SUMO User Conference 2025
Conference paper
https://doi.org/10.52825/scp.v6i.2638

Published: 15 Jul. 2025

SUMO's Interpretation of the Krauß Model

Jakob Erdmann[1,*] and Peter Wagner[1,2]

[1]DLR Institute for Transportation Systems, Germany
[2]TU Berlin, Institute for Land- and Sea Traffic, Germany
*Correspondence: Jakob Erdmann, jakob.erdmann@dlr.de

Abstract. This work gives a short impression on the long history of SUMO's default car-following model, the one according to Stefan Krauß. It tries to summarize some of the ideas that were needed to change a beautiful theoretical model into a versatile and reliable tool that does its job in a wide variety of different situations, almost without producing obnoxious surprises.

Keywords: Car Following, Krauß Model, SUMO

1. Introduction

The simulation models built into traffic microsimulation software can be viewed as an approximation to the original model, with models built into SUMO [1] being no exception. Differently from the original source, they need to perform correctly and reliably under a huge amount of different conditions, or being used for different purposes than originally intended [2]. This leads to adoptions and changes to the original model that are often not well documented. In addition, they are enhanced with lane-changing [3], [4] and intersection behavioral models [5], [6] that were absent from the original model, again introducing subtle modifications.

Here, a short account of SUMO's default model, the SK model (after Stefan Krauß) introduced in [7], is presented. This article is restricted to the car-following (CF) part. The CF is viewed as the most important part of any microscopic traffic model, it describes how a following vehicle reacts to the actions of a leading vehicle (acceleration, deceleration, stopping etc). Basically, the following vehicle should not crash into the lead vehicle, and it should not lag too far behind, at least if it is truly in CF-mode. The traffic literature is full of such models, and still new ones are invented and described in the literature.

The model [7] which serves as the default model in SUMO and was further investigated in [8], is based on the idea, that a vehicle must perform a crash-free trajectory. It is crash-free, if the distance between leader's rear and follower's front (usually called $g(t)$ ("gap")) is always positive for all times t. This can be achieved by computing a safe speed v_{safe} and request that the updated speed $v(t + \Delta t) \leq v_{\text{safe}}$, where v_{safe} may depend on g, the speeds of the leader V and the follower v, respectively, and on some additional parameters. In the original article [7], this results in a complicated formula based on the discrete braking distances, which involved a square-root.

Here, the presentation follows a simpler path, but note that a discrete formula for the braking distance is used in SUMO as well, see the function `MSCFModel::maximumSafeStopSpeedEuler()` in the code. A safe speed can also be derived from a continuous representation, which is close to what some of us have learned in driving school. It states:

$$\frac{v^2}{2b} + v\tau \leq \frac{V^2}{2B} + g, \tag{1}$$

where the first term is the braking distance of the following car (it brakes with braking deceleration b), the second term is the distance traveled during the minimum time headway τ, and the first term on the r.h.s. is the braking distance of the leading car, which decelerates with B. Similar ideas have been put forward already 20 years before, see, e.g. [9]. This condition should be fulfilled in each time step t (in the following, t is omitted as often as possible to make the equations easier to read), where time proceeds in integer multiples of the time-step size Δt. Eq. (1) can be solved for v to lead to a safe value of the speed $v(t+\Delta t) \leq v_{\text{safe}}$, the original work used $B = b$:

$$v_{\text{safe}} = -b\tau + \sqrt{(b\tau)^2 + V^2 + 2bg}, \tag{2}$$

which again contains a square-root. This equation is still in the SUMO code, but no longer in the default model, see `MSCFModel_KraussOrig1::vsafe()`. To avoid the costly call to the square-root (which was an issue at the computing machines in the late 1990ies[1]), [8] came up with a simplification, which can be reached by a Taylor expansion of Eq. (1) around $\bar{v} = (v+V)/2$. He had two additional arguments in his favor, namely that to ensure safety it might not needed to actually 'compute' the full braking distance, and, since the Taylor expansion involves the derivative of the braking distance with respect to v, there is no need in principle for having a fixed b, it might be speed dependent. This then ends in Eq. (3), which might have been in very early versions of SUMO's code:

$$v_{\text{safe}} = V + \frac{g - V\tau}{\frac{v+V}{2b} + \tau} \tag{3}$$

Clearly, Eqs (2) and (3) are not identical, however the difference especially for small Δv is small enough to be not of importance. See Figure 1 for a comparison of the two formulas, and in fact, the differences are small except for very extreme values of Δv, when the following car is approaching the leader rapidly (the total MAE in this Figure is $0.054\,m/s$, and $0.014\,m/s$ for $\Delta v > -20\,m/s$).

To complete this, the full update equations of the model read:

$$\tilde{v} = \min\{v + a_{\text{max}}\Delta t, v_{\text{safe}}, v_{\text{max}}\}, \tag{4}$$

$$v(t+\Delta t) =: v' = \tilde{v} - a_{\text{max}}\Delta t \varepsilon \xi, \tag{5}$$

$$x(t+\Delta t) =: x' = x + \Delta t v'. \tag{6}$$

This was being used in SUMO for a long time (and is now `gSemiImplicitEulerUpdate`). Here, the short-cuts (v', x') for the updated variables $(v(t+\Delta t), x(t+\Delta t))$ have been introduced. Furthermore, ξ is a random number drawn from the interval $[0,1]$ and ε is the strength of the noise, again from the interval $[0,1]$. This models driver imperfections, and it renders the deterministic model ($\varepsilon = 0$) into a (dynamical) stochastic one.

[1] On modern machines the computation time difference does not existent anymore, we have checked. And: it is not of importance for SUMO, since only about $1/3$ of the compute time goes into the actual update of the speeds. The other time is spent on intersection behavior and lane changing.

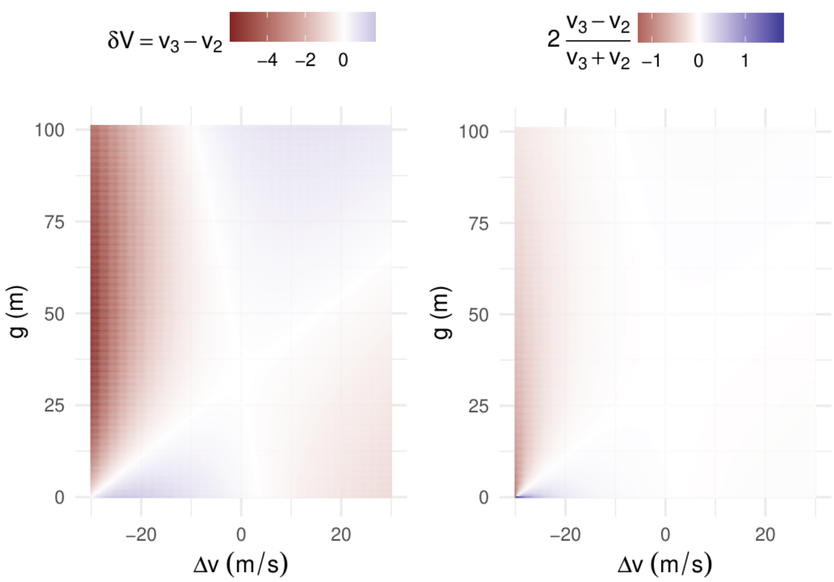

Figure 1. *The difference between the two equations for v_{safe} (left) and the relative difference (i.e. 1 = 100%, right), plotted as a function of g and Δv, for a fixed follower speed $v = 30\,m/s$. The indices relate to the equation numbers in this text. The parameters used are $b = 3.5\,m/s^2$ and $\tau = 1.1\,s$.*

It has two additional parameters a_{max}, v_{max} for the maximum acceleration and speed, respectively. Crash-freeness is guaranteed only if $\Delta t \le \tau$, and if $b \le B$ (this was not mentioned in [8], it has been learned later); and it has been proved in [8] only for the approximated formula in Eq. (3). Note, that the proof of crash-freeness is not a general crash-freeness but states that if a system is started in a safe state (i.e. where all speeds are below their safe speeds), then the update does not change this. If the initial condition violates this condition, then the result is not clear, vehicles may crash. This lesson has to be learned when it comes to vehicle insertion, and to lane-change, where additional care is needed.

Implicit in the formulation of the gap is also the vehicle length, and the additional parameter g_0 (aka `minGap` in SUMO) which models the distance between two stopped vehicles. This completes the small set of seven parameters of the model, $p = (a_{\text{max}},\, v_{\text{max}},\, \tau, b,\, \varepsilon,\, \ell_{\text{car}}, g_0)$. All parameters, except ε, have a clear physical meaning, and could in principle measured. Note, however, that in a calibration task it might be better to use them as fudge parameters and try to minimize a given objective function without forcing an exact match between the parameters in the model, and the parameters as they could be extracted from real data. Also worth mentioning is the fact that in SUMO each vehicle can have its own set of parameters, making it a heterogeneous fleet to simulate.[2] This is the default in SUMO since at least the maximum speeds are distributed; and it is important, for instance if one wants to get the right fundamental diagrams so that it can be compared with frameworks like the HCM (Highway Capacity Manual [10]) or the German HBS. See [11] for SUMO's performance in this regard. Note, too, that the common parlance in the community is not uniform: in some work, the fact that the parameters are distributed constitute already a stochastic model, which, from a dynamical perspective is still a deterministic model (provided SUMO is used with $\varepsilon = 0$). The same holds for the insertion of vehicles (demand): this may be stochastic, and therefore two simulation runs produce different results if a stochastic demand is used, but as long as $\varepsilon = 0$, the simulation itself is deterministic.

[2]We still have to review papers where researchers claim this to be their main innovation.

2. SUMO's extended SK model

Although some problems with the approach so far have been noted before, they became urgent when the ballistic update was introduced into SUMO. Eq. (6) is a so called semi-implicit Euler update in SUMO, and as a side note it should be mentioned that this is an example of a symplectic integrator[3]. Its precision is $\mathcal{O}(\Delta t)$, while the so called ballistic update is $\mathcal{O}(\Delta t^2)$:

$$x' = x + \frac{\Delta t}{2}(v + v') = x + v\Delta t + \frac{1}{2}a\Delta t^2. \tag{7}$$

This innocuous change has some consequences, which are discussed in depth in [12]. Two points are important here:

- The ballistic update introduces and additional term in Eq. (1), it now reads $v'^2/(2b) + \tau(v+v')/2 \leq V^2/(2B) + g$, when $\tau = \Delta t$. This changes the computation of the safe speed, it now becomes (again, $b = B$ is assumed):

$$v_{\text{safe}} = -\frac{b\tau}{2} + \sqrt{\left(\frac{b\tau}{2}\right)^2 + V^2 + 2bg - b\tau v}, \tag{8}$$

- Stopping of a vehicle: Suppose, a vehicle brakes with b and has in time-step t a speed $v(t) < b\Delta t$. Then, the new speed $v(t+\Delta t) = v(t) - b\Delta t$ will be negative. The Euler update above does silently set $v(t+\Delta t) = 0$, which is consistent within this approach. The ballistic update cannot be handled in this manner. When forcing $v(t+\Delta t) = 0$ the new position $x(t+\Delta t)$ is larger than what would come out of a correct treatment by either computing the exact time of stopping $v(t)/b$, or by using the braking distance formula $v(t)^2/(2b)$ which was also used in the derivation of the safe speed. Since $v(t+\Delta t) < 0$ the new position that came out of the ballistic update is smaller than the one obtained by simply setting $v(t+\Delta t) = 0$, therefore leading to crashes.

Another point has been mentioned already: the crash-freeness of the model hinges on $b \leq B$. This is due to the fact, that the safety condition Eq. (1) is only a terminal condition. It is a fine irony, that on improving the braking capabilities of the follower by making $b > B$, crashes are introduced. They happen during the braking process, while still fulfilling the final condition after stopping that the follower stops behind the leader. See again [12] for more details on this, and Figure 2 for a visualization.

And on top of that, there are sometimes floating point issues that lead to crashes, because the final gap becomes -10^{-6} or so. To deal with this, SUMO uses a refined approach. Here, we describe only the important points and ignore code that deals with insertion, lane-changing, or emergency braking, which are all in there and make the code difficult to grasp at once. Note, that the $b > B$ issue has a very simple solution in that the following vehicle assumes that the leader brakes with b, not with B. This leads to a slightly sub-optimal result, since the correct solution (which can be found, but is in fact complicated, see again [12]) would allow for higher speeds.

3. SUMO's approach

To approach the points above, the CF update in SUMO for the default model is described into more detail below. Note, that still a lot of points are skipped, we hope we

[3] A symplectic integrator is a numerical integration scheme, which preserves the energy of the system. This is not important in CF models, nevertheless the difference between this and even the ballistic update (which is not symplectic) can be seen in the very simple CF-model $\dot{v} = g - V$, which is a harmonic oscillator in $(\Delta v, g)$-space.

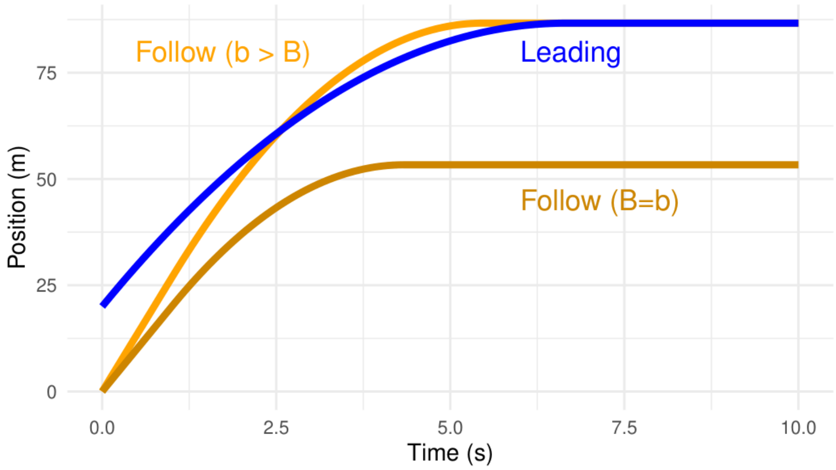

Figure 2. *Three braking curves: the leading vehicle (blue line) brakes with $B = 3\,m/s^2$, while the follower (orange line) used $b = 6\,m/s^2$. In the final position, they are fine, but during the braking, a collision has happened. SUMO avoids this by the follower assuming $B = b$ (brown line), therefore ending up a considerable distance behind the optimal stopping distance. However, the simulation (not shown) corrects for this by the re-evaluation of v_{safe} in any time-step, so that $g(t) \geq 0$ is fulfilled for all t.*

have picked the most important ones. Note, too, that some constants like the time-step size is omitted in this description, to keep things as clear as possible. Most of the collision-freeness has been moved into `MSCFModel`, since it could be of use in other models as well, most notably it is used in the ACC and CACC models. Furthermore, the code must not only handle the CF itself, but also the insertion of a new vehicle into a running simulation, as well as the lane-changing, and the stopping process.

- The CF process starts with `MSCFModel_Krauss::followSpeed()`, which delegates directly to `MSCFModel::maximumSafeFollowSpeed()`. This way, the functionality can serve as a *safety feature of last resort* for other car-following models.
- By computing the brake gap of the leader car, the original safety equation Eq. (1) is simplified to $v^2/(2b) + v\tau \leq C$. Where C is computed with the help of function `brakeGap()` and the equation is solved with function `maximumSafeStopSpeed()`.
- The `brakeGap()`-function computes the distance a vehicle needs to come to a complete stop, based on the current speed v, the headway τ, and the braking deceleration b. It branches into a piece for the Euler, and a piece of code for the ballistic update. The ballistic update uses the continuous formula above, i.e. $v(\tau + 0.5v/b)$, while the Euler uses a discrete version, $n\,v - b\,n(n+1)/2) + v\tau$, where $n = \lfloor v/b \rfloor$ (the floor) is the number of steps needed to reach $v < b$.
- `maximumSafeStopSpeed()` uses this brake gap to compute the `safeStopSpeed()`, which again requires two separate functions and formulas. While the Euler version is only one step with a complicated formula (accounting for piecewise constant speeds), the ballistic update must take care of the fact that a stop occurs within the next update. This is handled so as to compute a substitute deceleration so that the vehicle comes to an exact stop at the end of the next time-step. This could have been handled differently, see [13] for a slighly different approach.
- As mentioned already, the update takes care of the issue where $b > B$ by assuming in the various equations that the leader is braking with b, not B.
- Once the updated speed of the following vehicle is computed, the rest of code sets the boundaries (of acceleration and speed) and finally in fact uses either Eq. (6) or (7) to update the positions of the vehicles.

4. Collisions

Early versions of SUMO implemented Eq. (2) and this formula can still be found in `MSCFModel_KraussOrig1` (activated as `carFollowModel="KraussOrig1"`). However, this formula does not match the vehicle dynamics of the original Euler-update rule of Eq. (6) and could actually produce collisions. In the earliest version of SUMO this was circumvented by permitting arbitrary decelerations in dangerous situations, which was also the original approach used in [8]: there, decelerations were not limited to b, albeit the deceleration used by the model was in most cases smaller than b. Later versions of SUMO limited deceleration to the value b assumed by the formula and thereby uncovered the collision problem. This prompted introduction of the Euler-variant-formulas for `brakeGap()` and `safeStopSpeed()`. Still later versions introduced another model parameter to serve as a hard limit on deceleration (`emergencyDecel`).

However, collisions that are intentionally caused by the model design rather than by conceptual problems or bugs are a desirable feature for traffic simulations since they increase the breadth of possible situations that can be simulated. Thus far, the following features are able to cause rear-end collisions with the default Krauss model in SUMO:

- Setting a value of τ that is lower than the simulation step size. Since the simulation step size sets a practical limit on the reaction time of the cars, vehicles continue at their chosen speed for longer than anticipated by the safety formula.
- Activating a model for perception errors (`device.driverstate`) which applies symmetrical noise to the leader gap and leader speed before computing the safe velocity [14].
- Activating model parameter (`apparentDecel`) which replaces the true braking capabilities of the leader vehicle. Setting this to a lower value than the actual leader deceleration can cause unsafe situations. Due to the fact that the leader deceleration is increased up to the value of the follower deceleration, this only has an effect if the vehicle fleet uses heterogeneous deceleration values.

Further collisions and dangerous situations can be created by suitable configuration for the intersection model and lane change-changing model.

5. Conclusion

This text has given a short account on the cliffs that had to be circumvented to make SUMO’s default model as it is now. Especially the mechanism that ensures crash-freeness is also used with most of the other models implemented in SUMO (some of them are not crash-free by construction, like the ACC models), therefore it is quite general and versatile.

There is another kind of irony involved here: when modeling traffic safety, the crash-freeness is a show-stopper in the first place. This can be handled with SUMO’s tooling as well, most notably with the action-step and the perception error mechanism. Note, however, that considerable modeling work is needed to transform these basics into a fully fledged traffic safety model. For instance, especially with the models for the perception errors, a completely different world is entered, since almost nothing could be measured: we only see the reaction of the driver on changes in the input variables, but we cannot observe how they came up with the decision to change the acceleration. Clearly, this is a paper for another time.

Data availability statement

No data have been used in this work.

Author contributions

Jakob Erdmann: Software, Validation, and Investigation. **Peter Wagner**: Conceptualization, Methodology, Writing-, and Original draft preparation. **All**: Writing-, Reviewing, and Editing.

Competing interests

The authors declare that they have no competing interests.

Funding

This work used the basic funding of the DLR (German Aerospace Center).

Acknowledgements

The many persons who have contributed to the development of SUMO are gratefully acknowledged here.

References

[1] P. Alvarez Lopez et al., *Simulation of urban mobility (SUMO) (version 1.22.0)*, Feb. 2025. [Online]. Available: https://elib.dlr.de/212503/.

[2] D. Krajzewicz, J. Erdmann, J. Härri, and T. Spyropoulos, "Including pedestrian and bicycle traffic into the traffic simulation SUMO," in *10th ITS European Congress*, 2014. [Online]. Available: https://elib.dlr.de/90621/.

[3] J. Erdmann, "Lane-changing model in sumo," in *Proceedings of the SUMO2014 Modeling Mobility with Open Data*, ser. Reports of the DLR-Institute of Transportation SystemsProceedings, vol. 24, Deutsches Zentrum für Luft- und Raumfahrt e.V., Mai 2014, pp. 77–88. [Online]. Available: https://elib.dlr.de/89233/.

[4] J. Erdmann, "Sumo's lane-changing model," in *LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES*, M. Behrisch and M. Weber, Eds., ser. Lecture Notes in Control and Information Sciences, vol. 13, Springer Verlag, 2015, pp. 105–123. [Online]. Available: https://elib.dlr.de/102254/.

[5] D. Krajzewicz and J. Erdmann, "Road intersection model in sumo," in *1st SUMO User Conference - SUMO 2013*, ser. Reports of the DLR-Institute of Transportation Systems, vol. 21, DLR, Mai 2013, pp. 212–220. [Online]. Available: https://elib.dlr.de/84363/.

[6] J. Erdmann and D. Krajzewicz, "Sumo's road intersection model," in *Simulation of Urban Mobility*, ser. Lecture Notes in Computer Science VII, M. Behrisch, D. Krajzewicz, and M. Weber, Eds., vol. 8594, Springer, Nov. 2014, pp. 3–17. [Online]. Available: https://elib.dlr.de/93669/.

[7] S. Krauß, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E*, vol. 55, no. 5, pp. 5597–5602, 1997. DOI: 10.1103/PhysRevE.55.5597.

[8] S. Krauß, "Microscopic modelling of traffic flow: Investigation of collision free vehicle dynamics," Ph.D. dissertation, University of Cologne, 1998.

[9] P. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Part B*, vol. 15, no. 2, pp. 105–111, 1981, ISSN: 0191-2615. DOI: http://dx.doi.org/10.1016/0191-2615(86)90012-3. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0191261586900123.

[10] Transportation Research Board and National Academies of Sciences, Engineering, and Medicine, *Highway Capacity Manual 7th Edition: A Guide for Multimodal Mobility Analysis*. Washington, DC: The National Academies Press, 2022, ISBN: 978-0-309-08766-7. DOI: 10.17226/26432. [Online]. Available: https://nap.nationalacademies.org/catalog/26432/highway-capacity-manual-7th-edition-a-guide-for-multimodal-mobility.

[11] J. Erdmann, J. Rummel, and P. Wagner, "SUMO and the German Handbook for the Dimensioning of Highways (HBS)," in *Towards Simulation for Autonomous Mobility*, ser. Lecture Notes in Mobility, Springer, 2017. [Online]. Available: https://elib.dlr.de/112657/.

[12] L. Lücken, *Resolving collisions for the gipps car-following model*, 2019. arXiv: 1902.04927 [nlin.AO]. [Online]. Available: https://arxiv.org/abs/1902.04927.

[13] M. Treiber and V. Kanagaraj, "Comparing numerical integration schemes for time-continuous car-following models," *Physica A: Statistical Mechanics and its Applications*, vol. 419, pp. 183–195, 2015, ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2014.09.061. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437114008383.

[14] L. Lücken, E. Mintsis, N. P. Kallirroi, R. Alms, Y.-P. Flötteröd, and D. Koutras, "From automated to manual - modeling control transitions with sumo," in *SUMO User Conference 2019*, M. Behrisch, L. Bieker-Walz, R. Hilbrich, and M. Weber, Eds., ser. EPiC Series in Computing, vol. 62, EasyChair, Aug. 2019, pp. 124–144. [Online]. Available: https://elib.dlr.de/129416/.