



# TU Clausthal

## Bachelorarbeit

### Eine agentenbasierte Erweiterung des Nagel-Schreckenberg-Fahrzeugfolgemodells

Sven Albert-Pedersen

11. April 2018

Institut für Informatik

Technische Universität Clausthal

Erstgutachter:

Prof. Dr. Jörg P. Müller

Zweitgutachter:

Priv.-Doz. Dr. Umut Durak

Betreuer:

Dipl.-Inf. Philipp Kraus

Studiengang: Wirtschaftsinformatik

Matrikelnummer: 295798

---

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| <b>1 Einleitung</b>   | <b>1</b>  |
| 1.1 Gliederung der Arbeit . . . . .   | 1         |
| <b>2 State-of-the-Art</b>   | <b>2</b>  |
| 2.1 Zellulärautomaten . . . . .   | 3         |
| 2.2 Stochastische Ansätze . . . . .   | 6         |
| 2.3 Das Social-Force-Vehicle-Modell . . . . .                                   | 6         |
| <b>3 Die Simulationsumgebung und „LightJason“</b>                               | <b>9</b>  |
| 3.1 Multiagentensysteme . . . . .   | 9         |
| 3.1.1 Intelligente Agenten . . . . .  | 10        |
| 3.2 Das Konzept der Agentensprache „Agentspeak(L++)“ von „LightJason“ . . . . . | 11        |
| <b>4 Realisierung</b>   | <b>12</b> |
| 4.1 Entwicklung des Agentenverhaltens . . . . .                                 | 12        |
| 4.1.1 Zustandsautomat . . . . .   | 12        |
| 4.1.2 Agentenpläne . . . . .  | 14        |
| 4.2 Verfeinerung des Agentenverhaltens . . . . .                                | 15        |
| 4.2.1 Einstellmöglichkeiten in der Szenariokonfiguration . . . . .              | 16        |
| 4.2.2 Einstellmöglichkeiten im Agentenverhalten (Parameter) . . . . .           | 17        |
| 4.3 Anpassungen der Simulationssoftware . . . . .                               | 18        |
| 4.3.1 Schwierigkeiten bei der Simulation der Ringform der Strecke . . . . .     | 18        |
| 4.3.2 Unterteilung des Sichtfeldes der Fahrzeuge . . . . .                      | 22        |
| 4.4 Setups für die Langzeittests . . . . .                                      | 23        |
| 4.4.1 Fahrzeuganzahl . . . . .  | 23        |
| 4.4.2 Zellgröße und Zeitschrittlänge . . . . .                                  | 24        |
| 4.5 Testläufe des Einspurzenarios . . . . .                                     | 29        |
| 4.5.1 Verkehrsmenge A7: 34 Fahrzeuge . . . . .                                  | 30        |
| 4.5.2 Verkehrsmenge A38: 15 Fahrzeuge und Verkehrsmenge A71: 8 Fahrzeuge .      | 30        |
| 4.5.3 Beobachtungen . . . . .   | 30        |
| 4.6 Erkenntnisse aus Entwicklung und Simulationsläufen . . . . .                | 32        |
| 4.6.1 Ursache Zellgröße . . . . .   | 32        |
| 4.6.2 Ursache Zeitschrittlänge . . . . .  | 33        |
| <b>5 Überlegungen für die Multilane-Szenarios</b>                               | <b>34</b> |
| 5.1 Veränderungen für die Mehrspurigkeit . . . . .                              | 34        |
| 5.1.1 Zustandsautomat . . . . .   | 34        |
| 5.1.2 Agentenverhalten . . . . .  | 35        |
| 5.2 Testläufe des Mehrspurmodells . . . . .                                     | 36        |
| 5.3 Zusätzliche Erweiterungsmöglichkeiten . . . . .                             | 38        |
| 5.3.1 Zufälligkeit des Aus- und Einscherens . . . . .                           | 38        |
| 5.3.2 Dosiertes Abbremsen . . . . .   | 40        |
| 5.3.3 Vermeidung des Rechts-Überholens . . . . .                                | 40        |
| <b>6 Fazit</b>  | <b>41</b> |

---

|                             |   |           |
|-----------------------------|---|-----------|
| 6.1                         | Ausblick . . . . .                        | 41        |
| 6.2                         | Das NaSch-Modell in der Zukunft . . . . . | 43        |
| <b>Appendices</b>           |   | <b>44</b> |
| Literatur . . . . .         |   | 46        |
| Code-Listings . . . . .     |   | 47        |
| weitere Diagramme . . . . . |   | 55        |

## Abbildungsverzeichnis

|    |  |    |
|----|--|----|
| 1  | Überblick über die Entwicklung der Verkehrsflussmodelle . . . . .                            | 2  |
| 2  | Fundamentalrelationen in verschiedenen Ebenen . . . . .                                      | 3  |
| 3  | Beispiel für die Griddarstellung einer Straße . . . . .                                      | 3  |
| 4  | Darstellung des Verkehrs in einem geschlossenen System . . . . .                             | 5  |
| 5  | „Social Forces“ für Grid-Zellen . . . . .  | 7  |
| 6  | gewichtetes Roulette-Rad (fitness-proportionate selection) . . . . .                         | 8  |
| 7  | ein Agent in seiner Umwelt . . . . .   | 10 |
| 8  | Zustandsautomat für das Agentenverhalten . . . . .   | 13 |
| 9  | Beispielkurven eines Testdurchlaufes . . . . .   | 15 |
| 10 | Geschwindigkeitskurven bei $p_{linger} = 0,7$ . . . . .                                      | 18 |
| 11 | Beispiele für Geschwindigkeitskurven bei unterschiedlichen Werten für $p_{linger}$ . . . . . | 18 |
| 12 | Punktehäufung im negativen Bereich . . . . .   | 20 |
| 13 | Umbrechen des Sichtfelds eines Fahrzeugs am Ende der Lane . . . . .                          | 21 |
| 14 | Fehlerbehebung: Übergang vom Streckenende zum Anfang . . . . .                               | 21 |
| 15 | mögliche Unterteilung des Sichtfeldes . . . . .  | 22 |
| 16 | Simulationen mit Zellgröße 7,5 m und Zeitschrittänge 0,1 min . . . . .                       | 25 |
| 17 | Simulationen mit Zellgröße 7,5 m und Zeitschrittänge 0,05 min . . . . .                      | 26 |
| 18 | Simulationen mit Zellgröße 7,5 m und Zeitschrittänge 0,025 min . . . . .                     | 26 |
| 19 | Simulationen mit Zellgröße 5,0 m und Zeitschrittänge 0,05 min . . . . .                      | 27 |
| 20 | Simulationen mit Zellgröße 5,0 m und Zeitschrittänge 0,025 min . . . . .                     | 27 |
| 21 | Simulationen mit Veränderungen in den Agentenplänen . . . . .                                | 27 |
| 22 | Simulationen mit Zellgröße 2,5 m und Zeitschrittänge 0,05 min . . . . .                      | 28 |
| 23 | Simulationen mit Zellgröße 2,5 m und Zeitschrittänge 0,025 min . . . . .                     | 29 |
| 24 | Testlauf: 34 Fahrzeuge, 7,5 km Strecke, 2 Stunden . . . . .                                  | 31 |
| 25 | Zustandsautomat für das Spurverhalten . . . . .  | 34 |
| 26 | Überholen im Multilane-Szenario . . . . .  | 37 |
| 27 | Kurvenbeispiele der logistischen Funktion . . . . .  | 38 |
| 28 | Funktionskurve für die Ausscherwahrscheinlichkeit . . . . .                                  | 39 |
| 29 | Fahrzeugbewegung in Weg-Zeit-Linien aus Luftaufnahmen . . . . .                              | 41 |
| A1 | Positionsdiagramm mit rückwärts laufender Stauwelle . . . . .                                | 55 |
| A2 | weitere Simulationsläufe mit 34 Fahrzeugen, A7-Verkehrsmenge . . . . .                       | 56 |
| A3 | Geschwindigkeitskurven und Positionsdiagramm der erweiterten Testläufe . . . . .             | 57 |
| A4 | Diagramme, A38-Verkehrsmenge: 15 Fahrzeuge . . . . .   | 58 |
| A5 | Diagramme, A71-Verkehrsmenge: 8 Fahrzeuge . . . . .  | 59 |

## **Eidesstattliche Erklärung**

Ich versichere hiermit, die vorliegende Bachelorarbeit selbstständig und ausschließlich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, sind als solche gekennzeichnet. Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Clausthal-Zellerfeld, den 11. April 2018

---

(Sven Albert-Pedersen)

# **Erklärung zur Einsichtnahme**

„Hiermit erkläre ich mich damit einverstanden, dass meine Bachelorarbeit in der Instituts- und/oder der Universitätsbibliothek ausgelegt und zur Einsichtnahme aufbewahrt werden darf.“  
Diese Erklärung kann von mir jederzeit schriftlich beim Erstgutachter widerrufen werden.

Clausthal-Zellerfeld, den 11. April 2018

---

(Sven Albert-Pedersen)

# **Danksagung**

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Mein Dank gebührt Herrn Prof. Dr. Jörg Müller für die Bereitstellung des Themas, die Übernahme des Erstgutachtens und die Anregungen. Herrn Dr. Umut Durak danke ich für das Übernehmen des Zweitgutachtens.

Weiterhin vielen herzlichen Dank an meinen Betreuer, Herrn Dipl.-Inf. Philipp Kraus, für die hilfreichen Anregungen und die konstruktive Kritik bei der Erstellung dieser Arbeit.

Besonderer Dank gilt meiner Frau Marianne, meinen beiden Töchtern Francesca und Hannah, meinen Eltern Petra und Frank und meinen Großeltern Erika und Horst sowie Edith und Klaus für die Unterstützung während meiner Studienzeit.

Sven Albert-Pedersen  
Clausthal-Zellerfeld, am 11. April 2018

---

# 1 Einleitung

Mit dem Begriff Verkehr kann eine Vielzahl verschiedener Arten der Fortbewegung oder des Transports bezeichnet werden. Dazu dienen unterschiedliche Verkehrsmittel, mit denen sich an Land, auf dem Wasser oder in der Luft fortbewegt werden kann. An Land unterscheidet man wiederum in straßen-/wege- oder schienenbasierten Verkehr.

Um Straßen so auszulegen, dass sie auch noch prognostizierten Verkehr bewältigen können, bedarf es geeigneter Modelle und Simulationsmethoden. Bereits in der ersten Hälfte des vergangenen Jahrhunderts gab es Bestrebungen, Verkehrsmuster zu erkennen, erklären und nachzubilden.

Am Anfang der 1990er Jahre gelang es Kai Nagel und Michael Schreckenberg mit einfachen mathematischen Regeln das Verhalten eines einzelnen Fahrzeug(führer)s so abzubilden, dass die Interaktion mehrerer dieser Fahrzeuge ein realistisches Verkehrsbild zeigte. Das damals erstellte Modell trägt inzwischen den Namen seiner Entdecker, das „Nagel-Schreckenberg-Modell“. Im Jahre 1995 konnte damit z.B. das deutsche Autobahnnetz in Echtzeit simuliert werden und wird außerdem im Rahmen der Software TRANSIMS zur Simulation des Straßenverkehrs der Schweiz genutzt (siehe [Spa02, S. 33]).

Das Modell arbeitet allerdings mit gleichartigen Verkehrsteilnehmern. Dies stellt sich natürlich in der Realität vollkommen anders dar. Vom „Schleicher auf der linken Spur“ bis hin zum „Verkehrsrowdy“ gibt es eine Vielzahl von Autofahrertypen und von der „Ente“ bis zum Porsche die unterschiedlichsten Fahrzeuge.

In Multiagentensystemen ist diese Gleichartigkeit nicht zwingend erforderlich [Rit15, S. 2]. Anstatt homogenen Verkehrs kann eine große Vielfalt unterschiedlicher Verkehrssubjekte (und -objekte) dargestellt werden und diese können miteinander agieren.

Mit dieser Arbeit soll gezeigt werden, ob sich das Multiagentenframework „LightJason“ mit „AgentSpeak(L++)“ dazu eignet, das Grundkonzept des Nagel-Schreckenberg-Modells für die Steuerung von Agenten zu übernehmen und Verkehrsverhalten unter Beachtung realer Geschwindigkeits-, Abstands- und Beschleunigungswerte nachzubilden.

## 1.1 Gliederung der Arbeit

Diese Bachelorarbeit besteht aus insgesamt sechs Kapiteln. Das Kapitel 2 enthält einige Arbeiten, die Historie und Stand der Technik auf dem behandelten Gebiet wiederspiegeln. Kapitel 3 beschäftigt sich mit der Thematik Multiagentensysteme und stellt das Agenten-Framework vor. Dem Entwicklungsprozess, der das Verhalten des Nagel-Schreckenberg-Modells (NaSch-Modell) als Agentenversion als Ergebnis hat, sowie dessen Test widmet sich Kapitel 4. Im darauf folgenden Kapitel werden Erweiterungsideen für eine Mehrspurversion gegeben. Das letzte Kapitel beinhaltet Fazit und Ausblick.

## 2 State-of-the-Art

Den Straßenverkehr theoretisch abzubilden, ist fachübergreifend eine große Herausforderung in der Forschung. Hier ergeben sich u. a. Aufgaben im Bereich der Mathematik (Routenplanung z.B. durch graphentheoretische Ansätze), der Psychologie (Erforschung menschlichen Verhaltens) und nicht zuletzt der Informatik (Simulationen).

Die Simulation von Fahrverhalten ist aufgrund des unsicheren Faktors Mensch, der als Hauptentscheider bisher nur durch Wahlmöglichkeiten mit vorgegebenen unterschiedlich hohen Wahrscheinlichkeiten modelliert werden kann, ein Bereich in dem seit den 1930er Jahren geforscht wird.

In [vWKvLVH15] wird ein umfassender Überblick über die verschiedenen Ansätze gegeben, siehe Abbildung 1, die seit den ursprünglichen Annahmen der „fundamental relation“ von Bruce D. Greenshields (1934/35) verfolgt wurden. Diese besagte, dass es einen Zusammenhang zwischen dem Abstand  $s$  zwischen Fahrzeugen (genauer zweier Fahrzeugfronten) und deren Geschwindigkeit  $v$  geben muss.

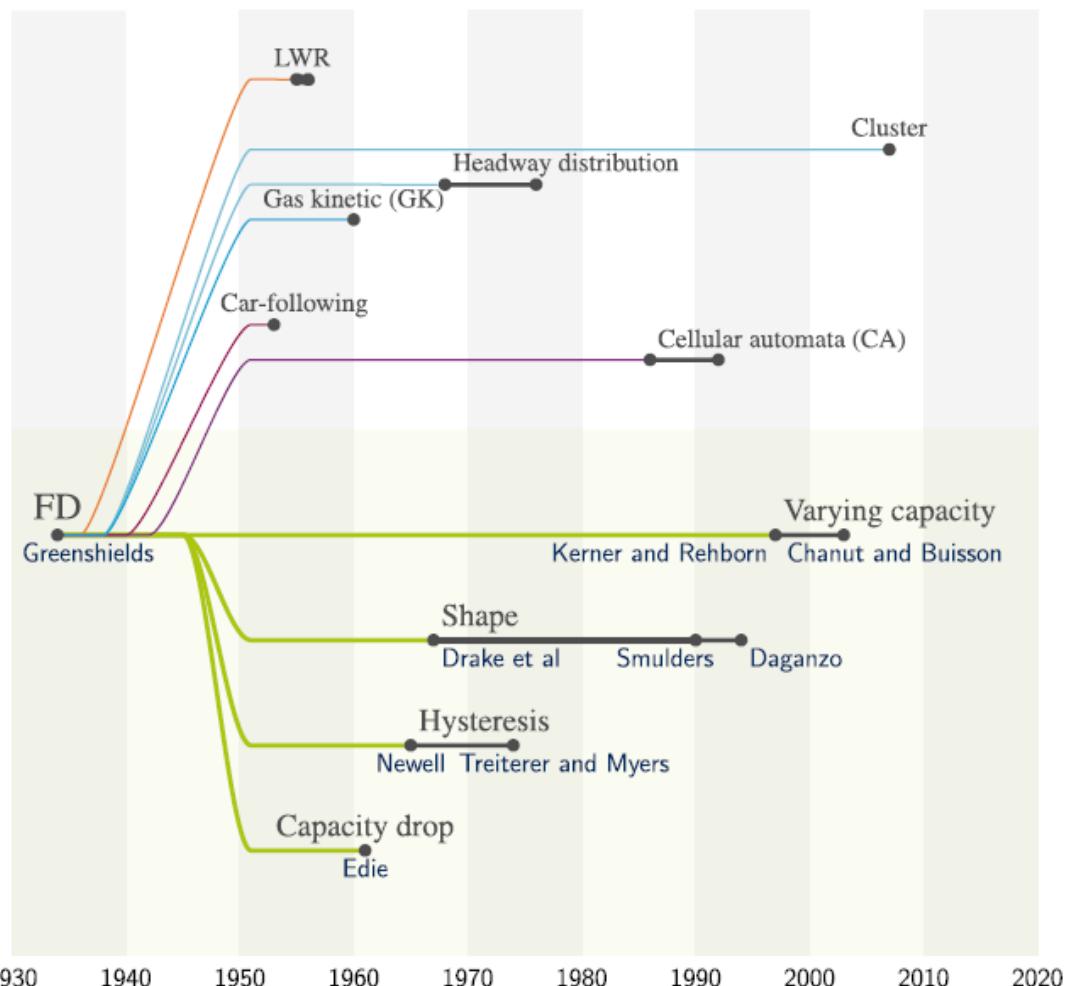


Abbildung 1: „Stammbaum“ der Verkehrsflussmodelle, aus [vWKvLVH15, Fig. 2]

Die Fundamentalrelationen, heute bekannt als Ausprägungen des Fundamentaldiagramms, kann aber auch mit Hilfe anderer Variablen, wie z.B. der Verkehrsdichte  $\rho$ , der durchschnittlichen Anzahl von Fahrzeugen auf einer bestimmten Strecke, und dem Verkehrsfluss  $q$ , der durchschnittlichen Anzahl von Fahrzeugen pro Zeiteinheit, wiedergegeben werden, siehe Abbildung 2.

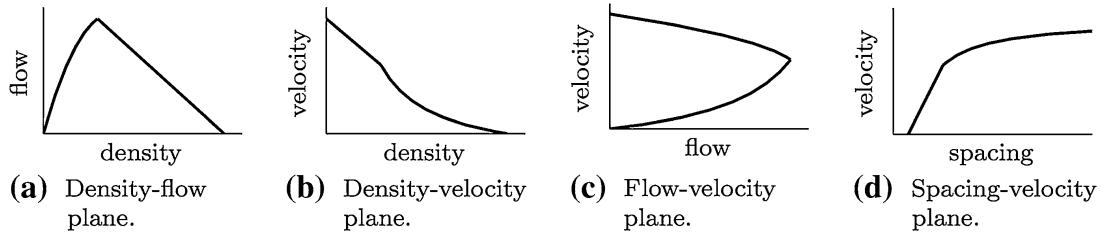


Abbildung 2: Fundamentalrelationen in verschiedenen Ebenen, aus [vWKvLVH15, Fig. 3]

Verschiedene Ansätze, Verkehrsflüsse zu simulieren, die auch unterschiedliche Sichtweisen auf Verkehr und Verkehrsteilnehmer ermöglichen, wurden und werden verfolgt, vgl. [Din11].

- *Mikroskopische Modelle* simulieren einzelne Fahrzeug-Fahrer-Einheiten, basierend auf dem Fahrerverhalten. Zwei Modellierungsansätze können hier unterschieden werden: das Fahrzeugfolgemodell („car-following model“) und das Zellularautomatenmodell. Zu letzterem mehr in Kapitel 2.1.
- *Makroskopische Modelle* konzentrieren sich auf Charakteristiken des Verkehrsflusses, wie mittlere Geschwindigkeit („average velocity“), Verkehrsdichte, -fluss und Durchschnittsgeschwindigkeit des Verkehrsstromes („mean speed of a traffic stream“).
- *Mesoskopische Modelle* kombinieren die beiden vorgenannten Modelle, simulieren die Fahrzeuge einzeln, nutzen aber die makroskopische Sicht, um deren Aktivitäten und Interaktionen zu beschreiben. Ein klassischer Vertreter ist das gaskinetische Modell.

## 2.1 Zellularautomaten

Zellularautomaten (oder zelluläre Automaten) werden seit etwa 25–30 Jahren für die Verkehrssimulation verwendet.

Ein zellulärer Automat ist eine regelmäßige Annordnung von Zellen. Jede Zelle kann eine endliche Zahl von Werten / Zuständen annehmen und hat eine begrenzte Zahl von Nachbarzellen, die sie beeinflussen können. Das Muster des gesamten zellulären Automaten ändert sich in einzelnen Schritten, die durch eine Reihe von Übergangsregeln bestimmt werden, die für alle Zellen gelten. (aus [Bec03])

Vereinfacht kann eine Fahrspur einer Straße als Aneinanderreihung vieler Zellen gesehen werden (siehe Abbildung 3) und damit durch den Zellularautomaten abgebildet werden. Dies führt schließlich zu einer gridbasierten Simulationsumgebung.

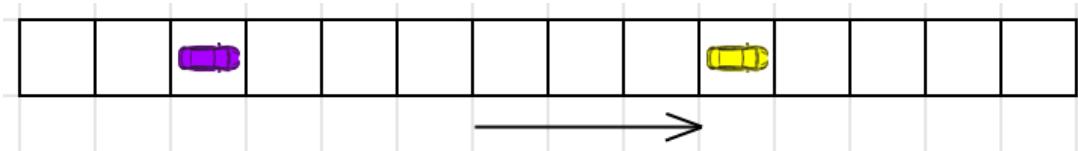


Abbildung 3: symbolische Griddarstellung einer Straße mit einer Fahrspur mit Fahrzeugen, nach [WNW96]

Quelle Auto-Silhouette: vecteezy.com

## Das Nagel-Schreckenberg-Modell:

Nagel und Schreckenberg war es 1992 gelungen, auf Basis solcher Automaten mit einfachen Regeln das mikroskopische Verhalten jedes Fahrzeugführers so abzubilden, dass sich die makroskopische Sicht auf den Verkehrsfluss als für den Autobahnverkehr realistisch darstellte. Jedes Fahrzeug durchläuft in jedem Zeitschritt die folgenden Übergangsregeln, vgl. [NS92].

1. *Beschleunigung*: Solange ein Fahrzeug nicht seine max. Geschwindigkeit  $v_{max}$  (Zellen/Zeitschritt) erreicht hat und ein vorausfahrendes Fahrzeug weit genug entfernt ist, erhöht es die Geschwindigkeit um 1,  $v \rightarrow v + 1$ ;
2. *Abbremsen*: Hat ein Fahrzeug ein anderes Fahrzeug  $j$  Zellen vor sich, dann reduziert es die Geschwindigkeit auf  $j - 1$ ,  $v \rightarrow j - 1$ ;
3. *Zufallsgröße*, auch „Trödelwahrscheinlichkeit“: Mit einer Wahrscheinlichkeit  $p$  wird die Geschwindigkeit ( $v > 0$ ) eines Fahrzeugs um 1 reduziert,  $v \rightarrow v - 1$ ;
4. *Fahrzeugbewegung*: Jedes Fahrzeug wird um  $v$  Zellen nach vorn gesetzt.

Erstmals konnte das Entstehen von „Stau aus dem Nichts“ und das Vorhandensein von „Stauwellen“, die sich rückwärts durch einen solchen Stau bewegen, simulatorisch dargestellt werden. Die Rechenumgebung wurde als eindimensionales Array mit  $L$  Zellen definiert,  $L$  ist dabei eine ganze Zahl. Jede der Zellen kann entweder von einem Fahrzeug belegt oder frei sein. Eine Simulation in dieser Grid-Welt hat den Vorteil, dass die Erkenntnisse auf die reale Welt skaliert werden können. Es wurde eine Zelllänge von 7,5 m angenommen (vgl. [NS92, S. 2227]), was ungefähr dem beanspruchten Platz eines Pkw - von Fahrzeugfront zu Fahrzeugfront - in einer Stausituation entspricht (Fahrzeulgänge + Abstand). Jedes Fahrzeug hat eine ganzzahlige Geschwindigkeit zwischen null und  $v_{max}$ .

| $v^{sim}$ in $\frac{\text{Zellen}}{\text{Zeitschritt}}$ | $\cong v^{real}$ in $\frac{\text{m}}{\text{s}}$ | = in $\frac{\text{km}}{\text{h}}$ |
|---|---|-----------------------------------|
| 0   | 0   | 0                                 |
| 1   | 7,5   | 27                                |
| 2   | 15  | 54                                |
| 3   | 22,5  | 81                                |
| 4   | 30  | 108                               |
| 5   | 37,5  | 135                               |
| 6   | 45  | 162                               |

Tabelle 1: Umrechnung der Geschwindigkeiten: Gridwelt → reale Welt

Die Simulation erfolgte unter Nutzung periodischer Randbedingungen. Während bei offenen Randbedingungen an den Rändern ständig Fahrzeuge in das System strömen und es verlassen, bezeichnen periodische Randbedingungen eine zu einem Ring geschlossene Strecke, vgl. [Spa02, S. 6]. Die Fahrzeuge werden somit vom ‚Ende‘ der Gridstrecke wieder an den ‚Anfang‘ eingesetzt. Damit entstand die Möglichkeit einer kontinuierlich arbeitenden Simulationsumgebung, siehe Abbildung 4.

Zu Beginn eines Simulationslaufes wurden  $N$  Fahrzeuge zufällig auf die  $L$  Zellen verteilt. Dies führte dazu, dass über den Durchlauf hinweg im System eine konstante Fahrzeudichte  $\rho$  garantiert werden konnte:

$$\rho = \frac{N}{L} = \frac{\text{Anzahl der Fahrzeuge im System}}{\text{Anzahl der Zellen des Systems}}$$

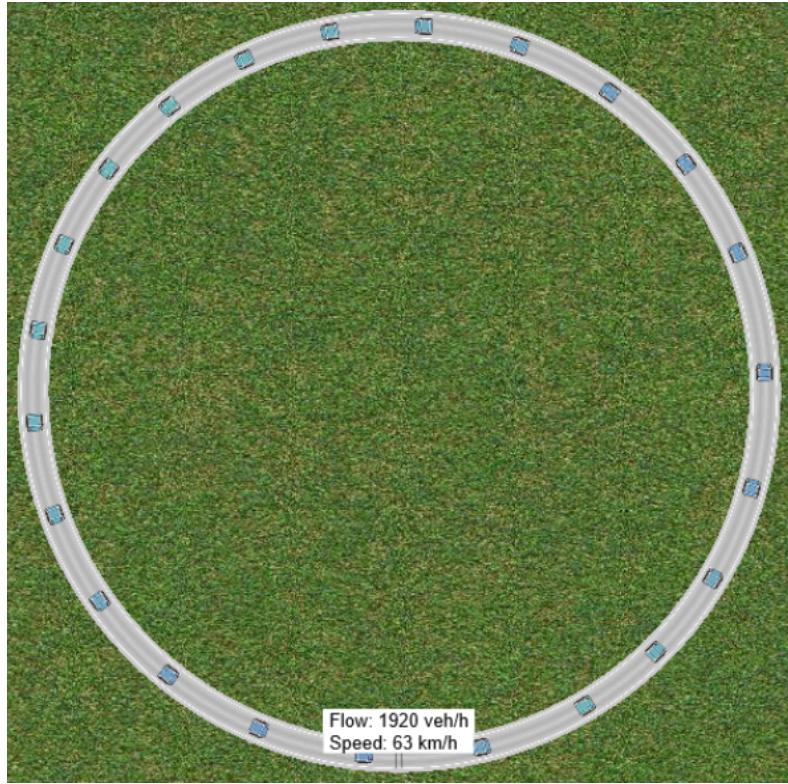


Abbildung 4: Symbolbild: Verkehr in einem Kreis (geschlossenes System)  
Screenshot: <http://www.traffic-simulation.de>

### Erweiterung auf Mehrspurigkeit:

Das Originalmodell war nicht geeignet, um Verkehr auf mehrspurigen Fahrbahnen zu simulieren. 1996 folgte deswegen eine Ausdehnung auf Mehrspurigkeit [WNW96] - zwei Spuren. Regeln, die prüfen, ob ein Spurwechsel vorteilhaft ist und einen sicheren, weil kollisionsfreien, und nicht behindernden Spurwechsel ermöglichen, wurden hinzugefügt. Die Regeln 1–3 wurden übernommen. Nach Simulationsläufen, die unerwünschtes Verhalten zeigten, lauteten die finalen Regeln wie folgt:

*Hinweis:*  $\Delta x \hat{=} \text{dem Abstand zum vorausfahrenden Fahrzeug, die Indizes } b \text{ („behind“ - für rückwärtigen Verkehr) und } o \text{ („other“ - für Verkehr auf der Spur auf die gewechselt werden soll)}$  können auch kombiniert werden.

- *Sicherheitsbedingung:*  
die Geschwindigkeit des Fahrzeugs auf der Spur auf die gewechselt werden soll, darf nicht größer als der Abstand zu jenen Fahrzeug sein (keine Behinderung des Nachfolgeverkehrs),  
 $v^{o,b} \leq \Delta x^{o,b} - 1$ ;
- *Intention zum Spurwechsel:*  
Fahrzeug kann nicht so schnell fahren, wie es möchte,  $v_{max} > \Delta x - 1$   
UND  
die andere Spur ist nicht schlechter als die aktuelle Spur,  $\Delta x^o \geq \Delta x$
- *nach dem Spurwechsel:*  
(diese Regeln werden mit einer gewissen Wahrscheinlichkeit  $p_{l2r}$  („Zurückwechselwahrscheinlichkeit“) ausgeführt)  
ausreichend großer Abstand zum hinteren Fahrzeug auf der rechten Spur,  $v_{max}^{o,b} \leq \Delta x^{o,b} - 1$   
UND  
ausreichend Platz nach vorn auf der rechten Spur,  $v \leq \Delta x^o - 1$

---

Wenn es die festgelegten Kriterien zulassen, erfolgt der Spurwechsel auf die linke Spur immer, da dieser nicht an eine Wahrscheinlichkeit geknüpft ist. Lediglich um die Überbevölkerung der Überholspur zu vermeiden, wurde eine Wahrscheinlichkeit  $p_{l2r}$  für das Zurückwechseln eingeführt.

Was dieser Modellierung fehlt, ist die Berücksichtigung irregulärer Verhaltensweisen der Fahrer - Unsicherheiten (z.B. Fehleinschätzungen oder Ängste), Ablenkungen (z.B. Gespräche mit anderen Fahrzeuginsassen, Telefonieren, Suche im Handschuhfach oder nach der fallen gelassenen Zigarette).

Zusätzlich ist die Bremswirkung, wie auch im ursprünglichen Nagel-Schreckenberg-Modell, „überdimensioniert“. Vollbremsungen aus Höchstgeschwindigkeit sind innerhalb kürzester Distanzen und Zeiträume möglich [HM08]. Daraus ergibt sich eine weitere fehlende Komponente der Realität. Denn auch Unfälle gehören zum Verkehrsgeschehen.

Studien zufolge sind drei Viertel der Autofahrer abgelenkt, wenn sie am Steuer sitzen. Jeder zehnte Verkehrsunfall in Deutschland wird durch unaufmerksame Autofahrer verursacht. In Österreich und der Schweiz geht man, aufgrund der Einordnung in eine eigene Kategorie für diese Art Unfälle, von einer Rate von etwa 30% der Unfälle mit Personenschäden oder gar Toten aus. (vgl. [Deu])

Laut Statistischem Bundesamt gab es 2016 auf Autobahnen 21.193 Unfälle, davon 1.435 innerhalb von Baustellen, siehe [Sta17b, S. 94]. Bei den etwa 13000 km Autobahn ([Sta17a, S. 607]) entspricht dies durchschnittlich etwa einem Unfall pro 200 km Strecke pro Tag. In Wirklichkeit wird es aber Unfallschwerpunkte geben, die diesen Durchschnittswert überschreiten, sodass an anderen Stellen unterdurchschnittlich viele Unfälle passieren.

## 2.2 Stochastische Ansätze

Es existieren Ansätze, die das mikroskopische Verhalten der Fahrzeugführer aus Realdaten stochastisch modelliert haben, z.B. [KWT16]. Dort wurde in den aggregierten Daten eine Laplace-Verteilung der Beschleunigungswerte erkannt. Ein daraus entwickeltes Modell wurde getestet. Die Reproduktion von Sicherheitsparametern, wie ‚Zeit bis zur Kollision‘ (TTC, „time to collision“) und der ‚Bremsrate um einen Unfall zu verhindern‘ (DRAC, „deceleration rate to avoid crash“) gelang. Allerdings konnte das Fundamentaldiagramm nicht nachgestellt werden.

Von [YCLR12] wird der Ansatz der Mehrspurigkeit in einem Multiagenten-Fuzzy-System integriert. Die Spuren werden als Vereinigung mehrerer miteinander kommunizierender einspuriger Zellularautomaten („continuous cellular automata“) beschrieben. Diese Kommunikation zwischen den Spuren beschränkt sich auf die Sicherheitskriterien, wenn ein Fahrzeug die Spur wechseln möchte. Intentionen zum Wechsel zwischen den Spuren werden durch fortwährend aktualisierte „Stresslevel“ je Fahrzeug beeinflusst, die Wahrscheinlichkeiten dafür durch einen Bernoulli-Prozess berechnet. Dieser Ansatz arbeitet mit Gridzellen und achtet auf Kollisionsfreiheit.

## 2.3 Das Social-Force-Vehicle-Modell

In [Tra17] wurde mit dem „Social-Force-Vehicle-Modell“ ein neuer Vorschlag für die Simulation von Spurwechselvorgängen entwickelt. Das Modell orientiert sich bei den Fahrzeubewegungen an den Grundlagen des Nagel-Schreckenberg-Modells, vermeidet Unfälle aber in erster Linie nicht durch Abbremsen, sondern durch Ausweichen, also Spurwechsel.

Im Gegensatz zur ursprünglichen erdachten Nutzung der „Social Forces“ für Fußgänger, wie z.B. in [HM98] beschrieben, wird dieses System auf die Gridzellen angewandt. Hier wirken die Kräfte als eine Art „Anziehungskraft“ und liefern bei Auswahl einen Impuls für die Beibehaltung oder Änderung der Fahrspur. Von den ein Fahrzeug umgebenden acht Zellen müssen, aufgrund der Bewegungsrichtung, nur die drei betrachtet werden, die sich in Fahrtrichtung befinden.

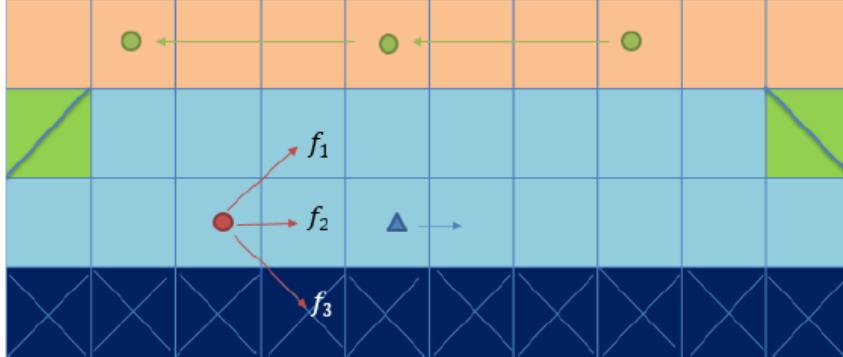


Abbildung 5: Vorschlag der „Social Forces“ für Grid-Zellen (Fahrzeug auf Normalspur), aus [Tra17, Abb. 22]

Da die Position eines Fahrzeugs zum Zeitpunkt  $t + 1$  von der Position im vorhergehenden Zeitschritt abhängt, werden die Kräfte der Zellen, siehe Abbildung 5, für jedes Fahrzeug zu jedem Zeitschritt  $t$  über verschiedene Funktionen berechnet.

Die letztendliche Auswahl der Zielzelle für einen „Impuls zur Spurwahl“ wird mittels „Fitness-proportionate Selection“ durchgeführt.

**„Fitness-proportionate Selection“ oder „Roulette-wheel sampling“:** Das Konzept der „Fitness-proportionate Selection“ kommt aus dem Bereich der genetischen Algorithmen. [Gol89] beschreibt es in einem Beispiel anhand einer Population von  $n = 4$  Strings, die man repräsentativ auch als Individuen oder Wahlmöglichkeiten ansehen kann. Jedem dieser Strings ist ein Funktionswert  $f$  (Fitnesswert), ein Maß für Profit, Nutzwert oder Güte zugeordnet, den es zu maximieren gilt. Strings mit einem höheren Fitnesswert haben eine höhere Wahrscheinlichkeit, einen oder mehrere Nachkommen für die Folgegeneration beizutragen. Demnach ist dies eine künstliche Version des Darwin'schen „Survival of the Fittest“.

| No.    | Fitnesswert $f$ | % von Gesamt |
|--------|-----------------|--------------|
| 1      | 169             | 14,4         |
| 2      | 576             | 49,2         |
| 3      | 64              | 5,5          |
| 4      | 361             | 30,9         |
| Gesamt | 1170            | 100,0        |

Tabelle 2: Beispiel nach [Gol89, Table 1.1]

Ein einfacher Weg *Fitness-proportionate Selection* zu veranschaulichen ist *Roulette-wheel sampling*. Dabei wird jeder Option ein Bereich (Slot) auf einem Rouletterad zugewiesen, welcher von der Größe her deren Fitnesswert entspricht, siehe Abbildung 6.

Benötigt man z.B. auch in der Nachfolgegeneration vier Strings, würde das Rouletterad viermal gedreht. Das Individuum in dessen Slot die Kugel liegen bleibt, würde ausgewählt.

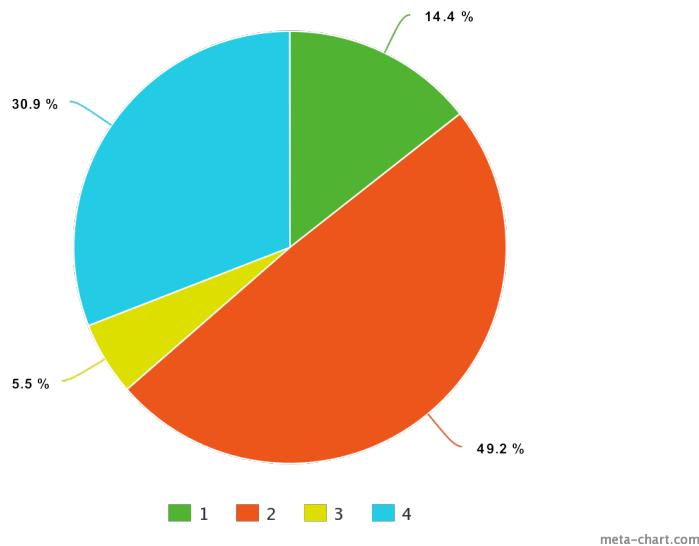


Abbildung 6: schematische Darstellung eines gewichteten Rouletterades für das Beispiel in Tabelle 2

Aufgrund der Größe der Slots von String 2 und String 3 wäre es z.B. möglich, dass String 2 für die Kindsgeneration zweimal und String 3 hingegen gar nicht ausgewählt wird.

#### **Verwendung im Social-Force-Vehicle-Modell:**

Im Social-Force-Vehicle-Modell gibt es zwei Möglichkeiten, aus denen gewählt werden kann. Zum einen jeweils das Geradeausfahren, zum anderen die Möglichkeit aus- respektive einzuscheren. Für die Wahl der Alternative ist nur eine Entscheidung zwischen den jeweils zwei Möglichkeiten nötig.

Entgegen des Zwangs zum Ausscheren oder einer vorgegebenen Wahrscheinlichkeit für das Einscheren in der mehrspurigen Version des Nagel-Schreckenberg-Modells wird eine stochastische Simulationsmöglichkeit der Neigung zum Spurwechsel aufgezeigt. Es entsteht eine Ungewissheit, ob die wahrscheinlichste Alternative ausgewählt wird oder ob die andere, weniger wahrscheinliche, das Verhalten bestimmen wird.

#### **Möglichkeit von Kollisionen im Social-Force-Vehicle-Modell:**

Da Kollisionen im Modell durch Ausweichen verhindert werden sollen, aber die „Bremskomponente“ aus dem Ursprungsmodell entfernt wurde, sind in bestimmten Situationen Kollisionen (Auffahrunfälle) denkbar.

---

### 3 Die Simulationsumgebung und „LightJason“

Als Simulationsumgebung für die geplanten Tests der Modelle wurde eine Software verwendet, die Elemente einer Java-Software nutzt, welche als „Traffic Simulation Game“ im September 2017 für einen Workshop<sup>1</sup> mit Doktoranden entwickelt wurde. Die Steuerung des Agentenverhaltens wird dabei von LightJason, vorgestellt in [AKM16], übernommen.

Es handelt sich bei LightJason um ein BDI-Multi-Agenten-Framework zum Erstellen von Multiagentensystemen mit Java, welches nebenläufige Ausführung unterstützt. Dabei wurde versucht einen Rahmen zu schaffen, der es ermöglicht, AI-Algorithmen zu einer bestehenden Softwarearchitektur hinzuzufügen. Das Framework kombiniert klassische Methoden der Künstlichen Intelligenz mit Optimierungs- und Fuzzy-Logik-Konzepten. (nach [ADK])

LightJason beschreibt einen BDI-Agenten als parallel ausgeführten regelbasierten Zustandsautomaten. Somit eignet sich das grundlegende Ausführungskonzept für den Einsatz im Rahmen eines zellulären Automaten (siehe Kapitel 2.1).

#### 3.1 Multiagentensysteme (nach [Woo09])

Wann immer in der Vergangenheit verteilte Systeme entwickelt wurden, stellte genau diese Verteilung ein Problem dar. Wenn ein Computersystem, das in unserem Auftrag handelt, mit einem anderen interagiert, das die Interessen einer anderen Person vertritt, dann ist es ziemlich sicher, dass diese Interessen nicht die gleichen sind. Es wurde notwendig, die Systeme mit der Fähigkeit auszustatten, mit anderen Systemen zu kooperieren und mit ihnen Übereinkünfte zu erzielen - mehr oder minder in der gleichen Weise wie wir Menschen dies im täglichen Leben tun.

Die Idee eines Multiagentensystems ist dabei recht einfach. Ein Agent ist ein Computersystem, welches in der Lage ist, selbstständig im Interesse seiner Benutzer/Eigner zu handeln. Der Agent kann eigenständig herausfinden, welche Handlungen nötig sind, um seine Designziele zu erreichen, ohne dass ihm immer wieder vorgegeben werden müsste, was in welchem Moment zu tun ist.

Der Agent ist die aktuell letzte Stufe der Entwicklung in der Reihe der Programmierkonzepte - von Maschinensprache und Assemblercode bis hin zu abstrakten Datentypen und Objekten.

Ein Multiagentensystem besteht in der Regel aus einer Anzahl von Agenten, deren Kommunikation über eine Computernetzwerkinfrastruktur abläuft. Die Zielsetzung und Motivation der einzelnen Agenten kann grundverschieden sein, da jeder Agent sein Ziel bzw. das seines Benutzers verfolgt. Für eine Interaktion müssen die Agenten die Fähigkeit haben, zu kooperieren, zu koordinieren und zu verhandeln.

Jeder Agent befindet sich dabei in einer Umwelt, die er mit verschiedenen geeigneten Sensoren wahrnehmen kann, siehe Abbildung 7. Dies können bei Agenten in einer realen Umwelt physische Sensoren oder bei Softwareagenten Softwaresensoren sein.

Aufgrund dieser Umweltwahrnehmung trifft er Entscheidungen und handelt entsprechend. Durch seine Handlungen (und natürlich auch die von anderen Agenten in der gleichen Umwelt) kann diese Umwelt verändert werden. Dies kann auch während der Entscheidungsfindung des Agenten passieren, so dass die getroffene Entscheidung nicht mehr optimal zum dann aktuellen Zustand der Umwelt ist.

---

<sup>1</sup>siehe <https://lightjason.github.io/news/2017-09-workshop/>

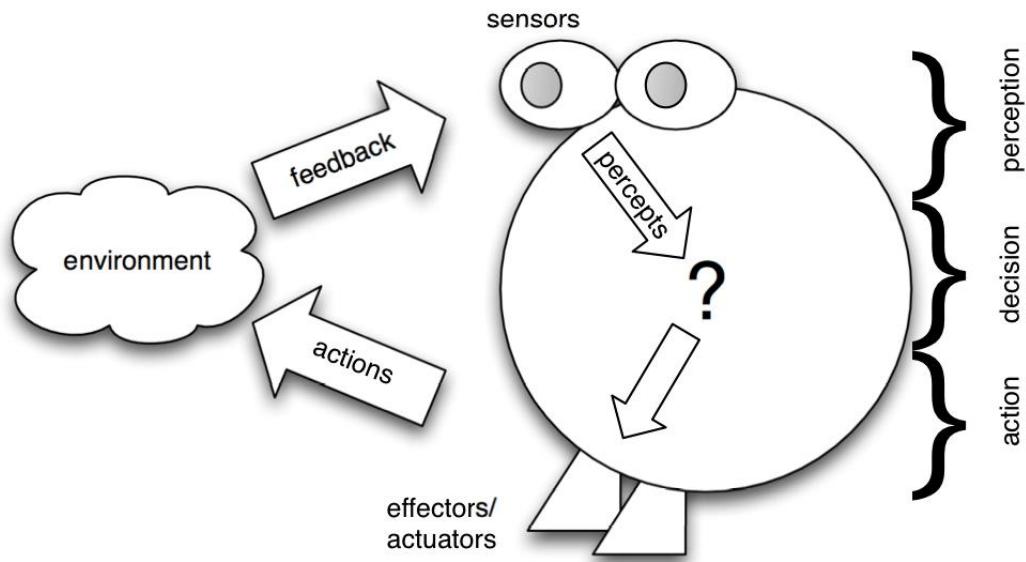


Abbildung 7: ein Agent in seiner Umwelt, aus [Woo09, Figure 2.1]

Ein einfaches System, welches außerhalb der Digitalität als Agent angesehen werden kann, ist z.B. ein Heizungsthermostat - [RN10] bezeichnet dies mit dem Begriff *reaktiver Agent*. An diesem lassen sich leicht verständlich die Begrifflichkeiten erklären.

Die Umwelt des Thermostats ist der Raum, in dem sich die Heizung befindet. Der Thermostat hat einen Sensor, welcher die Raumtemperatur misst. Aufgrund dieser Wahrnehmung entscheidet er, ob das Heizungsventil geöffnet oder geschlossen (gehalten) wird, wobei das Öffnen des Ventils gewöhnlich eine Erwärmung des Raumes zur Folge hat.

Dieser Effekt kann aber nicht garantiert werden. Ist etwa eine Tür oder ein Fenster geöffnet, kann eine Erwärmung des Raumes ausbleiben.

### 3.1.1 Intelligente Agenten

Ein intelligenter Agent zeigt in der Regel drei Arten von Verhalten: Reaktivität, Proaktivität und soziale Fähigkeit.

**Reaktivität:** Da sich die Umwelt des Agenten mit großer Wahrscheinlichkeit ständig verändert, muss er in der Lage sein, diese Änderungen aufzunehmen und in angemessener Zeit darauf zu reagieren, um seine vorgegebenen Ziele zu erreichen.

**Proaktivität:** Das Verhalten des Agenten ist nicht auf pure Reaktion beschränkt, sondern soll auch eigenes zielführendes Verhalten zeigen. Er soll die Initiative übernehmen, um sein Ziel zu erreichen.

**soziale Fähigkeit:** Da einige Ziele nicht allein zu erreichen sind, muss der Agent die Fähigkeit haben, mit anderen Agenten (evtl. Menschen) zu interagieren, um sein gesetztes Ziel zu erreichen.

---

### 3.2 Das Konzept der Agentensprache „Agentspeak(L++)“ von „LightJason“ (nach [AKM16] und [ADK])

Für die Entwickler von LightJason war es wichtig, dass die Agentensprache auch von Nicht-Informatikern verwendet werden kann. Auf der einen Seite brauchte man eine Syntax um Verhalten zu definieren, auf der anderen Seite musste diese Syntax leicht verständlich sein. Die Syntax von „Agentspeak(L++)“ (ASL) wurde als logische Programmiersprache designt, wobei alle Elemente auf Terme und Literale reduziert werden konnten.

Dies erlaubte das Modellieren eines Multiagentensystems, welches es ermöglicht, bei der Agentenprogrammierung deren Verhalten durch Umwelteinindrücke (beliefs), Regeln (rules), Pläne (plans) und Aktionen (actions) zu beschreiben.

**belief:** Beliefs (oder auch Fakten) sind die Eindrücke des Agenten von seiner Umwelt. Aufgrund dieser Daten trifft der Agent seine Entscheidungen.

Beliefs können in der Definition des Agentenverhaltens als Standardwerte vorgegeben, von Sensoren erfasst oder vom Agenten selbst aus eigenen Daten oder den Daten anderer Agenten kombiniert werden.

Ein Belief muss nicht faktisch richtig sein. Er kann also der Realität der Umwelt widersprechen. Ursache hierfür können defekte Sensoren sein oder die aufgenommenen Informationen, die z.B. andere Agenten bereitstellen, sind fehlerhaft.

Ein einfaches Beispiel für einen Belief im Verkehrskontext wäre die wahrgenommene Lichtfarbe einer Ampel - „`light(red)`“.

**plan:** Pläne sind ähnlich statischen Methoden oder Funktionen in imperativen Programmiersprachen mit einer Ausführungsbedingung und einem boolean Rückgabewert.

Der Aufruf von Plänen erfolgt über die Festlegung von Zielen (goals). So kann etwa aus dem Plan „`+!main`“ über das Ziel „`!nextplan`“ im nächsten Durchlauf der Plan „`+!nextplan`“ aufgerufen werden.

Innerhalb eines Planes kann es durch Auswertung von Beliefs (ähnlich IF-Bedingungen) zu unterschiedlichen Resultaten beim Durchlauf ein und desselben Planes kommen. Diesen unterschiedlichen Ausgängen könnten z.B. unterschiedliche Ziele und damit die Ausführung unterschiedlicher Pläne zugewiesen werden.

Für den Fall, dass die Ausführung eines Planes fehlschlägt, kann mit dem Trigger „`-!nextplan`“ ein Alternativverhalten definiert werden.

Für ausführliche Erklärungen zur Syntax, Beispiele usw. sei an dieser Stelle auf [ADK] verwiesen.

---

## 4 Realisierung

Aufgrund enger zeitlicher Vorgaben wurde die Simulationssoftware nicht selbst entwickelt. Eine komplette Neuentwicklung wäre wesentlich aufwändiger gewesen, als auf ein bestehendes System aufzusetzen. In einer vom Betreuer dieser Arbeit für einen Workshop mit Doktoranden<sup>2</sup> entwickelten Software („Traffic Simulation Game“) waren einige der gewünschten Funktionalitäten für die Simulation im Verkehrsumfeld bereits vorhanden.

In diesem Workshop mussten die Teilnehmer mit Hilfe der Agentensprache ein Fahrzeug steuern. Dieses sollte ein 50 km langes Straßenstück mit unterschiedlichen zufälligen Geschwindigkeitsbegrenzungen absolvieren. Ziel war es, wenige Verkehrsverstöße zu begehen und dennoch möglichst schnell zu sein.

Aufgrund von Spezifikationsvorgaben für die Simulationsumgebung wurde vom ursprünglichen Entwickler eine Software realisiert, die einen Teil der Architektur des Workshops übernehmen konnte, Teile, wie z.B. das User-Interface, ausschloss und neue Komponenten wie z.B. die Kreisform der Strecke implementierte.

Hauptaugenmerk lag auf der Herstellung des Fahrzeugverhaltens. So wurden u.a. die Zustandsautomaten in Kapitel 4.1.1 und 5.1.1 entwickelt und die Feinabstimmung der Szenariodatei (Listing A1) vorgenommen. Außerdem wurde das Agentenverhalten in den Agentenplänen (Listings A2 und A3) definiert.

Während der Tests der ersten Agentenpläne mussten Korrekturen an der Simulationssoftware vorgenommen werden, da eine fehlerfreie Simulation sonst nicht möglich gewesen wäre. Mehr dazu nachfolgend in Kapitel 4.3. Fehlersuche und kleinere Änderungen wurden dabei selbst durchgeführt.

### 4.1 Entwicklung des Agentenverhaltens

Jeder Agent in den Simulationen, die dieser Arbeit zugrunde liegen, stellt ein individuelles Fahrzeug dar. Die mathematischen Gleichungen in [NS92] beschreiben gut, welches Verhalten dem Agenten zu geben ist:

- wenn möglich beschleunigen
- zufällige Reduktion der Geschwindigkeit (Trödeln)
- wenn nötig abbremsen

#### 4.1.1 Zustandsautomat

Das oben beschriebene Verhalten wurde zuerst mit Hilfe eines endlichen Automaten beschrieben, siehe Abbildung 8.

---

<sup>2</sup><https://lightjason.github.io/news/2017-09-workshop/>

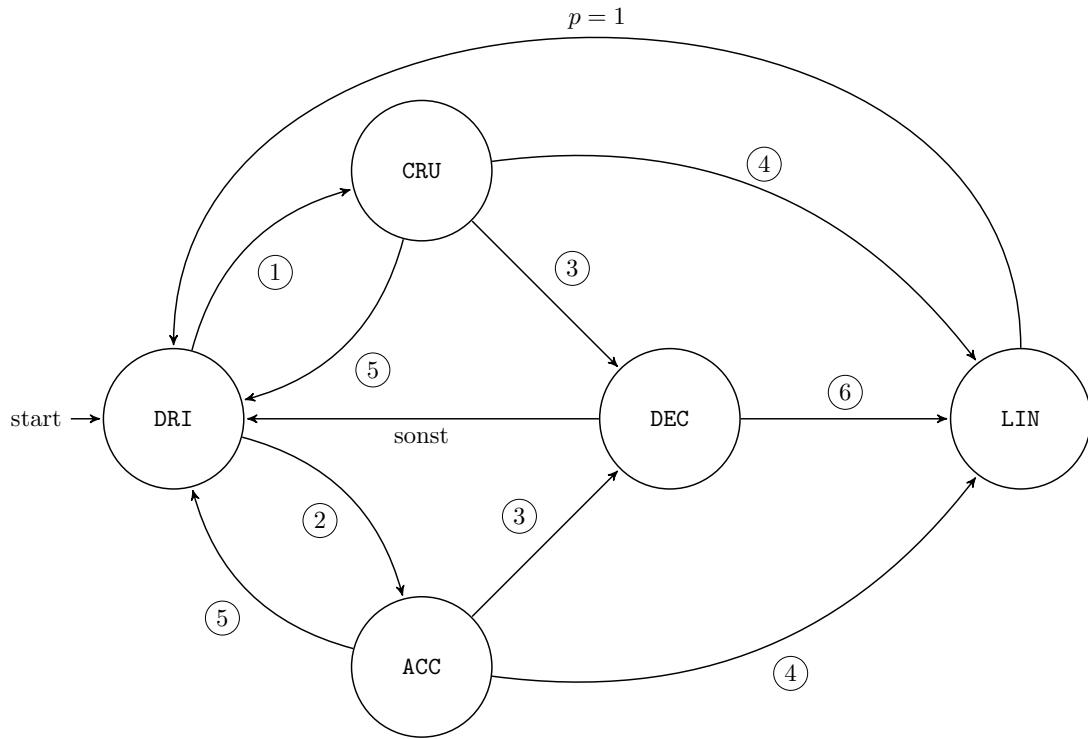


Abbildung 8: Zustandsautomat für das Agentenverhalten nach dem Nagel-Schreckenberg-Modell

Erläuterungen zu Abbildung 8:

1.  $v_{aktuell} \geq v_{limit}$
2.  $v_{aktuell} < v_{limit}$
3. „Verkehr voraus“
4. „kein Verkehr voraus“ UND  $plinger$
5. „kein Verkehr voraus“ UND  $1 - plinger$
6.  $v_{aktuell} > 0$  UND  $plinger$

Startzustand des Automaten ist ein allgemeiner Zustand DRI (Drive/Fahren), in dem nichts am Verhalten des Agenten geändert wird. Von hier aus ist der Zustand ACC (Accelerate/Beschleunigen) zu erreichen, wenn die aktuelle Geschwindigkeit unter der max. zulässigen liegt. Der Zustand CRU (Cruise/Geschwindigkeit halten) wird erreicht, wenn die max. zulässige Geschwindigkeit erreicht oder überschritten ist.

Die Übergangsmöglichkeiten aus den zwei letztgenannten Zuständen sind identisch. Ist kein vorausfahrender Verkehr vorhanden und wird nicht getröstet erfolgt der Übergang zurück in den normalen DRI-Zustand. Wird stattdessen zufällig mit Wahrscheinlichkeit  $plinger$  das Trödeln angestoßen, wird der entsprechende Zustand LIN (Linger/Trödeln) erreicht. Ist Verkehr voraus, wird der Zustand DEC (Decelerate/Verzögern, Bremsen) erreicht.

Aus dem DEC-Zustand kann zufällig ein Übergang in den LIN-Zustand erfolgen ansonsten zurück in den allgemeinen DRI-Zustand. Aus dem LIN-Zustand wird zwangsläufig wieder in den DRI-Zustand übergegangen.

Ein Zeitschritt der Simulation entspricht dabei einem Zyklus vom Startzustand ausgehend bis zur Rückkehr dorthin.

### 4.1.2 Agentenpläne

Der o.g. Automat wurde in die Agentensprache „übersetzt“. Die Agentenversion unterscheidet sich dabei leicht von der Automatenversion.

Der Agent beginnt im ersten Zeitschritt mit dem Ziel `!cruise`, siehe Zeile 1 in Listing 1. Der Plan `cruise` erfüllt dieses Ziel und wird demzufolge ausgeführt. Dort werden alle weiteren Pläne (`accelerate`, `linger` und `decelerate`) für den nächsten Zeitschritt zur Ausführung hinzugefügt. Außerdem ruft sich der Plan selbst wieder auf.

Es entsteht somit ein wiederholter Aufruf der Unterpläne, die das Verhalten des Agenten steuern. Bedingungen werden erst im jeweiligen Plan geprüft.

Während der Simulationsläufe diente der `cruise`-Plan zudem dem Loggen der Statusdaten. Das komplette Listing siehe Listing A2.

Listing 1: Auszug aus Agentenscript: single lane-Version

```
1 !cruise.  
2  
3 // --- start all other plans ---  
4 +!cruise <-  
5   !accelerate;  
6   !decelerate;  
7   !linger;  
8   !cruise  
9 .
```

Der `accelerate`-Plan war ursprünglich mit der Bedingung ausgestattet, dass die Beschleunigung nur erfolgen soll, wenn die aktuelle Geschwindigkeit unter der zulässigen liegt. Hier ergab sich das Problem, dass auch bei vorwärtigem Verkehr eine Beschleunigung durchgeführt wurde. Die nachfolgende Verzögerung musste diesen Geschwindigkeitszuwachs in jedem Zeitschritt zusätzlich abbauen. Die Fähigkeit zu verzögern war entsprechend schlecht.

Abgestellt wurde dies durch den Zusatz, dass eine Beschleunigung ebenfalls ausbleiben soll, wenn sich ein Fahrzeug im vorwärtigen Sichtbereich befindet. Mit der Erweiterung des Sichtfeldes, siehe Kapitel 4.2.1, wurde es nötig, außerdem die Entfernung, ab der nicht mehr beschleunigt werden soll, mit in die Bedingung aufzunehmen.

Um einen realen Folgeverkehr zu ermöglichen, wurde erlaubt, dass beschleunigt werden darf, solange die Geschwindigkeit des Vorausfahrenden höher ist.

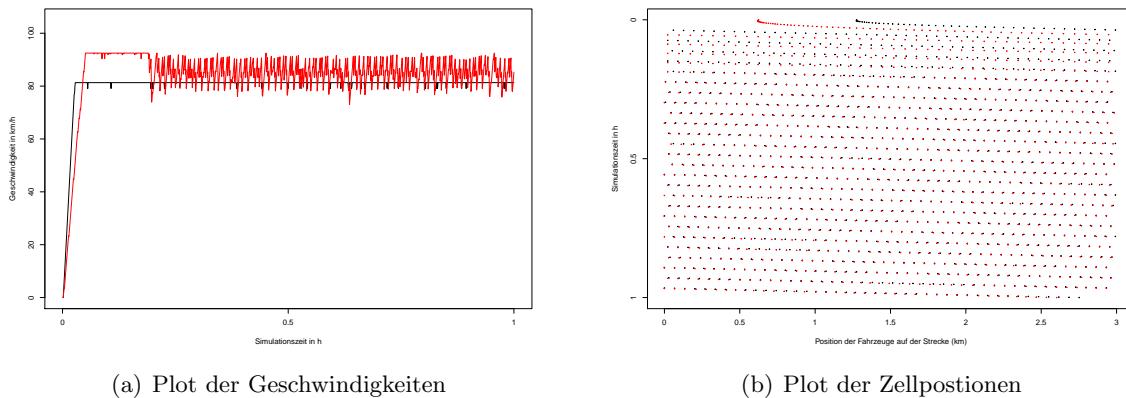
Der `linger`-Plan modelliert das zufällige Beibehalten bzw. Verringern der Geschwindigkeit. Zunächst wurde das Trödeln mit einer Wahrscheinlichkeit von 10% und einer Intensität von 50% der Bremskraft ausgeführt. Für die Dauertests wurden diese Werte später noch angepasst, siehe Kapitel 4.2.2.

Das Bremsverhalten des Agenten wird durch den `decelerate`-Plan gesteuert. Dort wird die Verzögerung bei vorwärtigem Verkehr geregelt. Auch hier wurden durch die bereits angemerkt Erweiterung des Sichtfelds Änderungen nötig. Das Abbremsen sollte nicht übermäßig stark und die entstehende Lücke nicht zu groß ausfallen. Diese Vorgaben konnten durch zusätzliche Bedingungen für die relative Geschwindigkeit und einen Mindestabstand erreicht werden.

Die realistische Modellierung der Geschwindigkeitsänderungen macht es unmöglich, dass die Fahrzeuge in aufeinanderfolgenden Zellen hintereinander fahren. Vielmehr muss, wie im realen Verkehr auch, ein gewisser Abstand, innerhalb dessen der Nachfolgeverkehr reagieren kann, eingehalten werden. Ein Wert von 100 m erschien in der Simulation sinnvoll, um die Kollisionshäufigkeit zu verringern. Für eine Geschwindigkeit von 100 km/h entspricht diese Entfernung nach der Faustformel, die in Fahrschulen gelehrt wird, dem Bremsweg, vgl. [ADA].

Ist der Geschwindigkeitsunterschied zwischen Fahrzeugen allerdings zu groß, konnten Kollisionen bzw. Kollisionsereignisse nicht verhindert werden.

**Tests des Agentenscripts:** Um die Wirksamkeit der Anweisungen für die Agenten in dieser Phase zu testen, wurden in der Szenarienvereinbarung zwei Arten von Fahrzeugen festgelegt. Diese waren mit identischen Plänen ausgestattet. Der einzige Unterschied zwischen den Fahrzeugen war, dass eines seine Geschwindigkeit aus einem niedrigeren Intervall zugewiesen bekam. Somit war sichergestellt, dass das schnellere Fahrzeug im Laufe der für Testzwecke kurz gehaltenen Simulationzeit auf das langsamere aufholen würde. Es war wiederholbar zu testen, ob das Abbremsen bei vorausfahrendem Verkehr funktionierte, siehe Abbildung 9.



**Abbildung 9:** Beispieldiagramme eines Testdurchlaufs

Dauer: 1 h, Länge: 3 km, Zellgröße: 7,5 m, Zeitschritt: 0,05 min

Die Geschwindigkeiten werden in einem Zeit-Geschwindigkeits-Diagramm (Abbildung 9(a)) dargestellt, die Bewegung in einem Positions-Zeit-Diagramm (Abbildung 9(b)).

Dort entspricht jeder Punkt pro Zeile der Position eines Fahrzeugs im jeweiligen Zeitschritt. Die Simulationszeit beginnt am oberen Ende der Zeitachse. Durch die Ringform der Strecke werden mehrere Durchläufe der Strecke dargestellt.

## 4.2 Verfeinerung des Agentenverhaltens

Das Szenario für eine Simulation kann über eine YAML-Datei individuell festgelegt werden, siehe Listing A1. Dort besteht u.a. die Möglichkeit für die Angabe von:

- Streckenlänge (`length_in_km`),
- Anzahl der Fahrspuren (`lanes`),
- Simulationsdauer (`simulationtime_in_minutes`),
- Zellgröße (`cellsize_in_meter`),
- Zeitschrittgröße (`timestep_in_minutes`),
- Anzahl der Fahrzeuge (`count`),
- Ausdehnung des Sichtfeldes (`viewrange`)

Durch die Angabe von Intervallen können die Geschwindigkeit (`speed`) sowie die Beschleunigungs- und Verzögerungswerte (`acceleration/deceleration`) der Fahrzeuge eingestellt werden.

---

#### 4.2.1 Einstellmöglichkeiten in der Szenariokonfiguration

**Zellgröße:** Die Entwicklung der Agentenpläne wurde ausschließlich in einer Simulationsumgebung mit einer Zellgröße von 7,5 m durchgeführt. Als das Verhalten der Agenten zufriedenstellend war, wurde u.a. auch wegen der in Kapitel 4.6.1 beschriebenen Bremsprobleme getestet, inwiefern sich die Zelllänge verkürzen lässt und welchen Einfluss dies auf die Simulation hat.

**Zeitschrittlänge:** Die Länge der Zeitschritte lässt sich in der Konfiguration in Minuten oder Bruchteilen davon angeben. Dies stellte sich als unpraktikabel heraus, da z.B. eine Dauer von einer Sekunde durch den Dezimalbruch für  $\frac{1}{60}$  angegeben werden müsste. Dieser weist aber eine Periodizität auf.

Begonnen wurde mit einer Zeitschrittlänge von 0,1 min, was sechs Sekunden entspricht. Diese Länge erwies sich als nicht sinnvoll, da zwischen den einzelnen Entscheidungszeitpunkten eine zu lange Zeitspanne liegt.

Die Entwicklung der Agentenpläne wurde mit einer Schrittlänge von drei Sekunden, 0,05 min, durchgeführt. Diese Länge genügte für diesen Zweck völlig. Für die Tests auf Leistungsfähigkeit wurde die Schrittlänge dann auf 0.025 min, 1,5 Sekunden, reduziert, siehe Kapitel 4.4.2.

**Beschleunigung/Verzögerung:** Das originale Modell von Nagel und Schreckenberg legt Beschleunigung, Verzögerung und Sichtweite sehr einfach fest. Bedingt durch die Zellgröße von 7,5 m, die Ganzzahligkeit der Geschwindigkeit und die Zeitschrittlänge von 1 Sekunde ergeben sich Beschleunigungswerte von  $7,5 \frac{m}{s^2}$ . Durch die Möglichkeit, die Geschwindigkeit innerhalb eines einzigen Zeitschrittes von der Maximalgeschwindigkeit „5“ auf Null zu verringern, ergibt sich theoretisch eine Verzögerung von  $37,5 \frac{m}{s^2}$ . Dies ist unrealistisch.

Die Simulationsumgebung arbeitet hier mit frei wähl- und einstellbaren Werten. In [BM09] sind die Werte für eine Vielzahl von Fahrzeugen aufgelistet. Für die Simulation wurden als real mögliche Werte Intervalle zwischen 3,5 und  $7 \frac{m}{s^2}$  für die Beschleunigung und zwischen 8 und  $10 \frac{m}{s^2}$  für die Verzögerung gewählt. Der Wert wird für jeden Agenten bei der Initialisierung im Rahmen dieser Intervalle zufällig festgelegt.

Dabei sind auch ungewöhnliche Kombinationen - z.B. ein guter Beschleunigungswert gepaart mit einem schlechten Verzögerungswert - in der realen Welt nicht unmöglich. Die o.g. Konstellation trafe etwa auf einen Sportwagen mit defekten Bremsen zu.

Die Dosierung der möglichen Beschleunigung bzw. Verzögerung wird im Agentenscript festgelegt.

**Sichtweite:** Für die Sichtweite ergibt sich im NaSch-Modell der theoretische Wert von  $5 \times 7,5$  m, also 37,5 Meter (bzw. je nach max. möglicher Geschwindigkeit  $v_{max}$ :  $v_{max} \times 7,5$  m).

Für die in der Agentensimulation vorliegenden „realen“ Bremsvorgänge ist diese Entfernung unzureichend.

Die Sichtweite im Straßenverkehr kann laut [Hri09] aufgrund von physiologischen und psychologischen Gründen in drei Zonen eingeteilt werden:

- „Fernorientierung/Information“, auf Autobahnen etwa zwischen 600 und 360 Metern
- „Bereitschaft/Entscheidung“, etwa zwischen 360 und 110 Metern
- „Nahorientierung/Handlung“, unter 110 Metern

---

Die für Reaktionsträgheit maßgebliche Komponente Mensch fehlt im Agentensystem. Darum muss die Zone der Fernorientierung nicht ausgeschöpft werden. Ein Wert für die Sichtweite mittig im Intervall der Entscheidungszone genügt, um bis zum Beginn der Handlungszone „tätig zu werden“.

Für die Sichtweite wurde als Wert 250 m gewählt. Dieser trifft die o.g. Vorgabe. Den Agenten sollte diese Strecke ausreichend Raum für Reaktionen geben. In den Agentenplänen wird schließlich die Handlung durch Bedingungen auf einen Bereich unter 110 m beschränkt.

#### 4.2.2 Einstellmöglichkeiten im Agentenverhalten (Parameter)

**Beschleunigungs- und Verzögerungsraten:** Die Beschleunigungs- und Verzögerungsraten können durch einen Wert zwischen Null und Eins angegeben werden. Im `accelerate`-Plan wurde 0,5 gewählt, was 50% der möglichen Beschleunigung entspricht. Für die Verzögerung im `decelerate`-Plan wurde der größtmögliche Wert 1 festgelegt. Kleinere Werte führen nicht zu ausreichend großer Bremsleistung.

**Festlegen der Trödelparameter:** Im Originalmodell wird dem jeweiligen Fahrzeug im Trödelfall eine Geschwindigkeitseinheit wieder abgezogen. In jenem Zeitschritt erfolgt somit keine Beschleunigung bzw. bei Fahren mit max. möglicher Geschwindigkeit wird diese reduziert. Für das Abbremsen hinter einem anderen Fahrzeug wirkt das Trödeln wie eine Art „Überbremsen“, da die Geschwindigkeit ohnehin an die Größe der vorhandenen Lücke angepasst wird.

Für die Simulation musste ein Wert gefunden werden, der dieses Verhalten näherungsweise nachbildet. Im Laufe mehrerer Testdurchgänge schien eine Intensität der Verzögerung von 0,3 die getätigte Beschleunigung am besten zu egalisieren. Hierbei ist noch die Streuung der Beschleunigungs- und Verzögerungswerte zu beachten, sodass das Trödeln unterschiedlich große Effekte haben kann.

Für das Abbremsen in der Agentensimulation kann sich das Trödeln positiv auf den Bremsweg auswirken, da ein zusätzlicher Bremsimpuls erzeugt wird. Hier wird im allgemeinen aber in Summe eine Verzögerung über dem max. real möglichen Wert erzielt.

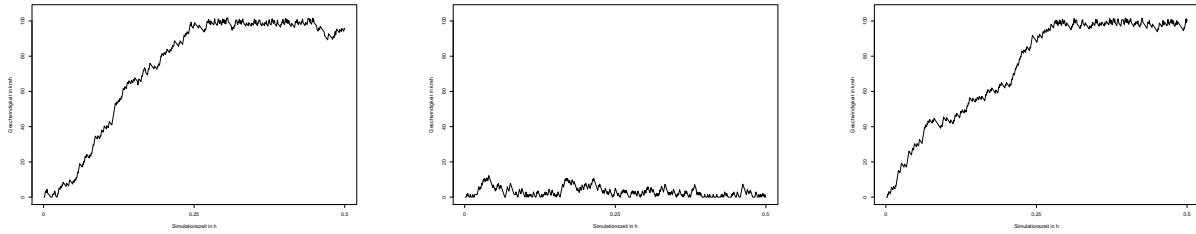
Aufgrund der „Bremsprobleme“, siehe Kapitel 4.6.1, wurde dies aber vorerst in Kauf genommen.

Für die Trödelwahrscheinlichkeit  $p_{linger}$  wurde beobachtet, dass deren Höhe bei Werten  $\leq 0,5$  nur einen kleinen Einfluss auf das Erreichen der max. möglichen Geschwindigkeit hatte. In den Testdurchläufen konnte diese Geschwindigkeit jeweils erreicht werden.

Bei einer größeren Wahrscheinlichkeit wurde die mögliche Geschwindigkeit innerhalb der 30 Minuten Simulationszeit teilweise nicht oder nur schwer erreicht. Bei einem Wert von 0,7 ergab sich ein Durchlauf, bei dem nicht einmal die 20 km/h-Marke erreicht wurde, siehe Abbildung 10(b).

Eine merkliche Veränderung trat auch beim Beschleunigungsvermögen der Fahrzeuge auf. Je höher der Wert für  $p_{linger}$ , desto flacher wurde im allgemeinen der Anstieg bei der Beschleunigung, siehe Abbildung 11. Außerdem bilden sich mit zunehmender Trödelwahrscheinlichkeit deutlichere Geschwindigkeitseinbrüche aus - z.B. Abbildung 11(a) vs. Abbildung 11(b).

Weiterhin hat die Trödelwahrscheinlichkeit bei gleicher Höhe, scheinbar abhängig von den gefahrenen Geschwindigkeiten, unterschiedliche Einflüsse auf den Verkehrsfluss mehrerer Fahrzeuge. Bei einer max. möglichen Geschwindigkeit von 50 km/h erzeugt erst eine Wahrscheinlichkeit von 0,5 einen merklichen Effekt. Für eine Geschwindigkeit von 100 km/h ist dies bei gleicher Fahrzeuganzahl bereits bei einem Wert von 0,3 der Fall.

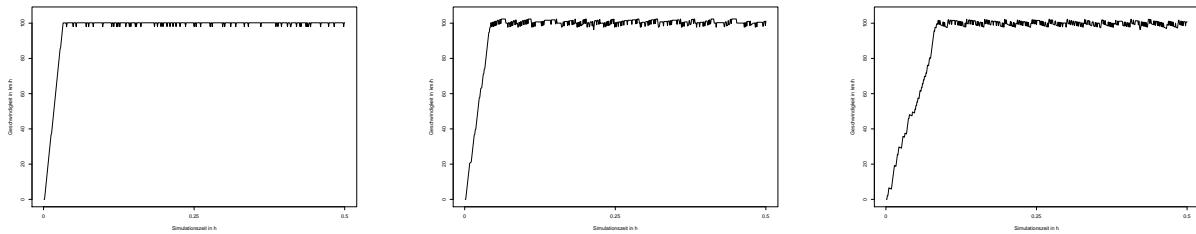


(a) 1. Durchlauf

(b) 2. Durchlauf

(c) 3. Durchlauf

Abbildung 10: Geschwindigkeitskurven bei  $p_{linger} = 0,7$



(a) Durchlauf  $p_{linger} = 0,1$

(b) Durchlauf  $p_{linger} = 0,3$

(c) Durchlauf  $p_{linger} = 0,5$

Abbildung 11: Beispiele für Geschwindigkeitskurven bei unterschiedlichen Werten für  $p_{linger}$

Bei Werten darüber hinaus wurde beobachtet, dass die Beschleunigung eines oder mehrerer Fahrzeuge so gehemmt wurde, dass ein Erreichen der möglichen Geschwindigkeiten für alle im System befindlichen Fahrzeuge nicht möglich war.

Für die Dauertests mit einer max. möglichen Geschwindigkeit von 100 km/h wurden die Parameter im `linger`-Plan somit auf Wahrscheinlichkeit von 0,3 und eine Bremsintensität von 0,3 festgelegt. Der Wahrscheinlichkeitswert musste allerdings nochmals korrigiert werden (Kapitel 4.5).

### 4.3 Anpassungen der Simulationssoftware

Einer der Hauptunterschiede zur Softwareversion des Workshops ist, dass die Simulationen hier auf einer unendlichen Strecke stattfinden, wie bereits in Kapitel 2.1 für die Versuche von Nagel und Schreckenberg beschrieben.

Diese Unendlichkeit existiert dabei allerdings nur virtuell. Fahrzeuge, die das Ende der Strecke passieren, werden mit gleichen Verhaltenswerten am Anfang wieder eingesetzt. Der Abstand vom Anfang der Strecke richtet sich dabei nach der letzten Position am Streckenende und der aktuellen Geschwindigkeit.

Das Umsetzen selbst bereitete keine Probleme, allerdings resultierten solche aus dieser Unterbrechung der Strecke.

#### 4.3.1 Schwierigkeiten bei der Simulation der Ringform der Strecke

Die Streckenunterbrechung führte dazu, dass es eine Pause in der Sichtbarkeit der Fahrzeuge untereinander gab, wenn sich diese an unterschiedlichen Enden der Strecke befanden. Aufgrund dieser „Unsichtbarkeit“ war es unmöglich, eine Abstandsberechnung zwischen den einzelnen Fahrzeugen durchzuführen. Dies führte zu einer freien Beschleunigung, da eine freie Strecke vermutet wurde.

---

Nachfolgend ist dies anhand des Log-Auszuges einer Simulation mit zwei Fahrzeugen, `vehicle0` und `vehicle20`, dargestellt. („...“ bezeichnet Kürzungen der Ausgabe.)

In Zeitschritt 315 war das Fahrzeug `vehicle20` gerade noch in der letzten Zelle der Fahrbahn und überschreitet diese Grenze einen Zeitschritt später. Die Beliefliste beider Fahrzeuge ist somit ab dem Schritt 316 leer. Erst vier Zeitschritte später befindet sich auch das Folgefahrzeug `vehicle0` am Anfang der Fahrspur und ist ab Zeitschritt 320 wieder für Fahrzeug `vehicle20` sichtbar (andersherum ebenso).

```
----- step 315 -----
  vehicle0  -> BELIEFLIST  [view/vehicle[id[vehicle20], ... direction[forward[]]]]
  vehicle20  -> BELIEFLIST  [view/vehicle[id[vehicle0], ... direction[backward[]]]]
  vehicle0  in lane  1  in cell  331.0  @  87.11188616490347  kph
  vehicle20  in lane  1  in cell  400.0  @  88.65039571533352  kph
----- step 316 -----
  vehicle0  -> BELIEFLIST  []
  vehicle20  -> BELIEFLIST  []
  vehicle0  in lane  1  in cell  345.0  @  87.89778423377561  kph
  vehicle20  in lane  1  in cell  14.0  @  88.65039571533352  kph
----- step 317 -----
...
----- step 318 -----
...
----- step 319 -----
  vehicle0  -> BELIEFLIST  []
  vehicle20  -> BELIEFLIST  []
  vehicle0  in lane  1  in cell  387.0  @  89.01152857206483  kph
  vehicle20  in lane  1  in cell  56.0  @  88.65039571533352  kph
----- step 320 -----
  vehicle0  -> BELIEFLIST  [view/vehicle[id[vehicle20], ... direction[forward[]]]]
  vehicle20  -> BELIEFLIST  [view/vehicle[id[vehicle0], ... direction[backward[]]]]
  vehicle0  in lane  1  in cell  1.0  @  89.79742664093698  kph
  vehicle20  in lane  1  in cell  70.0  @  88.65039571533352  kph
```

Für die Simulation mit mehreren Fahrzeugen stellte dies insbesondere ein Problem dar, weil eine Stauwelle, die sich in ihrer Rückwärtsbewegung dem Anfang der Strecke näherte, von Fahrzeugen am Ende der Strecke nicht erkannt werden konnte. Demzufolge gab es auch keine Reaktion auf das Hindernis.

Dies führte zu dem Verhalten, dass Fahrzeuge vom Ende der Strecke an den Anfang gesetzt werden sollten, dort aber nicht in eine freie Zelle gesetzt werden konnten. Die Simulationsumgebung sieht hier vor, dass das Fahrzeug auf die freie Zelle (in Fahrtrichtung) hinter das jeweils letzte Fahrzeug gesetzt wird.

Softwareseitig ist Zelle 0 die erste zu besetzende Zelle einer Lane. Die freie Zelle hinter dem Fahrzeug war hier aber Zelle -1 und nicht die letzte Zelle am Lane-Ende. Zudem wurde ein Fahrzeug, das sich bereits dort befand, jeweils noch ein Feld weiter nach hinten gesetzt und das neue Fahrzeug dazwischen einfügt. Dies setzte sich kaskadierend in den negativen Bereich fort. Hierbei handelt es sich um einen separaten Fehler, der durch ersten erkannt werden konnte.

Es ergab sich bei einigen Simulationsläufen eine Häufung von Positionspunkten im negativen, eigentlich nicht vorhandenen, Teil der Strecke. In der Simulation in Abbildung 12 trat das Problem direkt am Anfang des Durchlaufs aufgrund der langen Standzeit (siehe auch Kapitel 4.6.1) des

direkt am Anfang der Lane verorteten Fahrzeuges auf und konnte über die Simulationsdauer nicht abgebaut werden.

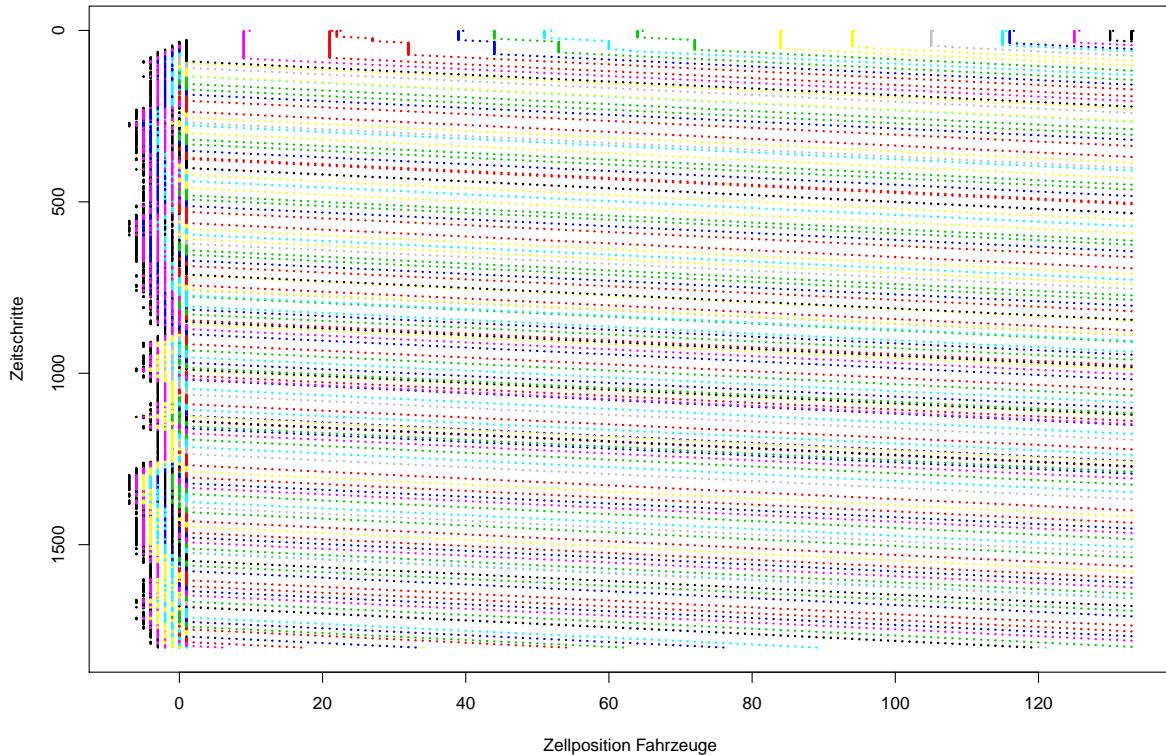


Abbildung 12: Häufung der Punkte im negativen Bereich (Position-Zeit-Diagramm)

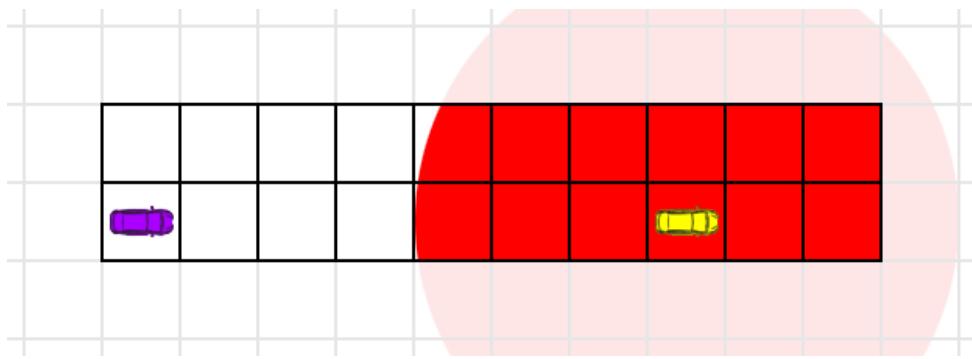
In Abbildung 13(a) wird die Problematik des „Nicht-Sehens“ beispielhaft am Sichtfeld des gelben Fahrzeugs grafisch verdeutlicht. Gleiches gilt analog für die rückwärtige Sicht des violetten Fahrzeugs.

Für die Erstellung der sichtbaren Umgebung der Fahrzeuge werden die Zellen zur „view range“, die sich innerhalb der im Szenario angegebenen Entfernung befinden. Diese werden als relative Angaben mit dem jeweiligen Fahrzeug als Mittelpunkt gespeichert und in jedem Zeitschritt auf reale Koordinaten umgerechnet. Befindet sich dann ein anderes Fahrzeug in einer dieser Zellen, so ist es im Sichtfeld und eine Entfernung und Richtung werden berechnet.

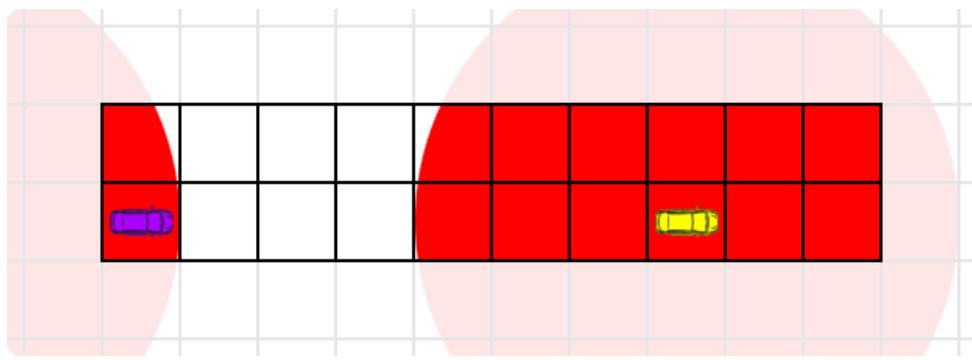
Die Schwierigkeit bestand darin, die Zellen am jeweils anderen Ende der Strecke mit in die Liste der relativen Zellpositionen aufzunehmen, siehe Abbildung 13(b), und virtuelle Zellpositionen korrekt auf reale Positionen umzurechnen.

Durch Testläufe unterschiedlicher Versionsstände der Simulationssoftware mit identischen Szenariovorgaben - ein Fahrzeug wurde mit einer so kleinen Geschwindigkeit generiert, dass es in seiner Startzelle verblieb - konnte die Behebung des Fehlverhaltens nachgewiesen werden, siehe Abbildung 14.

Danach konnten dann auch Stauwellen beobachtet werden, die sich rückwärts vom Streckenanfang her darüber hinaus am Streckenende fortsetzten.



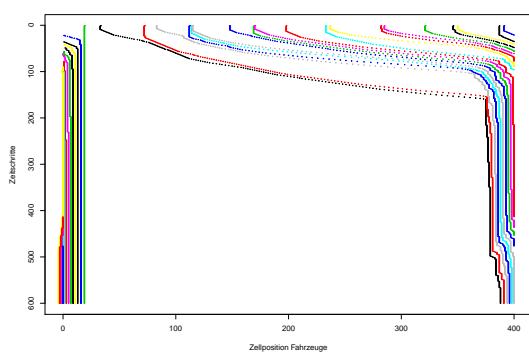
(a) ohne Umbruch



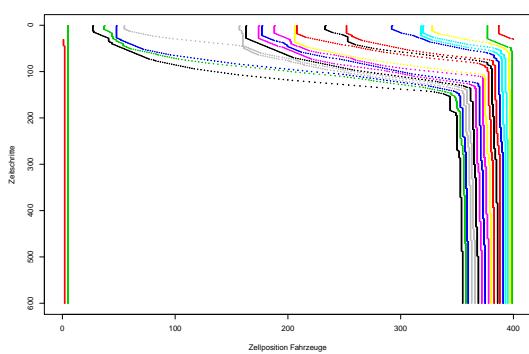
(b) mit Umbruch

Abbildung 13: Sichtfeld des gelben Fahrzeugs am Ende der Lane

Quelle Auto-Silhouette: vecteezy.com



(a) vorher: Durchlauf mit Fehler



(b) nachher: Durchlauf ohne Fehler

Abbildung 14: Fehlerbehebung des Übergangs vom Streckenende zum Anfang

#### 4.3.2 Unterteilung des Sichtfeldes der Fahrzeuge

Um die relative Position eines anderen Fahrzeugs in der Umwelt festzustellen, kann die Richtung, in der dieses sich befindet, bestimmt werden. Die Simulationssoftware hatte für das Sichtfeld der Fahrzeuge eine Unterteilung in acht Sektoren vorgesehen, sodass neben den Richtungen vorwärts, rückwärts, links und rechts auch Unterscheidung in Zwischenschritte links vorwärts, rechts vorwärts usw. möglich war, siehe Abbildung 15(a).

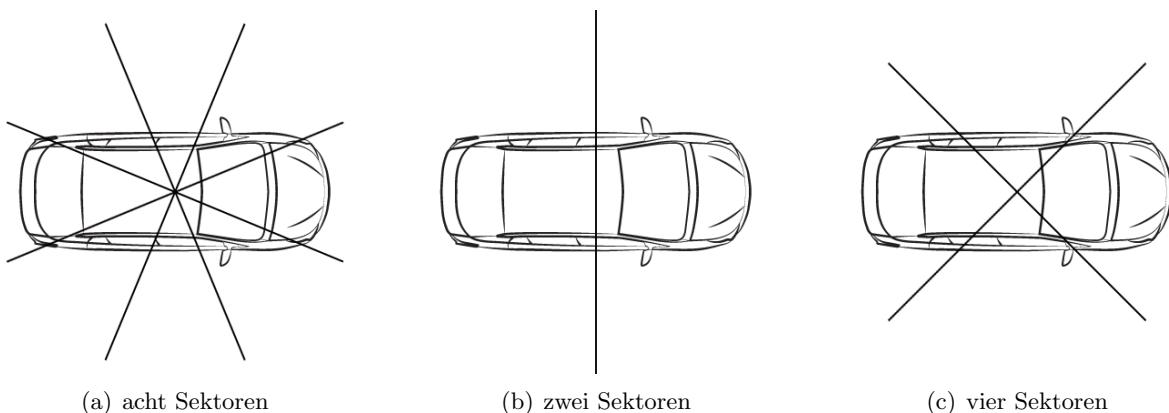


Abbildung 15: mögliche Unterteilung des Sichtfeldes

Quelle Auto-Silhouette: vecteezy.com

Eine solch feine Unterscheidung war nicht nötig. Es sollte genügen, feststellen zu können, ob sich ein anderes Fahrzeug relativ zum betrachteten Fahrzeug im vorwärtigen oder rückwärtigen Raum befindet, siehe Abbildung 15(b). Die Abdeckung der Umwelt sollte dabei ohne Unterbrechung sein, d.h. jeder Position sollte einem der beiden Bereiche zugewiesen werden können. Für die Simulation nach dem Einspurmodell von Nagel-Schreckenberg funktionierte dies ohne Probleme.

Bei einem Testlauf des Mehrspurmodells mit vier Fahrzeugen kam es allerdings zur Kollision zweier Fahrzeuge. Bei der Kontrolle des Logs wurde festgestellt, dass ein Fahrzeug (`vehicle0`), welches sich auf der Überholspur (lane 2) befand, ein anderes Fahrzeug (`vehicle20`) direkt neben sich nicht ‚gesehen‘ hatte. Die ist an der Zeile „PIA1“ zu erkennen, die nur ausgegeben wurde, wenn sich keinerlei Verkehr im Sichtfeld des Fahrzeuges befindet. Es gab in dieser Konstellation demnach keine Zuordnung dieser Position zu einem der beiden Bereiche. Daraufhin, so wie es das Verhalten des Agenten vorsah, wurde zum nächsten Schritt (366) in die Hauptspur gewechselt. In diesem Zeitschritt kam es dann zum Auslösen des Kollisionsereignisses, da der Abstand zwischen `vehicle0` und `vehicle20` nicht ausreichend groß war, um das hintere Fahrzeug mit seiner aktuellen Geschwindigkeit die volle Strecke zu bewegen.

*Anmerkung:* Gleichzeitig reagierte `vehicle0` korrekt auf das Hindernis direkt vor sich und leitete für den folgenden Zeitschritt einen Überholvorgang ein.

---

```

--- step 365 -----
    vehicle20  in lane  1  in cell   333.0  @  74.21582362319865  kph
    vehicle2   in lane  1  in cell   282.0  @  129.99324819683062  kph
    vehicle0   in lane  2  in cell   333.0  @  99.94723620551235  kph
PIA1     vehicle0  sees no traffic at all -> Pull-in
    vehicle1   in lane  1  in cell   90.0  @  130.8321253026147  kph
--- step 366 -----
    vehicle20  in lane  1  in cell   4.0   @  74.21582362319865  kph
    vehicle2   in lane  1  in cell   289.0  @  131.11124225593747  kph
    vehicle0   in lane  1  in cell   3.0   @  100.80986539456516  kph
TFC100   vehicle0  has vehicle in-front of -> decelerate
COS      vehicle0  STOPPED -> collision
OUT      vehicle0  -> Pull-out attempt successful
    vehicle1   in lane  1  in cell   97.0  @  130.64144521630223  kph

```

Das Sichtfeld der Fahrzeuge wurde daraufhin zu einem Vier-Sektoren-Modell abgewandelt, siehe Abbildung 15(c), und die Pläne für Ein- und Ausscheren, der hiervon gleichermaßen betroffen sein dürfte, angepasst. Es werden nun jeweils die beiden Sektoren kontrolliert, die die Bewegungs- bzw. Orientierungsrichtung darstellen. Nach vorn und hinten wird jeweils auch auf Entfernung geprüft, seitlich genügt es auf reine Präsenz/Nichtpräsenz von anderen Fahrzeugen zu prüfen.

Im Rahmen dieser Umstellung wurde festgestellt, dass es bei der Richtungsbestimmung einen weiteren Fehler in der Simulationssoftware gab. Die Berechnung lieferte für beide seitlichen Richtungen ein und denselben Wert und somit auch die gleiche Blickrichtung. Der Fehler war durch die Verwendung der Methode `acos` entstanden. Es wurde nicht beachtet, dass der `arccos` nur im Bereich 0 bis  $\pi$  definiert ist, hier aber der Vollkreis zugrunde liegt.

## 4.4 Setups für die Langzeittests

Für die Durchführung der Dauersimulationen galt es sinnvolle Einstellungen zu finden. Für das Agentenverhalten wurde dies während der Verhaltensentwicklung realisiert, siehe Kapitel 4.2. Nachfolgend sind Festlegungen für das Szenario zu finden.

### 4.4.1 Fahrzeuganzahl

Um einen Anhaltspunkt für die Anzahl der zu simulierenden Fahrzeuge zu erhalten, wurden reale Daten der Bundesanstalt für Straßenwesen aus dem Jahr 2016 für je eine automatische Dauerzählstelle auf den Autobahnen A7, A38 und A71 in der Harzregion herangezogen, siehe Tabelle 3. Das Teilstück auf der A7 war zu dieser Zeit nur zweispurig ausgebaut. Die A38 und die A71 sind generell nur zweispurig.

Für eine Messstelle auf der A395 zwischen Bad Harzburg und Braunschweig lagen nur Werte aus dem Jahr 2014 vor. Die Zahlen lagen nur wenig über den aktuelleren der Messstelle auf der A38. Aus diesem Grund wurde auf die Einbeziehung verzichtet.

Bei der Berechnung der Durchschnitte wurde, wenn nötig, auf volle Fahrzeuge aufgerundet.

Die o.a. Zahlen spiegeln eine Verkehrsstärke  $q$  (in Fahrzeuge pro Zeit) wieder. Nach [SL11] kann diese mit Hilfe der mittleren momentanen Geschwindigkeit  $\bar{v}_m$  in eine Verkehrsdichte  $\rho$  (in Fahrzeuge pro Streckenabschnitt) umgerechnet werden. Der Zusammenhang lautet:

$$\rho = \frac{q}{\bar{v}_m}$$

| Zählstelle<br>(Autobahn, Quelle) | Düderode<br>(A7, [Buna])   | Röstebachtalbrücke<br>(A38, [Bunb]) | Tunnel Schmücke<br>(A71, [Bunc]) |
|----------------------------------|----------------------------|-------------------------------------|----------------------------------|
| Lage zw. AS                      | Seesen (Harz)<br>und Echte | Breitenworbis<br>und Leinefelde     | Heldrungen<br>und Kölleda        |
| Verkehr Fahrtrichtung 1          | 25.881                     | 11.851                              | 6.418                            |
| Verkehr Fahrtrichtung 2          | 26.340                     | 11.877                              | 6.386                            |
| Durchschnitt pro Tag             | 26.111                     | 11.864                              | 6.402                            |
| Durchschnitt pro Stunde          | 1.088                      | 495                                 | 267                              |

Tabelle 3: durchschnittliche tägliche Verkehrsdaten

Die weiteren Berechnungen erfolgten jeweils mit den Durchschnittswerten der beiden Fahrtrichtungen. Als  $\bar{v}_m$  wird für Pkw 120 km/h angenommen. Somit ergibt sich auf diesen Autobahnstücken eine Verkehrsdichte von 9,066 Fahrzeugen pro km für die A7, 4,125 für die A38 und 2,225 für die A71.

Aufgrund der Einspurigkeit muss dieser Wert für das Szenario halbiert werden. Die Anzahl der zu simulierenden Fahrzeuge erhält man dann durch Multiplikation mit der Streckenlänge. Die Rundung erfolgte zum nächsten ganzen Fahrzeug.

Ausgangswerte für die Simulation einer 7,5 km langen Strecke sind die in Tabelle 4 angegebenen Fahrzeugzahlen.

| Autobahn | A7 | A38 | A71 |
|----------|----|-----|-----|
| Anzahl   | 34 | 15  | 8   |

Tabelle 4: Fahrzeuganzahl für Streckenlänge 7,5 km

### Anmerkung zum Teilstück auf der A7

Die Autobahn A7, im Land Niedersachsen, wurde in den vergangenen Jahren bereits an vielen Stellen auf drei Spuren je Fahrtrichtung verbreitert. Das Teilstück zwischen den Anschlussstellen Seesen und Nörten-Hardenberg war eines der letzten zweispurigen. Der Umbau hat im September 2017 begonnen<sup>3</sup>. Aufgrund seiner Lage zwischen zwei bereits dreispurig ausgebauten Teilstücken mussten hier sicherlich größere Verkehrslasten bewältigt werden als auf anderen zweispurigen Autobahnen.

Für die Simulation war dies ein glücklicher Zufall, da im speziellen  $p_{linger}$  mit der so vorgegebenen Fahrzeuganzahl weiter präzisiert werden konnte, siehe Kapitel 4.5.

#### 4.4.2 Zellgröße und Zeitschrittänge

Den hier folgenden Ausführungen liegt das in Tabelle 5 beschriebene Szenario zugrunde.

Mit dem bereits erwähnten Vorgehen - identische Agentenpläne, unterschiedliches Geschwindigkeitsintervall bei der Initialisierung (schnelleres Fahrzeug  $v \in [80; 130]$  und langsameres Fahrzeug  $v \in [70; 90]$ ) - wurde das Verhalten der Agenten, ausgehend von einer Zellgröße von 7,5 Metern (in 2,5 m-Schritten absteigend) und einer Zeitschrittgröße von 0,1 Minuten (jew. halbiert), in jew. drei Durchgängen getestet.

<sup>3</sup>siehe [https://www.strassenbau.niedersachsen.de/download/45795/Sechsstreifiger\\_Ausbau\\_der\\_A\\_7\\_Uebersichtskarte.pdf](https://www.strassenbau.niedersachsen.de/download/45795/Sechsstreifiger_Ausbau_der_A_7_Uebersichtskarte.pdf), abgerufen am 09. März 2018

---

|                                 |            |
|---------------------------------|------------|
| Anzahl Fahrspuren:              | 1          |
| Streckenlänge:                  | 1 km       |
| Simulationsdauer:               | 30 min     |
| Anzahl Fahrzeuge:               | 2          |
| max. zulässige Geschwindigkeit: | 100 km/h   |
| Trödelwahrscheinlichkeit:       | 0,1        |
| Verhalten bei Kollision:        | Totalstopp |

---

Tabelle 5: Szenariodata für Versuche zur Bestimmung von Zellgröße und Zeitschrittlänge

Das schnellere Fahrzeug wird in den Abbildungen 16 bis 23 jeweils durch die rote Kurve dargestellt, das langsamere durch die schwarze.

Für die Auswertung, ob eine Zellgröße/Zeitschrittlänge-Kombination für eine Simulationsdurchführung nutzbar ist, wurde der Geschwindigkeitsverlauf der beiden Fahrzeuge über die Simulationszeit genutzt.

**Zellgröße 7,5 m/Zeitschritt 0,1 min:** In allen drei Kurvenverläufen in Abbildung 16 sind mehrfach Reduktionen der Geschwindigkeit auf Null zu sehen. Dies zeigt, dass an diesen Stellen eine Kollision stattgefunden hat. Das Kollisionsergebnis wird vom Simulationstool ausgelöst, wenn ein Fahrzeug nicht in der Lage wäre, im nächsten Zeitschritt die für seine Geschwindigkeit entsprechende Streckenlänge zurückzulegen.

Die Zeitschrittlänge von 0,1 Minuten, was sechs Sekunden entspricht, genügt nicht, um Geschwindigkeit in ausreichendem Maße abzubauen, um einen hinreichend großen Abstand vom Vordermann einzuhalten.

Auf die Ausführung von Durchgängen mit dieser Zeitschrittgröße und kleinerer Zellgröße wurde verzichtet.

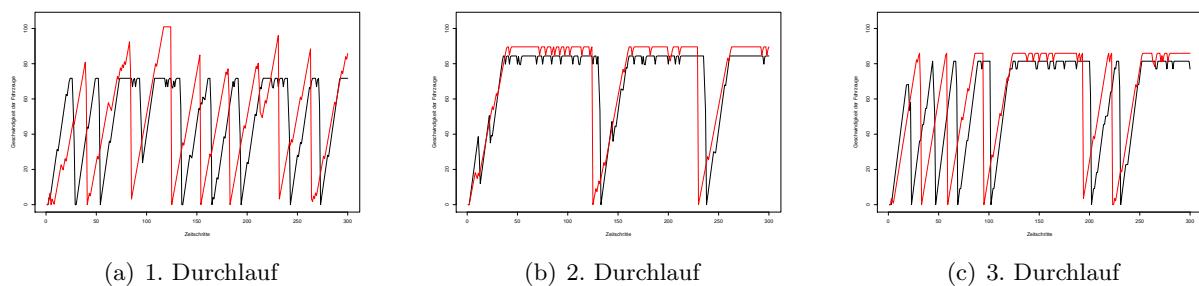


Abbildung 16: Simulationen mit Zellgröße 7,5 m und Zeitschrittlänge 0,1 min

**Zellgröße 7,5 m/Zeitschritt 0,05 min:** Die Verläufe der Kurven in Abbildung 17 zeigen in zwei von drei Fällen, dass sich die Geschwindigkeit des auffahrenden Fahrzeuges um die des langsameren Fahrzeugs einpendelt.

Im Plot des zweiten Durchlaufes ist zu sehen, dass auch in dieser Konstellation Kollisionen stattgefunden haben. Die Geschwindigkeit des langsameren Fahrzeuges lag in diesem Durchlauf im Vergleich zu den anderen beiden um einiges niedriger.

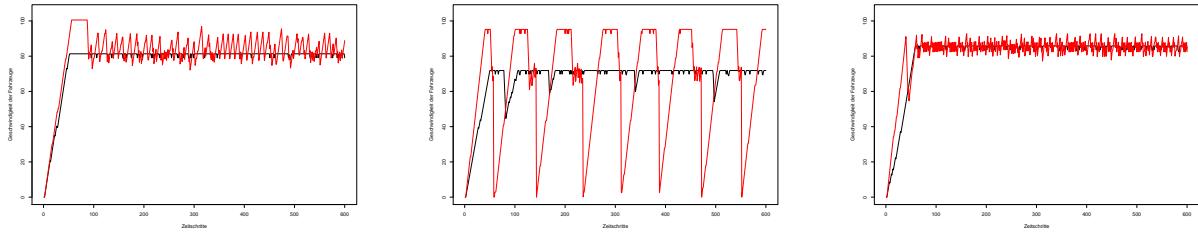


Abbildung 17: Simulationen mit Zellgröße 7,5 m und Zeitschrittlänge 0,05 min

**Zellgröße 7,5 m/Zeitschritt 0,025 min:** Die Geschwindigkeitskurve des schnelleren Fahrzeugs weist in Abbildung 18(a) ein sägezahnähnliches Muster auf. Dies resultiert aus den, im Vergleich zu den herbeigeführten Geschwindigkeitszuwachsen, großen Zellen. Ein Geschwindigkeitsunterschied von zehn bis 15 km/h reicht bei entsprechend zugrunde liegender Ausgangsgeschwindigkeit nicht aus, eine Verkürzung des Abstands herbeizuführen. Da sich die Abbrems- und Beschleunigungsmanöver nahezu identisch wiederholen, kommt es zu dieser Musterbildung. Das Plot in Abbildung 18(b) zeigt ein Einordnen hinter dem langsameren Fahrzeug und die Beibehaltung von dessen Geschwindigkeit. Abbildung 18(c) ist nahezu eine Mischform aus den ersten beiden Kurven.

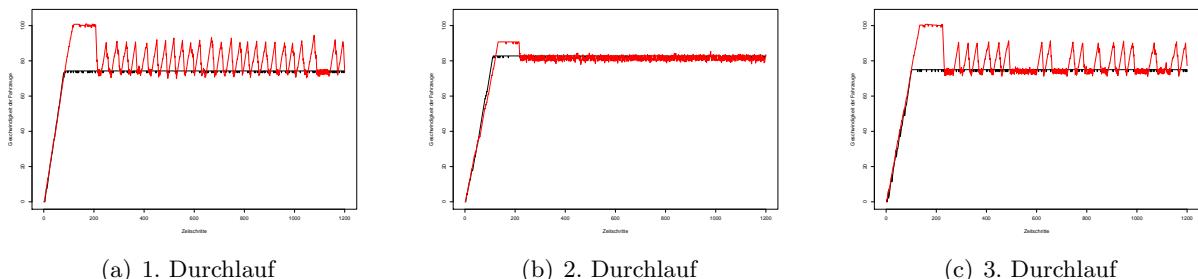


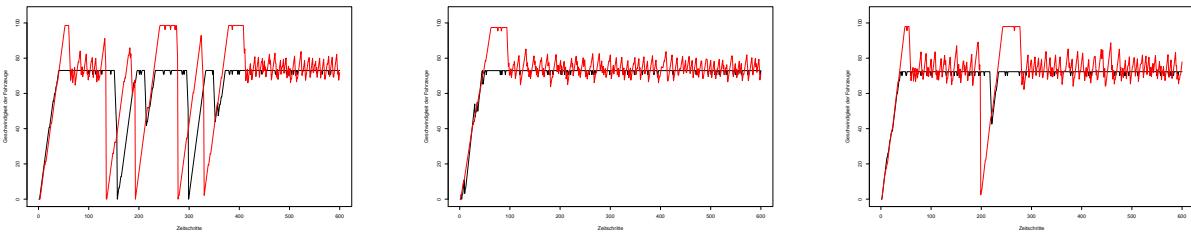
Abbildung 18: Simulationen mit Zellgröße 7,5 m und Zeitschrittlänge 0,025 min

**Zellgröße 5,0 m/Zeitschritt 0,05 min:** Die Diagramme in Abbildung 19 zeigen wiederholt, dass der Abstand, der für den Bremsvorgang zur Verfügung steht, mit einem recht langen Zeitraum zwischen den Zeitschritten, nicht ausreicht.

Allerdings ist im ersten und im dritten Durchlauf zu erkennen, dass das langsamere Fahrzeug für das gerade beschleunigende schnellere Fahrzeug abbremsen konnte, ohne dass es zu einer Kollision kommen muss.

**Zellgröße 5,0 m/Zeitschritt 0,025 min:** Die drei Durchläufe mit einer Zeitschrittlänge von 0,025 min ergaben ein uneinheitliches Bild. In Abbildung 20(a) konnte das schnellere Fahrzeug wieder mit höherer Geschwindigkeit hinter dem langsameren herfahren, scheinbar ohne dass ein Raumgewinn passierte, der zu einer Verzögerung geführt hätte.

Evtl. fuhr das schnellere Fahrzeug auch mit seiner max. möglichen Geschwindigkeit und der Geschwindigkeitsunterschied war so gering, das es innerhalb des Simulationszeitraumes das langsamere Fahrzeug nicht erreichte.



(a) 1. Durchlauf

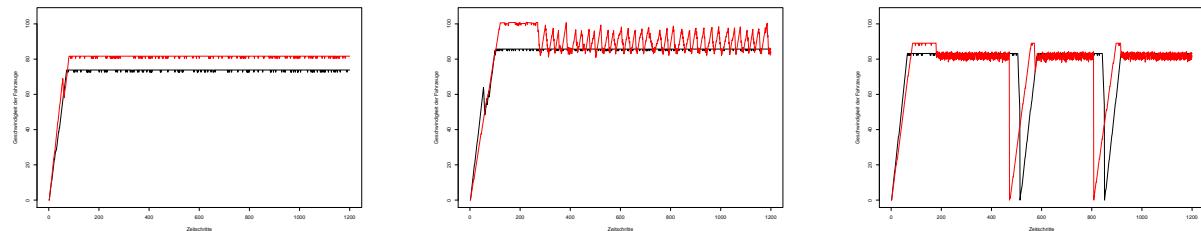
(b) 2. Durchlauf

(c) 3. Durchlauf

Abbildung 19: Simulationen mit Zellgröße 5,0 m und Zeitschrittlänge 0,05 min

Der Durchlauf im Diagramm in Abbildung 20(b) zeigt wieder ein sägezahnähnliches Muster, wie es mit den größeren Zelldimensionen bereits der Fall war.

Unerwarteterweise kam es im dritten Durchlauf zu je zwei Kollisionen. Besonders die jeweils auslösende Kollision des ursprünglich schnelleren Fahrzeuges war nicht zu erklären, da der Geschwindigkeitsunterschied in beiden Fällen minimal gewesen war.



(a) 1. Durchlauf

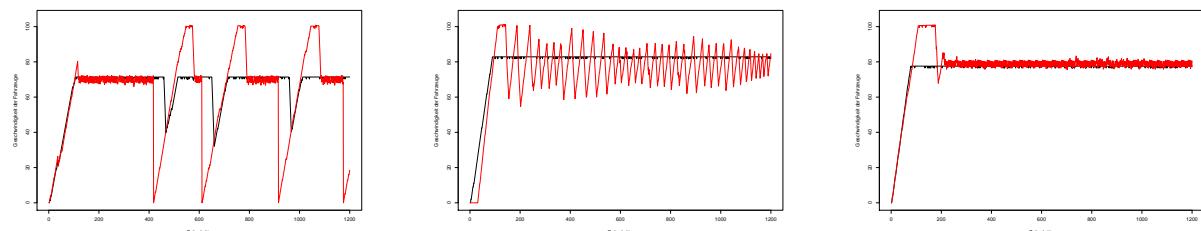
(b) 2. Durchlauf

(c) 3. Durchlauf

Abbildung 20: Simulationen mit Zellgröße 5,0 m und Zeitschrittlänge 0,025 min

**Verhaltensanpassung:** Ein weiterer Durchlauf brachte einen ähnlichen Ablauf hervor, siehe Abbildung 21(a). Da dabei die Ausgabe der Fahrzeugdaten im Kontrollfenster erfolgte, konnte der Grund für das Verhalten aufgeklärt werden. Die Bedingung im Agentenplan waren so formuliert, dass ein Fahrzeug dem vorausfahrenden folgen konnte ohne dass ein „Sicherheitsabstand“ gebildet wurde, wenn die Geschwindigkeit jenes Fahrzeugs nicht größer als die des vorausfahrenden Fahrzeugs war.

Somit genügte, unter gewissen Voraussetzungen, eine kleine Beschleunigung, ggf. bei Zusammentreffen mit einem Trödelvorgang des langsameren Fahrzeugs, um im nachfolgenden Zeitschritt die für die Geschwindigkeit erforderliche Wegstrecke nicht mehr umsetzen zu können.



(a) ursprüngliches Verhalten

(b) Modifikation 1

(c) Modifikation 2

Abbildung 21: Simulationen mit Veränderungen in den Agentenplänen

Eine erste Modifikation führte wieder zurück zu einem Verhalten, siehe Abbildung 21(b), aufgrund dessen die Geschwindigkeitskomponente mit in die Verzögerungslogik eingefügt worden war. Das nachfolgende Fahrzeug bremste so lange ab, bis der gewünschte Abstand erreicht war, um daraufhin mit einem „Zwischensprint“ wieder auf das vorausfahrende Fahrzeug aufzuschließen. Durch den Geschwindigkeitsüberschuss, der abgebaut werden muss, wurde der Abstand wieder extrem verlängert, dass der Kreislauf von vorn beginnt.

In einigen Simulationsläufen wurde die Geschwindigkeit sogar derart massiv abgebaut, dass das Folgefahrzeug fast zum Stillstand kam.

Es musste eine Möglichkeit gefunden werden, die Geschwindigkeit eines Fahrzeuges in Zusammenhang mit der Entfernung vom vorausfahrenden Fahrzeug zu bringen.

Ansätze, bei denen in Auswertungslogik Berechnungen mit vorliegenden Werten durchzuführen gewesen wären, wurden von der Agentensprache nicht unterstützt.

Da aber den Agenten sowohl die eigene Geschwindigkeit als auch die Entfernung als numerische Werte bekannt sind, werden diese nun zu einander in Beziehung gesetzt.

Die Geschwindigkeit des betrachteten Fahrzeugs muss nun gleichzeitig der jeweilige Mindestabstand vom Vordermann sein. Hier wurde zu einem späteren Zeitpunkt noch der Faktor 1,5 eingefügt.

Eine gleichwertige Bedingung wurde auch für die Simulation der Beschleunigung übernommen. Der reine Geschwindigkeitsvergleich wurde entfernt.

Aufgrund der Veränderung in den Agentenplänen wurde die erneut verringerte Zellgröße ebenfalls mit den beiden noch verbliebenen Zeitschrittewerten simuliert.

**Zellgröße 2,5 m/Zeitschritt 0,05 min:** Die ersten beiden Simulationen mit 2,5 m/0,05 min, siehe Abbildungen 22(a) und 22(b), zeigten ein Bild, welches nach den Veränderungen zu erwarten war. In der dritten Durchführung traten allerdings wieder Kollisionsereignisse auf.

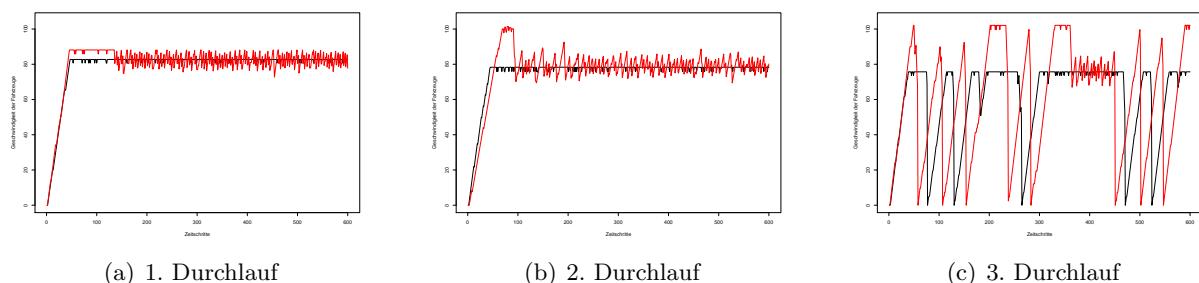
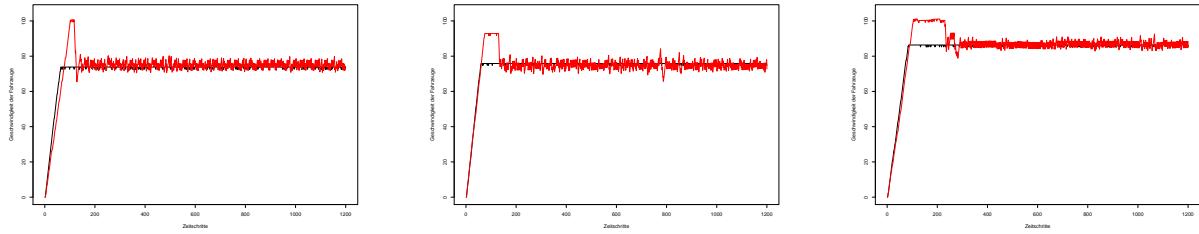


Abbildung 22: Simulationen mit Zellgröße 2,5 m und Zeitschrittlänge 0,05 min

**Zellgröße 2,5 m/Zeitschritt 0,025 min:** Die Simulationen der 2,5 m/0,025 min-Variante (Abbildung 23) zeigten ein, nach den bisherigen Erfahrungen, erwartetes und gewünschtes Verhalten. Das schnelle Fahrzeug konnte ohne Kollision hinter dem langsamen Fahrzeug abbremsen und sich dessen Geschwindigkeit anpassen.

Es wurden insgesamt zehn weitere Durchgänge mit diesem Setup durchgeführt. Dabei wurde zweimal je eine Kollision beobachtet, die aber immer im Bereich der Beschleunigungsphase am Anfang der Simulation stattfand.



(a) 1. Durchlauf

(b) 2. Durchlauf

(c) 3. Durchlauf

Abbildung 23: Simulationen mit Zellgröße 2,5 m und Zeitschrittlänge 0,025 min

**Zellgröße < 2,5 m:** Bei vorhergehenden Simulationsläufen außerhalb dieser Testreihe wurde festgestellt, dass es bei einer Zellgröße von weniger als 2,5 Meter zu Inkonsistenzen in den Beliebbases der Agenten kommt. Dies ist wahrscheinlich eine Folge der Parallelverarbeitung. Entscheidungen müssten dann aufgrund unterschiedlicher Voraussetzungen getroffen werden. Aus diesem Grund wurde auf den Einsatz von kleineren Zellgrößen verzichtet.

## 4.5 Testläufe des Einspurszenarios

Die Testläufe für den Einspurfall erfolgten mit den in Tabelle 6 aufgelisteten Einstellungen:

|                            |                               |
|----------------------------|-------------------------------|
| Simulationsdauer:          | 2 h                           |
| Anzahl Fahrzeuge:          | variabel, siehe Kapitel 4.4.1 |
| Anzahl Fahrspuren:         | 1                             |
| Streckenlänge:             | 7,5 km                        |
| Zellgröße:                 | 2,5 m                         |
| Zeitschrittlänge:          | 0,025 min (1,5 sek)           |
| Intervall Beschleunigung:  | [3, 5; 7]                     |
| Intervall Verzögerung:     | [8; 10]                       |
| Intervall Geschwindigkeit: | [80; 130]                     |
| max. Geschwindigkeit:      | 100 km/h                      |
| Verhalten bei Kollision:   | Abbremsen (kein Totalstopp)   |

Tabelle 6: Szenariodata für Einspur-Versuche

**Anpassung der Trödelwahrscheinlichkeit  $plinger$ :** Da es sich bei den Statistikwerten in Kapitel 4.4.1 um Tagesdurchschnitte handelt, darf angenommen werden, dass die daraus berechnete Verkehrsdichte für das Teilstück der A7 zwar eine hohe, aber noch zu bewältigende Verkehrsmenge ist. In der Simulation sollte es also möglich sein, fließenden Verkehr zu generieren. Der zuvor präferierte Wahrscheinlichkeitswert von 0,3 führte mit 34 Fahrzeugen allerdings mehrfach zu Stockungen und damit nicht zu einem ordentlichen Verkehrsfluss. Aus diesem Grund wurden mit einer auf eine Stunde verkürzten Simulationsdauer kleinere Werte (erst 0,1-, dann 0,025-Schritte) und Veränderungen am Szenario getestet.

Die folgenden Beobachtungen wurden dabei getätigt:

- Bei Beseitigung der Geschwindigkeitsunterschiede (Anpassung des Geschwindigkeitsintervalls) waren Simulationsläufe mit dem angedachten  $plinger$ -Wert möglich.

- 
- 2. Bei Deaktivierung des `linger`-Planes (unter Beibehaltung o.g. des Geschwindigkeitsintervall) kam es teilweise zu starken Abbremsvorgängen.

Durch die Inhomogenität der für die Fahrzeuge möglichen Geschwindigkeiten existiert immer (mind.) ein langsamstes Fahrzeug, auf das aufgelaufen wird. Die Bremsreaktionen der Agenten in der Simulation scheinen zu einem dem Resultat des Trödelns vergleichbaren Verkehrsbild zu führen.

- 3. Bei aktiviertem `linger`-Plan konnte erst bei einem  $p_{linger}$ -Wert ab 0,125 oder darunter fließender Verkehr simuliert werden.

Aufgrund dieser Feststellung werden die Testläufe mit einer Trödelwahrscheinlichkeit von 0,125 durchgeführt.

#### 4.5.1 Verkehrsmenge A7: 34 Fahrzeuge

Mit dem o.g. Wert gelangen mehrere Durchläufe, bei denen fließender Verkehr mit Fluktuation der Geschwindigkeiten der Fahrzeuge simuliert werden konnte. Die gefundenen Einstellungen scheinen das Verkehrsgeschehen abbilden zu können. Abbildung 24 zeigt die Diagramme eines Durchlaufes, bei dem es nicht zu sichtbaren Stockungen des Verkehrs kam. Andere Testläufe, siehe Abbildung A2, zeigen, dass das Agentenverhalten unter gleichen Voraussetzungen auch zu anderen Verkehrsbildern führen kann.

Es wurde weiterhin mit einstündigen Simulationen getestet, ab welcher Fahrzeuganzahl das Szenario ins Stocken gerät. Dabei wurde festgestellt, dass noch 2-3 weitere Fahrzeuge hinzugefügt werden können, ohne den Verkehrsfluss deutlich zu beeinträchtigen. Erst ab insgesamt 38 Fahrzeugen auf der Strecke bringen die Wechselwirkungen untereinander die entstehende Fahrzeugschlange zum Stoppen. (siehe Abbildung A3)

Dies entspricht einer Verkehrsdichte von 5,066 Fahrzeugen pro Kilometer (auf einer Spur) - 10,13 für zwei Spuren. Eine Erhöhung der Verkehrsdichte um ein Fahrzeug pro Kilometer macht demzufolge bereits einen Unterschied.

#### 4.5.2 Verkehrsmenge A38: 15 Fahrzeuge und Verkehrsmenge A71: 8 Fahrzeuge

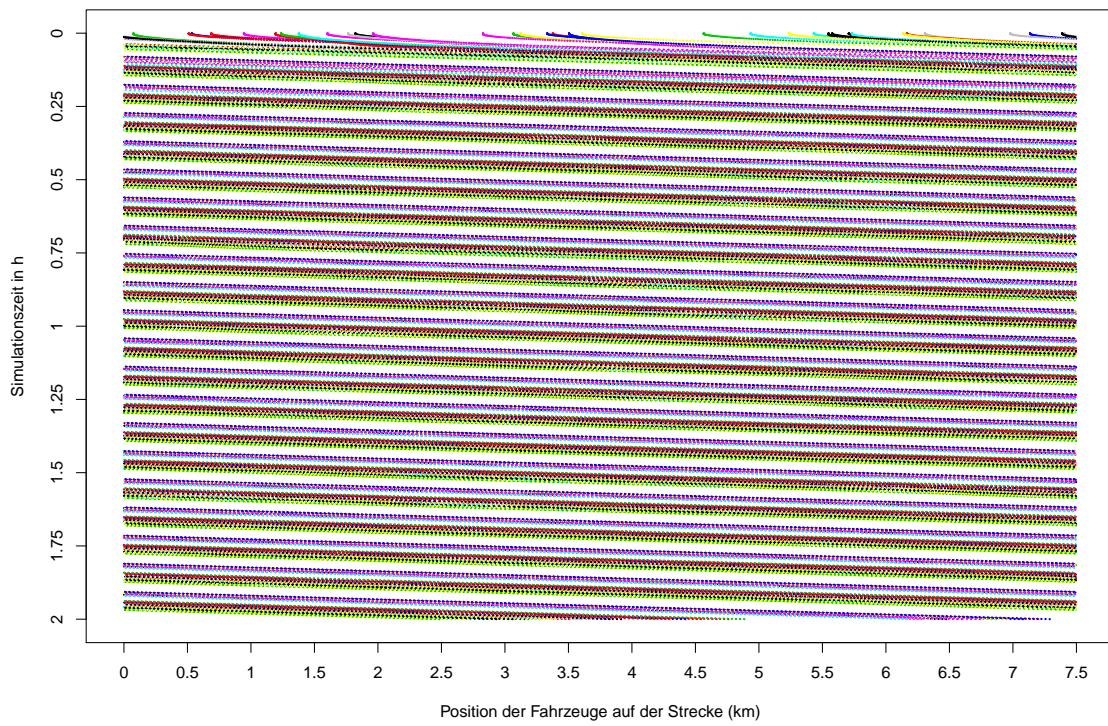
Da die Simulationsläufe mit der Verkehrsmenge der Messstelle auf der A7 zeigten, dass die Strecke mit den gewählten Einstellungen mehr als das Doppelte, bzw. Vierfache, an Verkehr verarbeiten konnte, waren aus den ausgeführten Durchläufen keine weiteren Erkenntnisse zu gewinnen.

In beiden Szenarien zeigte sich, dass sich schnellere Fahrzeuge mehrfach hinter jeweils langsameren Fahrzeugen „einordneten“, bis sich schließlich alle Fahrzeuge der Geschwindigkeit des langsamsten Fahrzeugs im System anpassten, siehe Abbildungen A4 und A5.

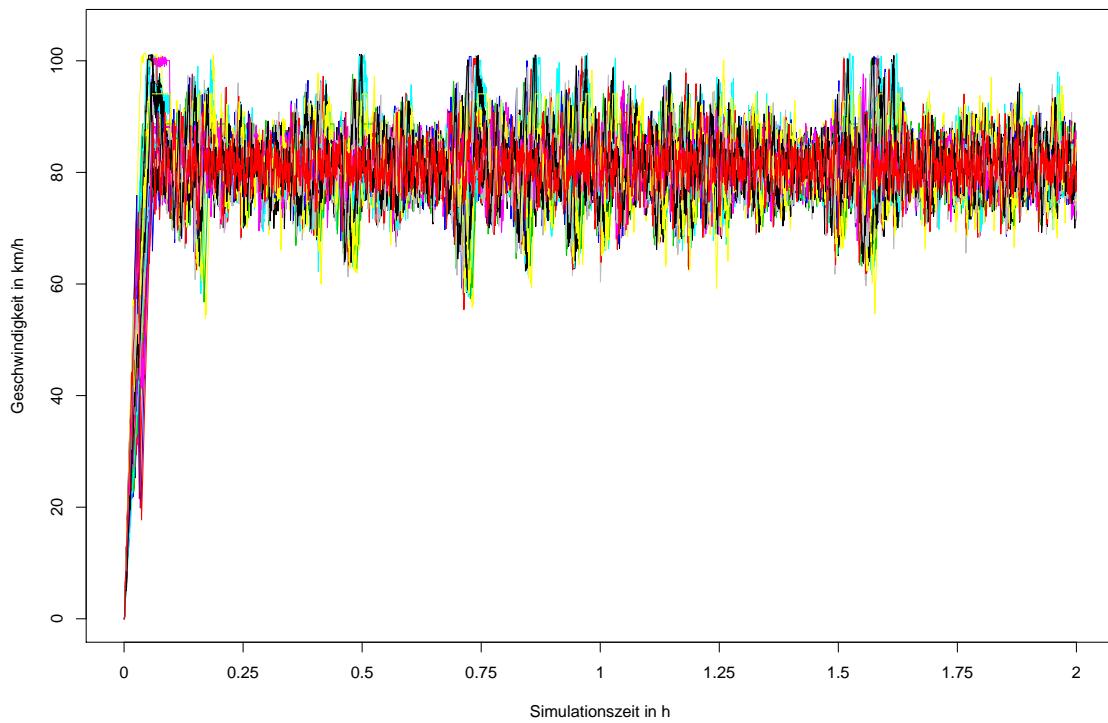
#### 4.5.3 Beobachtungen

Während bei der Entwicklung durchaus rückwärts laufende Stauwellen beobachtet werden konnten, siehe z.B. Abbildung A1, ist die Phase der Totalstopps, die dieses Verhalten hervorruft, bei der Simulation des fließenden Verkehrs im allgemeinen nicht vorhanden.

Dadurch dass die Bremsmanöver feiner ausgeführt werden, nähern sich die Positionen der Fahrzeuge einander an und fächern sich dann aber, nachdem die Ursache für das Abbremsen beseitigt ist, wieder auf.



(a) Fahrzeugpositionen



(b) Geschwindigkeiten

Abbildung 24: Testlauf: 34 Fahrzeuge, 7,5 km Strecke, 2 Stunden

---

In den Positionsplots bilden sich in den meisten Fällen weiße Streifen aus. Dies sind Stellen auf der Strecke, an denen sich kein Fahrzeug befindet.

## 4.6 Erkenntnisse aus Entwicklung und Simulationsläufen

Durch die einerseits real simulierten Beschleunigungs-, Verzögerungs- und Geschwindigkeitswerte, andererseits aber „grob“ unterteilten Zellgrößen und Zeitabschnitte kommt es in der Simulation zu Verhalten, welches durch diese Diskrepanz erklärbar ist.

### 4.6.1 Ursache Zellgröße

Für die folgenden Phänomene wurde die Größe der Zellen als Ursache ausgemacht.

**träges Anfahren:** Aufgrund der vergleichbar zum NaSch-Modell gewählten Zellgröße von 7,5 Metern, im Gegensatz dazu aber die Simulation realitätsnaher Beschleunigungswerte, kommt es am Beginn des Simulationslaufs mehrere Zeitschritte lang zu einem Stillstand der Fahrzeuge. Da jedes Fahrzeug softwareseitig in der Mitte der belegten Zelle platziert wird (bzw. im ersten Zeitschritt dorthin gesetzt wird) wird eine Geschwindigkeit von mehr als  $\frac{1}{2} \times 7,5 \frac{m}{s} = 13,5 \text{ km/h}$  benötigt, um eine sichtbare Bewegung durchzuführen. Durch die unterschiedliche Ausprägung der Beschleunigungswerte und möglicher Trödelphasen ist keine Nennung eines festen Zeitpunkts möglich, an dem dies vollzogen wird. Die Verkleinerung der Zellen schwächt dieses Verhalten ab. Im weiteren Verlauf der Simulation fällt es in der Weg-Zeit-Kurve nicht weiter auf, dass die Fahrzeuge an Zellen mit einer bestimmten Länge gebunden sind.

**schlechte Bremswirkung:** Die Simulationsplattform bietet durch realistisch modellierte Beschleunigungen und Verzögerungen den Vorteil, dass der Verkehr feingranularer abgebildet werden kann. Die analog zu den Simulationen von Nagel und Schreckenberg gewählte Zellgröße von 7,5 Metern führt allerdings dazu, dass theoretisch dargestellte Reduktionen der Geschwindigkeit nicht direkt in nächsten Zeitschritt umgesetzt werden. Vielmehr muss, ähnlich wie beim soeben beschriebenen Anfahren, ein gewisser Wert unterschritten werden, damit die gezeigte Bewegung auch real kürzer gesetzt wird.

Reicht der Platz nach vorn nicht aus, um das Fahrzeug entsprechend seiner Geschwindigkeit zu bewegen und in die freie Zelle hinter das vorausfahrende Fahrzeug zu setzen, wird ein Kollisionsereignis ausgelöst.

Die Reaktion darauf kann über einen Agentenplan gesteuert werden. Zu Beginn der Erarbeitung wurde die Kollision mit einem extremen Abbremsen modelliert, später dann als Totalstopp. Die Reduktion der Geschwindigkeit auf Null ist insofern vorteilhaft, weil der Extremausschlag nach unten in der Plotkurve sichtbar ist. Außerdem wird der Agent wirklich in der aktuellen Zelle gestoppt. Das Simulationstool würde sonst über (mehrere) Zeitschritte hinweg die Geschwindigkeit abbauen, ohne dass, bedingt durch vorausliegende Hindernisse, eine reale Bewegung stattfinden könnte.

Nachteilig ist zu sehen, dass ein Totalstillstand aufgrund der großen Geschwindigkeitsunterschiede meist weitere Kollisionen nach sich zieht.

Die Schwierigkeiten beim Bremsen können durch Reduktion der Zellgröße bei ansonsten gleichen Voraussetzungen verringert werden. Eine Simulation mit reduzierten Zellgrößen führte in diesem

Aspekt zu einem harmonischeren Verkehrsbild. Eine komplette Vermeidung der Kollisionsergebnisse ist nicht gelungen.

#### 4.6.2 Ursache Zeitschrittlänge

Für das nachfolgend beschriebene Verhalten ist hauptsächlich die Länge der Zeitschritte verantwortlich.

**Worst Case-Analyse für Mindestabstände:** Das Kollisionsergebnis wird nicht erst bei einer tatsächlichen Kollision ausgelöst, wenn sich also zwei Fahrzeuge in einer Zelle treffen, sondern bereits wenn der Platz nach vorn zum aktuellen Zeitpunkt nicht mehr zum Bewegen ausreichen würde. Hierbei spielt es für das Simulationstool keine Rolle, dass das Hindernis ggf. in Bewegung ist und sich von dieser Stelle entfernt.

Ein Verzögerungsvorgang - Abbremsen/Heranbremsen - muss also innerhalb einer kürzeren Strecke abgeschlossen werden, als dies in der Realität der Fall wäre.

In Tabelle 7 sind die Mindestabstände für eine kollisionsfreie Bewegung für eine Zellgröße vom 2,5 m und einer Zeitschrittgröße von 0,025 und 0,05 min aufgeführt. Zwischenschritte der Berechnung wurden jeweils aufgerundet.

Anzahl zu setzender Zellen:

$$x = \left\lceil \frac{\text{Geschwindigkeit}}{3,6} \times \frac{\text{Zeitschrittlänge}}{\text{Zellgröße}} \right\rceil$$

Mindestabstand:

$$s_{min} = \lceil x \times \text{Zellgröße} \rceil$$

Fasste man die ungerundeten Gleichungen zusammen, würde die Zellgröße aus der Gleichung entfernt.

| (Rest-) Geschwindigkeit | $s_{min}$ in m bei ts=0,025 min | $s_{min}$ in m bei ts=0,05 min |
|-------------------------|---------------------------------|--------------------------------|
| 130                     | 55                              | 110                            |
| 100                     | 43                              | 85                             |
| 80                      | 35                              | 68                             |
| 70                      | 30                              | 60                             |
| 60                      | 25                              | 50                             |
| 50                      | 23                              | 43                             |
| 30                      | 13                              | 25                             |
| Schritt                 | 3                               | 5                              |

Tabelle 7: Mindestabstände für Restgeschwindigkeit

Da die Zellgröße bei entsprechender Umstellung der obigen Gleichung keinerlei Einfluss auf die Berechnung hat, verändert eine Änderung des Wertes die Ergebnisse nicht grundlegend. Es wird lediglich auf ein Vielfaches der Zellgröße gerundet.

Durch die Entfernungen, die in einem längeren Zeitschritt zurückgelegt werden, erhöht sich durch eine Anhebung der Zeitschrittlänge auch der Mindestabstand entsprechend.

## 5 Überlegungen für die Multilane-Szenarios

Aufgrund zweier Fehler<sup>4</sup> im Agenten-Framework, die bis zum Termin dieser Arbeit nicht behoben wurden, war es nicht möglich, auch belastende Tests für die Mehrspurszenarien durchzuführen. Dennoch sind theoretische Überlegungen und in Teilen getestete praktische Vorarbeiten vorhanden, die in diesem Kapitel präsentiert werden sollen.

### 5.1 Veränderungen für die Mehrspurigkeit

Gegenüber der Einspurversion muss das Agentenverhalten um die Spurwechselkomponenten erweitert werden.

#### 5.1.1 Zustandsautomat

Für die Simulation mehrspuriger Fahrbahnen kann der ursprüngliche Zustandsautomat (Abbildung 8), der die Veränderung der Geschwindigkeit beinhaltet, unverändert bleiben. Das Spurwechselverhalten wird in einem separaten Automaten abgebildet, siehe Abbildung 25.

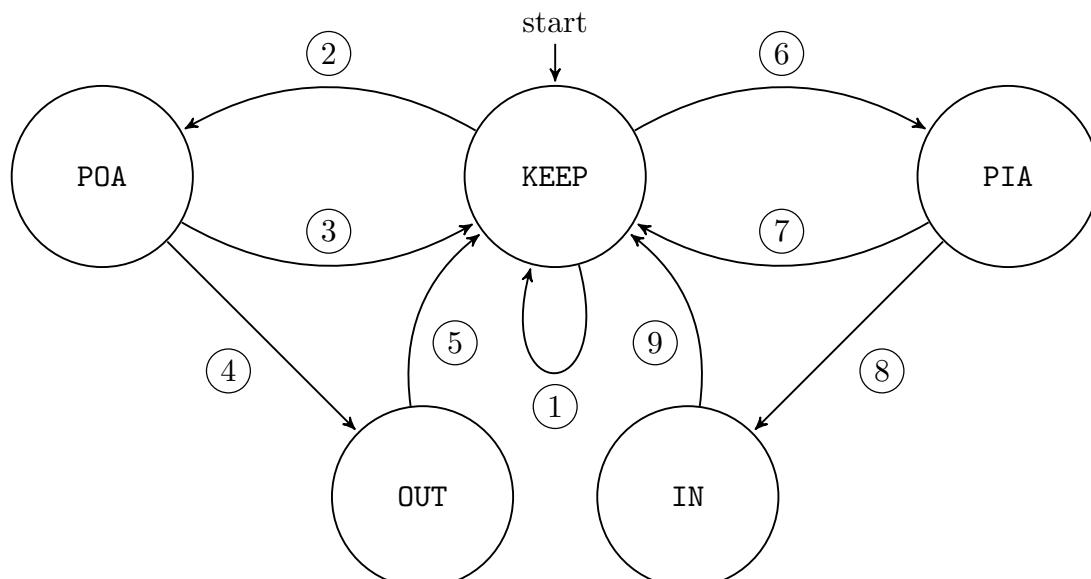


Abbildung 25: Zustandsautomat für das Agentenverhalten Spur halten/Spur wechseln

Der Startzustand des Automaten, KEEP (keep lane/Spur halten), steht für das Verhalten, die aktuelle Spur beizubehalten.

Aus diesem Zustand heraus sind die Zustände POA (Pull-out attempt/Ausscherversuch) und PIA (Pull-in attempt/Einscherversuch) zu erreichen. Ein Ausscherversuch kann z.B. unternommen werden, wenn das Fahrzeug auf ein anderes Fahrzeug aufläuft.

Ist keiner der Spurwechselversuche möglich, kann der Zustand KEEP über eine Schleife auch wieder selbst erreicht werden.

Sowohl aus POA, als auch aus PIA ist ein Übergang zurück zu KEEP möglich. Dieser Pfad wird genommen, wenn der jeweilige Versuch abgebrochen werden muss.

Für den Ausscherversuch könnte dies beispielsweise aufgrund von seitlichem Verkehr auf der Überholspur der Fall sein.

<sup>4</sup>siehe <https://github.com/LightJason/AgentSpeak/issues/47> und <https://github.com/LightJason/AgentSpeak/issues/50>

Sind die Voraussetzungen für einen sicheren Spurwechsel gegeben, erfolgt jeweils der Übergang in die Zustände OUT (pull out/Ausscheren) bzw. IN (pull in/Einscheren) und von dort aus wieder in den KEEP-Zustand.

Auch hier entspricht ein Zeitschritt der Simulation einem Zyklus vom Startzustand ausgehend bis zur Rückkehr dorthin.

### 5.1.2 Agentenverhalten

Die Veränderungen im Grundplan des Agentenverhaltens sind minimal. Es müssen lediglich zwei Ziele für das Ausscheren (Pull-out) und das Einscheren (Pull-in) hinzugefügt werden. Dies wird durch die jeweiligen Pläne ergänzt. Für das komplette Agentenscript siehe Listing A3.

Der Aufruf des jeweiligen Plans und das Abarbeiten der Bedingungen entspricht dabei dem Spurwechselversuch. Die Ausführung wird aus dem Agentenplan an die Simulationssoftware weiter geleitet.

**Listing 2:** Auszug aus Agentenscript: multi lane-Version

```

1 !cruise.
2
3 // --- start all other plans ---
4 +!cruise <-
5     !accelerate;
6     !decelerate;
7     !linger;
8     !pullout
9     !pullin
10    !cruise
11 .
12
13 // --- pull-out, change lane to overtake ---
14 +!pullout
15     : // --- Bedingungen gekuerzt ---
16     <- vehicle/pullout
17 .
18
19 // --- pull-in, change lane after overtake is finished ---
20 +!pullin
21     : // --- Bedingungen gekuerzt ---
22     <- vehicle/pullin
23 .

```

#### Ausscherverhalten im pullout-Plan

Der Plan für das Ausscheren ist mit insgesamt fünf Bedingungen versehen, siehe Listing A4.

1. Befindet sich das Fahrzeug nicht auf der am weitesten linken Fahrspur?
2. Ist in der aktuellen Spur Verkehr voraus?
3. Ist der Verkehr in der Spur, in die gewechselt werden soll, nicht näher?
4. Befindet sich kein Verkehr direkt links neben dem Fahrzeug in der Nachbarspur?
5. Ist der rückwärtige Verkehr auf der Zielspur nicht zu nah?

Die erste Bedingung ist zwingend notwendig, da sonst keine Spur zum hineinwechseln vorhanden wäre. Die Auslösung eines Kollisionsereignisses wäre in der Simulation die Folge.

Die zweite Bedingung macht den Ausschervorgang erst nötig. Wäre die aktuelle Spur nicht blockiert, könnte auf ihr weitergefahren werden.

Ein Spurwechsel ist nur sinnvoll, wenn auf der zukünftigen Spur mindestens genauso viel Raum nach vorn ist, wie auf der aktuellen. Dem trägt Bedingung 3 Rechnung.

Durch die Unterteilung des Sichtfeldes in vier Sektoren, siehe auch Kapitel 4.3.2, wurde die vierte

---

Bedingung notwendig.

Um den nachfolgenden Verkehr nicht zu behindern, ist die fünfte Regel zu beachten.

Durch die Fehler im Framework ist die gleichzeitige Auswertung aller Bedingungen unmöglich. Bei keiner der Bedingungen ist es möglich, diese wegzulassen oder mit einer anderen zusammenzufassen, ohne dass unerwünschtes Verhalten auftritt.

Würden zum Beispiel die Bedingungen 2 und 3 zusammengefasst und nur auf vorwärtigen Verkehr (in beiden Spuren) geprüft, so würde dies zwar der Bedingung „nicht schlechter“ für die Überholspur genügen, aber jedes überholende Fahrzeug würde zum Auslösen dieser Bedingung führen.

Eine Auslagerung der Bedingungen 1 und 2 in den *cruise*-Plan erscheint auch nicht sinnvoll, weil diese Kontrolle einen Zeitschritt vor Ausführung des Planes geschehen würde. Der Spurwechsel selbst würde dann mit noch einem Zeitschritt Verzögerung ausgeführt werden.

Außerdem könnte es zu Inkonsistenzen bzgl. der aktuellen Fahrspur kommen. So könnte der Agent in einem Zeitschritt den Spurwechsel ausführen, gleichzeitig aber nochmals den *pullout*-Plan anstoßen. Dort würde dann nicht wieder geprüft, in welcher Spur sich der Agent befindet. Somit würde ein weiteres Ausschermanöver ausgeführt, ohne dass eine Zielspur vorhanden wäre.

Ein „Workaround“ wäre, Verkehr auf der Zielspur generell auszuschließen. Dieser Plan würde aber nicht optimal arbeiten.

### **Einscherverhalten im pullin-Plan**

Im Plan für das Einscheren sind vier Bedingungen zu beachten, siehe auch Listing A5.

1. Befindet sich das Fahrzeug nicht auf der am weitesten rechten Fahrspur?
2. Ist in der Spur, in die gewechselt werden soll, genügend Abstand nach vorn?
3. Befindet sich kein Verkehr direkt rechts neben dem Fahrzeug in der Nachbarspur?
4. Ist der rückwärtige Verkehr auf der Zielspur ausreichend weit entfernt?

Wie auch beim Ausscheren muss sich in Orientierungsrichtung mindestens eine weitere Spur befinden, da es sonst zum Auslösen des Kollisionseignisses durch die Simulationssoftware kommt. Hierfür sorgt die erste Bedingung.

Bedingung 2 sorgt dafür, dass das Fahrzeug selbst einige Zeit freie Fahrt auf der Zielspur hat. Die Bedingungen 3 und 4 bewirken, dass kein anderer Verkehr behindert wird.

Auch die Ausführung dieses Planes ist durch die Fehler im Agenten-Framework betroffen. Allerdings gibt es einen „Workaround“, der aber kein optimales Verhalten liefern kann.

Das kollisionsfreie Wiedereinordnen kann garantiert werden, wenn sich keinerlei Verkehr im Sichtfeld des Fahrzeuges befindet. Dies ist jedoch bei Simulationen mit vielen Fahrzeugen nahezu unmöglich.

## **5.2 Testläufe des Mehrspurmodells**

Da die erdachte Logik nicht funktionierte, wurde überlegt, mit welchen Alternativen bzw. Workarounds dennoch ein Teil des Verhaltes getestet werden könnte. Die o.a. Alternativen wurden jew. in einem separaten Plan umgesetzt. Belastende Tests konnten mit diesen allerdings nicht durchgeführt werden.

Bereits bei drei Fahrzeugen im System kam es zu Situationen, bei denen das Einscheren ausblieb.

Es konnte aber zumindest in einem Szenario mit zwei Fahrzeugen gezeigt werden, dass ein schnelles Fahrzeug durch Wechsel auf eine andere Fahrspur ein anderes Fahrzeug überholen kann, siehe Abbildung 26.

Das Diagramm der Spurnutzung für das schnellere Fahrzeug (Abbildung 26(a)) zeigt deutlich, dass das Manöver kurz nach der 0,5 h-Marke durchgeführt wird. In der Geschwindigkeitskurve (Abbildung 26(b)) ist dabei nur ein minimaler Geschwindigkeitsverlust zu erkennen. Dies liegt daran, dass sowohl mit dem Überholvorgang, als auch dem Abbremsen bei der gleichen Entfernung begonnen werden sollte. Im Positionsplot (Abbildung 26(c)) kommt es zur gleichen Zeit zu einer Überlappung der beiden Punktlinien.

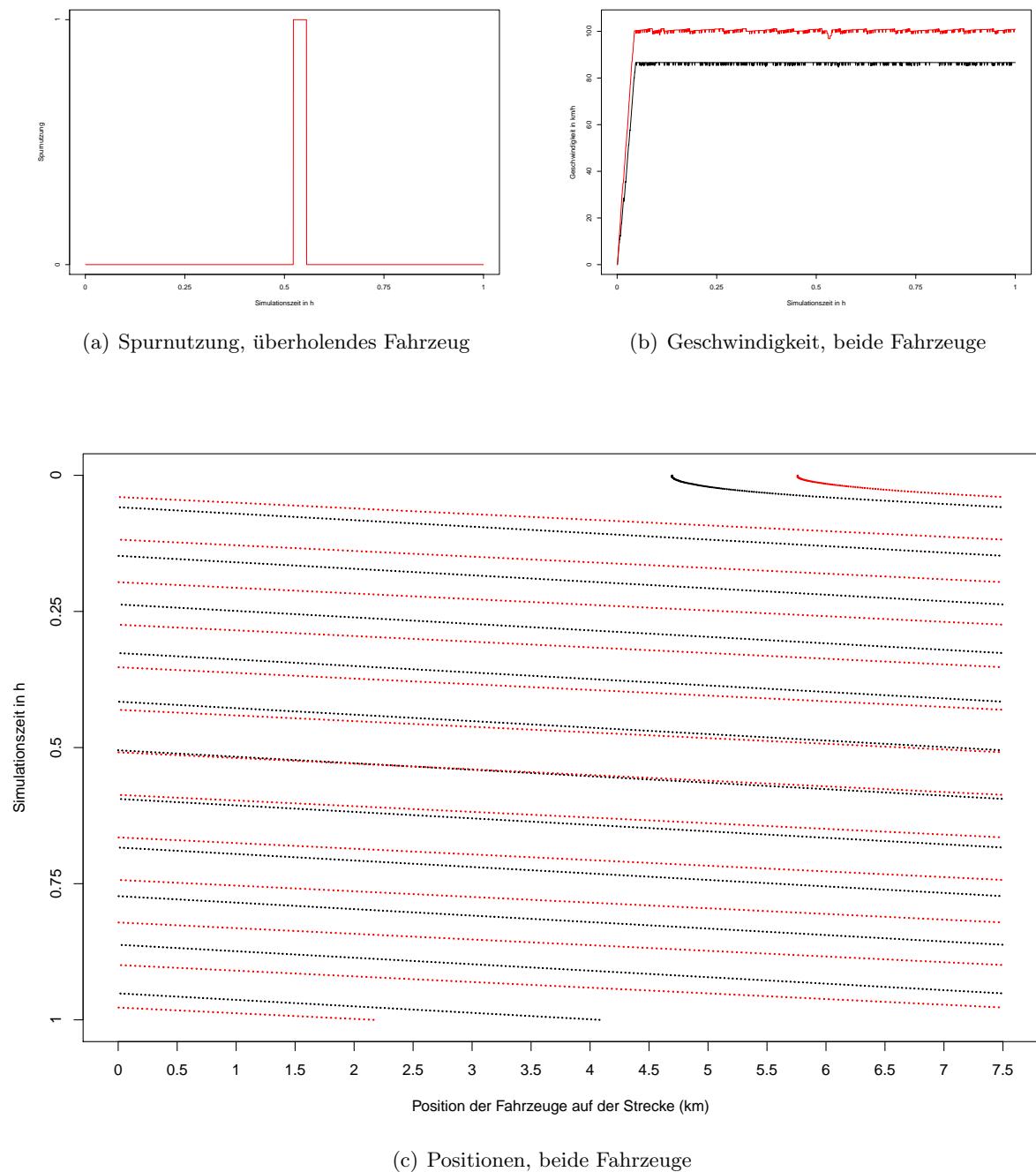


Abbildung 26: Überholen im Multilane-Szenario

## 5.3 Zusätzliche Erweiterungsmöglichkeiten

Für das Agentenverhalten im Mehrspurszenario waren zusätzliche Erweiterungen angedacht, die unregelmäßiges Verhalten der Fahrzeuge simulieren könnten.

### 5.3.1 Zufälligkeit des Aus- und Einscherens

Um unterschiedliche Verhaltensweisen beim Überholen im Autobahnverkehr abbilden zu können, benötigt man einen Mechanismus, Wahrscheinlichkeiten für jew. eine der Handlungen zu generieren.

Dieses wurde z.B. in [Tra17] anhand des „Social-Force-Vehicle-Modells“ versucht. Der dort erarbeitete Ansatz scheint jedoch das gewünschte Verhalten übermäßig zu bevorzugen.

Für den Anreiz zum Spurwechsel nach links (Pull-out) oder rechts (Pull-in) genügt im Zweispurfall die Berechnung der jeweiligen Wahrscheinlichkeit. Die Wahrscheinlichkeit, die Spur zu halten, wäre jeweils die Gegenwahrscheinlichkeit.

Für Szenarien mit mehr als zwei Spuren ist eine Exklusivität (XOR) der beiden Ereignisse zu gewährleisten. Hier könnte die Wahl auf die Alternative mit der größeren Wahrscheinlichkeit fallen bei exakter Gleichheit würde es Sinn machen, aufgrund des zu überholenden Verkehrs und des Verbotes des rechts Überholens, den Spurwechsel nach links zu präferieren.

#### Wahrscheinlichkeitsfunktion

Für eine Bestimmung von Wahrscheinlichkeiten für diesen Zweck erscheint die logistische Funktion

$$f(x) = \frac{1}{1+e^{-x}}$$

siehe [Roo], eine Sigmoid-Funktion (auch S- oder Schwanenhalsfunktion), ein geeigneter Kandidat. Diese wird gewöhnlich zur Simulierung von Wachstumsprozessen eingesetzt. Sie ist für alle Zahlen in  $\mathbb{R}$  definiert und ihr Wertebereich liegt zwischen 0 und 1.

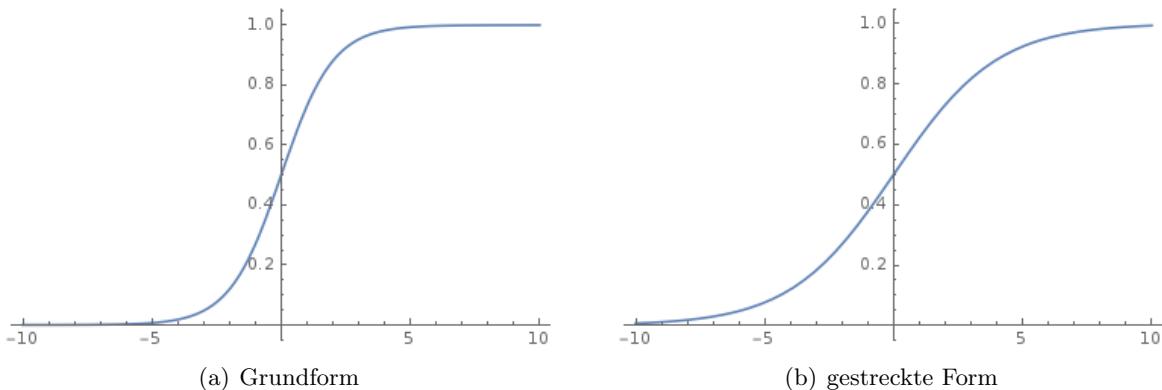


Abbildung 27: Kurvenbeispiele logistischer Funktionen im identischen Intervall

Die Grundfunktion (Abbildung 27(a)) steigt im Intervall  $[-5; 5]$  ziemlich rasch von nahe 0 auf nahe 1. Dies kann durch Streckung der Funktion etwas abgemildert werden. So verläuft z.B.

$$f(x) = \frac{1}{1+e^{-\frac{1}{2}x}}$$

weniger steil (Abbildung 27(b)) und hat den für diesen Zweck sinnvollen Intervall  $[-10; 10]$ . Für Werte außerhalb dieses Intervallbereichs ergibt sich jeweils entweder eine kleine von Null, bzw. große von Eins verschiedene Wahrscheinlichkeit.

## Wahrscheinlichkeit des Ausscherens (Pull-out)

Für das Ausscheren wurden die folgende Zusammenhänge, bezogen auf ein vorausfahrendes Fahrzeug, erkannt:

- Abstand  $\searrow \Rightarrow$  Ausscherwunsch  $\nearrow$  (und umgekehrt)
- relative Geschwindigkeit  $\nearrow \Rightarrow$  Ausscherwunsch  $\nearrow$  (und umgekehrt)

Es galt sinnvolle Werte zu finden und diese auf den o.g. Intervall zu normieren.

Für den Abstand  $s$  wurde die Grenze der Zone der Nahortientierung/Handlung, siehe Kapitel 4.2.1 - 110 m - und ein Punkt, an dem nach Möglichkeit spätestens ein Überholvorgang eingeleitet sein sollte - 50 m - gewählt.

Dies führt zu der Funktion

$$f(s) = -\frac{s}{3} + \frac{80}{3}$$

Somit ergibt sich für die Überholwahrscheinlichkeit in Abhängigkeit vom Abstand zum vorausfahrenden Fahrzeug  $p_{po}(s)$  die Funktion

$$p_{po}(s) = \frac{1}{1+e^{-\frac{1}{2}f(s)}}$$

Für die relative Geschwindigkeit  $v_{rel}$  wurden die Grenzen bei 0 und 25 km/h festgelegt. Ab 0 km/h bzw. darunter ist kein Überholen mehr nötig. Für die obere Grenze wurde das Verhalten auf Landstraßen, siehe [ADA11, S. 27], auf den Autobahnverkehr übertragen.

Dies führt zu der Funktion

$$f(v_{rel}) = \frac{4v_{rel}}{5} - 10$$

Für die Überholwahrscheinlichkeit in Abhängigkeit von der relativen Geschwindigkeit zum vorausfahrenden Fahrzeug  $p_{po}(v_{rel})$  ergibt sich

$$p_{po}(v_{rel}) = \frac{1}{1+e^{-\frac{1}{2}f(v_{rel})}}$$

Eine Kombination der beiden Gleichungen und Normierung auf den Intervall  $[0; 1]$

$$p_{po} = \frac{1}{2}(p_{po}(s) + p_{po}(v_{rel})) = \frac{1}{2}p_{po}(s) + \frac{1}{2}p_{po}(v_{rel})$$

liefert die Wahrscheinlichkeit für das Ausscheren  $p_{po}$ . Hier wären die beiden Wahrscheinlichkeiten gleich gewichtet (Faktor  $\frac{1}{2}$ ). Eine unterschiedliche Gewichtung kann durch Faktoren, welche in Summe 1 ergeben müssen, herbeigeführt werden.

Es ergibt sich eine Funktionskurve mit der Ausbildung von drei erkennbaren Plateaus um die Wahrscheinlichkeitswerte 0, 0,5 und 1, siehe Abbildung 28.

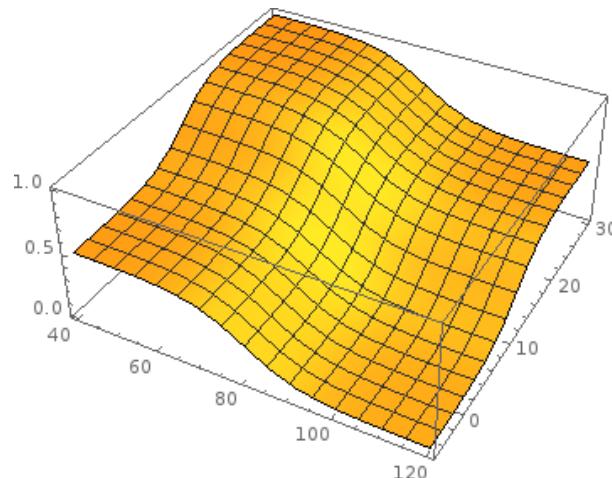


Abbildung 28: Plateaubildung der Funktionskurve der Ausscherwahrscheinlichkeit  
(für  $p_{po} \in [0; 1]$ ,  $s \in [40; 120]$ ,  $v_{rel} \in [-5; 30]$ )

## Wahrscheinlichkeit des Einscherens (Pull-in)

Eine ähnliche Herangehensweise ist für das Einscheren möglich. Hier müssten allerdings sowohl der vorausfahrende Verkehr  $T_f$ , als auch der rückwärtige Verkehr  $T_b$  beachtet werden. Aufgrund der fehlenden Möglichkeit eines Tests, wird hier nur das Konzept umrissen.

- Abstand  $T_f \nearrow \Rightarrow$  Einscherdrang  $\nearrow$  (und umgekehrt)
- relative Geschwindigkeit  $T_f \searrow \Rightarrow$  Einscherdrang  $\nearrow$  (und umgekehrt)
- Abstand  $T_b \nearrow \Rightarrow$  Einscherdrang  $\nearrow$  (und umgekehrt)
- relative Geschwindigkeit  $T_b \nearrow \Rightarrow$  Einscherdrang  $\nearrow$  (und umgekehrt)

Beim Zusammenfassen der entstehenden einzelnen Wahrscheinlichkeiten kann ebenfalls wieder über Faktoren eine individuelle Gewichtung vorgenommen werden.

### 5.3.2 Dosiertes Abbremsen

In den zugrunde liegenden Tests wurde das Abbremsen mit maximal möglicher Intensität ausgeführt. Denkbar wäre auch hier die Verwendung der logistischen Funktion, um die Bremsstärke zu dosieren. Ähnlich wie beim Ausscheren scheinen Abstand und relative Geschwindigkeit als zu beachtende Größen sinnvoll.

### 5.3.3 Vermeidung des Rechts-Überholens

Um Fahrzeuge in der Simulation vom Überholen auf der rechten Seite abzuhalten ist nur eine einfache Bedingung im `decelerate`-Plan notwendig, siehe Listing 3.

Listing 3: Auszug aus Agentenscript: Überholverbot rechts

```
1 !decelerate
2 // --- gekürzt ---
3   : >>( view/vehicle( _, data( _, static( lane( Lane ), cell( Cell ),
4     speed( Speed ), distance( Dist ), direction( Dir ) ) ) ),
5     bool/equal( generic/type/tostring( Dir ), "forward[]" )
6     && math/floor( Lane ) > CurrentLane
7     && Speed < CurrentSpeed
8     && Dist < 1.5*CurrentSpeed
9   ) <-
10   CurrentSpeed > 80 ? vehicle/decelerate(0.9) : vehicle/decelerate
11   (0.5);
12 .
```

Je weiter links eine Fahrspur ist, desto größer ist ihre Id. Befindet sich also im Sichtfeld von Fahrzeug A ein Fahrzeug B auf einer Fahrspur mit größerer Id, so wird von Fahrzeug A ein Bremsmanöver eingeleitet.

Weitere Einschränkungen ergeben sich durch die Hinzunahme einer Geschwindigkeits- und einer Abstandskomponente.

Außerdem ist eine Erweiterung im `accelerate`-Plan erforderlich, dass eine Beschleunigung auch bei vorausfahrendem Verkehr auf Spuren mit größerer Id ausbleiben kann.

Das Verhalten, was diese Bedingung auslöst, konnte getestet werden, indem ein langsameres Fahrzeug auf der Überholspur fuhr und das schnellere auf der Normalspur aufschloss. Das schnellere Fahrzeug bremste ab und passte sich der Geschwindigkeit des langsameren Fahrzeugs an.

## 6 Fazit

Anders als bei den Simulationen zur ursprünglichen Arbeit von Nagel und Schreckenberg [NS92] findet bei den Simulationen mit dem Agentenkonzept die Interaktionen zwischen den Agenten/Fahrzeugen ausschließlich aufgrund von Größen wie Abstand, (relativer) Geschwindigkeit und relativer Position statt. Dies ist vergleichbar zur Situation im realen Straßenverkehr.

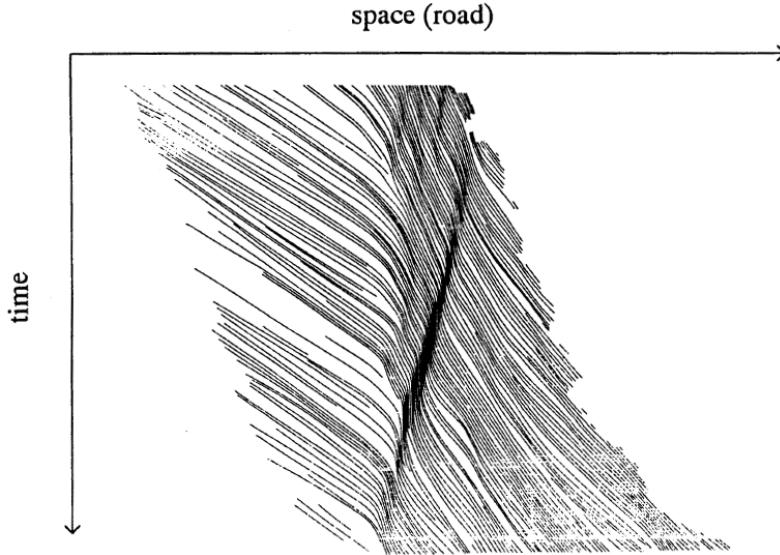


Abbildung 29: Fahrzeugbewegung in Weg-Zeit-Linien aus Luftaufnahmen, aus [NS92, Fig. 3]

Trotz dieser Veränderungen ergeben sich z.B. in der zweiten Hälfte der Simulation in Abbildung A1 vergleichbare Muster zu Weg-Zeit-Linien von Fahrzeugen, siehe Abbildung 29, die aus Luftaufnahmen gewonnen wurden.

Die Gridumgebung dient ausschließlich als Darstellung der Straße.

Die Fahrspuren sind, auch bei Mehrspurigkeit, anders als bei [YCLR12], keine miteinander kommunizierenden Automaten, sondern die Zellen darauf „Verankerungspunkte“ für die Fahrzeuge. Zwischen den Fahrspuren kann hin und her gewechselt werden. Die Verantwortung für die Einhaltung von Vorgaben liegt bei jedem Agenten.

Die Zellgröße wurde gegenüber dem ursprünglichen Nagel-Schreckenberg-Modell verkleinert. Eine weitere Verkleinerung scheint möglich, wenn die Inkonsistenzen in den Beliefbases abgestellt werden können.

Die Kollisionsfreiheit des Originalmodells konnte bei zu großen Geschwindigkeitsunterschieden nicht dargestellt werden. Dadurch dass die Auslösung des Kollisionseignisses bereits bei Unterschreitung eines Mindestabstandes stattfindet, der abhängig von der aktuell gefahrenen Geschwindigkeit ist, gelang es ebenfalls nicht, eine realistische Unfallhäufigkeit zu erreichen. Ebenso wurde die Möglichkeit einer engen Fahrzeugfolge dadurch beeinflusst.

Bei der Verkehrsfluss/Verkehrsdichte-Ebene des Fundamentaldiagrammes gelang nur die Darstellung des aufsteigenden Astes, die frei fließenden Verkehr bedeutet.

### 6.1 Ausblick

Durch eine weitere Reduzierung der Größe der Gridzellen und Verkleinerung der Zeitschritte ist die Diskretisierung der Simulationsumgebung möglich. Problematisch ist hier allerdings die aktuelle Entstehung von Inkonsistenzen in den Entscheidungsgrundlagen der Agenten zu sehen.

---

Durch die Verkleinerung der Zellgröße wird es nötig sein, dass für Fahrzeuge bei der Initialisierung eine Fahrzeuglänge generiert wird. Diese würden dann mehrere dieser dann kleineren Zellen überdecken. Folge wäre ein diversifizierteres Fahrzeugbild (Fahrzeugklassen), das auch in Sachen Platzbedarf realistisch dargestellt werden kann.

Für die Zukunft muss ein Weg gefunden werden, die Fahrzeuge innerhalb real möglicher Abstände so abzubremsen, dass keine Kollisionen entstehen.

Wenn es die Weiterentwicklung von „LightJason“ zulässt, sollte mit den in Kapitel 5 getätigten Überlegungen geprüft werden, ob die Mehrspurversion des Agentenscriptes unter Belastung eingesetzt werden kann.

Das Agentenmodell selbst kann ebenfalls noch um Komponenten der Realität erweitert werden.

## Fahrertypen

Im Jahr 2015 wurde bei einer Studie der London School of Economics and Political Science (LSE) im Auftrag des Reifenherstellers Goodyear, siehe [Lon], festgestellt, dass man Autofahrer in sieben Typen einteilen kann.

- „the teacher“, der Lehrer: möchte, dass andere Fahrer wissen, was sie falsch gemacht haben und erwartet Anerkennung für seine Anstrengungen andere zu belehren
- „the Know-it-all“, der Besserwisser: denkt, dass er von inkompetenten Schwachköpfen umgeben ist und schreit diese gelegentlich auch schon einmal an, während er sicher in seinem Auto sitzt
- „the Competitor“, der Wettkämpfer: muss vor allen anderen Fahrern sein und reagiert verärgert, wenn sich diesem Ziel jemand entgegen stellt, beschleunigt evtl. falls jemand zu überholen versucht oder schließt die Lücke um ein Einfädeln unmöglich zu machen
- „the Punisher“, der Bestrafer: möchte jeden anderen Fahrer für gefühltes Fehlverhalten bestrafen, steigt ggf. aus dem Auto aus, um andere Fahrer direkt zu konfrontieren
- „the Philosopher“, der Philosophische: akzeptiert Fehlverhalten problemlos und versucht dieses rational zu erklären, versteht es seine Gefühle im Auto zu kontrollieren
- „the Avoider“, der Vermeider: behandelt Fahrer, die sich daneben benehmen, unpersönlich, tut sie als Gefahr ab
- „the Escapee“, der Flüchter: hört Musik oder spricht am Telefon, um sich zu isolieren, lenkt sich mit ausgewählten sozialen Beziehungen ab, um sich nicht mit anderen Fahrern auf der Straße beschäftigen zu müssen, in erster Linie ist es aber eine Strategie, das Aufkommen von Frust zu vermeiden

Für zukünftige Entwicklungen der Simulation könnte man für diese oder ähnliche Fahrertypen Profile ausarbeiten und anlegen, die sich auf Abstands-, Beschleunigungs- und Verzögerungsverhalten sowie die Risikobereitschaft beim Aus- und Einscheren auswirken. So könnte noch eine weitere Komponente mit Hinsicht auf Realitätsnähe geschaffen werden.

## Stresslevel

Weiterhin könnte „Hupen“ als Broadcast- oder Message-Ereignis festgelegt werden. Dies oder auch wiederholtes zu dichtes Auffahren und „geschnitten werden“, könnten sich auf eine Art Stresslevel auswirken, das als interner Belief im Agenten hinterlegt wird.

Das Stresslevel wiederum könnte die Art und Weise verändern, wie das jeweilige Fahrzeug am

---

Straßenverkehr teilnimmt. Ggf. auch in unterschiedlicher Art und Weise, je nach Fahrertyp.  
Über eine gewisse Zeit, baut sich erworbbener Stress wieder ab.

### Außenwahrnehmung der Fahrzeuge

Die Außenwahrnehmung eines Fahrzeuges könnte durch einen öffentlich sichtbaren Belief gesteuert werden.

Ist ein Fahrer betrunken, fällt dies anderen Fahrern normalerweise u.a. aufgrund der Fahrweise auf. Hier könnte z.B. der Hinweis, dass der Fahrer betrunken ist, andere Fahrzeuge vorsichtiger handeln lassen.

Ein weiterer Aspekt könnte die Fahrtrichtungsanzeige sein. Blinkt ein Fahrzeug, um einen Spurwechsel anzudeuten, wird ein nachfolgender Fahrer evtl. nicht weiter beschleunigen.

## 6.2 Das NaSch-Modell in der Zukunft

Wird das Nagel-Schreckenberg-Modell zukünftig noch von Bedeutung sein? Viele Entwicklungen im Automobilbereich gehen aktuell in Richtung hochassistentes oder gar autonomes Fahren. Wie wird man diese Art der individuellen Fortbewegung simulieren können?

Laut [SAE] kann neben dem manuellen Fahren in fünf Stufen der Automation unterschieden werden. Zur Zeit befindet man sich am Übergang von Stufe 2 zu Stufe 3, ab der die Umwelt automatisiert überwacht wird. Dies findet durch sog. Assistenzsysteme statt. Diese können den Fahrer unterstützen - Notbremsassistenten, Spurhalteassistenten - entbinden diesen aber nicht von seiner eigenen Sorgfaltspflicht. Zudem ist etwa ein Abbremsen eines Assistenzsystems abrupt und wesentlich stärker als das eines Menschen. Somit käme man wesentlich eher an den Punkt, an dem ein Fahrzeug so stark abbremsen muss, dass es still steht.

Bei aktuell erhältlichen, serienmäßigen Fahrzeugen ist - zumindest durch die Gesetzgebung - immer noch ein menschlicher Fahrer nötig. Selbst die Firma Tesla, die das eigene „Autopilot“-System auf ihrer Webseite mit „Full Self-Driving Capability“/„Volles Potenzial für autonomes Fahren“ anpreist (vgl. [Tesb] und [Tesa]), verweist in der eigenen Internetpräsentation und bei Reaktionen auf Unfälle mit ihren Fahrzeugen darauf, dass der Fahrer selbst auf die Straße achten muss. [VD]

Sollten diese Systeme in der Zukunft zu 100% sicher funktionieren und gesetzlich erlaubt werden, kann man den Faktor Mensch, der meist die Unsicherheiten in das System Verkehr einbringt, aus der Gleichung entfernen. Von diesem Punkt ist die Technik aktuell noch ein Stück weit entfernt (vgl. [HR]), sodass der Mensch hinter dem Lenkrad wohl noch einige Zeit erhalten bleiben wird ... und sei es als Backup bei Fehleinschätzungen der Technik. Somit hat das Nagel-Schreckenberg-Modell auch zukünftig seine Berechtigung.

Und selbst im Falle der vollständigen Automation gelten für die Fahrzeuge die gleichen physikalischen Voraussetzungen. Da die Straßenfahrzeuge nicht untereinander gekoppelt werden, sind gewisse Verzögerungen und Abweichungen der Reaktionen nach wie vor vorhanden.

Wichtig wäre die Kommunikation der Fahrzeuge untereinander. Dann könnten Verzögerungen durch Reaktionszeiten ggf. noch weiter gesenkt werden.

---

# Anhang

## Literatur

- [ADA] ADAC e.V. Formeln für die Theorieprüfung. Webseite. <https://www.jungesportal.de/fuehrerschein/faustformeln-fuer-die-theorie.php>; abgerufen am 25. Februar 2018.
- [ADA11] ADAC e.V. Überholen auf Landstraßen - Motive, Nutzen und Risiken von Überholmanövern. PDF, 2011. [https://www.adac.de/\\_mmm/pdf/fachdossier-ueberholen\\_auf\\_landstra%C3%9Fen\\_68414.pdf](https://www.adac.de/_mmm/pdf/fachdossier-ueberholen_auf_landstra%C3%9Fen_68414.pdf); abgerufen am 07. März 2018.
- [ADK] Malte Aschermann, Sophie Dennisen, and Philipp Kraus. Lightjason. Webseite. <http://www.lightjason.org>; abgerufen am 01. März 2018.
- [AKM16] Malte Aschermann, Philipp Kraus, and Jörg P. Müller. LightJason: A BDI Framework Inspired by Jason. *IfI Technical Report Series*, IfI-16-04, 2016.
- [Bec03] Hans-Georg Beckmann. Zelluläre Automaten. PDF, 2003. VLIN - Virtuelle Lehrerfortbildung im Fach Informatik in Niedersachsen, <http://www.vlin.de/material/ZAutomaten.pdf>; abgerufen am 03. November 2017.
- [BM09] Heinz Burg and Andreas Moser. *Handbuch Verkehrsunfallrekonstruktion: Unfallaufnahme, Fahrdynamik, Simulation*. Vieweg+Teubner, Wiesbaden, 2nd edition, 2009.
- [Buna] Bundesanstalt für Straßenwesen (BASt). Automatische Zählstellen 2016 - Dauerzählstelle: Düderode. Webseite. [http://www.bast.de/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016\\_1/Jawe2016.html?nn=626916&cms\\_detail=4602&cms\\_map=0](http://www.bast.de/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016_1/Jawe2016.html?nn=626916&cms_detail=4602&cms_map=0); abgerufen am 09. März 2018.
- [Bunb] Bundesanstalt für Straßenwesen (BASt). Automatische Zählstellen 2016 - Dauerzählstelle: Röstebachtalbrücke. Webseite. [http://www.bast.de/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016\\_1/Jawe2016.html?nn=626916&cms\\_detail=4372&cms\\_map=0](http://www.bast.de/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016_1/Jawe2016.html?nn=626916&cms_detail=4372&cms_map=0); abgerufen am 09. März 2018.
- [Bunc] Bundesanstalt für Straßenwesen (BASt). Automatische Zählstellen 2016 - Dauerzählstelle: Tunnel Schmücke. Webseite. [http://www.bast.de/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016\\_1/Jawe2016.html?nn=626916&cms\\_detail=4340&cms\\_map=0](http://www.bast.de/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Daten/2016_1/Jawe2016.html?nn=626916&cms_detail=4340&cms_map=0); abgerufen am 09. März 2018.
- [Deu] Deutscher Verkehrssicherheitsrat e.V. Ablenkung im Straßenverkehr. Webseite. <http://www.abgelenkt.info/infos.htm>; abgerufen am 09. November 2017.
- [Din11] Ding Ding. Modeling and simulation of highway traffic using a cellular automaton approach. Examensarbete, Uppsala Universitet, 2011.
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1st edition, 1989.
- [HM98] Dirk Helbing and Peter Molnar. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51, 05 1998.
- [HM08] Samer H. Hamdar and Hani S. Mahmassani. From existing accident-free car-following models to colliding vehicles exploration and assessment. *Transportation Research Record*, (2088):45–56, 2008.

- 
- [HR] Thomas Harloff and Felix Reek. Süddeutsche Zeitung - So weit ist das autonome Fahren. Webseite. <http://www.sueddeutsche.de/auto/verkehrssicherheit-so-weit-ist-das-autonome-fahren-1.3913983>; abgerufen am 20. März 2018.
- [Hri09] Borislav Hristov. Untersuchung des Blickverhaltens von Kraftfahrern auf Autobahnen. Dissertation, Technische Universität Dresden, 2009.
- [KWT16] Andreas Kendziorra, Peter Wagner, and Tomer Toledo. A stochastic car following model. *Transportation Research Procedia*, 15:198–207, 6 2016.
- [Lon] London School of Economics and Political Science (LSE). You Auto Know: Which Driving Personality Are You? Webseite. <http://www.lse.ac.uk/website-archive/newsAndMedia/newsArchives/2015/09/YouAutoKnowWhichDrivingPersonalityAreYou.aspx>; abgerufen am 26. Februar 2018.
- [NS92] Kai Nagel and Michael Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2(12):2221–2229, 1992.
- [Rit15] Steffen Ritter. Autonome Multi-Agenten-Systeme in der Industrie. Masterarbeit, Hochschule Offenburg, 2015. <https://opus.hs-offenburg.de/files/963/Master-Thesis.pdf>; abgerufen am 09. April 2018.
- [RN10] Stuart Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, 3rd edition, 2010.
- [Roo] Roolfs, G. (Niedersächsischer Bildungsserver). Logistische Funktion. PDF. <http://www.nibis.de/~lbs-gym/AnalysisTeil2pdf/LogistischeFunktion.pdf>; abgerufen am 07. März 2018.
- [SAE] SAE International. Automated Driving - Levels od Driving Automation are defined in new SAE INTERNATIONAL Standard J3016. PDF. [https://www.sae.org/misc/pdfs/automated\\_driving.pdf](https://www.sae.org/misc/pdfs/automated_driving.pdf); abgerufen am 11. November 2017.
- [SL11] Werner Schnabel and Dieter Lohse. *Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung: Band 1 - Straßenverkehrstechnik*. Beuth Verlag, 2011.
- [Spa02] Michael Spahn. Reisezeiten im Nagel-Schreckenberg-Modell mit velocity dependent randomization und weiteren eindimensionalen Verkehrsflussmodellen auf einer offenen Kante. Diplomarbeit, Humboldt-Universität zu Berlin, 2002.
- [Sta17a] Statistisches Bundesamt. Statistisches Jahrbuch 2017. PDF, 2017. [https://www.destatis.de/DE/Publikationen/StatistischesJahrbuch/TransportVerkehr.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/Publikationen/StatistischesJahrbuch/TransportVerkehr.pdf?__blob=publicationFile); abgerufen am 01. Dezember 2017.
- [Sta17b] Statistisches Bundesamt. Verkehrsunfälle - Zeitreihen 2016. PDF, 2017. [https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/Verkehrsunfaelle/VerkehrsunfaelleZeitreihenPDF\\_5462403.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/Verkehrsunfaelle/VerkehrsunfaelleZeitreihenPDF_5462403.pdf?__blob=publicationFile); abgerufen am 01. Dezember 2017.
- [Tesa] Tesla GmbH. Autopilot. Webseite. [https://www.tesla.com/de\\_DE/autopilot](https://www.tesla.com/de_DE/autopilot); abgerufen am 18. März 2018.
- [Tesb] Tesla, Inc. Autopilot. Webseite. <https://www.tesla.com/autopilot>; abgerufen am 18. März 2018.

- 
- [Tra17] Khac Dat Tran. Social-Force-Modell zur Steuerung von Fahrzeugen in Grid-Umgebungen. Bachelorarbeit, Technische Universität Clausthal, 2017.
  - [VD] Peter Valdes-Dapena. CNN tech - Tesla in Autopilot mode crashes into fire truck. Webseite. <http://money.cnn.com/2018/01/23/technology/tesla-fire-truck-crash/index.html>; abgerufen am 18. März 2018.
  - [vWKvLVH15] Femke van Wageningen-Kessels, Hans van Lint, Kees Vuik, and Serge Hoogen-doorn. Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*, 4(4):445–473, 12 2015.
  - [WNW96] Peter Wagner, Kai Nagel, and Dietrich E. Wolf. Realistic Multi-Lane Traffic Rules for Cellular Automata. *PHYSICA A*, 234:96–2586, 1996.
  - [Woo09] Michael Wooldridge. *An Introduction to MultiAgent Systems*. Wiley Publishing, 2nd edition, 2009.
  - [YCLR12] Öznur Yeldan, Alberto Colorni, Alessandro Luè, and Emanuele Rodaro. A stochastic continuous cellular automata traffic flow model with a multi-agent fuzzy system. *Procedia - Social and Behavioral Sciences*, 54:1350–1359, 2012.

---

## Code-Listings

Listing A1: Scenariovereinbarung

```
1 # global section of configuration
2 main:
3
4     # statistic result ("summary" = aggregated values, "descriptive" =
5     # aggregated & single values)
6     statistic: "descriptive"
7
8     # simulated time
9     simulationtime_in_minutes: 60
10
11    # prettyprint of the result json
12    prettyprint: true
13
14    # time in ms for printing a message to avoid CircleCI timeout (CircleCI time
15    # can be set to 480000)
16    alive: 0
17
18    # line breaker after each step
19    linebreak: "----- step {0} -----"
20
21    # runtime
22    runtime:
23
24        # type of the runtime, parameter is the thread number: ->synchronized<-
25        # workstealing, fixedsize(T), cached, scheduled(T), single
26        type: "synchronized"
27
28        # number of threads of the runtime (fixedsize, scheduled)
29        threads: 1
30
31        # unit of the simulation
32        unit:
33
34            # cell size / size of a vehicle in meter
35            cellsize_in_meter: 2.5
36
37            # time of a single timestep in simulation
38            # 0.1 -> 6 sec, 0.05 -> 3 sec, 0.025 -> 1.5 sec, 0.0125 -> 0.75 sec
39            timestep_in_minutes: 0.025
40
41 # environment / street setting
42 environment:
43
44     # number of lanes
45     lanes: 1
46
47     # length if the lane in kilometer
48     length_in_km: 7.5
49
50 # agent vehicle configuration
51 agent:
52
53     # global constants for all vehicles (constant name
54     # must start with an upper-case letter)
55     constant:
56         AllowedSpeed: 100
57
58     # all vehicle configuration, key is the filename of the filename
59     # AgentSpeak(L++) script relative to configuration file
60     source:
61         vehicle.asl:
62
63             # number of generated vehicles
64             count: 34
65
66             # show beliefs in each cycle
67             showlog: false
68
69             # view range in meters around a vehicle
70             # (should be greater than cell size)
71             viewrange: 250
72
73             # local constant values for this vehicle
74             constant:
75                 LocalValue: "nothing"
```

---

```
76      # speed range configuration for initialize and maximum
77      # speed (maximum value will be set by random)
78      speed:
79          min: 80
80          max: 130
81
82      # acceleration configuration (the value is
83      # calculated by random)
84      acceleration:
85          min: 3.5
86          max: 7
87
88      # decceleration configuration (the value is
89      # calculated by random)
90      deceleration:
91          min: 8
92          max: 10
```

Listing A2: Agentenscript: NaSch-Modell, einspurig

```

1  /*
2   *      ASL file          NaSch model traffic, single lane
3   */
4
5 !cruise.
6
7 // --- start all other plans ---
8 +!cruise <-
9     !accelerate;
10    !decelerate;
11    !linger;
12    !cruise
13 .
14
15 // --- acceleration ---
16 +!accelerate
17     // --- accelerate only, if no traffic ahead ---
18     // --- otherwise you have to brake against the acceleration ---
19     // --- resulting in too long braking distances ---
20     :
21     CurrentSpeed < AllowedSpeed
22     && ~>>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell ),
23                               ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) )
24             ,
25             bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
26             && FwdDist < 1.5*CurrentSpeed
27             && FwdSpeed < CurrentSpeed
28         )
29     <-
30         vehicle/accelerate(0.5);
31         !accelerate
32 .
33
34 // --- lingering ---
35 +!linger <-
36     L = math/statistic/randomsimple;
37     L < 0.125;
38     vehicle/decelerate(0.3)
39 .
40
41 // --- deceleration ---
42 +!decelerate
43 /*
44     // --- decelerate if max. allowed speed is reached ---
45     // --- if commented out vehicle will cruise slightly over speed limit ---
46     : CurrentSpeed > AllowedSpeed <-
47         vehicle/decelerate(0.05);
48         !decelerate
49 */
50     // --- decelerate if traffic is ahead ---
51     // --- and own speed is higher than distance ---
52     : >>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell ),
53                               ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) ),
54             bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
55             && FwdDist < 1.5*CurrentSpeed
56         )
57         <-
58         vehicle/decelerate(1);
59         !decelerate
60 .
61
62 // --- collision ---
63 +!vehicle/collision <-
64 /*
65     // --- brake as hard as possible ---
66     vehicle/decelerate( 1 );
67 */
68     // --- stop immediately ---
69     vehicle/stop;
70 .

```

Listing A3: Agentenscript: NaSch-Modell, mehrspurig

```

1  /*
2   *   ASL file      multilane traffic (NaSch)
3   */
4
5 !cruise.
6
7 // --- start all other plans ---
8 +!cruise <-
9     CorrectedLanes = Lanes-1;
10    !accelerate;
11    !decelerate;
12    !linger;
13 //    !pullout(CorrectedLanes);
14 //    !pullin;
15 //    !pulloutworkaround(CorrectedLanes);
16 //    !pullinworkaround;
17 //    !cruise
18 .
19
20 // --- acceleration ---
21 +!accelerate
22     // --- accelerate only, if no traffic ahead ---
23     // --- otherwise you have to brake against the acceleration ---
24     // --- resulting in too long braking distances ---
25     :
26     CurrentSpeed < AllowedSpeed
27     && ~>>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell
28         ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) )
29         ),
30         bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
31         && FwdDist < 1.5*CurrentSpeed
32         && FwdSpeed < CurrentSpeed
33     )
34     <-
35         vehicle/accelerate(0.5);
36     !accelerate
37 .
38
39 // --- lingering ---
40 +!linger <-
41     L = math/statistic/randomsimple;
42     L < 0.125;
43     vehicle/decelerate(0.3)
44 .
45
46 // --- PULL-OUT PLAN WORKAROUND ---
47 // --- pull out, when NO TRAFFIC AT ALL on CurrentLane+1 ---
48 +!pulloutworkaround(OverallLanes)
49     :
50     // --- cond #1: not in leftmost lane ---
51     CurrentLane < OverallLanes
52     // --- cond #2: vehicle in front in own lane is close ---
53     && ~>>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell
54         ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) )
55         ,
56         bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
57         && math/floor( FwdLane ) == CurrentLane
58         && FwdDist < 1.5*CurrentSpeed
59     )
60     // --- cond #4a: no other vehicle in sight in lane to pull into to the
61     // left ---
62     && ~>>( view/vehicle( _, data( _, static( lane( LeftLane ), cell(
63         LeftCell ), speed( LeftSpeed ), distance( LeftDist ), direction(
64         LeftDir ) ) ) ),
65         math/floor( LeftLane ) == CurrentLane+1
66     )
67     <-
68         vehicle/pullout
69 .
70
71
72 // --- pull-out, change lane to overtake ---
73 // --- should work, but doesn't ---
74 // --- BUG in framework
75 // --- use +!pulloutworkaround instead ---
76 +!pullout(OverallLanes)
77     :
78     // --- cond #1: not in leftmost lane ---
79     CurrentLane < OverallLanes
80     // --- cond #2: vehicle in front in own lane is close ---
81

```

```

71     && >>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell
72         ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) ) )
73         ,
74         bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
75         && math/floor( FwdLane ) == CurrentLane
76         && FwdDist < 110
77         )
78     // --- cond #3: other lane is not worse (in driving direction) ---
79     // --- this includes an empty lane ---
80     && ~>>( view/vehicle( _, data( _, static( lane( FwdLane2 ), cell(
81         FwdCell2 ), speed( FwdSpeed2 ), distance( FwdDist2 ), direction(
82         FwdDir2 ) ) ) ),
83         bool/equal( generic/type/tostring( FwdDir2 ), "forward[]" )
84         && math/floor( FwdLane2 ) == CurrentLane+1
85         && FwdDist2 < FwdDist
86     /* --- commented out last two conditions, bug in framework ---
87
88     // --- cond #4: no other vehicle directly to the left in lane to pull
89     // --- into ---
90     && ~>>( view/vehicle( _, data( _, static( lane( LeftLane ), cell(
91         LeftCell ), speed( LeftSpeed ), distance( LeftDist ), direction(
92         LeftDir ) ) ) ),
93         bool/equal( generic/type/tostring( LeftDir ), "left[]" )
94         && math/floor( LeftLane ) == CurrentLane+1
95         )
96     // --- cond #5: don't hinder car behind in lane to pull into ---
97     // --- includes empty lane ---
98     && ~>>( view/vehicle( _, data( _, static( lane( BwdLane ), cell( BwdCell
99         ), speed( BwdSpeed ), distance( BwdDist ), direction( BwdDir ) ) )
100
101        ,
102        bool/equal( generic/type/tostring( BwdDir ), "backward[]" )
103        && math/floor( BwdLane ) == CurrentLane+1
104        && BwdDist < 100
105        && BwdSpeed > CurrentSpeed
106    */
107    <-
108        vehicle/pullout
109 .
110
111 // --- PULL-IN PLAN WORKAROUND ---
112 // --- pull in, when NO TRAFFIC AT ALL on CurrentLane-1 ---
113 +!pullinworkaround
114     :
115     // --- cond #1: not in rightmost lane ---
116     CurrentLane > 0
117     // --- cond #3a: no vehicle in sight in lane to pull into to the right
118     // --- ---
119     && ~>>( view/vehicle( _, data( _, static( lane( RightLane ), cell(
120         RightCell ), speed( RightSpeed ), distance( RightDist ), direction(
121         RightDir ) ) ) ),
122         math/floor( RightLane ) == CurrentLane-1
123         )
124     <-
125        vehicle/pullin
126 .
127
128 // --- pull-in, change lane after overtake is finished
129 // --- should work, but doesn't ---
130 // --- BUG in framework
131 // --- use +!pullinworkaround instead ---
132 +!pullin
133     :
134     // --- cond #1: not in rightmost lane ---
135     CurrentLane > 0
136     // --- cond #2: space forward in lane to pull into ---
137     // --- (no other vehicle in view range) ---
138     && ~>>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell
139         ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) )
140         ),
141         bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
142         && math/floor( FwdLane ) == CurrentLane-1
143         )
144     // --- cond #3: no vehicle directly to the right ---
145     && ~>>( view/vehicle( _, data( _, static( lane( RightLane ), cell(
146         RightCell ), speed( RightSpeed ), distance( RightDist ), direction(
147         RightDir ) ) ) ),
148         bool/equal( generic/type/tostring( RightDir ), "right[]" )

```

---

```

132             && math/floor( RightLane ) == CurrentLane-1
133             )
134         // --- cond #4: don't hinder car behind in lane to pull into ---
135         && ~>>( view/vehicle( _, data( _, static( lane( BwdLane ), cell( BwdCell
136             ), speed( BwdSpeed ), distance( BwdDist ), direction( BwdDir ) ) )
137             ),
138             bool/equal( generic/type/tostring( BwdDir ), "backward[]" )
139             && math/floor( BwdLane ) == CurrentLane-1
140             && BwdDist < 125
141         //             && BwdSpeed > CurrentSpeed
142         )
143     <-
144     vehicle/pullin
145 .
146 // --- deceleration ---
147 +!decelerate
148 /*
149     // --- decelerate if max. allowed speed is reached ---
150     : CurrentSpeed > AllowedSpeed <-
151         vehicle/decelerate(0.05);
152     !decelerate
153 */
154     // --- decelerate if traffic is ahead ---
155     // --- and own speed is higher than distance ---
156     : >>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell ),
157         speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) ) ),
158         bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
159         && math/floor( FwdLane ) == CurrentLane
160         && FwdDist < 1.5*CurrentSpeed
161     ) <-
162     vehicle/decelerate(1);
163     !decelerate
164 .
165 // --- collision ---
166 +!vehicle/collision <-
167     // --- brake as hard as possible ---
168     vehicle/decelerate( 1 );
169 /*
170     // --- stop immediately ---
171     vehicle/stop;
172 */

```

Listing A4: Agentenscript: Auszug Bedingungen pullout-Plan

```

1  /*
2   * ASL file          pullout conditions (separate for testing)
3   */
4
5  +!pullouttest <-
6      // --- TEST SETUP FOR THE CONDITIONS ---
7      // --- each condition has an output line for control purposes ---
8
9  Cond1 = CurrentLane < Lanes;
10 generic/print("vehicle is not on leftmost lane:", Cond1);
11
12 Cond2 = >>( view/vehicle( _, data( _, static( lane( FwdLane ), cell( FwdCell
13     ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) ) ) ),
14     bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
15     && math/floor( FwdLane ) == CurrentLane
16     && FwdDist < 175
17 );
18 generic/print("traffic ahead in own lane:", Cond2);
19
20 Cond3 = ~>>( view/vehicle( _, data( _, static( lane( FwdLane2 ), cell(
21     FwdCell2 ), speed( FwdSpeed2 ), distance( FwdDist2 ), direction( FwdDir2
22     ) ) ) ),
23     bool/equal( generic/type/tostring( FwdDir2 ), "forward[]" )
24     && math/floor( FwdLane2 ) == CurrentLane+1
25     && FwdDist2 < FwdDist
26 );
27 generic/print("lane to pull into is not worse:", Cond3);
28
29 Cond4 = ~>>( view/vehicle( _, data( _, static( lane( LeftLane ), cell(
30     LeftCell ), speed( LeftSpeed ), distance( LeftDist ), direction( LeftDir
31     ) ) ) ),
32     bool/equal( generic/type/tostring( LeftDir ), "left[]" )
33     && math/floor( LeftLane ) == CurrentLane+1
34     );
35 generic/print("no traffic directly to the left on lane to pull into:", Cond4
36 );
37
38 Cond5 = ~>>( view/vehicle( _, data( _, static( lane( BwdLane ), cell(
39     BwdCell ), speed( BwdSpeed ), distance( BwdDist ), direction( BwdDir ) ) ) ),
40     bool/equal( generic/type/tostring( BwdDir ), "backward[]" )
41     && math/floor( BwdLane ) == CurrentLane+1
42     && BwdDist < 100
43     && BwdSpeed > CurrentSpeed
44 );
45 generic/print("not hindering traffic behind in lane to pull into", Cond5);
46
47 Combined = Cond1 && Cond2 && Cond3 && Cond4 && Cond5;
48 generic/print("all combined", Combined)
49 .

```

Listing A5: Agentenscript: Auszug Bedingungen pullin-Plan

```

1  /*
2   * ASL file          pullin conditions (separate for testing)
3   */
4
5  +!pullintest <-
6      // --- TEST SETUP FOR THE CONDITIONS ---
7      // --- each condition has an output line for control purposes ---
8
9      Condi = CurrentLane > 1;
10     generic/print("vehicle is not on rightmost lane:", Condi);
11
12     Cond2 = ~>>( view/vehicle( _, data( _, static( lane( FwdLane ), cell(
13         FwdCell ), speed( FwdSpeed ), distance( FwdDist ), direction( FwdDir ) )
14         ) ),
15             bool/equal( generic/type/tostring( FwdDir ), "forward[]" )
16             && math/floor( FwdLane ) == CurrentLane-1
17             );
18     generic/print("no other vehicle in forward view on lane to pull into", Cond2
19         );
20
21     Cond3 = ~>>( view/vehicle( _, data( _, static( lane( RightLane ), cell(
22         RightCell ), speed( RightSpeed ), distance( RightDist ), direction(
23         RightDir ) ) ),
24             bool/equal( generic/type/tostring( RightDir ), "right[]" )
25             && math/floor( RightLane ) == CurrentLane-1
26             );
27     generic/print("no other vehicle directly to the right on lane to pull into",
28         Cond3);
29
30     Cond4 = ~>>( view/vehicle( _, data( _, static( lane( BwdLane ), cell(
31         BwdCell ), speed( BwdSpeed ), distance( BwdDist ), direction( BwdDir ) )
32         ) ),
33             bool/equal( generic/type/tostring( BwdDir ), "backward[]" )
34             && math/floor( BwdLane ) == CurrentLane-1
35             && BwdDist < 125
36             );
37     generic/print("not hindering traffic behind on lane to pull into", Cond4);
38
39     Combined = Condi && Cond2 && Cond3 && Cond4;
40     generic/print("all combined", Combined)
41 .

```

---

## weitere Diagramme

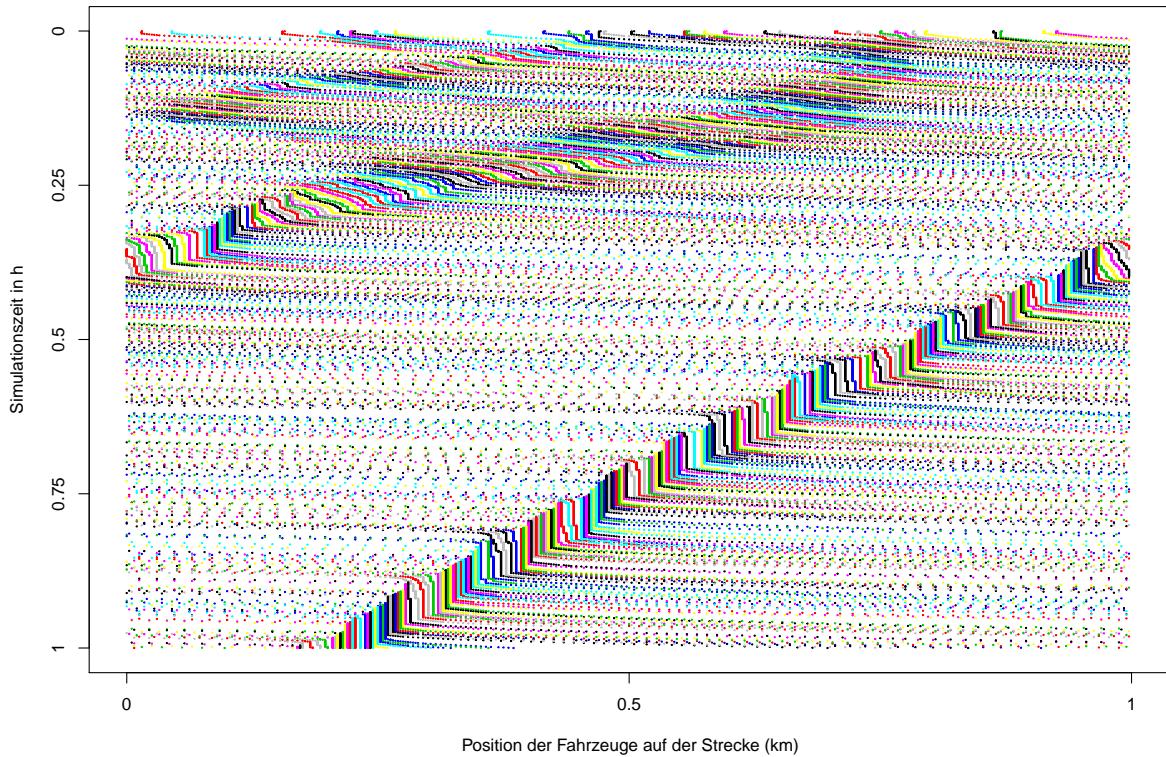


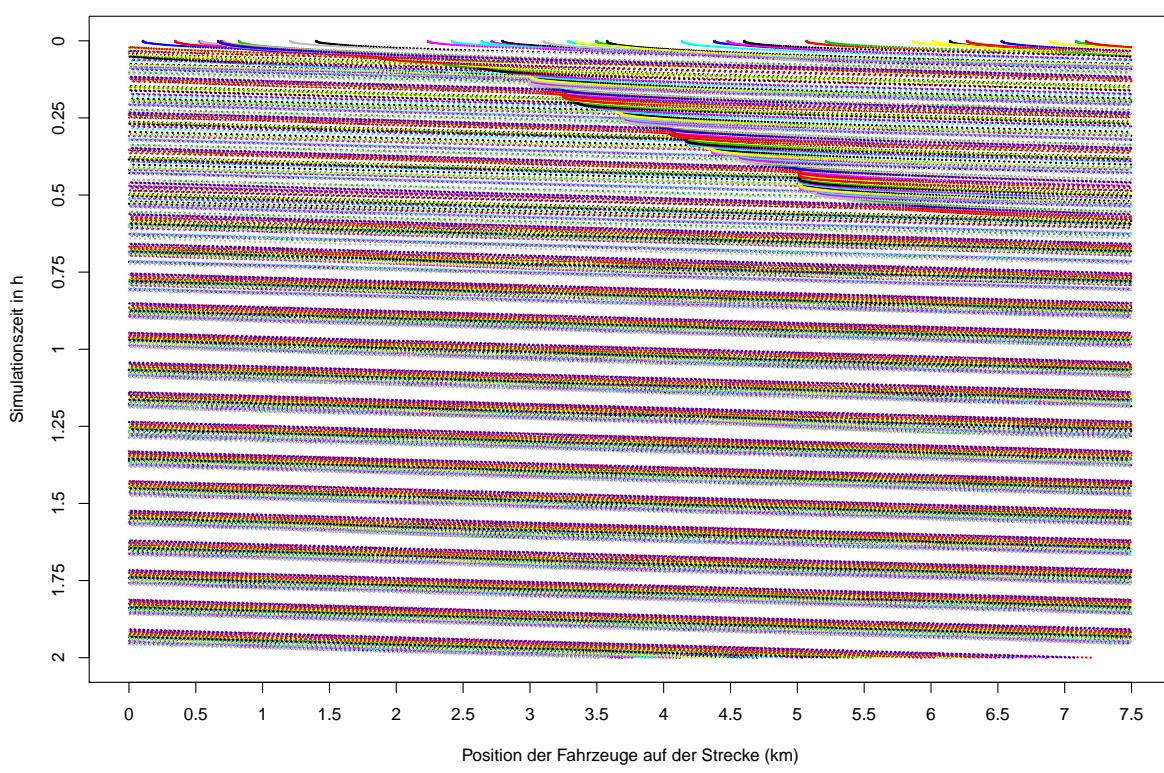
Abbildung A1: Positionsdiagramm mit rückwärts laufender Stauwelle

Abbildung A1 entstand aufgrund einer Fehleinstellung in der Szenariodatei. Hier fuhren 33 Fahrzeuge auf einer Strecke von 1 km. Für die kurze Strecke ist diese Anzahl Fahrzeuge nicht zu bewältigen.

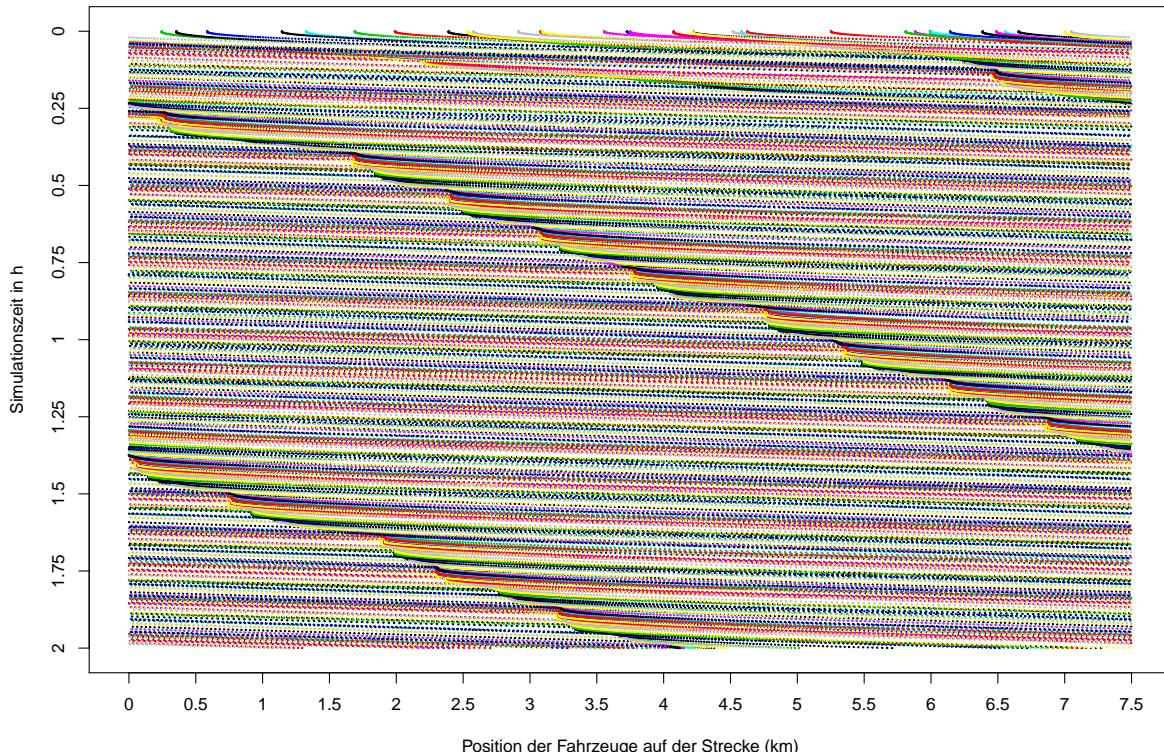
Zu Beginn sieht man aufgrund der enger werdenden Punktmuster, dass sich die Geschwindigkeit der Fahrzeuge nur verlangsamt. Dann kommt es zu einer länger anhaltenden Stillstandsphase der Fahrzeuge, die auch zu Simulationsende noch anhält.

Hier erkennt man die in den Versuchen in [NS92] erkannte Rückwärtsbewegung der Stauwelle.

Die Positionsplots weiterer Simulationsläufe mit der A7-Verkehrsmenge (siehe Kapitel 4.5.1) in Abbildung A2 zeigen, dass auch mit 34 Fahrzeugen Stockungen auftreten können. Oft passierte dies zu Beginn des Durchlaufes, pendelte sich dann aber ein.

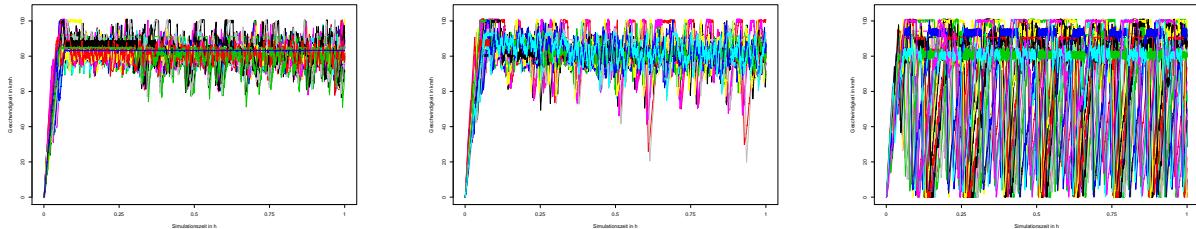


(a) Stockungen am Anfang der Simulation



(b) wiederholte Stockungen über die Dauer der Simulation

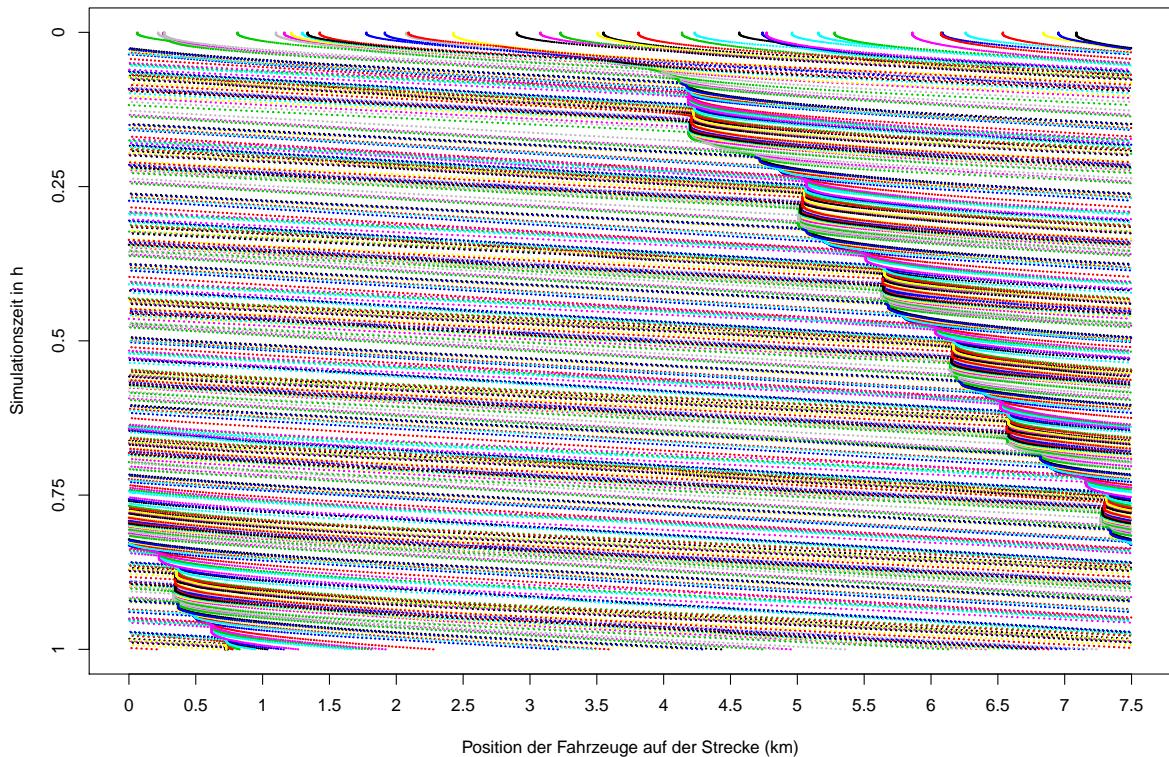
Abbildung A2: weitere Simulationsläufe mit 34 Fahrzeugen, A7-Verkehrsmenge



(a) mit 36 Fahrzeugen

(b) mit 37 Fahrzeugen

(c) mit 38 Fahrzeugen



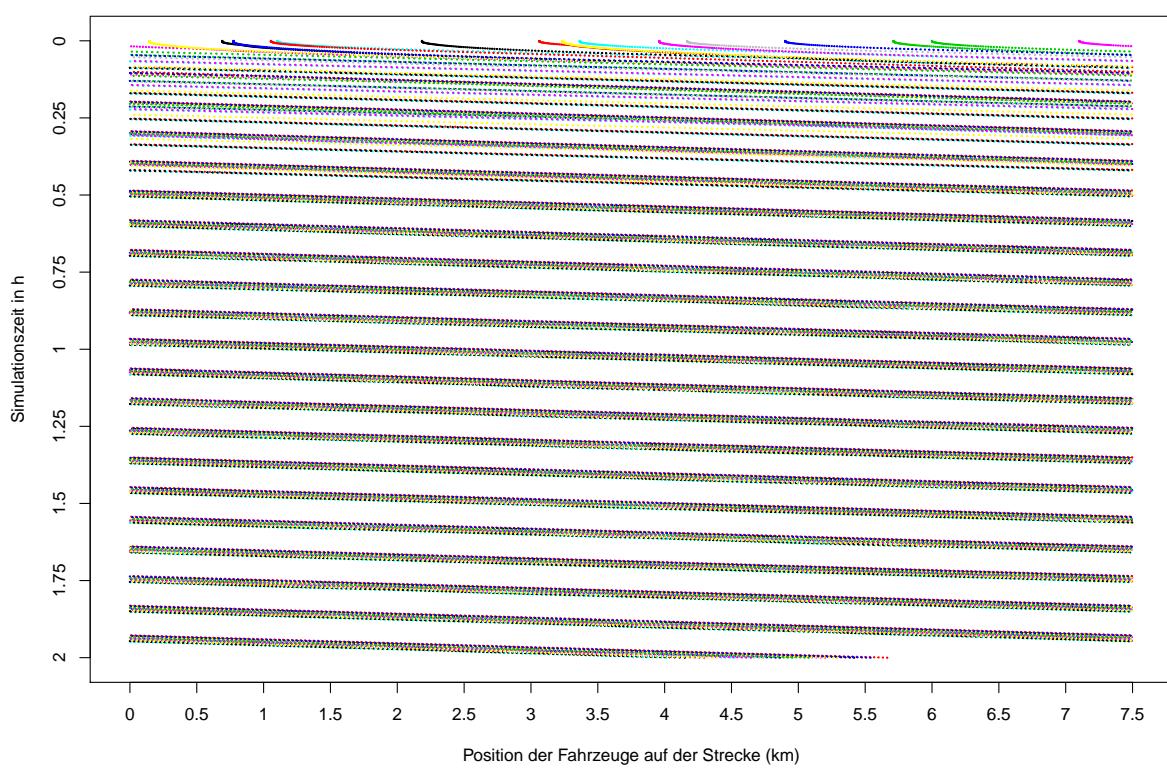
(d) mit 38 Fahrzeugen (Positionen)

Abbildung A3: Geschwindigkeitskurven und Positionsdiagramm der erweiterten Testläufe

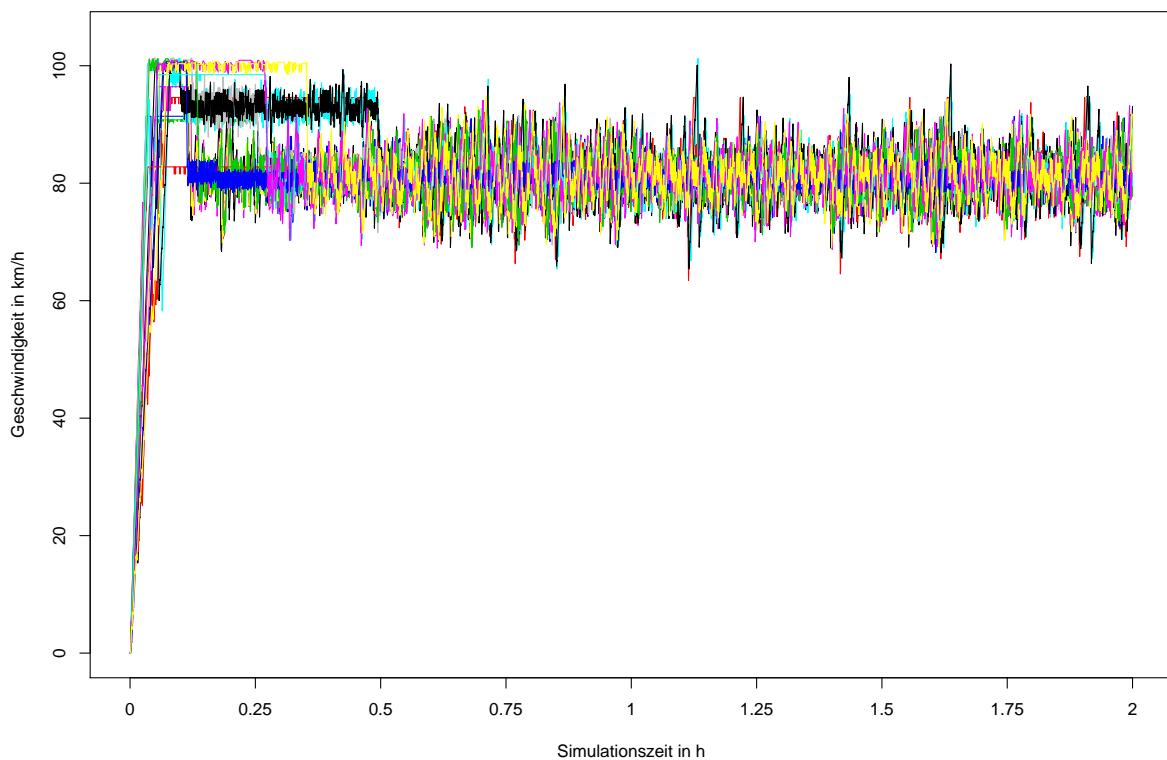
Abbildung A3 zeigt drei Geschwindigkeitskurven und ein Positionsplot für die Simulationsläufe der fahrzeugweisen Erhöhung der Verkehrsmenge der A7 aus Kapitel 4.5.1.

Bereits ab einer Menge von 37 Fahrzeugen sind die Einbrüche der Geschwindigkeitskurven wesentlich stärker ausgeprägt (Abbildung 3(b)). Der Verkehrsfluss kann aber noch aufrecht erhalten werden.

Mit 38 Fahrzeugen und mehr kommt es dann zu mehreren kleineren Stauereignissen. Während sich die Ereignisse auf der Fahrbahn vorwärts bewegen, ist jeweils die Rückwärtsbewegung der Stauwelle erkennbar (Abbildung 3(d)).

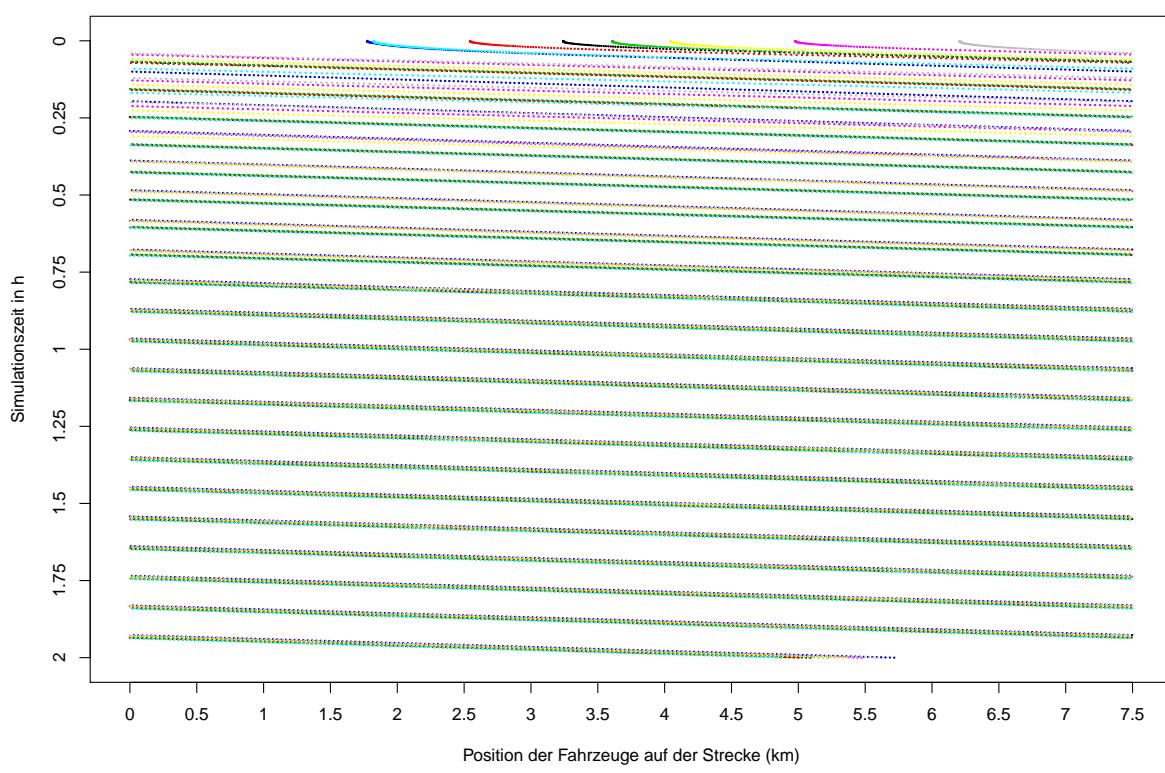


(a) Position

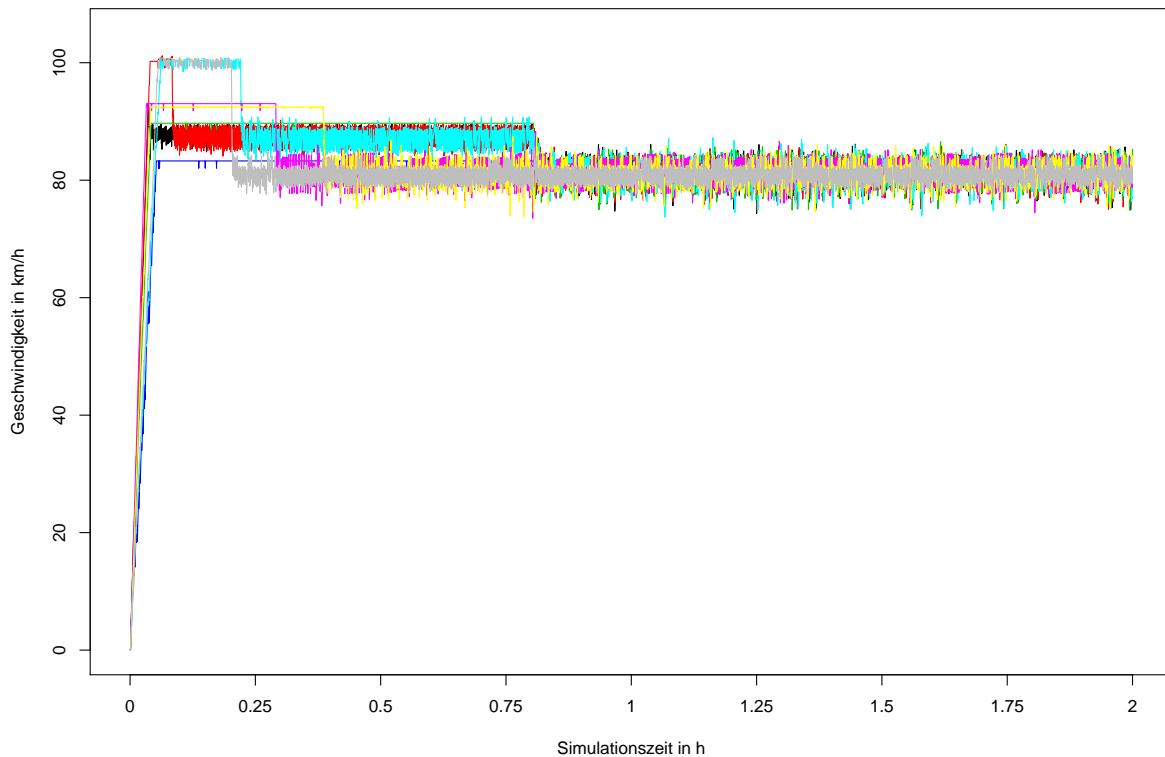


(b) Geschwindigkeit

Abbildung A4: Diagramme, A38-Verkehrsmenge: 15 Fahrzeuge



(a) Position



(b) Geschwindigkeit

Abbildung A5: Diagramme, A71-Verkehrsmenge: 8 Fahrzeuge

---

## **CD-ROM**

Die beiliegende CD-ROM enthält diese Bachelorarbeit im PDF-Format - Ordner **thesis**. Zudem sind die Szenariodateien für Single- und Multilane-Versuche (.YML) und jew. eine entsprechende Datei mit den Agentenplänen (.ASL) vorhanden - Ordner **single-lane** und **multi-lane**. Das Script mit dem die Diagramme generiert wurden, ist ebenfalls enthalten (.R) - Ordner **r-script**. Zusätzlich enthält die CD-ROM aus Gründen der Nachvollziehbarkeit die Simulationssoftware als ausführbare JAR-Datei in der Version 0.1.86 vom 06. März 2018 - Ordner **simulation**. Mit dieser Version wurden die Versuche getätigt.