

Verkehrssimulation im Hardware-in-the-Loop-Steuergerätetest

Dr.-Ing. Peter **Reinold**, Norbert **Meyer**, M. Sc.

dSPACE GmbH

Dominik **Buse**, M. Sc., Florian **Klingler**, M. Sc.,
Prof. Dr.-Ing. Christoph **Sommer**, Prof. Dr.-Ing. habil. Falko **Dressler**

Heinz Nixdorf Institut und Institut für Informatik, Universität Paderborn

Markus **Eisenbarth**, M. Sc., Prof. Dr.-Ing. Jakob **Andert**

Mechatronische Systeme am Verbrennungsmotor, RWTH Aachen University

Einleitung und Motivation

In der Fahrzeugentwicklung stellen Hardware-in-the-Loop (HIL)-Systeme den Stand der Technik für die Absicherung von Regler- und Steuergerätealgorithmen dar [1]. Für entsprechende Closed-Loop-Szenarien werden echtzeitfähige Simulationsmodelle eingesetzt. Zum Testen von Advanced Driver Assistance Systems (ADAS) oder auch autonomen Fahrfunktionen sind Simulationsmodelle erforderlich, die neben dem Fahrzeug notwendigerweise auch ein geeignetes Modell der Fahrzeugumgebung mit anderen Verkehrsteilnehmern umfassen. Für klassische Assistenzfunktionen werden diese Verkehrsteilnehmer dabei in konkreten, ausgewählten und vorab definierten Situationen simuliert. So lassen sich eindeutig spezifizierbare Testsznarien abprüfen. Die Einbindung realer Antriebsstrangkomponenten, beispielsweise von Verbrennungsmotoren mitsamt entsprechenden Prüfständen, ist dabei eine Möglichkeit, um den realen Kraftstoffverbrauch oder auch das Emissionsverhalten bei Fahrten in einer simulierten Umgebung messbar zu machen [2].

Die Entwicklung und Absicherung von Funktionen hochautomatisierter und miteinander kommunizierender Fahrzeuge stellt steigende Anforderungen an Umgebungsmodelle. Diese resultieren zum einen aus den quantitativ steigenden Aufwänden, um entsprechende Algorithmen und Steuergeräte in unterschiedlichsten Situationen zu testen und zu validieren. Zum anderen steigen auch die qualitativen Anforderungen an solche Tests, da die zu testenden Szenarien nicht nur vielfältiger, sondern auch deutlich komplexer und weniger klar definierbar werden. Es ist erforderlich, realitätsnahe Verkehrssituationen zu simulieren, bei denen eine Interaktion zwischen den Verkehrsteilnehmern erfolgt [3].

Mikroskopische Verkehrssimulatoren simulieren Verkehr unter Berücksichtigung der Interaktion zwischen den einzelnen Verkehrsteilnehmern. Dabei wird eine große Anzahl an Fahrzeugen fahrdynamisch vereinfacht simuliert. Die Kommunikation der Verkehrsteilnehmer wird über eine geeignete Netzwerksimulation dargestellt. Integrierte Simulatoren kombinieren eine solche Verkehrssimulation mit einer Netzwerksimulation und ermöglichen es so, die erforderlichen Testsituationen abzubilden [4].

Sowohl HIL-basierte Systeme als auch die integrierte Verkehrs- und Netzwerksimulation werden üblicherweise unabhängig voneinander in ihren Einsatzfeldern als Einzelsysteme verwendet und bieten den hier erforderlichen Funktionsumfang. Doch erst eine Kopplung beider Werkzeuge ermöglicht die Absicherung von ADAS-Funktionen in komplexen, interagierenden Szenarien unter Einbindung von Hardware-Komponenten.

Dieser Beitrag erläutert, wie durch eine Kopplung die Möglichkeiten der Verkehrs- und Netzwerksimulation für den Test von Steuergeräten am HIL-Simulator mit etablierten Werkzeugketten unter Echtzeitbedingungen erschlossen werden können.

Prüfstandskonzept und –topologie

Ziel des Forschungsprojektes Hy-Nets ist es, den Energiebedarf eines Hybridfahrzeugs durch intelligente Fahralgorithmen, die die Möglichkeiten der Fahrzeugkommunikation nutzen, zu senken. Dies erfordert eine ganzheitliche Betrachtung, bei der die Komponenten- und Funktionsentwicklung nicht nur im Kontext des Fahrzeugs, sondern des Gesamtverkehrs betrachtet werden. Diese Betrachtung erfolgt am Beispiel der Stadt Paderborn als einer typischen, mittelgroßen, europäischen Stadt [5].

Zur Bereitstellung einer hierzu geeigneten Entwicklungs- und Testplattform wird eine echtzeitfähige Fahrzeugsimulation auf einem HIL-Simulator mit einer integrierten Verkehrs- und Netzwerksimulation basierend auf dem Simulationsframework Veins [4] in Form einer Co-Simulation kombiniert. Neben den eigentlichen Fahrzeugbewegungen werden auch Fahrzeugkommunikationssignale in die Echtzeitsimulation übertragen und können hier einem DUT (Device under Test) bereitgestellt werden. Als Echtzeitsimulator wird dSPACE SCALEXIO verwendet.

Um die Auswirkungen der Fahrfunktionen auf den Kraftstoffverbrauch insbesondere im transienten Betrieb durch Messungen evaluieren zu können, werden reale Antriebsmaschinen des Hybridantriebsstrangs (ein Verbrennungs- und ein Elektromotor) in die Co-Simulation eingebunden. Die Antriebsmaschinen werden auf räumlich verteilten Prüfständen am Center for Mobile Propulsion der RWTH Aachen University betrieben und deren Steuergeräte mit dem HIL-Simulator verbunden [6]. Steuergeräte und Maschinen sind so mit dem HIL-Simulationsmodell gekoppelt, dass sie Teil eines virtuellen Hybridantriebs in P3-Topologie sind. Bild 1 illustriert die entsprechende Prüfstandstopologie.

Sowohl der Verbrennungsmotor als auch der Elektromotor sind mechanisch mit den Dynamometern ihrer Prüfstände verbunden, die die in der Simulation berechneten Lasten mit Hilfe der zugehörigen Prüfstandsautomatisierungssysteme aufprägen. Das Restfahrzeug wird auf dem HIL-Simulator mit seinen Umfeldsensoren und seiner lokalen Umgebung (Straße, Fahrer und Verkehr) mit Hilfe der *Automotive Simulation Models (ASM)* von dSPACE simuliert. Das DUT ist im Projektkontext der betrachtete Fahralgorithmus. Diese Fahrfunktion bestimmt nicht nur die Hybridbetriebsstrategie, sondern übernimmt auch die Fahrzeugführung, um den Energiebedarf zu minimieren. Prinzipiell können am gleichen Aufbau auch reale ADAS-Steuergeräte getestet werden, die sich über entsprechende elektrische Schnittstellen mit dem HIL-Simulator verbinden lassen.

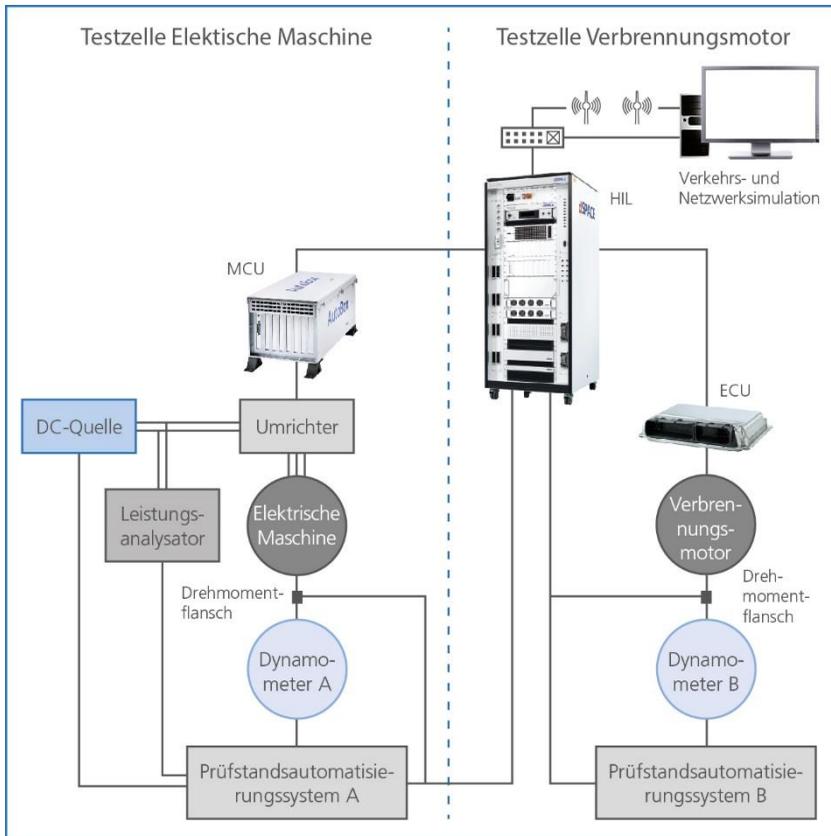


Bild 1: Prüfstandstopologie am Center for Mobile Propulsion in Aachen

Durch diese Kopplung entstehen Wechselwirkungen, sodass der Einfluss der simulierten Fahrzeugkommunikation und des Verkehrs auf den Energiebedarf und die Emissionen am Prüfstand als reale physikalische Größen messbar werden. Auf diese Weise können Fahrfunktionen in komplexen, realitätsnahen Verkehrsszenarien getestet werden, deren einzelne Teilnehmer miteinander interagieren und kommunizieren. Dies ermöglicht den für das hochautomatisierte Fahren erweiterten Umfang an Steuergerätetests. Andere Anwendungen eines solchen Aufbaus können im Bereich von Real-Driving-Emissions (RDE)-Tests liegen, bei denen es wichtig ist, den realen Kraftstoffverbrauch und Emissionen in verschiedensten realistischen Verkehrssituationen zu messen.

Verkehrssimulation am HIL-Prüfstand

Um an einen solchen HIL-Prüfstand eine Verkehrs- und Netzwerksimulation anbinden zu können, ist ein geeignetes Co-Simulations- und Synchronisationskonzept erforderlich. Dieses Konzept, die Modellkomponenten und ihr Zusammenwirken werden im Folgenden erläutert.

Das Ego-Fahrzeug und seine nähere Umgebung werden durch die dSPACE Modellbibliothek ASM auf dem Echtzeitsimulator dSPACE SCALEXIO simuliert, um den Anschluss der Realkomponenten zu ermöglichen. Die integrierte Verkehrs- und Netzwerksimulation der Fahrzeugkommunikation basiert auf dem Simulationsframework Veins. Sie wird nicht auf einem dedizierten Echtzeitsystem ausgeführt sondern auf einem separaten Linux-Standardrechner. Diese Simulation wird als globale Umgebungssimulation betrachtet, mit der das Nahumfeld um das Ego-Fahrzeug, also die lokale Umgebungssimulation auf SCALEXIO, synchronisiert werden muss.

Die Kopplung zwischen diesem Linux-PC mit der globalen Umgebungssimulation und dem Echtzeitsystem erfolgt bilateral über Ethernet, basierend auf Google Protocol Buffers. Der Takt und die Uhr des Echtzeitsystems werden als die einzigen verlässlichen Zeitgeber im Gesamtsystem betrachtet, sodass die globale Umgebungssimulation nach ihnen ausgerichtet werden muss. Das echtzeitfähige Ego-Fahrzeugmodell wird mit einer konstanten Schrittweite von 1 ms gerechnet, während das globale Umgebungsmodell mit deutlich größeren Schrittweiten (hier mit 100 ms) simuliert werden kann. Zur Synchronisation wird die Bewegung des Ego-Fahrzeugs alle 100 Schritte, das heißt im 100-ms-Raster, an die globale Umgebungssimulation gesendet und hier vom *Ego Vehicle Interface (EVI)* entgegengenommen. Das EVI sendet vorab berechnete Bewegungen der relevanten Teilmenge von Umgebungsfahrzeugen aus dem Nahumfeld des Ego-Fahrzeugs zurück, die zum gleichen Zeitpunkt wie die erhaltene Ego-Fahrzeugbewegung Gültigkeit besitzen. Auf diese Weise synchronisiert sich die globale Umgebungssimulation mit dem Takt des Echtzeitsystems.

Die aus der globalen Umgebungssimulation erhaltenen Bewegungen werden genutzt, um die Bewegung der Umgebungsfahrzeuge (Fellows) in der Umgebungssimulation des Ego-Fahrzeugs vorzugeben. Diese lokale Umgebungssimulation des Ego-Fahrzeugs wird auf dem HIL-Simulator ausgeführt. Somit wird im Nahumfeld des Ego-Fahrzeugs die gleiche Verkehrssituation abgebildet wie in der globalen Umgebungssimulation. Durch die konkrete Ausprägung der Fellows in ASM können diese nun durch Sensoren des Ego-Fahrzeugs detektiert und für die Fahrzeugführung berücksichtigt werden. Hierdurch wird die Co-Simulationsschleife zwischen Ego-Fahrzeug- und Verkehrssimulation geschlossen, sodass eine wechselseitige Interaktion entsteht. Das simulierte Ego-Fahrzeug kann durch eine Umgebung mit interagierenden Umgebungsfahrzeugen bewegt werden.

Innerhalb der globalen Umgebungssimulation stellt das EVI diese Ego-Fahrzeugposition zum einen der mikroskopischen Verkehrssimulation SUMO zur Verfügung, die das Ego-Fahrzeug als einen Teilnehmer im globalen Verkehrsfluss berücksichtigt, zum anderen wird sie auch in der Netzwerksimulation verwendet. Diese generiert standard-konforme Car2X-Signale, die als Funksignale „over-the-air“ gesendet werden. Am SCALEXIO-System werden diese Signale empfangen und den Fahralgorithmen durch das dSPACE V2X Blockset zur Verfügung gestellt. Hierzu stehen sowohl am SCALEXIO-System als auch am Rechner mit der globalen Umgebungssimulation Antennen zur Verfügung, über die Fahrzeugkommunikationssignale (sog. CAMs) gesendet werden.

Bild 2 illustriert die resultierende Topologie des Gesamtmodells.

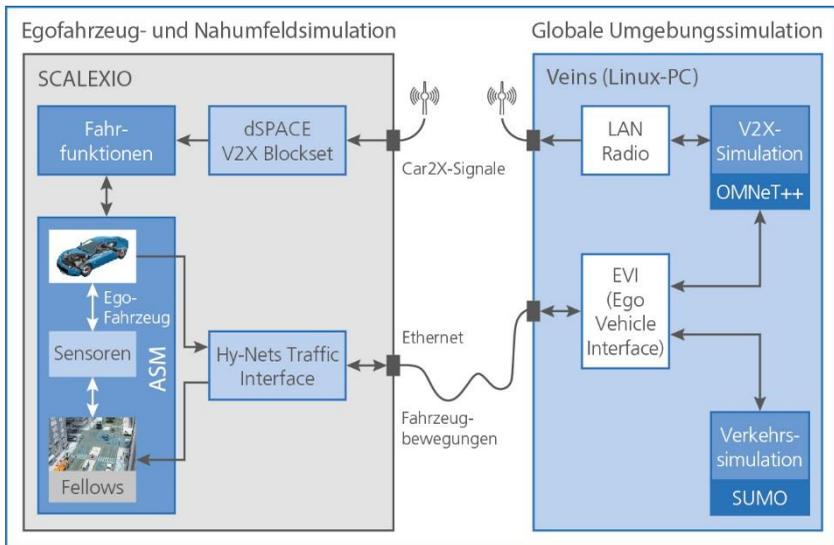


Bild 2: Gesamtmodell-Topologie

Die Ethernetverbindung wird neben der Bewegungsübertragung auch für die Synchronisation weiterer Größen genutzt, zum Beispiel für die der Ampelsignale. Zusätzlich können den Fahr-funktionen hierüber Signale, die in der globalen Umgebungssimulation erzeugt werden, direkt zur Verfügung gestellt werden. Im Projektkontext sind dies Durchschnittsgeschwindigkeiten auf konkreten Straßen oder Ampelschaltzeiten. Der Datenaustausch über diese Schnittstelle kann um weitere Größen erweitert werden.

Ego-Fahrzeugsimulation und Umgebungsparametrierung

Bei den für die Simulation des Ego-Fahrzeugs verwendeten *Automotive Simulation Models (ASM)* von dSPACE handelt es sich um eine Toolsuite auf Basis offener MATLAB®/Simulink®-Modelle. Diese offene Modellstruktur ermöglicht es, Anpassungen vorzunehmen und die in diesem Projekt betrachtete P3-Hybridtopologie abzubilden. In gleicher Weise könnten auch andere Antriebsstrangtopologien aufgebaut werden. Die Flexibilität der offenen Modelle erlaubt es außerdem, die zu testenden Fahralgorithmen leicht zu integrieren und die für die erläuterte Prüfstandsankopplung erforderlichen Schnittstellen für den Signalaustausch mit Steuergeräten und den Prüfstandsautomatisierungssystemen zu schaffen.

Neben der Synchronisation der Fahrzeugbewegung zur Simulationszeit, auf die im nachfolgenden Abschnitt detaillierter eingegangen wird, ist es erforderlich, die Offline-Daten der Umgebungssimulation in ASM und der Verkehrssimulation SUMO, die Teil der globalen Umgebungssimulation ist, zu synchronisieren. Im Wesentlichen betrifft das die Straßendaten. Aus der freien Geodatenbank *OpenStreetMap (OSM)* können sowohl für ASM als auch für SUMO Karten konvertiert werden [3], [5]. Allerdings beschreiben die in OSM enthaltenen Kartendaten nur Stützstellen und nicht den exakten geometrischen Verlauf der realen Straße. Die Anforderungen an die Straßengeometrie sind aufgrund des verschiedenen Einsatzes von ASM und SUMO unterschiedlich; entsprechend unterscheidet sich die aus den gleichen Stützstellen generierte Straßengeometrie nach der Konvertierung im Detail voneinander. In SUMO wird der Straßenverlauf durch Polygonzüge abgebildet. Ein so resultierender Polygonzug ist jedoch für die Simulation mit einer Fahrdynamiksimulation ungeeignet, da es an den Eckpunkten des Polygonzugs zu Krümmungssprüngen kommt. Auf die abstrahierte Fahrzeugbewegung in der Verkehrssimulation haben sie keinen Einfluss. In realen Straßenverläufen existieren Krümmungssprünge nicht. Aus diesem Grund werden in ASM Straßen aus den Grundformen Gerade, Kreisbogen, Klothoide und Spline mit stetigen Übergangsbedingungen definiert. Das Resultat einer Konvertierung aus OSM ist in ASM eine aus Splines aufgebaute Straße. Diese weist jedoch zwischen den Stützstellen einen etwas anderen Verlauf auf, als die aus Polygonzügen aufgebaute SUMO-Straße. Um zwei möglichst ähnliche Straßengeometrien zu erhalten, werden die Straßen daher nicht unabhängig voneinander aus OSM in eine SUMO- und eine ASM-Straße konvertiert, sondern der in Bild 3 illustrierte Workflow angewandt.

Zunächst wird der relevante Kartenausschnitt aus OSM in eine SUMO-Straße konvertiert (1). Hintergrund ist, dass die SUMO-Straße in eine ASM-Straße konvertiert werden soll, da in typischen Anwendungsfällen in der globalen Umgebungssimulation ein Netzwerk betrachtet wird, das ebenso groß oder größer als das in der Simulation des Nahbereichs um das Ego-Fahrzeug ist. Diese aus Polygonzügen aufgebaute SUMO-Straße wird anschließend einem Postprocessing unterzogen (2). Dabei werden zwischen

den Stützstellen des Polygonzugs Zwischenwerte im Verlauf eines Splines berechnet. Die resultierende Straße wird für die Simulation in SUMO verwendet (3). Sie hat den Verlauf eines Polygonzugs, der in seiner Form aber nahezu dem Verlauf von Splines entspricht. In der unteren Darstellung von Bild 3 ist der Unterschied durch dieses Post-processing zu erkennen. Der relevante Bereich dieser Straße wird in eine ASM-Straße konvertiert (4). So ergeben sich annähernd identische Straßengeometrien in ASM und SUMO, die sich für die Co-Simulation eignen. Da die Stützstellen erhalten bleiben und nur durch berechnete Zwischenwerte angereichert werden, findet keine Verfälschung der Rohdaten statt. Auch das OpenDrive-Format kann zum Austausch der Straßen verwendet, da beide Tools sowohl den Import als auch den Export von OpenDrive-Straßen unterstützen. Da Straßennetze in SUMO aus Einbahnstraßen aufgebaut sind, führt eine Konvertierung aus SUMO über das OpenDrive-Format in ASM zu einem Straßennetzwerk, das ebenfalls aus Einbahnstraßen aufgebaut ist. So entsteht ein unnötig komplexes Netzwerk um Straßen mit beiden Fahrtrichtungen abzubilden.

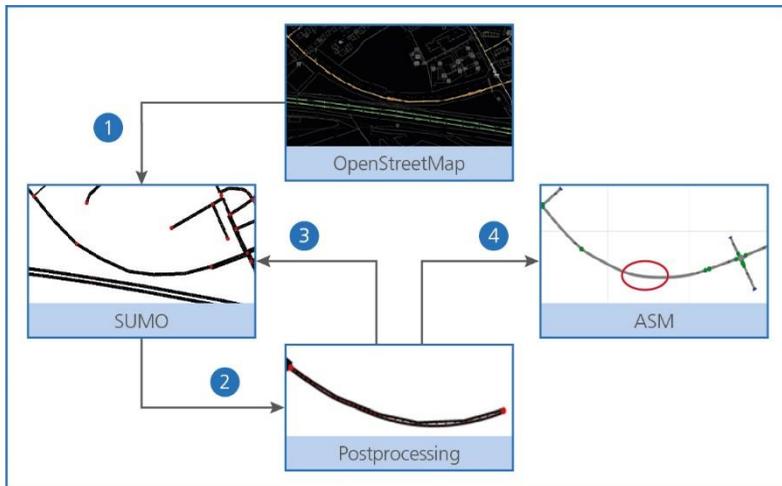


Bild 3: Straßenkonvertierung¹

¹ Contains information from Open Street Map (openstreetmap.de) which is made available here under the Open Database License (ODbL) opendatacommons.org/licenses/odbl/

Datenaustausch zur Simulationszeit

Um auf den im vorherigen Kapitel synchronisierten Straßen eine Co-Simulation durchzuführen, ist es erforderlich, zur Simulationszeit Daten auszutauschen. Nach einer Initialisierung wird die Bewegung des Ego-Fahrzeugs alle 100 ms an die globale Umgebungssimulation übertragen. Hierzu werden die relevanten Daten serialisiert als Google Protocol Buffers Message versendet. Im Gegenzug werden ebenfalls alle 100 ms die Bewegungssignale des relevanten Umgebungsverkehrs empfangen. Diese Google Protocol Buffers Messages werden dekodiert und als Simulink-Busse im Modell zur Verfügung gestellt.

Zur Abbildung des Verkehrs in der näheren Umgebung des Ego-Fahrzeugs wird ein Verkehrsszenario mit 30 Umgebungsfahrzeugen vorgehalten. Dabei können verschiedene Fahrzeugtypen berücksichtigt werden. ASM ermöglicht es, unendlich viele Umgebungsfahrzeuge zu simulieren, sodass auch eine deutlich größere Anzahl der Fahrzeuge aus der Verkehrssimulation synchronisiert werden können. Da jedoch die Anzahl der Umgebungsfahrzeuge in der unmittelbaren Umgebung des Ego-Fahrzeugs begrenzt ist und nur diese im Erfassungsbereich der Sensorik liegen bzw. mit diesen eine unmittelbare Interaktion erfolgt, wird die Anzahl zu Gunsten der Performance begrenzt. Aus der globalen Umgebungssimulation werden die relevanten, maximal 30 Fahrzeuge übertragen. Ein geeigneter Algorithmus stellt sicher, dass ein identisches Fahrzeug aus der globalen Umgebungssimulation immer dem gleichen Fahrzeug im ASM-Verkehr zugeordnet wird, solange dieses in der Synchronisation berücksichtigt wird. Auf der ASM-Seite erfolgt eine Reinstanziierung, wenn ein Fahrzeug nicht mehr übertragen wird, sodass es für weitere Fahrzeuge frei wird.

Da die Fahrzeugbewegung aus der globalen Umgebungssimulation nur alle 100 ms übertragen wird, stehen auch nur zu diesen diskreten Zeitpunkten aktualisierte Werte für die lokale ASM-Umgebung zur Verfügung. Würden diese Bewegungen unmittelbar den Fellows aufgeprägt, würden sowohl die hier simulierten als auch möglicherweise real angeschlossene Sensoren einen unnatürlichen und nicht kontinuierlichen Positionsverlauf detektieren. Um dies zu verhindern, muss die Bewegung der Umgebungsfahrzeuge zwischen zwei Synchronisationsschritten präzisiert werden. So resultiert in der durch ASM mit einer Schrittweite von 1 ms simulierten lokalen Umgebungssimulation eine quasi-kontinuierliche Bewegung. Diese kann dann durch die Sensormodelle adäquat detektiert werden, die Teil des Ego-Fahrzeugmodells sind. Der Antriebsstrang des Ego-Fahrzeugs besteht zum Teil aus realen Antriebsmaschinen auf ihren Prüfständen (Bild 1). Die Sensorinformationen können zur Fahrzeugführung im DUT genutzt werden, sodass das Ego-Fahrzeug geeignet auf den Umgebungsverkehr reagiert. Bild 4 illustriert diesen Datenaustausch.

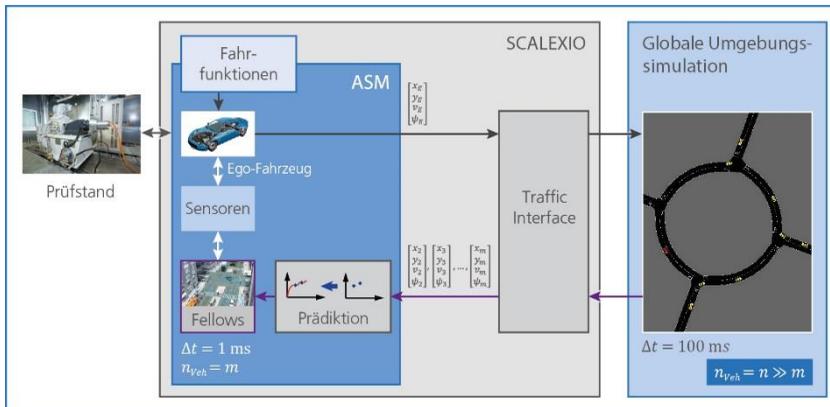


Bild 4: Datenaustausch

Verkehrs- und Car2X-Simulation

Das globale Umgebungsmodell im vorgestellten Gesamtsystem wird von Software übernommen, die auf der Simulationsplattform Veins [4] aufbaut. Im Wesentlichen besteht das Umgebungsmodell aus vier Komponenten:

- Simulation von Car2X-Kommunikation
- Simulation von Straßenverkehr
- Echtzeitfähige Kopplung des Umgebungsmodells mit dem des Ego-Fahrzeugs
- Senden simulierter Car2X-Funkkommunikation mittels realer Hardware

Alle vier Komponenten werden im Folgenden kurz vorgestellt.

Zur Simulation von Car2X-Kommunikation und Straßenverkehr existieren in der Literatur eine Vielzahl von Ansätzen [7]. Das hier vorgestellte Gesamtsystem setzt auf die Simulationsplattform Veins², eine in der heutigen Forschungslandschaft etablierte Referenz zur Simulation von Fahrzeugnetzen. Veins integriert insbesondere detaillierte Simulationsmodelle des Medienzugriffs und Drahtloskanals. Damit bildet Veins nicht nur Latenz- und Sättigungseffekte, sondern etwa auch Ausbreitungs- und Abschattungseffekte in der Stadt ab, denen die Funkkommunikation nach derzeitigen Standards für Car2X-Kommunikation [8] unterworfen ist. Alle verwendeten Modelle in Veins

² <http://veins.car2x.org>

wurden in Feldversuchen validiert, teils basierend auf der quelloffenen und frei verfügbaren Entwicklungsplattform für Car2X-Protokolle in Europa: *OpenC2X* [9] [10].

Für die Simulation von Fahrzeugbewegungen verwendet Veins das Simulationswerkzeug SUMO [11]. Es modelliert auf mikroskopischer Ebene, das heißt auf Granularitätsstufe einzelner Autos, den Verkehr und die Infrastruktur einer Stadt. Dabei werden sowohl Abfahrtzeiten und Routenwahl modelliert als auch die Reaktion jedes einzelnen Autos auf umgebenden Verkehr. Ebenso implementiert SUMO Überholmanöver sowie Beachtung von Vorfahrtsregelungen und Lichtsignalanlagen, darunter auch solche mit bedarfsabhängiger Schaltung durch Induktionsschleifen. Das Verhalten einzelner Autos lässt sich dabei im Laufe einer Simulation entsprechend den zu untersuchenden Ansätzen dynamisch anpassen, um gezielt zu untersuchende Verkehrssituationen herbeizuführen. Beispielsweise kann so das Eintreffen von Querverkehr passgenau zur Ankunft des simulierten Ego-Fahrzeugs an einer Einmündung erreicht werden. Die exakte Modellierung des Straßenverkehrs einer Stadt gestaltet sich durch die nötige Detailtreue sehr zeitintensiv, weshalb nur wenige frei verfügbare Szenarien, etwa LuST [12], existieren. Im Projekt wird ein selbst entwickeltes Modell des Straßenverkehrs der Stadt Paderborn eingesetzt.

Die echtzeitfähige Kopplung des Umgebungsmodells mit dem des Ego-Fahrzeugs übernimmt eine dedizierte Software-Komponente von Veins, das *Ego Vehicle Interface (EVI)* [13]. Das EVI koordiniert den Datenaustausch und den Zeitfortschritt von Car2X-Simulation, Verkehrssimulation und echtzeitfähigem System. Im Rahmen dieses Projekts handelt es sich beim Echtzeitsystem um die submikroskopische Ego-Fahrzeugsimulation inklusive Umgebungsmodell. Zur Koordination des Zeitfortschritts wartet das EVI kontinuierlich auf Nachrichten der Echtzeitkomponente. Diesen entnimmt es aktuelle Zustandsdaten des Ego-Fahrzeugs und den Zeitstempel des Echtzeitsystems, welcher als Referenzzeit für die Simulation verwendet wird. Diese Daten werden aufbereitet und an Car2X-Simulation und Verkehrssimulation weitergeleitet. Als Antwort erhält die Echtzeitkomponente vorausberechnete Zustandsdaten des Umgebungsverkehrs. Bis zur nächsten Nachricht der Echtzeitkomponente koordiniert das EVI die Simulation des nächsten Zeitschritts und den kontinuierlichen Fortschritt der Car2X-Simulation. Alternativ kann das so eingebundene Echtzeitsystem beispielsweise auch ein Fahr Simulator [14] sein.

Die direkte Funkkommunikation des Ego-Fahrzeugs mit seiner simulierten Umgebung über Car2X-Protokolle übernimmt eine zweite dedizierte Software-Komponente von Veins, das *LAN Radio* [15]. Im Folgenden wird lediglich eine kurze Übersicht der Funktionsweise gegeben. Im Wesentlichen besteht die Komponente aus zwei Teilen: einem physischen Teil in Form eines Embedded-Systems mit 802.11p-Netzwerkkarte und einem simulierten Gegenpart, der Teil einer Car2X-Simulation ist und das Ego-Fahrzeug darstellt. Jede Nachricht, die das simulierte Ego-Fahrzeug innerhalb der Simulation,

detektiert wird aus der Simulation extrahiert und über den Drahtloskanal physisch versandt (vgl. Bild 2). Diese physische Nachricht wird streng nach den europäischen Standards für Car2X-Kommunikation kodiert und durch eine reale Hardware-Komponente empfangen, die einen Teil des auf dem SCALEXIO simulierten Ego-Fahrzeugs darstellt.

Anwendungsbeispiel – Entwicklung prädiktiver Fahrfunktionen

Energieoptimale Geschwindigkeitswahl

Die vorgestellte Testumgebung ermöglicht die Bereitstellung realitätsnaher Verkehrsszenarien unter Berücksichtigung zukünftiger Fahrzeugkommunikationstechnologien. Darüber hinaus kann das Fahrzeugverhalten in einer kontrollierten Umgebung detailliert untersucht und der Kraftstoffverbrauch sowie die Schadstoffemissionen im dynamischen Betrieb am Prüfstand gemessen werden. Somit eignet sich die Testumgebung zur Entwicklung und Validierung prädiktiver Fahrfunktionen mit dem Ziel einer Reduktion von Energiebedarf und Schadstoffemissionen.

Im Forschungsprojekt Hy-Nets wurde die Testumgebung verwendet, um einen prädiktiven Optimierungsalgorithmus zur energieeffizienten Geschwindigkeitswahl zu entwickeln. Dieser berücksichtigt Verkehrsrandbedingungen wie Ampelschaltzeiten und das zukünftige Geschwindigkeitsprofil vorausfahrender Fahrzeuge und verhindert somit unnötige Beschleunigungen und Verzögerungen, sodass Antriebsenergie eingespart werden kann. Die Umsetzung des Optimierungsalgorithmus erfolgt durch einen modellprädiktiven Beschleunigungsregler im linearen Zustandsraum.

Um das Einsparpotenzial dieser prädiktiven Fahrfunktion hinsichtlich der auf das Rad bezogenen Traktionsenergie sowie die im Kraftstoff und in der Batterie gespeicherte Primärenergie für verschiedene Antriebsstrangtopologien zu untersuchen, wurde eine Fallstudie für ein Fahrzeug mit konventionellem Antriebsstrang (Konv.), für ein batterieelektrisches Fahrzeug (engl. Battery Electric Vehicle, BEV) und für ein P3-Hybridfahrzeug (engl. Hybrid Electric Vehicle, HEV) simulationsbasiert durchgeführt. Die Antriebsstrangkomponenten wurden dazu so dimensioniert, dass bei allen Fahrzeugvarianten eine gleiche Antriebsystemleistung erreicht wird. Als Basis wird ein Fahrzeug aus dem C-Segment gewählt (VW Golf).

Die Fallstudie hat ergeben, dass für alle Antriebsstrangvarianten durch eine optimierte Geschwindigkeitswahl eine deutliche Reduktion der benötigten Traktionsenergie in einem Größenordnungsbereich von 40 % erreicht werden kann (vgl. Bild 5).

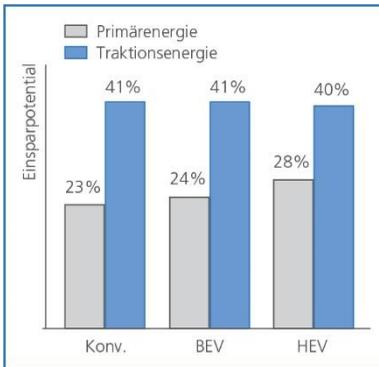


Bild 5: Energieeinsparpotenzial prädiktiver Optimierungsalgorithmen zur Geschwindigkeitswahl für konventionelle (Konv.), batterieelektrische (BEV) und hybridelektrische Fahrzeuge (HEV)

Darüber hinaus konnte aus der Fallstudie abgeleitet werden, dass der konventionelle Antriebsstrang in Bezug auf die Primärenergie im Vergleich zu den anderen Antriebsstrangtopologien am wenigsten von der optimierten Geschwindigkeitsregelung profitiert. Dies ist mit dem aus dem energetisch optimierten Geschwindigkeitsprofil resultierenden größeren Teillastanteil und den geringen Teillastwirkungsgraden des Verbrennungsmotors zu begründen. Der Wirkungsgrad des Elektromotors ist prinzipiell höher als der des Verbrennungsmotors, nimmt jedoch auch mit kleineren Lasten ab. Somit kann hier ein leicht höheres Einsparpotenzial (24 %) realisiert werden. Das Hybridfahrzeug profitiert am meisten (28 %) von einer optimierten Geschwindigkeitswahl, da bereits durch eine einfache regelbasierte Hybridstrategie der ineffiziente Teillastbetrieb zu einem großen Teil vermieden werden kann (vgl. Bild 5). Detaillierte Ausführungen zu den Ergebnissen sind in [16] zu finden.

Prädiktion des Umgebungsverkehrs

Nur bei exakter Kenntnis der zu berücksichtigenden zukünftigen Verkehrsrandbedingungen kann das volle Potenzial zur Energieeinsparung erschlossen werden. In der Realität treten jedoch unvorhersehbare Ereignisse auf, sodass beispielsweise die Bewegungstrajektorien der anderen Verkehrsteilnehmer nur mit einer begrenzten Genauigkeit präzifizierbar sind. Um den Einfluss dieser zuvor nicht berücksichtigten Abweichungen auf das Energieeinsparpotenzial zu untersuchen, wurde im zweiten Schritt ein modellbasierter Algorithmus zur Prädiktion des Umgebungsverkehrs an Kreuzungen ohne Lichtsignalanlagen entwickelt und mit Messdaten validiert [16].

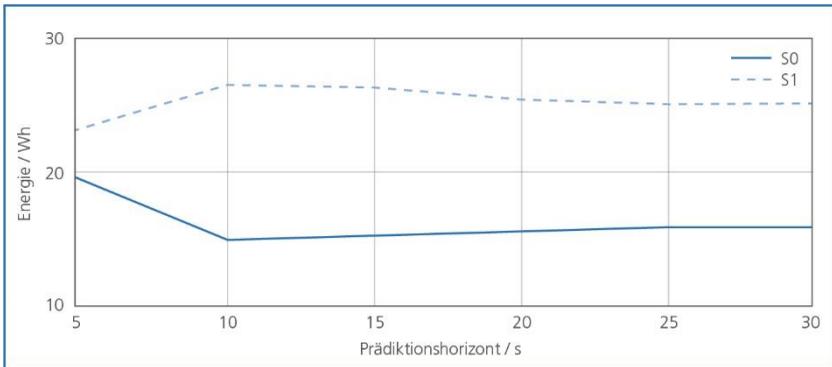


Bild 6: Energiebedarf in Abhängigkeit des Prädiktionshorizonts

Die Abhängigkeit des Energiebedarfs einer Kreuzungsüberfahrt vom Prädiktionshorizont wurde für zwei Fälle verglichen:

- S0: mit A-priori-Kennntnis der Referenztrajektorien
- S1: Prädiktion auf Basis der aktuellen Position und Geschwindigkeit

Die Untersuchungen haben gezeigt, dass der Vorausschauhorizont einen entscheidenden Parameter zur Reduktion der Traktionsenergie darstellt, sodass ein größerer Horizont zunächst zu einem größeren theoretischen Energieeinsparpotenzial führt (vgl. Bild 6). Als Konsequenz des begrenzt prädizierbaren menschlichen Fahrverhaltens nimmt die Genauigkeit der Vorhersage für den Fall S1 mit steigendem Prädiktionshorizont ab und der Energiebedarf wird durch Fehlprädiktionen sogar erhöht. In einigen Situationen wirkt sich somit ein größerer Vorausschauhorizont sogar insgesamt negativ auf den Energiebedarf aus. So kommt es in dem beispielhaften Szenario bei einem Prädiktionshorizont von 10 s bereits zu einer Steigerung des Traktionsenergiebedarfs um 80 % gegenüber einer optimierten Geschwindigkeitswahl mit exakter Kenntnis der Verkehrsrandbedingungen.

Die Ergebnisse zeigen, dass der gekoppelte Co-Simulationsansatz großes Potenzial bietet, um das Einsparpotenzial prädiktiver Fahrfunktionen unter realen Bedingungen zu quantifizieren.

Zusammenfassung und Ausblick

In diesem Beitrag wurde aufgezeigt, wie die Möglichkeiten einer integrierten Verkehrs- und Netzwerksimulation für den Einsatz in Hardware-in-the-Loop-Tests mit etablierten

Werkzeugketten erschlossen werden können. Hierzu wurde eine mit der Fahrzeugmodellsuite ASM aufgebaute Fahrzeug- und Umgebungssimulation echtzeitfähig mit einer globalen Umgebungssimulation gekoppelt. Teil des globalen Umgebungsmodells, das auf dem integrierten Simulator Veins basiert, ist die mikroskopische Verkehrssimulation SUMO, die eine Vielzahl an Fahrzeugen und deren Interaktionen simulieren kann. Außerdem wurde als Teil der globalen Umgebungssimulation die Fahrzeugkommunikation simuliert und der Ego-Fahrzeugsimulation als Car2X-Nachrichten „over-the-air“ zugesandt. Die so aufgebaute Co-Simulation erlaubt die Simulation des physikalischen Ego-Fahrzeugmodells auf einem Echtzeitsimulator, dessen lokales Umgebungsmodell mit der globalen Umgebungssimulation synchronisiert wird.

An den HIL-Simulator sind als reale Komponenten ein Verbrennungsmotor und eine elektrische Maschine auf ihren Prüfständen angeschlossen. Beide sind so miteinander verbunden, dass sie als Komponenten eines virtuellen Hybridantriebs agieren, dessen Rest mit dem Ego-Fahrzeugmodell simuliert wird. So werden die Auswirkungen des simulierten Verkehrs und der Fahrzeugkommunikation auf den Energiebedarf und die Emissionen in Form realer Größen am Prüfstand mess- und validierbar.

Dieser Aufbau erfolgte im Rahmen des Projektes Hy-Nets mit dem Ziel, intelligente Fahralgorithmen für Hybridfahrzeuge zu entwickeln, die den Energiebedarf durch die Nutzung von Car2X-Informationen reduzieren. Mit der vorgestellten Co-Simulationsumgebung steht eine geeignete Entwicklungs- und Testplattform für solche Algorithmen zur Verfügung. Erste idealisierte Umsetzungen auf Basis der Co-Simulationsumgebung zeigen, dass der Energiebedarf durch solche Algorithmen signifikant gesenkt werden kann. In den nächsten Schritten sollen diese weiterentwickelt und mit Prüfungsmessungen validiert werden.

Durch die Kopplung des HIL-Simulators mit angeschlossenen Prüfständen und der Verkehrs- und Netzwerksimulation mit ihren jeweiligen Stärken ist zudem eine Testplattform entstanden, die die Absicherung von ADAS-Funktionen in komplexen, interagierenden Verkehrsszenarien unter Einbindung realer Antriebskomponenten ermöglicht.

Danksagung

Das Projekt Hy-Nets wird durch die EU und das Land NRW im Rahmen des „Europäischen Fonds für regionale Entwicklung“ gefördert.



EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung



EFRE.NRW
Investitionen in Wachstum
und Beschäftigung

Literaturverzeichnis

- [1] P. Wältermann, „Hardware-in-the-Loop: Die Technologie zum Test elektronischer Steuerungen und Regelungen in der Fahrzeugtechnik,“ in *6. Paderborner Workshop „Entwurf mechatronischer Systeme“*, Paderborn, 2009.
- [2] N. Meyer, S. Walter, D. Guse und S. Klein, „Vehicle-Comparative Calibration – Crucial factors for efficient in-the-Loop operation of dynamic engine test benches,“ in *4. Int. Conf. Diesel Powertrains 3.0*, Meriden, Solihull, UK, 2018.
- [3] S. Walter, N. Meyer, T. Schulze, J. Andert, S. Klein und D. Guse, „Durchgängig von der Straße auf den Prüfstand bis zur Simulation – Eine qualitative Analyse am Beispiel RDE,“ in *Simulation und Test 2017*, Wiesbaden, 2018.
- [4] C. Sommer, R. German und F. Dressler, „Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,“ *IEEE Transactions on Mobile Computing*, Bd. 10, pp. 3-15, 1 2011.
- [5] M. Eisenbarth, K. Etzold, J. Andert, T. Plum, P. Reinold, U. Schwarz und R. Gries, „Eine ganzheitliche Methodik zur Entwicklung von vernetzten Hybridfahrzeugen,“ in *7. Internationales Symposium für Entwicklungsmethodik*, Wiesbaden, 2017.
- [6] J. Andert, S. Klein, R. Savelsberg, S. Pischinger und K. Hameyer, „Virtual shaft: Synchronized motion control for real time testing of automotive powertrains,“ *Control Engineering Practice*, Bd. 56, pp. 101-110, November 2016.
- [7] C. Sommer, J. Härrri, F. Hrizi, B. Schünemann und F. Dressler, „Simulation Tools and Techniques for Vehicular Communications and Applications,“ in *Vehicular ad hoc Networks - Standards, Solutions, and Research*, C. Campolo, A. Molinaro und R. Scopigno, Hrsg., Springer, 2015, pp. 365-392.
- [8] C. Sommer und F. Dressler, *Vehicular Networking*, Cambridge University Press, 2014.
- [9] S. Laux, G. S. Pannu, S. Schneider, J. Tiemann, F. Klingler, C. Sommer und F. Dressler, „OpenC2X – An Open Source Experimental and Prototyping Platform Supporting ETSI ITS-G5,“ in *8th IEEE Vehicular Networking Conference (VNC 2016), Demo Session*, Columbus, 2016.

- [10] F. Klingler, G. S. Pannu, C. Sommer, B. Bloessl und F. Dressler, „Field Testing Vehicular Networks using OpenC2X,“ in *15th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys 2017), Poster Session, Niagara*, 2017.
- [11] D. Krajzewicz, J. Erdmann, M. Behrisch und L. Bieker, „Recent Development and Applications of SUMO – Simulation of Urban MObility,“ *International Journal On Advances in Systems and Measurements*, Bd. 5, pp. 128-138, 12 2012.
- [12] L. Codeca, R. Frank und T. Engel, „Luxembourg SUMO Traffic (LuST) Scenario: 24 Hours of Mobility for Vehicular Networking Research,“ in *7th IEEE Vehicular Networking Conference (VNC 2015)*, Kyoto, 2015.
- [13] D. S. Buse, M. Schettler, N. Kothe, P. Reinold, C. Sommer und 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, 2018.
- [14] D. S. Buse, C. Sommer und F. Dressler, „Integrating a Driving Simulator with City-Scale VANET Simulation for the Development of Next Generation ADAS Systems,“ in *37th IEEE Conference on Computer Communications (INFOCOM 2018), Demo Session*, Honolulu, 2018.
- [15] F. Klingler, G. S. Pannu, C. Sommer und F. Dressler, „Connecting Simulation and Real World: IEEE 802.11p in the Loop,“ in *23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017), Poster Session*, Snowbird, 2017.
- [16] M. Eisenbarth, T. Plum und J. Andert, „Energieoptimierte Fahrstrategie unter Berücksichtigung der Verkehrsteilnehmer an Kreuzungen ohne Lichtsignalanlage,“ *Forschung im Ingenieurwesen*, Bd. 82, Nr. 1, pp. 71-81, 2018.
- [17] T. Plum, M. Wegner, M. Eisenbarth, Z. Ye, K. Etzold, S. Pischinger und J. Andert, „A simulation-based case study for powertrain efficiency improvement by automated driving functions,“ *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 2018, DOI: <https://doi.org/10.1177/0954407018775182>.