

Proof-of-Concept einer komplexen Co-Simulationsumgebung für einen Fahrsimulator zur Untersuchung von Car2X-Kommunikations-Szenarien

M.Sc. **Sven Henning**, M.Sc. **Dominik S. Buse**, B.Sc. **Mario Franke**, Prof. Dr.-Ing. habil. **Ansgar Trächtler**, Dr.-Ing. **Sandra Gausemeier**, Prof. Dr.-Ing. habil. **Falko Dressler**, Heinz Nixdorf Institut, Universität Paderborn

1. Zusammenfassung

Für die voranschreitenden Entwicklungen moderner Fahrerassistenzsysteme (Advanced Driving Assistance Systems, kurz: ADAS) werden heutzutage verschiedenste Simulationsumgebungen eingesetzt. So existieren Fahrsimulatoren unterschiedlichster Komplexitätsgrade für virtuelle Testfahrten, Verkehrssimulationen mit Fremdverkehr zur Untersuchung umfangreicher Straßennetze und Simulationsumgebungen für die Kommunikation zwischen Fahrzeugen (Car2X-Kommunikation). Jede dieser Simulationsumgebung erfüllt für sich einen speziellen Zweck, jedoch existieren bisher keine Lösungen, alle genannten Funktionen in einer Gesamtsimulation zu vereinen. Dieser Beitrag zeigt daher einen Proof-of-Concept einer Co-Simulationsumgebung, welche die genannten Funktionen vereint. Die vorgestellte Simulationsumgebung ermöglicht einem menschlichen Fahrer ein virtuelles Fahrerlebnis in einem Fahrsimulator mit intelligentem Fremdverkehr sowie Car2X-Kommunikation. Es werden Herausforderungen bei der Synchronisierung der Einzelsimulationen und bei der Einhaltung von Echtzeitbedingungen an das Gesamtsystem aufgezeigt sowie eine Bewertung der Leistungsfähigkeit vorgenommen. Ein einfaches Testszenario für ein vernetztes ADAS schließt den Beitrag ab.

2. Abstract

For the progressive development of modern Advanced Driving Assistance Systems (ADAS), a wide variety of simulation environments are used nowadays. Driving simulators of various degrees of complexity exist for virtual test drives, traffic simulations involving external traffic for the investigation of complex road networks as well as simulation environments for communication between vehicles (Car2X communication). Each of these simulation environments serves a specific purpose, but so far there are no solutions to combine all mentioned functions in one overall simulation. This article therefore shows a proof-of-concept of a co-simulation environment that combines the above mentioned functions. The presented simulation environment provides a human driver to have a virtual driving experience in a driving simulator with intelligent external traffic and Car2X communication. Challenges for the synchronization of the individual simulations and the fulfillment of real-time conditions to the

overall system are shown and an evaluation of the performance is carried out. A simple test scenario for a networked ADAS completes the contribution.

3. Einleitung und Motivation

Moderne ADAS steigern Sicherheit, Effizienz und Komfort im Straßenverkehr. Klassischerweise vom Fahrer übernommene Fahrfunktionen können durch den Einsatz von Sensoren, Aktoren und entsprechender Logik durch das Fahrzeug unterstützt oder übernommen werden. In [10] und [11] geben die Autoren eine Übersicht über die Vielzahl klassischer und moderner ADAS, die die Sicherheit oder den Komfort steigern. Die Verwendung von Fahrsimulatoren unterschiedlichster Ausprägung zum Test von ADAS während der Entwicklung stellt bereits heute in vielen Fällen den Stand der Technik dar. Gerade in Hinsicht auf moderne Funktionen für (teil-)autonom fahrende Fahrzeuge, deren Entwicklung neue Herausforderungen mit sich bringt, weist die modellbasierte und simulationsgestützte Entwicklung Vorteile gegenüber aufwendigen und in der Regel kostenintensiven Prototypentests auf. Beispielsweise werden am Heinz Nixdorf Institut der Universität Paderborn mehrere Fahrsimulatoren für verschiedene Anwendungszwecke betrieben; von einfachen Bildschirmen mit Lenkrad, Gas- und Bremspedalen bis hin zu einem Fahrsimulator mit einem 5-Freiheitsgrade-Bewegungssystem, einem Fahrzeug-Chassis sowie einer Rundprojektion (vgl. Abbildung 1).

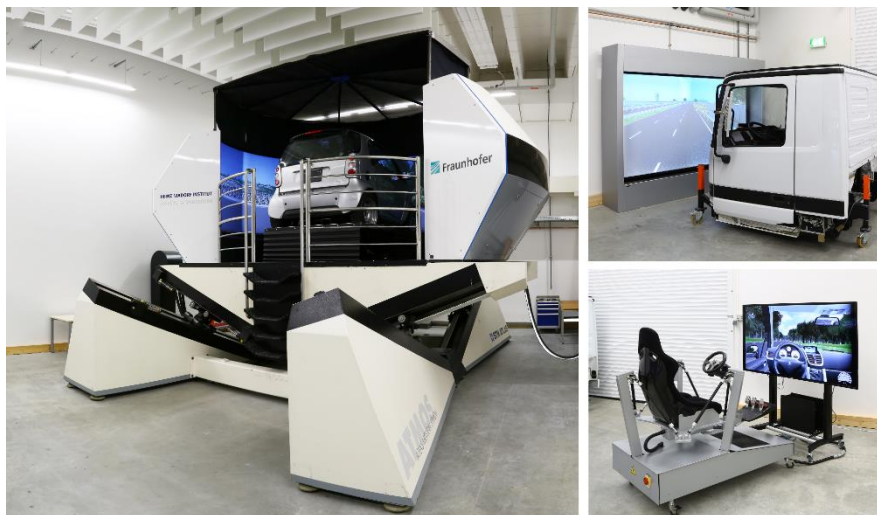


Abbildung 1: Fahrsimulatoren unterschiedlicher Komplexität am Heinz Nixdorf Institut

Für den Fahrsimulator, in Abbildung 1 links dargestellt, wurde in [8] und [9] exemplarisch eine rekonfigurierbare Simulationsumgebung vorgestellt, die aus mehreren Komponenten besteht. Die Komponenten umfassen eine Bewegungsplattform mit realem Fahrzeugchassis, in dem ein Mensch die Möglichkeit hat, haptisch in einem Fahrsimulationsszenario zu interagieren und die wirkenden Beschleunigungen und Drehraten einer Fahrt zu erleben (vestibuläre Reize). Hierzu werden sog. Motion-Cueing-Algorithmen mit dem Ziel eingesetzt, die

Beschleunigungen und Drehraten unter Berücksichtigung der Beschränkungen des Fahrsimulators möglichst realitätsnah wiederzugeben. In [7] wurde zur Umsetzung eines solchen Algorithmus ein vielversprechender Ansatz präsentiert, der auf einer echtzeitfähigen modellprädiktiven Regelung basiert und eine sehr hohe Regelgüte erzielt. Außerdem wird ein echtzeitfähiges Fahrdynamikmodell für das Ego-Fahrzeug¹ eingesetzt, welches zum einen die simulierten Beschleunigungsgrößen und Drehraten für den zuvor erwähnten Motion-Cueing-Algorithmus berechnet und zum anderen die Position und Lage des Ego-Fahrzeugs bereitstellt, die für eine 3D-Darstellung notwendig sind. Die Rundprojektion stellt dem Fahrer mit Hilfe einer selbstentwickelten 3D-Visualisierung (Unity3D²) automatisch erzeugte 3D-Szenen dar während eine rudimentäre Fremdverkehrssimulation weitere Fahrzeuge in der Szene simuliert. Mit Systemen wie diesem können Tests früher im Entwicklungszyklus durchgeführt und durch kontrollierte Testbedingungen vergleichbar reproduziert werden, was Entwicklungskosten im Vergleich zu klassischen Prototypentests senken kann. In [25] wurde mit der vorgestellten Simulationsumgebung beispielsweise erfolgreich ein lichtbasiertes Fahrerassistenzsystem entwickelt und simulativ erprobt, bevor die abschließende Entwicklungsphase mit realen Tests begonnen wurde. Jedoch ist die Einsatzmöglichkeit des Fahrsimulators beschränkt, da bisher kein interaktiver und intelligenter Fremdverkehr existiert und nicht alle notwendigen Technologien für zukünftige ADAS integriert sind. Ziel ist es daher, diese für zukünftige Untersuchungen am Fahrsimulator nutzbar zu machen.

Zu solchen Technologien zukünftiger ADAS zählen beispielsweise eine Drahtloskommunikation zwischen Fahrzeugen sowie der Verkehrsinfrastruktur (Car2X-Kommunikation). Damit ergeben sich Quellen für Umgebungsdaten jenseits der Reichweite der fahrzeugeigenen Sensoren sowie die Möglichkeit kooperativer Fahrmanöver. Die Vernetzung der Systeme erschwert durch die erhöhte Komplexität jedoch auch die Sicherstellung ihrer Funktion. Statt den Systemen von nur einem Fahrzeug müssen zukünftig Netzwerke von Fahrzeugen und ihre Interaktionen untereinander untersucht werden.

Die Kommunikation ist als Schlüsseltechnologie für vernetzte ADAS aufzufassen, die den zukünftigen Verkehr mithilfe von autonomen Fahrfunktionen sicherer und effizienter gestalten können. In [12] wird das Konzept eines selbstorganisierenden Verkehrsinformationssystems erläutert, welches auf dezentraler Car2X-Kommunikation basiert und die Möglichkeit eröffnet, dass die Fahrzeuge selbständig den aktuellen Verkehrszustand erfassen und diese Information an andere Fahrzeuge weitergeben. Solche Informationen bilden die Grundlage für vernetzte Verkehrsmanagementansätze. In [6] wurde als eines von vielen Beispielen in der Literatur ein Konzept für ein hierarchisches Verkehrsmanagement vorgestellt, das basierend auf der Annahme, dass der Verkehr vollständig autonom abgehandelt wird und eine Car2X-Kommunikation existiert, eine Balancierung der Verkehrswegeauslastung ermöglicht und

¹ Als Ego-Fahrzeug wird das eigene Fahrzeug aus Sicht des Fahrers genannt (i.d.R. ist dies ein Mensch)

² Unity3D ist eine Laufzeit- und Entwicklungsumgebung für Spiele

somit Risiken von Stauungen sowie die mittleren Reisezeiten der Fahrzeuge reduziert. In dem Beitrag wurde die Kommunikation zwischen den Fahrzeugen jedoch vereinfacht als ideal und frei von Latenzen und Störungen angenommen und spiegelt damit die in der Realität vorhandenen Gegebenheiten nicht korrekt wider. Eine ganzheitliche Untersuchung von vernetzten ADAS muss daher auch die Car2X-Kommunikation berücksichtigen. Mit speziellen Simulationsumgebungen zur Untersuchung von Fahrzeugnetzwerken können Effekte der Informationsübertragung bereits simuliert und bei der Umsetzung von ADAS einbezogen werden. Ebenso bleibt der menschliche Fahrer ein wichtiger Einflussfaktor – sowohl als Nutzer des ADAS als auch als Fahrer anderer Fahrzeuge, denn der Übergang zum autonomen Fahren wird ein fließender Übergang mit teil-autonomen Mischverkehr (autonom / nicht autonom) sein. Die Berücksichtigung des menschlichen Verhaltens in Form von zusätzlichem intelligentem Fremdverkehr in der Fahrsimulation ist ebenfalls bereits Stand der Technik.

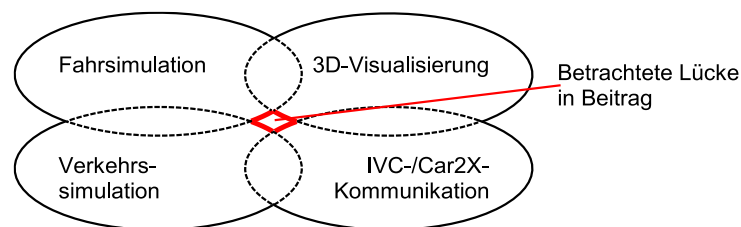


Abbildung 2: Teilgebiete der Fahrsimulation sowie deren Überlappungen, in Rot: In diesem Beitrag geschlossene Lücke

Für Teilgebiete sind also bereits Lösungen vorhanden. Jedoch existiert bisher keine Möglichkeit, eine komplexe Fahrsimulation mit intelligentem Fremdverkehr, integrierter Car2X-Kommunikation sowie dem menschlichen Fahrer mit seinem schwer vorhersagbaren Verhalten in einer einzigen Gesamtsimulation zu untersuchen oder für die Evaluierung von ADAS nutzbar zu machen. Diese Lücke zeigt sich an den Überlappungen dieser Teilgebiete (siehe auch Abbildung 2). Dieser Beitrag erläutert einen Proof-of-Concept für eine Co-Simulationsumgebung, die den Funktionsumfang mehrerer Simulationsumgebungen bzw. -konzepte vereint. Hierbei werden besondere, in diesem Kontext auftretende Problemstellungen aufgezeigt und mögliche Lösungen gegeben (bspw. Kopplung mehrerer Simulationen).

4. Stand der Technik

Wie bereits in Abbildung 2 zu erkennen ist, besteht eine umfassende Fahrsimulation oft aus mehreren Einzelkomponenten, die zu einer Gesamtsimulation kombiniert werden. In der Literatur sind einige Kombinationen aus Einzelsimulationen bereits diskutiert worden und sollen kurz vorgestellt und verglichen werden.

In [2] stellen die Autoren die bidirektional gekoppelte Simulationsumgebung Veins vor, die auf der diskreten ereignisorientierten Simulation OMNeT++ (**O**bjective **M**odular **N**etwork **T**estbed in **C++**) [15] und der Verkehrssimulation SUMO (**S**imulation of **U**rban **M**obility) [11] basiert.

Veins implementiert mithilfe von OMNeT++ eine komplexe Car2X-Kommunikation, die eine Simulation selbiger über mehrere Abstraktionsschichten hinweg ermöglicht. Dabei wird die bereitgestellte Netzwerkschnittstelle TraCI [14] zum synchronisierten Datenaustausch verwendet (vgl. Abbildung 3). Veins wird zur simulativen Untersuchung von sog. VANETs (**V**ehicular **a**d-hoc **n**etworks) eingesetzt.

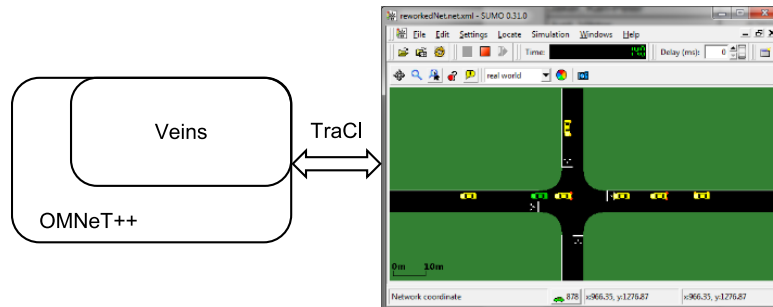


Abbildung 3: Kombination der Car2X-Kommunikationssimulation Veins (links) mit der Verkehrssimulation SUMO (rechts)

Veins ist in der Lage, physikalische Effekte der Kommunikation zu simulieren. Dazu gehören bspw. die Wellenausbreitung, Interferenzen, Antennengeometrien, Abschattungen durch Objekte oder die Leistungsaufnahme von Sendern und Empfängern für alle in der Simulation vorkommenden Fahrzeuge [2]. Darüber hinaus werden Protokolle und Standards auf MAC-Ebene (**M**edia **A**ccess **C**ontrol) berücksichtigt, die für die Kommunikation notwendig sind sowie die Möglichkeit gegeben, darauf aufsetzend Kommunikationsanwendungen zu implementieren. Diese umfassen u.a. IEEE 802.11p, Wi-Fi und DSRC (**D**edicated **S**hort-**R**ange **C**ommunications), welche in [1] detaillierter diskutiert werden. Ein Fahrsimulator, ein detailliertes Fahrdynamikmodell sowie eine interaktive 3D-Umgebung sind jedoch nicht enthalten. In [3] stellen die Autoren ein sog. Ego-Vehicle-Interface (EVI) vor, welches ein echtzeitbasiertes Hardware-in-the-Loop-System (HiL) mit Veins synchronisieren kann und stellt somit die Grundlage zur Einbringung eines komplexen Fahrdynamikmodells, das z.B. auf einer echtzeitfähigen Hardware berechnet wird. Somit konnten die Autoren in [3] die angesprochene Lücke zum Teil schließen. Der EVI-Ansatz wird in [4] und [5] weiter ausgearbeitet, welche Grundlagen für diesen Beitrag aufweisen. Die Autoren in [16] stellen ein gekoppeltes System aus einer 3D-Visualisierung und einer eigenen Verkehrssimulation vor. In [17] wird ein System präsentiert, welches, wie auch in [8] gezeigt, ein komplexes Fahrdynamikmodell einsetzt und ebenfalls eine 3D-Visualisierung umfasst, wobei jedoch Monitore und keine Rundprojektion als Ausgabegeräte eingesetzt wurden. Eine weitere Veröffentlichung [18] zeigt die Kombination eines realen Fahrzeugs und der Verkehrssimulation SUMO, bei der der Informationsaustausch durch Smartphones umgesetzt wurde. Die Kombination aus einer 3D-Visualisierung, einer Verkehrssimulation, einer Car2X-Kommunikationssimulation und einer Fahrdynamiksimulation stellen die Autoren in [19] vor. Jedoch wurden alle Komponenten selbst entwickelt und aus Echtzeitgründen in Unity3D

implementiert, sodass der Funktionsumfang beschränkt ist. In [20] wird das System „Distributed Virtual Environmet“ (kurz DiVE) vorgestellt. Die in einer Server/Client-Struktur angelegte Simulationsumgebung ermöglicht es mehreren Menschen in einem gemeinsamen Szenario zu interagieren. Zusätzlich wird eine eigene Car2X-Kommunikationssimulation basierend auf OMNeT++ eingesetzt. Jedoch wird für das Ego-Fahrzeug auf ein in Unity3D integriertes und stark vereinfachtes Fahrdynamikmodell zurückgegriffen. An der Universität Buffalo wurde ein umfangreiches Gesamtsystem entwickelt, das aus dem proprietären Verkehrssimulator Paramics, dem Netzwerksimulator NS2 und einem Fahrsimulator mit Bewegungsplattform besteht (also auch ein entsprechend umfangreiches Fahrdynamikmodell beinhaltet) und damit alle notwendigen Funktionen erfüllt, jedoch durch die proprietären Softwarebestandteile wenig flexibel ist.

Zusammenfassend kann festgestellt werden, dass es durchaus Ansätze gibt, verschiedene Simulationsumgebung zu kombinieren, um eine realistischere Fahrsimulation zu ermöglichen. Allerdings ist ebenfalls ersichtlich, dass eine vollumfassende Kombination von Verkehrssimulation und Car2X-Kommunikationssimulation sowie einem detaillierten Fahrdynamikmodell und einer 3D-Visualisierung nicht existiert. In Tabelle 1 ist eine Übersicht der verschiedenen vorgestellten Ansätze zu sehen.

Tabelle 1: Vergleich verschiedener kombinierter Fahrsimulationsumgebungen

Quelle	3D-Visualisierung	Verkehrssimulation	Car2X-Kommunikationssimulation	Fahrdynamiksimulation
[2]	Nein	SUMO	OMNeT++/Veins	Nein
[3]	Erweiterbar	SUMO	OMNeT++/Veins	Erweiterbar
[8]	Unity3D	Nein	Nein	MATLAB/Simulink
[16]	Unity3D	Eigenes Modell	Nein	Nein
[17]	Ja	Nein	Nein	Eigenes Modell
[18]	Echtes Fahrzeug	SUMO	Nein	Echtes Fahrzeug
[19]	Unity3D	Modell in Unity3D	Modell in Unity3D	Modell in Unity3D
[20]	Unity3D	Eigenes Modell	Basierend auf OMNeT++	Modell in Unity3D
[21]	Ja	Paramics	NS2	Eigene Plattform

5. Konzept und Implementierung

Im Folgenden wird die Gesamtsimulationsumgebung vorgestellt, welche mit der erweiterten Synchronisationsschnittstelle EVI (Ego Vehicle Interface) die vier einzelnen Komponenten miteinander verkoppelt, aus denen das System besteht. Wie in Abbildung 4 links dargestellt, besteht das Gesamtsystem aus einer Kommunikationssimulation, einer Verkehrssimulation, einer Fahrdynamiksimulation für das Ego-Fahrzeug und einer 3D-Visualisierung mit Ausgabe. Die Verkehrssimulation übernimmt die Aufgabe, alle Fremdverkehrsfahrzeuge sowie deren Verhalten für einen Straßennetausschnitt zu simulieren. In diesem Beitrag wird die Simulationsumgebung SUMO [13] eingesetzt.

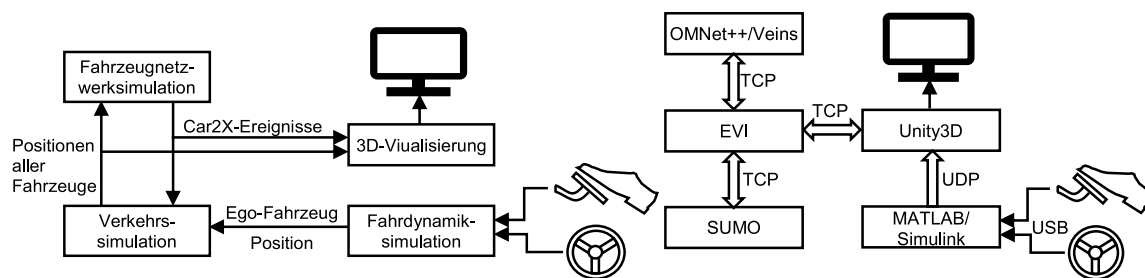


Abbildung 4: Links: Informationsfluss zwischen den Einzelsimulationen, Rechts: Umsetzung der Kombination durch die Implementierung von EVI

SUMO besitzt zu jedem Zeitpunkt die Informationen über die Position und Lage aller Fahrzeuge sowie über das Straßennetz (z.B. Straßengeometrien, Hindernisse wie Gebäude etc.) und kann diese den anderen Simulationsumgebungen zur Verfügung stellen. Zusätzlich können extern gesteuerte Fahrzeuge (Ego-Fahrzeuge) über Positionsinformationen eingebunden werden, auf die der Fremdverkehr entsprechend reagiert. Das in SUMO verwendete Straßennetz ist als globales Modell für alle Simulationsumgebungen zu betrachten und kann synthetischer Natur sein oder aus OpenStreetMap³-Kartenmaterial erzeugt werden. Als Kommunikationssimulation wird Veins/OMNeT++ eingesetzt, das wie zuvor erläutert eine sehr detaillierte Abbildung der Kommunikation zwischen Fahrzeugen erlaubt und zudem für die Verwendung mit SUMO ausgelegt ist (vgl. Abbildung 3). Die Car2X-Ereignisse, die in Veins/OMNeT++ auftreten (z.B. ADAS-Nachrichten), werden an SUMO zur Benachrichtigung der simulierten Fahrzeuge sowie an die 3D-Visualisierung für eine etwaige optische Benachrichtigung des menschlichen Fahrers weitergeleitet. Da der Beitrag eine Machbarkeit darstellt, wurde nicht der in Abbildung 1 links dargestellte Fahrsimulator als Plattform gewählt, sondern aus Gründen der Vereinfachung ein weniger komplexes System eingesetzt, welches im Folgenden erläutert wird. Die Erkenntnisse der Untersuchung sind jedoch übertragbar. Die 3D-Visualisierung besteht aus einer Unity3D-Anwendung, die automatisiert 3D-Szenarien auf Grundlage eines SUMO-Straßennetzmodells erzeugen kann und die Bewegungen aller Verkehrsteilnehmer inkl. des Ego-Fahrzeugs darstellt. Die Ausgabe der 3D-Visualisierung erfolgt in diesem Beitrag durch einfache Monitore oder durch eine Virtual-Reality-Brille. Als Fahrdynamikmodell wird ein komplexes Vollfahrzeugmodell verwendet, das hauptsächlich ein Zweispurmodell mit vier Freiheitsgraden (FHG) für die Horizontaldynamik (zwei translatorische FHG in der Ebene, der Gierwinkel und Schwimmwinkel) umfasst und mit einem Vertikaldynamikmodell mit elf FHG verkoppelt ist (Huben, Nicken, Wanken, vier Verdrehwinkel der Räder und vier vertikale translatorische FHG der Räder). Zusätzlich enthält das Vollfahrzeugmodell zur Abbildung der Reifenkräfte Reifenmodelle nach Pacejka sowie ein einfaches Antriebsstrangmodell, das aus Motor, Getriebe und Differential besteht. Das

³ OpenStreetMap ist ein freies Projekt, das frei nutzbare Geodaten sammelt, strukturiert und für die Nutzung durch die Allgemeinheit in einer Datenbank bereitstellt

Fahrdynamikmodell wurde in Matlab/Simulink implementiert und wird als Echtzeitsimulation mit einer Schrittweite von einer Millisekunde auf einem Windows-PC ausgeführt. Alternativ dazu ist ebenso eine Ausführung auf einer Echtzeithardware (vgl. [3]) möglich, um harte Echtzeit zu garantieren. Das Fahrdynamikmodell besitzt drei Eingänge, die von einem menschlichen Fahrer beeinflusst werden können: Den Lenkradwinkel, die Gaspedalstellung und die Bremspedalstellung (das Getriebe des Fahrdynamikmodells ist als Automatik ausgelegt). Diese Eingangsgrößen werden im einfachsten Fall über ein per USB an den PC angeschlossenes Lenkrad mit Pedalen erfasst. Die Ausgänge sind die Position und die Orientierung des Ego-Fahrzeugs, die an SUMO weitergeleitet werden.

Abbildung 4 rechts zeigt, an welcher Stelle nun EVI ansetzt wird und wie die notwendigen Informationen ausgetauscht werden. EVI nimmt hier die Rolle eines zwischengeschalteten Brokers ein, der für die Einzelsimulationen alle notwendigen Informationen erhält und sie für die Einzelsimulationen im entsprechenden Format weiterleitet. Hierbei wählt EVI die lokal zu berücksichtigen Verkehrsumgebung selbst aus. Diese Struktur ist möglich, da alle Einzelsimulationen eine Netzwerkschnittstelle per TCP/UDP bereitstellen. Als Besonderheit wird hier das von Matlab/Simulink berechnete Fahrdynamikmodell nicht direkt an SUMO, sondern an Unity3D gekoppelt, da an dieser Stelle reine Daten ohne Einsatz eines Protokolls übertragen werden. Die Implementierung von EVI nutzt ein Publish/Subscribe-Entwurfsmuster, welches durch die Nachrichtenaustausch-Bibliothek ZeroMQ [22] bereitgestellt wird.

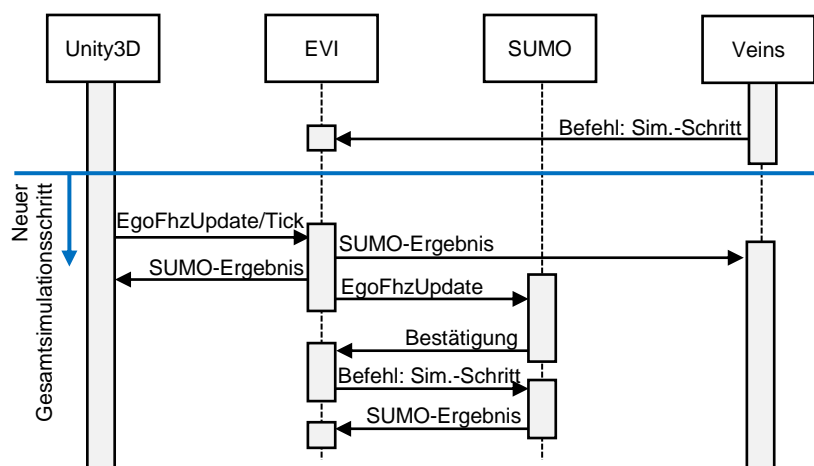


Abbildung 5: Sequenzdiagramm eines synchronisierten Simulationsschritts

Zur Strukturierung der Daten wird das zugrundeliegende TraCI-Protokoll [14] verwendet, welches die meisten notwendigen Funktionen umfasst und von SUMO und Veins bereits unterstützt wird. Jedoch sind zusätzlich Datenstrukturen notwendig, um die 3D-Visualisierung über Car2X-Ereignisse zu benachrichtigen, die durch TraCI nicht berücksichtigt werden. Da die Erweiterung des TraCI-Protokolls sehr aufwendig ist, wurde in EVI eine Datenserialisierung

mittels Google Protocol Buffers⁴ implementiert, die TraCI-Datenpakete einbettet und diese zwischen SUMO und Veins vermitteln kann, während zusätzlich beliebige weitere Daten anderer Struktur angehängt werden können. In Abbildung 5 ist das Sequenzdiagramm eines synchronisierten Simulationsschritts mit der verkoppelten Gesamtsimulation zu sehen. Zunächst sendet Veins aus dem vergangenen Simulationsschritt den Befehl für SUMO, den nächsten SUMO-Simulationsschritt zu berechnen. Diese Nachricht wird durch EVI entgegengenommen. EVI wartet solange, bis von Unity3D als Taktgeber das Signal für den nächsten Simulationsschritt eingeht. Der Befehl von Veins wird dann aber nicht an SUMO weitergeleitet, da EVI stets einen Simulationsschritt von SUMO im Voraus zwischenspeichert und das SUMO-Ergebnis (u.a. Fahrzeugpositionen) daher aus dem letzten Simulationsschritt direkt an VEINS zurückgeben kann (und zusätzlich an Unity3D). Dieser Ansatz wurde gewählt, da die Berechnungen in Veins sehr zeitaufwendig sind und die Verzögerung durch SUMO so gering wie möglich sein soll. Der Nachteil, dass Veins die Berechnungen daher auf einem zurückliegenden SUMO-Simulationsergebnis tätig, fällt für eine genügend kleine Simulationsschrittweite nicht negativ ins Gewicht. Zusätzlich erhält EVI von Unity3D die Position des Ego-Fahrzeugs, die an SUMO als Aktualisierung für das extern gesteuerte Fahrzeug weitergeleitet wird. Anschließend stößt EVI den nächsten Simulationsschritt in SUMO an und speichert daraufhin das Ergebnis für den nächsten Gesamtsimulationsschritt zwischen.

6. Untersuchung der Leistungsfähigkeit

Da SUMO und Veins ereignisorientierte Simulationen sind (d.h. es existiert keine Garantie für harte Echtzeit, da jeder Simulationsschritt unterschiedlich viel Zeit beanspruchen kann) und damit im Gegensatz zum Fahrdynamikmodell nicht echtzeitfähig Ergebnisse liefern, ist es notwendig, dass EVI diesen Umstand bei der Synchronisierung berücksichtigt. Als Echtzeitanforderung wurde für das System im Rahmen dieses Beitrags eine maximale Gesamtschrittdauer von 100 ms festgelegt, innerhalb der die Gesamtsimulation als echtzeitfähig betrachtet wird (die Immersion bleibt für den menschlichen Fahrer dann erhalten). Die Einhaltung dieser festgelegten Grenze wird im Folgenden untersucht. In [3] wurde bereits festgestellt, dass Veins im Vergleich zu SUMO (8–15 ms) einen hohen Rechenbedarf erfordert und daher neben dem Fahrdynamikmodell (1 ms pro Simulationsschritt) und der Unity3D-Visualisierung (ca. 17 ms) als kritischer Engpass für die Echtzeitanforderung zu sehen ist. Für die Untersuchung der Leistungsfähigkeit wurden zwei Versuche durchgeführt. Im ersten Versuch wurde in Veins eine Pseudo-Anwendung implementiert, die eine definierte Zeit pro Simulationsschritt wartet (statische und gleichverteilte Last). Dadurch wurde untersucht, ab welcher benötigten Rechenzeit das Gesamtsystem nicht mehr echtzeitfähig ist und die Immersion der Fahrsimulation stört.

⁴ <https://developers.google.com/protocol-buffers/>



Abbildung 6: Straßennetausschnitt aus dem nördlichen Paderborn
 Links: Ausschnitt aus OpenStreetMap mit gefahrener Strecke, Mitte: 3D-Umgebung in Unity3D,
 Rechts: SUMO-Straßennetz

Zusätzlich wurde als zweiter Versuch eine virtuelle Rundfahrt von einem menschlichen Testfahrer in einem Szenario mit intelligentem Fremdverkehr (5 bis 25 Fremdverkehrsfahrzeuge) und Car2X-Kommunikation durchgeführt und zu Vergleichszwecken aufgezeichnet (s. Abbildung 6 links). In Tabelle 2 sind die für die Untersuchung verwendeten PCs aufgelistet.



Abbildung 7: Screenshot der 3D-Visualisierung

Tabelle 2: Komponenten der Simulations-PCs

Komponente	Windows 7 PC	Ubuntu 16.04 LTS PC
CPU	Intel Core i7-3770K @ 3.5 GHz	Intel Core i7-7700K @ 4.2 GHz
RAM	16 GB	16 GB
GPU	Nvidia GTX 650 Ti	Keine dedizierte Grafik
Simulationen	Matlab/Simulink R2015b, Unity3D 5.6.0f3	SUMO 0.30.0, Veins/OMNeT++ 5.1.1

Die Untersuchung hat für eine statische Last ergeben, dass Veins bis zu 95 ms (95%) der Gesamtsimulationsschrittzeit nutzen kann, ohne dass die Gesamtsimulation die Echtzeitfähigkeit verliert (vgl. Abbildung 8 links). Dasselbe Ergebnis ist zu erkennen, wenn eine gleichverteilte Last für Veins angenommen wird, die im Mittelwert jedoch ebenfalls nicht mehr als 95 ms nutzt. So ist das Gesamtsystem in einem solchen Fall in der Lage, nach mehreren Simulationsschritten mit jeweils mehr als 95 ms an Zeitbedarf wieder die Echtzeit zu erlangen und Verluste zu kompensieren. Für die Untersuchung eines einfachen Anwendungsszenarios, in dem ein menschlicher Fahrer mittels Lenkrad, Pedalen und einer Virtual-Reality-Brille eine virtuelle Testfahrt mit Fremdverkehr und einer Car2X-Kommunikation durchführt, wurde eine einfache 10Hz-Beaconing-Anwendung (Lokaler Nachrichten-Broadcast) implementiert, die Informationen zwischen den Fahrzeugen vermittelt (wie beispielweise lokal auftretende Verkehrsstörungen). Hierbei konnte festgehalten werden, dass die Anzahl der Fremdverkehrsfahrzeuge und die dadurch benötigte Berechnungszeit der Car2X-Kommunikation von Veins die limitierenden Faktoren der Gesamtsimulation sind, wie

in Abbildung 8 rechts dargestellt. Für die verwendete Hardware (vgl. Tabelle 2) stellte sich heraus, dass bis zu 20 Fremdverkehrsfahrzeuge echtzeitfähig mittels der Gesamtsimulationsumgebung simuliert werden können.

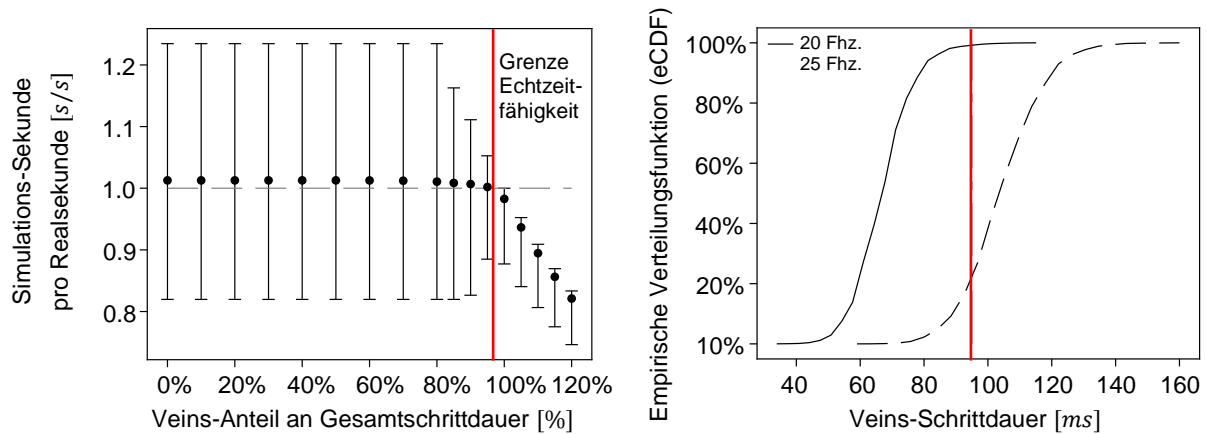


Abbildung 8: Ergebnisse der simulativen Untersuchung der Leistungsfähigkeit
Links: Statische Last, Rechts: Virtuelle Testfahrt

7. Fazit

In diesem Beitrag wurde das Konzept für eine komplexe Gesamtsimulationsumgebung gezeigt, die eine Kombination mehrerer Einzelsimulationsumgebungen ermöglicht. Hierzu zählen eine Simulation für intelligenten Fremdverkehr, eine detaillierte Car2X-Kommunikation für eine Beispielanwendung mit 10 Hz, ein komplexes Fahrdynamikmodell für das Ego-Fahrzeug und eine dynamischen 3D-Visualisierung. Die Komponenten unterscheiden sich in der Art der Simulation (ereignisorientierte vs. Echtzeit-Simulation) und wurden mittels einer implementierten Schnittstelle (EVI) synchronisiert und verkoppelt. Die resultierende Gesamtsimulationsumgebung konnte in Versuchen als echtzeitfähig für bis zu 20 Fremdverkehrsfahrzeuge betrachtet werden. Untersuchungen für den Einsatz mit komplexeren Fahrsimulatoren stellen die nächsten möglichen Herausforderungen dar.

Literatur:

- [1] C. Sommer, F. Dressler, „Vehicular Networking“, Cambridge University Press, Nov. 2014, DOI: 10.1017/CBO9781107110649
- [2] C. Sommer, R. German, F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis”, IEEE Transactions on Mobile Computing, vol. 10, no. 1, pp. 3–15, Jan. 2011. DOI: 10.1109/TMC.2010.133
- [3] D. S. Buse, M. Schettler, N. Kothe, P. Reinold, C. Sommer, F. Dressler, “Bridging Worlds: Integrating Hardware-in-the-Loop Testing with Large-Scale VANET Simulation“, in 14th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2018), Isola 2000, Frankreich: IEEE, Feb. 2018. DOI: 0.23919/WONS.2018.8311659
- [4] J. Heinovski, L. Stratmann, D. S. Buse, F. Klingler, M. Franke, M.-C. H. Oczko, C. Sommer, I. Scharlau, F. Dressler, "Modeling Cycling Behavior to Improve Bicyclists' Safety at Intersections – A Networking Perspective", Proceedings of 20th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2019), Washington, DC, Juni 2019
- [5] D. S. Buse, C. Sommer, F. Dressler, "Integrating a Driving Simulator with City-Scale VANET Simulation for the Development of Next Generation ADAS Systems," Proceedings of 37th IEEE Conference on Computer Communications (INFOCOM 2018), Demo Session, Honolulu, HI, April 2018

- [6] S. Henning, P. Biemelt, N. Rüdtenklau, S. Gausemeier, A. Trächtler, „A Simulation Framework for Testing a Conceptual Hierarchical Autonomous Traffic Management System including an Intelligent External Traffic Simulation“, Proceedings of the DSC 2018 Europe VR: New trends in Human in the Loop simulation and testing. Driving simulation and VR, S. 91-98, Frankreich, 2018 Driving Simulation Conference, Driving Simulation Association
- [7] P. Biemelt, S. Henning, N. Rüdtenklau, S. Gausemeier, A. Trächtler, „A Model Predictive Motion Cueing Strategy for a 5-Degree-of-Freedom Driving Simulator with Hybrid Kinematics“, Proceedings of the DSC 2018 Europe VR: New trends in Human in the Loop simulation and testing. Driving simulation and VR, S. 79-85, Frankreich, 2018 Driving Simulation Conference, Driving Simulation Association
- [8] K. Abdelgawad, B. Hassan, J. Berssenbrügge, J. Stöcklein, M. Grafe, „A Modular Architecture of an Interactive Simulation and Training Environment for Advanced Driver Assistance Systems“, International Journal On Advances in Software, vol. 8, no. 1 & 2, pp. 247–261, Jun. 2015
- [9] K. Abdelgawad, M. Abdelkarim, B. Hassan, M. Grafe, I. Gräßler, „A modular architecture of a PC-based driving simulator for advanced driver assistance systems development“, 15th International Workshop on Research and Education in Mechatronics (REM), El Gouna, Ägypten: IEEE, Sep. 2014. DOI: 10.1109/REM.2014.6920237
- [10] A. Shaout, D. Colella, S. Awad, "Advanced driver assistance systems-past, present and future", 2011 Seventh International Computer Engineering Conference (ICENCO'2011), Ägypten: IEEE, DOI: 10.1109/ICENCO.2011.6153935
- [11] A. Komzalov, N. Shilov, "Driver Assistance Systems: State-of-the-Art and Possible Improvements", Proceedings of the 20th Conference of Open Innovations Association FRUCT, LETI University, St.Petersburg, Russland, 2014, ISSN 2305-7254, ISBN 978-952-68653-0-0
- [12] L. Wischhof, A. Ebner, H. Rohling, M. Lott, R. Halfmann, „SOTIS – A Self-Organizing Traffic Information System“, 57th IEEE Vehicular Technology Conference (VTC2003-Spring), Jeju, Südkorea: IEEE, Apr. 2003, pp. 2442–2446. DOI: 10.1109/VETECS.2003.1208829
- [13] P. A. Lopez; M. Behrisch; L. Bieker-Walz; J. Erdmann; Y.-P. Flötteröd; R. Hilbrich; L. Lücken; J. Rummel; P. Wagner; E. Wießner, "Microscopic Traffic Simulation using SUMO", in 21st IEEE International Conference 2018, Maui, USA, 02.11.2018, <https://elib.dlr.de/124092/>
- [14] A. Wegener, M. Piorkowski, M. Raya, H. Hellbrück, S. Fischer, J.-P. Hubaux, „TraCI: An Interface for Coupling Road Traffic and Network Simulators“, in 11th Communications and Networking Simulation Symposium (CNS 2008), Ottawa, Kanada: ACM, Apr. 2008, pp. 155–163. DOI: 10.1145/1400713.1400740
- [15] A. Varga, „The OMNeT++ Discrete Event Simulation System“, in European Simulation Multiconference (ESM 2001), Prag, Tschechien, Jun. 2001
- [16] S. Guan, R. E. De Grande, A. Boukerche, „Real-time 3D Visualization for Distributed Simulations of VANets“, in 18th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, Toulouse, France: IEEE, Oct. 2014, pp. 138–146. DOI: 10.1109/DS-RT.2014.25
- [17] D. Gruyer, O. Orfila, V. Judalet, S. Pechberti, B. Lusetti, S. Glaser, „Proposal of a Virtual and Immersive 3D Architecture dedicated for Prototyping, Test and Evaluation of Eco-Driving Applications“, IEEE Intelligent Vehicles Symposium (IV'13), Gold Coast, QLD, Australien: IEEE, Jun. 2013, pp. 511–518. DOI: 10.1109/IVS.2013.6629519
- [18] W. M. Griggs, R. H. Ordóñez-Hurtado, E. Crisostomi, F. Häusler, K. Massow, R. N. Shorten, „A Large-Scale SUMO-Based Emulation Platform“, IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3050–3059, Dez. 2015. DOI: 10.1109/TITS.2015.2426056
- [19] F. Michaeler C. Olaverri-Monreal, „3D Driving Simulator with VANET Capabilities to Assess Cooperative Systems: 3DSimVanet“, in IEEE Intelligent Vehicles Symposium (IV'17), Redondo Beach, CA, USA: IEEE, Jun. 2017, pp. 999–1004. DOI: 10.1109/IVS.2017.7995845
- [20] H. Prendinger, M. Miska, K. Gajananan, A. Nantes, „A Cyber-Physical System Simulator for Risk-Free Transport Studies“, Computer-Aided Civil and Infrastructure Engineering, vol. 29, no. 7, pp. 480–495, Aug. 2014. DOI: 10.1111/mice.12068
- [21] Y. Hou, Y. Zhao, A. Wagh, L. Zhang, C. Qiao, K. F. Hulme, C. Wu, A. W. Sadek, X. Liu, „Simulation-Based Testing and Evaluation Tools for Transportation Cyber-Physical Systems“, IEEE Transactions on Vehicular Technology, vol. 65, no. 3, pp. 1098–1108, März 2016. DOI: 10.1109/TVT.2015.2407614
- [22] P. Hintjens, „ZeroMQ: Messaging for Many Applications“, O'Reilly Media, 2013
- [23] M. Haklay, P. Weber, „OpenStreetMap: User-Generated Street Maps“, IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, Oct. 2008. DOI: 10.1109/MPRV.2008.80
- [24] C. Obermaier, C. Facchi, „Observations on OMNeT++ Real-Time Behaviour“, in 4th OMNeT++ Community Summit (OMNeT++ 2017), Bremen, Germany: arXiv, Sep. 2017
- [25] J. Berssenbrügge, A. Trächtler, C. Schmidt, „Visualization of Headlight Illumination for the Virtual Prototyping of Light-Based Driver Assistance Systems“, Proceedings of the ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Boston, MA, USA, Aug. 2015