all vehicles, the intersection "decides" which vehicles are allowed to pass without braking and which vehicles have to yield. This is done using a right-of-way matrix. This matrix describes which connections cross each other and which one has the right of way in case of crossing connections. This concept is illustrated in the following using an example.

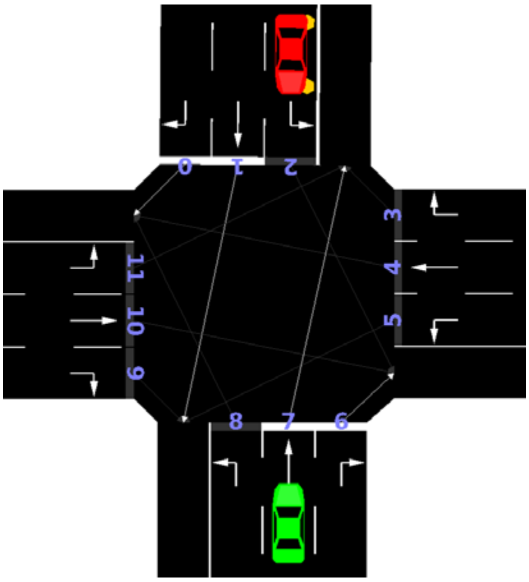

Figure 21-2: Intersection model terminology in SUMO. The intersection has id *X* and features the incoming roads *a,b,c,d* and the outgoing edges, *-a,-b,-c,-d* The connection from incoming lane *d_2* to outgoing lane *–b_0* crosses *X* on the internal lane *X_10_0*. The entry link with index 10 is circled in green. The exit link is circled in yellow.

Figure 21-2 shows an intersection which is approached by a red car on link 2 and a green car on link 7. Since the paths of both vehicles intersect and both wish to cross the intersection in overlapping time intervals, a right of way computation is performed. In Figure 21-3 the right-of-way matrix for this intersection is shown, emphasizing the discussed links. The matrix cell with row *i* and column *j* defines the right of way for a vehicle on link *i* in regard to a vehicle from link *j*. According to the colors (white/yellow/red) a vehicle on link *i* (ignores/has priority over/yields to) a vehicle on link *j*. In the example, the red car (link 2) yields to the green car (link 7) because of the red box in cell (2,7) which agrees with the common rules of traffic for left-turning vehicles. So, we see that the vehicle at link 2 has to wait for the vehicle at link 7.

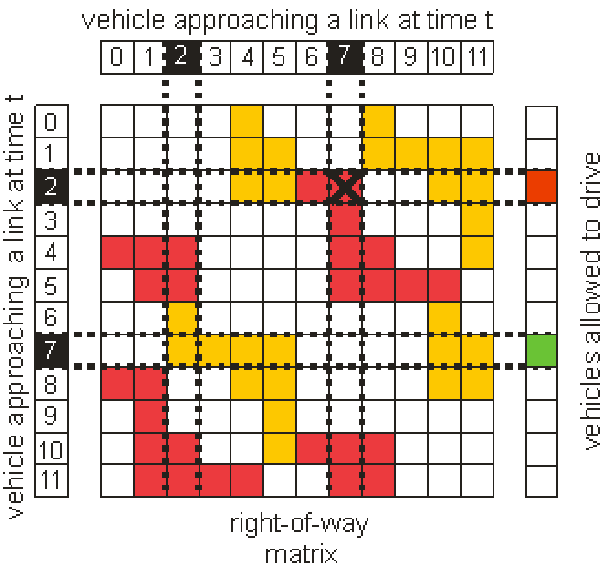

Figure 21-3: The right-of-way matrix of the intersection shown in Figure 21-2. Row i corresponds to the crossing/priority relation for link i. Link 7 crosses links 2,3,4,5,10,11 but has the right of way (yellow boxes). Link 2 crosses links 4,5,6,7,10,11 but must yield to 6 and 7 as indicated by red boxes. Since a vehicle approaches on link 7 (in the relevant time interval) the vehicle on link 2 has to brake.

The matrix itself is static and computed during the network import/generation. Traffic lights were modelled by removing the information about approaching the intersection for all vehicles that run at links that have a red light. Ignoring these vehicles during the right-of-way computation prevents them from blocking other vehicles and has the side-effect of not giving them the permission to pass the intersection.

Even though this model works well for simulation steps of one second, it caused problems when implementing sub-second time steps. Because the decision about letting a vehicle pass the intersection is performed in each time step, vehicles must not drive faster than their maximum braking ability multiplied with the step size time when being in front of the intersection. This is necessary to ensure that the vehicle can still brake if another vehicle with higher priority suddenly approaches. When decreasing the duration of simulation steps, this velocity is decreasing by the same factor, too, as depicted in Figure 21-4.

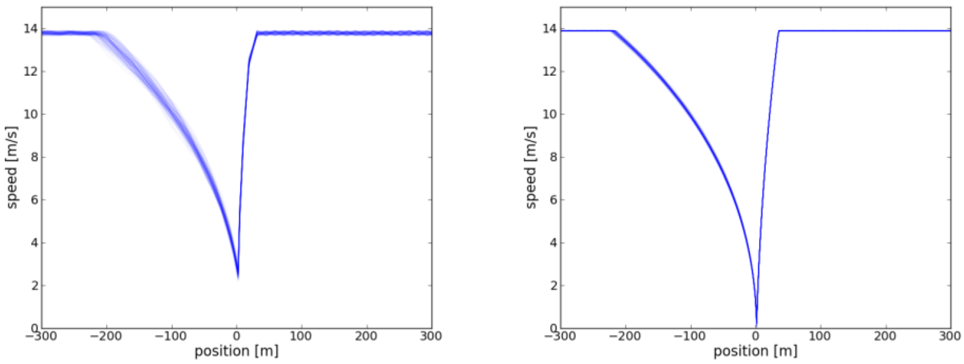

Figure 21-4: Vehicle speed when approaching an intersection in the old model; a) simulation steps of 1s, b) simulation steps of 0.1s.

This false behaviour for lower step times was the motivation to change the intersection control algorithm. Another motivation was the need to model the interaction between vehicles which occupy the intersection simultaneously. This became necessary after the introduction of internal lanes which allow the full range of longitudinal vehicle behaviour specifically unpredictable decelerations. At the end, transferring the logic for passing an intersection from the intersection model into the driver model is assumed to be a development step into the right direction, allowing further work on driver behaviour modeling.

21.3.2 Requirements for an improved Model

The goal for an improved intersection control model was to support all types of intersection typically found on European roads and to allow for realistic simulation dynamics. The following intersection types are deemed necessary:

- Intersections without prioritization (right-before-left),
- Prioritized intersections with
  - Different directions of the prioritized road (straight, turning),
  - Unprioritized lanes with yield or stop signs,
- Intersections controlled by traffic lights.

Important aspects of realistic intersection dynamics are the following:

- No deadlocks,
- No collisions,
- Efficient use of the intersection,
- Realistic acceptance gaps,
- Approaching unprioritized links without stopping,
- Qualitative dynamics independent of the simulation step length.

At the time of this writing all these goals have been met except the implementation of stop signs which is planned for a future release.

21.3.3 Current Model

The key to correct the deficiencies of the original model described in Section 21.3.1 was to not only consider the current time step, but give the right of way based on information about oncoming vehicles including an extrapolation of their time of arrival at the intersection into the future. To do so, each vehicle informs the entry link about its approach. In contrary to the initial model, not only the approach as such is stored, but also the expected time of arrival at the intersection and the speed of this arrival. Using this information, the time within which a certain link over the intersection is occupied can be computed. Each entry link also stores information about its "foe links". This corresponds to the red boxes in one row of the right-of-way matrix shown in Figure 21-3. When approaching an intersection (an entry link), a vehicle computes how long it will occupy the intersection and then checks against all approaching vehicles in all foe links of its entry link. If the requested time slot is separated from all approaching foe time slots by a suitable safety gap the vehicle is allowed to pass the entry link and thus enter the intersection. The size of the safety gap depends on the speed difference between the vehicle and its approaching foes and is set to a minimum value of 1 second.

A vehicle informs the entry links to the next few intersections on its current route (up to a distance of about 3000m) about its approach. Due to the advance knowledge of approaching foe vehicles, a vehicle approaching on an unprioritized link cannot be “surprised” by the sudden appearance of a foe. This allows decoupling the approach speed from the simulation step size. Instead, vehicles decelerate until they are one second away from the intersection. If braking is not necessary at this point they can safely accelerate and cross the intersection. The value of 1 second models an intersection with average visibility.

Once a vehicle enters an intersection by passing the entry link, this link is no longer informed. Since vehicles follow normal movement rules while on the intersection they may brake while on the intersection or even come to a stop. Therefore, other vehicles require an additional mechanism for keeping track of vehicles currently on the intersection and to avoid collisions. The goal is to use the existing functionality for letting vehicles follow each other at safe speeds. Normally, this functionality is active for vehicles which move on identical or subsequent lanes. At an intersection however, vehicles are on different lanes which cross somewhere on the intersection or merge into the same outgoing lane.

To be able to use the car following functions two things are required

1. A vehicle needs to know the lead vehicle;
2. There must be a well defined distance between the follower and the lead vehicle.

The first point is accomplished by declaring the first vehicle of any two vehicles to enter the intersection as the leader. This is particularly important, because several vehicles may be driving within the space of the intersection at the same time and there must be an antisymmetric, transitive and irreflexive leader relation among them to avoid deadlocks. The second point is accomplished by virtually superimposing both internal lanes up to the crossing point. If both internal lanes merge into the same outgoing lane, the crossing point is naturally the beginning of the outgoing lane. Let follower-vehicle F have distance d_F from the crossing point and let its leading vehicle L have distance d_L from the crossing point then the virtual gap g between both vehicles is defined as

$$g := d\_F - d\_L - length(L) - minGap(F)$$

where *length(L)* is the physical length of vehicle *L* and *minGap(F)* is the minimum gap that vehicle *F* intends to keep to its leader at all times. Note that *g* may be negative which causes vehicle *F* to stop.

In the current implementation each exit link maintains a list of “foe internal lanes”. These are the lanes which correspond to the yellow and red boxes in one row of the right-of-way matrix in Figure 21-3. In other words, these are the internal lanes which intersect with the internal lane the approaching vehicle intends to use.

A vehicle that wishes to pass an exit link on its route asks this link for any additional vehicles to which it must adapt its speed. These vehicles are called link leaders. The link checks all of its foe internal lanes for occupancy, computes the virtual gap and returns each found vehicle as a potential link leader to be followed.

Figure 21-5 shows the same intersection as Figure 21-1 with three vehicles green (G), blue (B) and red (R). Vehicle R wishes to pass the exit link that belongs to link 11. Both vehicles G and B are on the same internal lane which is a foe internal lane for link 11. On the left side of Figure 21-5, vehicles G and B are potential link leader for R. Since G and B have entered the intersection before R, they will both be followed. In this case only the speed adaption to B is relevant since B is already following G. On the right side of Figure 21-5 the situation is slightly different. Vehicle R has already entered the intersection before vehicle B and therefore, R only follows G.

In the current implementation each vehicle maintains a list of link leaders being followed for each exit link. This list is used when maintaining the antisymmetric link leader relation among vehicles (vehicle R only sets vehicle B as its link leader if B does not already have R as its link leader).

The link based model of detecting conflicting approach information coupled with the handling of link leaders allows for full vehicle dynamics on the intersection together with efficient use of the intersection as a natural extension of car following.

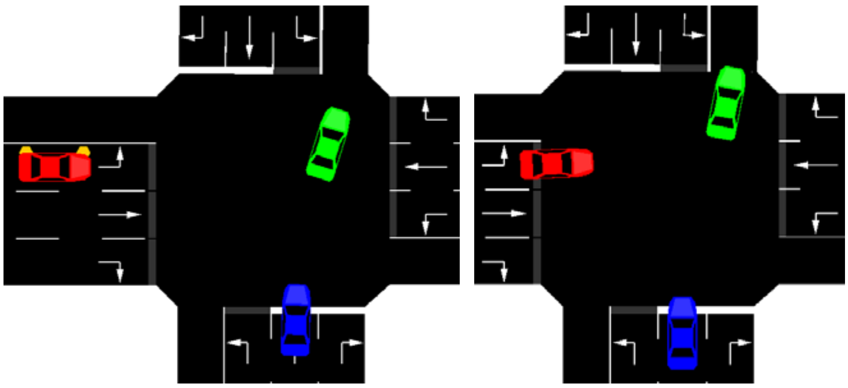

Figure 21-5: Examples of link leader relations for the vehicles green (G), blue (B) and red (R). In the left figure G is the leader of B and both are the leaders of R. In the right figure G is the leader of R and R is the leader of B because R entered the junction before B.

21.4 Approaching Links

The behaviour when approaching an intersection without having the right-of-way was consolidated for different time step sizes in the following way.

When the ego vehicle approaches an intersection where other vehicles may have the right of way, forcing ego stop, it decelerates to a velocity which allows braking in front of the intersection. One second before reaching the intersection, the ego vehicle decides whether the intersection may be crossed or not. If crossing is possible at this time, the vehicle may accelerate again, otherwise it decelerates to a velocity of zero.

Figure 21-6 shows that using this definition assures the similar behaviour for different simulation step sizes. The velocity used for approaching the intersection is the vehicle's deceleration capability multiplied with 1 s. For the standard Krauß parameters it is equal to 16.2 km/h, what was found to be empirically valid when compared to measures obtained from test drives with DLR test vehicles. Within the current model, the vehicle's maximum deceleration ability is used for all intersections and all directions of driving across them. Because in reality, this speed is mainly dictated by the possibility to look into foe lanes for determining whether the intersection may be crossed, further extensions of the model, in means of differing between approach velocities promise to improve the model's quality. It should be also noted that the simulated time line of deceleration and acceleration is not yet matching the reality.

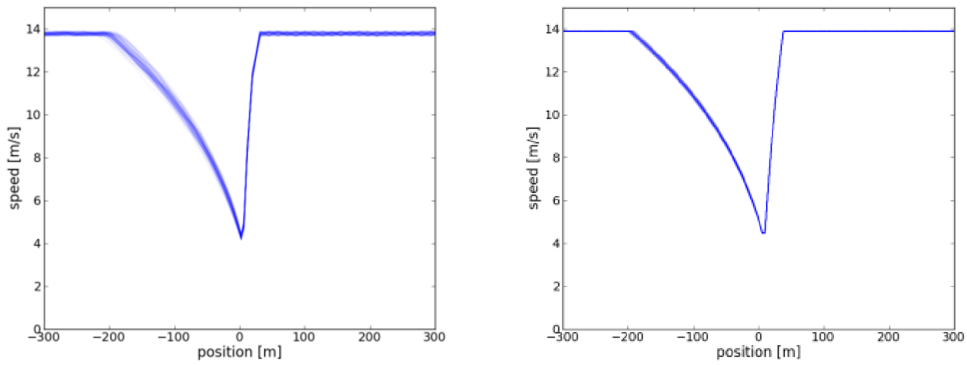

Figure 21-6: Vehicle speed when approaching an intersection in the new model. a) simulation steps of 1 s, b) simulation steps of 0.1 s.

21.5 Estimation of Link Entry/Exit Times

In the following, the estimation of times and speeds of arrival and leaving an intersection is discussed. Figure 21-7 shows the deviations of the estimated speeds and times over time for a major (high prioritized) link. These vehicles do not have to break. "deviation" denotes here the difference value obtained by subtracting the real from the estimated value in the following Figures. One may see that the times of arrival and leaving are both estimated too low and only increasingly move towards the correct value. This is due to the random "dawdling" behaviour of SUMO's default car-following model, see [4]. If the dawdling is disabled, the estimation is correct from the very begin on (not shown, here). The deviations in time are due to the same reason. They are straight, as in each time step, the estimation is based on the perfect speed (50 km/h in the shown example) and the dawdling is performed by the model afterwards. It should be noted, that the estimation could be more correct, if the dawdling, regarding its stochastic nature, would be taken into account during the computation of the times/speeds.

The additional error in estimating the leave time is probably due to taking into account the distance to the leader in jam/when standing (SUMO's "minGap" attribute of a vehicle type), which was set to 2m in the shown example; 2 m divided by 13.89 m/s gives the offset shown here, which is about 0.14 s.

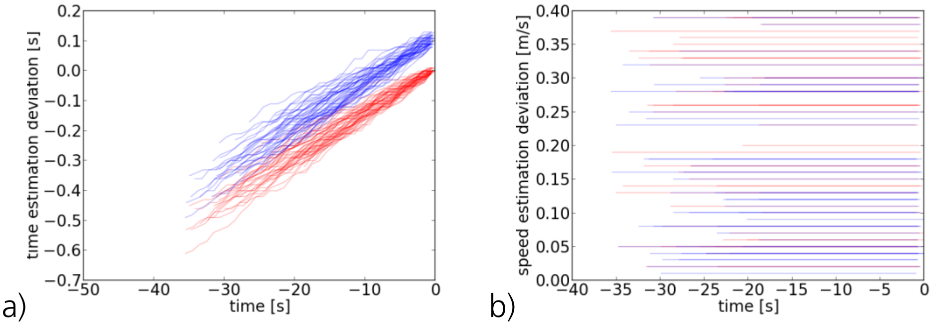

Figure 21-7: Deviations of the estimated time (left) and speeds (right) from their final counterpart for arrival at (red) and leaving an intersection (blue). High prioritized vehicles

The difference in starting times is due to using a random position for the place the vehicle departs at. This was done for adding randomness into the possible co-occurences of vehicles at high and at low prioritized roads. The behaviour of vehicles on prioritized roads is straightforward and can be easily explained, see above. But the behaviour of vehicles that have to react to crossing traffic are more complicated. Shown in Figure 21-8, the shape of time estimation development has three peculiarities. The first are large overestimations of the arrival and the leave time by about 260 s. The second can be seen better when focussing on the majority of traces, as done in Figures 21-8b and 21-8d. They show that the speed is – in addition to the continuous progress towards a correct value – oscillating with an amplitude of 2 s. The reason could be the dawdling, as discussed for vehicles approaching a major intersection. But, when looking at the same run with a dawdling value set to zero, as visualised in Figures 21-8c and 21-8d, some oscillations are still visible. The third peculiarity is an overestimation shortly before the link is reached.

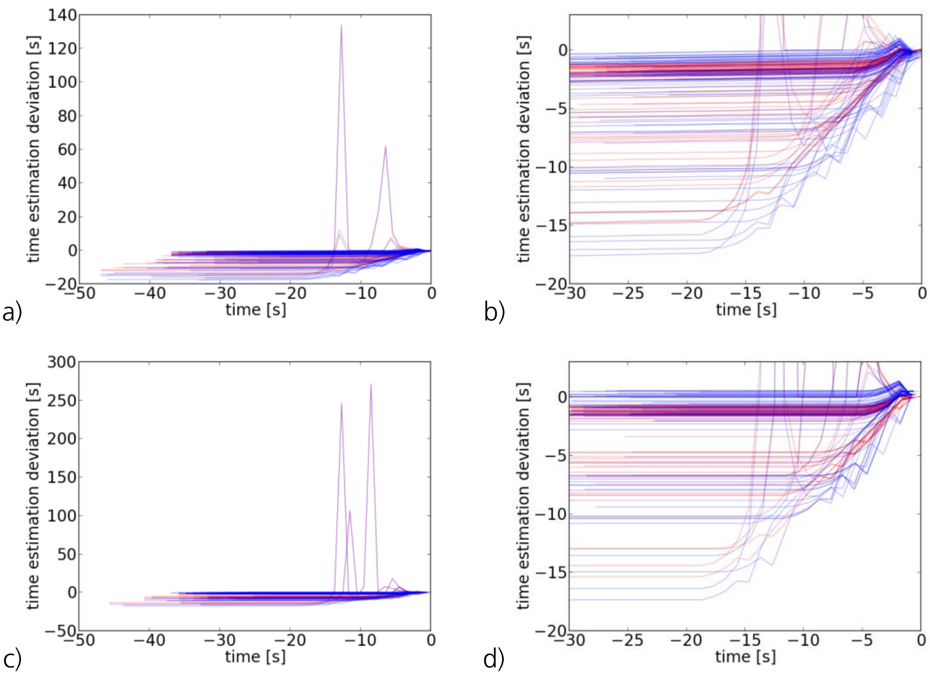

Figure 21-8: Deviations of the estimated time from their final counterpart for arrival at (red) and leaving an intersection (blue). Low prioritized vehicles. Top: with default dawdling, bottom: with no dawdling, left: the complete figure, right: focus on the majority of approaches

At the current time, these effects cannot be explained.

21.6 Summary

The currently implemented model for right-of-way rules at intersections was presented. It was shown that it is capable to work with different time steps. Additionally, some initial evaluations of the approaching behaviour were given. Besides additional explanations, missing at the current time, comparisons against real-world trajectories should be performed.

21.7 References

[1] Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO – Simulation of Urban MObility: An Overview. In: SIMUL 2011, The Third International Conference on Advances in System Simulation (2011)

[2] DLR and contributors: SUMO homepage. http://sumo.sourceforge.net/ (2013)

[3] Krajzewicz, D.: Traffic Simulation with SUMO – Simulation of Urban Mobility. In: Barceló, Jaume (Ed.): Fundamentals of Traffic Simulation, Series: International Series in Operations Research & Management Science, Vol. 145, Springer, ISBN 978-1-4419-6141-9 (2010)

[4] Krauß, S., Wagner, P., Gawron, C.: Metastable states in a microscopic model of traffic flow. Phys. Rev. E, American Physical Society, 1997, 55, pp. 5597-5602, (1997)