

Model-Based Engineering für die Automatisierung von Validierungsaktivitäten am Beispiel Fahrerassistenzsysteme

Constantin Mandel, Sebastian Lutz, Olivia Rau, Matthias Behrendt und Albert Albers

1 Einleitung und Motivation

Die Automatisierung des Fahrens verändert die individuelle Mobilität grundlegend. In diesem Kontext hat die Entwicklung von vergleichsweise einfachen Fahrerassistenzsystemen gezeigt, dass die neuen, noch kommenden Systeme zur Steuerung und Regelung der Fahraufgabe auch neue Entwicklungsmethoden zwingend notwendig machen. Ein zentraler Aspekt dieser Entwicklungsmethoden muss auf der Validierung der Fahrerassistenzsysteme liegen. Die Validierung ist insbesondere dort essenziell, wo Fahrerassistenzsysteme aktiv in die Führung des Fahrzeugs eingreifen und somit sicherheitskritische Funktionen übernehmen. Für eine effiziente Validierung im Produktentwicklungsprozess ist es notwendig, Validierungsaktivitäten nicht nur am finalen System, sondern kontinuierlich, von rein virtuellen Modellen bis hin zum Prototypen auf dem Prüfstand, durchzuführen.

Im Kontext immer komplexerer Systeme steigt die Vernetzung und gegenseitige Beeinflussung von Modellen und Systemen im Produktentwicklungsprozess. Ansätze mit dieser steigenden Komplexität und Vernetzung umzugehen finden sich im Model-Based Engineering (MBE). Dieser Beitrag untersucht Potenziale des Einsatzes von MBE-Ansätzen bei der Validierung von Fahrerassistenzsystemen. Ziel ist die Untersuchung der Machbarkeit des Aufbaus einer Validierungsumgebung zur Automatisierung von Tests im Rahmen der kontinuierlichen Validierung von Fahrerassistenzsystemen.

2 Stand der Forschung

2.1 Automatisiertes Fahren

Unaufmerksamkeit des Fahrers oder Fehleinschätzung in Ausnahmesituationen, z.B. beim Annähern an ein Stauende, sind laut Untersuchungen des Euro NCAP die häufigsten Ursachen von Auffahrunfällen im Straßenverkehr (EuroNCAP 2015). In diesem Zusammenhang können Fahrerassistenzsysteme wie der Abstandsregeltempomat (*Adaptive Cruise Control – ACC*) und der aktive Notbremsassistent (*Autonomous Emergency Brake – AEB*) den Fahrer bei der Längsführung des Fahrzeugs unterstützen und damit Unfälle vermeiden.

Vor allem das AEB-System ist darauf ausgelegt Auffahrunfälle zu verhindern, in dem es automatisiert eine frühzeitige Erkennung der Gefahrensituation ermöglicht. Auch das ACC-System vermeidet Unfälle. Einerseits wird der Abstand zum vorausfahrenden Fahrzeug permanent überwacht und somit Gefahrensituation frühzeitig erkannt. Damit können entsprechende Gegenmaßnahmen eingeleitet werden. Andererseits entlastet das ACC-System den Fahrer bei der ermüdenden Langstreckenfahrt, sodass dieser im Zweifelsfall in anderen Situationen wieder aufmerksam sein kann. Eine Kombination dieser beiden Vorrichtungen und in Kombination mit weiteren Fahrerassistenzsysteme kann also zu mehr Sicherheit im Straßenverkehr beitragen. (Steichele 2016)

Wie bereits das Wort „Fahrerassistenzsystem“ ausdrückt, assistieren diese Vorrichtungen dem Fahrer, wodurch das Zusammenspiel mit dem menschlichen Fahrer von großer Bedeutung ist. Grundlegend ist dabei zu klären, welche Aufgabe von wem am besten übernommen wird und wie die Schnittstelle dazwischen beschaffen sein muss. (Winner et al. 2015, Seite 28 nach Karl-Friedrich Kraiss 1998) Aus der Definition des Handbuch Fahrerassistenzsysteme geht demnach hervor, dass Fahrerassistenzsysteme hochautomatisiert handeln, wobei ihr Eingreifen zeitweise die Handlung des Fahrers überwiegt (Winner et al. 2015).

Fahrerassistenzsysteme wie das ACC-System können bewusst vom Fahrer aktiviert werden und haben somit unmittelbaren Einfluss auf die Fahrzeugsteuerung. Das AEB-System hingegen arbeitet üblicherweise im Hintergrund und greift erst bei einer erkannten Notsituation aktiv in das Fahrgeschehen

ein. Nimmt das System über die eingebaute Sensorik ein zu schnell näherkommendes Objekt wahr, wird der Fahrer zunächst akustisch und visuell gewarnt. Reagiert dieser jedoch nicht auf die Signale, greift das System in die Fahrzeugführung ein und leitet aktiv eine Bremsung ein, sodass entweder eine Kollision vollständig vermieden oder die Unfallschwere reduziert werden kann. Da zum aktuellen Zeitpunkt die Aufgabe und damit Verantwortung der Fahrzeugführung immer beim Fahrer liegt, muss dieser stets die Möglichkeit haben die Systeme zu überstimmen. (Winner et al. 2015; Daimler AG 2018)

2.2 Model-Based Engineering (MBE) und Model-Based Systems Engineering (MBSE)

Die im vorangegangenen Kapitel vorgestellten Fahrerassistenzsysteme können als Stellvertreter für den Trend verstanden werden, Systeme mit einem immer größeren Funktionsumfang zu entwickeln. Dieser Trend wird durch stetig steigende Kundenanforderungen getrieben. Gleichzeitig führt die Integration von immer mehr Funktionen zu einer steigenden Komplexität und internen Vernetzung der zu entwickelnden Systeme. (Beihoff et al. 2014)

Ansätze des Model-Based Engineering (MBE) und des Model-Based Systems Engineering (MBSE) können dabei helfen, mit der steigenden Komplexität umzugehen und diese beherrschbar zu machen (Kleiner und Husung 2017). Model-Based Systems Engineering beschreibt dabei den formalisierten Einsatz von Modellen zur Unterstützung des gesamten Produktlebenszyklus – von der Definition der Systemanforderungen, über die Konstruktion bis hin zur Produktnutzung im Markt (International Council on Systems Engineering (INCOSE) - Technical Operations 2007). Kernpunkt des MBSE ist dabei der Übergang von der Nutzung einer Vielzahl unabhängiger (Text-) Dokumente hin zur Nutzung von Modellen, welche die Bedarfe verschiedener Ingenieursdisziplinen (Mechanik, Elektrotechnik, Software etc.) und verschiedene Aspekte des Produktlebenszyklus (Anforderungen, Produktgestalt, Testfälle etc.) integriert und vernetzt abbilden (Kleiner und Husung 2017; International Council on Systems Engineering (INCOSE) - Technical Operations 2007).

Der Begriff Model-Based Engineering (MBE) bezeichnet als Sammelbegriff übergreifend modellbasierte Ansätze im Produktlebenszyklus (PivotPoint Technology Corp. 2018). Dazu können neben MBSE auch Geschäftsprozess-

Modellierung (BPM) oder Modellsimulation zählen (PivotPoint Technology Corp. 2018).

2.3 Validierung im Produktentwicklungsprozess

Nach Albers ist Validierung die zentrale Aktivität der Produktentstehung und Garant für ein erfolgreiches Produkt am Markt (Albers 2010). In der VDI Norm 2206 ist Validierung dabei *als „die Prüfung zu verstehen, ob das Produkt für seinen Einsatzzweck geeignet ist bzw. den gewünschten Wert erzielt“* (VDI Norm 2206, S. 39; VDI Norm 2206). Bei der Validierung *„geht die Erwartungshaltung des Fachexperten und des Anwenders ein“* (VDI Norm 2206, S. 39). Dies grenzt die Validierung von der Verifikation ab, welche eine reine Überprüfung beschreibt, *„ob eine Realisierung [...] mit der Spezifikation [...] übereinstimmt“* (VDI Norm 2206, S. 38).

Nach Ehrlenspiel erhöhen sich die Kosten zur Behebung von Fehlern im entwickelten System in jeder fortschreitenden Phase des Entwicklungsprozesses um den Faktor zehn (Ehrlenspiel und Meerkamm 2013; Reinhart et al. 1996). Daher sollten Aktivitäten zur Validierung des entwickelten Systems bereits in frühen Phasen des Entwicklungsprozesses begonnen und kontinuierlich während des gesamten Produktlebenszyklus durchgeführt werden (Albers et al. 2016).

In der Literatur werden bereits eine Vielzahl von Prozessmodellen und Ansätzen zur Integration von Validierungsaktivitäten in den Produktentwicklungsprozess beschrieben (vgl. Eigner et al. 2014). Während einige dieser Ansätze bereits die Bedeutung einer frühen Validierung betonen (siehe z.B. Paulweber et al. 2014), werden Validierungsaktivitäten oftmals phasengebunden und nicht kontinuierlich im Produktentwicklungsprozess beschrieben (vgl. Eigner et al. 2014).

Zur Unterstützung einer kontinuierlichen Validierung im Produktentwicklungsprozess wurde der IPEK-X-in-the-Loop-(IPEK-XiL)-Ansatz entwickelt (Albers et al. 2016). Der IPEK-XiL-Ansatz integriert bestehende Ansätze wie Software-in-the-Loop (SiL) und Hardware-in-the-Loop (HiL) und erweitert diese um die Bedarfe einer mechatronischen Entwicklung und die Einbindung der Sichten unterschiedlicher Fachdisziplinen (Albers et al. 2016). Am Beispiel der Antriebsstrangentwicklung beschreibt der IPEK-XiL-Ansatz die Validierung

des betrachteten Systems („X-in-the-Loop“) im Zusammenspiel mit Modellen für das Rest-Fahrzeug, der Umwelt und dem Fahrer. Übergreifend und abhängig vom Validierungsziel werden dabei relevante Fahrmanöver und Testfälle für das betrachtete System und die weiteren Modelle definiert.

3 Forschungslücke und Forschungsfragen

Wie im *Stand der Forschung* (siehe Kapitel 0) beschrieben, haben Fahrerassistenzsysteme ein hohes Potential die Sicherheit im Straßenverkehr zu erhöhen. Fahrzeughersteller und Zulieferer sind daher interessiert daran, neue und immer umfangreichere Systeme auf den Markt zu bringen. Zusätzlich sorgt gerade die Bewertung durch die EuroNCAP zur weiten Verbreitung von Notbremssystemen. Gleichzeitig wird durch die geforderte erhöhte Funktionalität und, damit einhergehend, erhöhte Komplexität solcher Systeme deren Entwicklung immer aufwändiger. Gerade der Validierungsprozess stellt eine große Herausforderung dar.

Simulation und automatisierte Testabläufe sind vielversprechende Ansätze, um den Validierungsprozess effizienter zu gestalten, da sie erlauben eine Reihe von Validierungsaktivitäten bereits in frühen Phasen der Entwicklung durchzuführen. Wie in Kapitel 0 beschrieben können somit Kosten im Entwicklungsprozess eingespart werden. Nach aktuellem Stand erfolgt dabei die Entwicklung solcher Systeme noch größtenteils im HiL- (Hardware-in-the-Loop) Aufbau an den Steuergeräten oder sehr kostenintensiv mit Prototypen auf dem Testgelände oder der Straße. Um in Zukunft Entwicklungskosten senken zu können, soll eine zunehmende Automatisierung der Testprozeduren erfolgen. Das hier dargestellte Vorgehen beschreibt einen Ansatz zum Aufbau einer Validierungsumgebung in welcher automatisierte Tests mit virtuellen Simulationen verbunden werden.

Im Rahmen dieser Arbeit sollen zwei Forschungsfragen untersucht werden:

- Kann eine (zunächst rein virtuelle) Validierungsumgebung, welche automatisierte Testabläufe im Rahmen der ACC und AEB Validierung ermöglicht, aufgebaut werden?
- Können in der Validierungsumgebung erzeugte Testergebnisse mit Anforderungen aus relevanten Normen in Verbindung gesetzt werden, um die Bewertung der Testergebnisse zu unterstützen?

Durch die Beantwortung der Forschungsfragen soll ein Beitrag dazu geleistet werden, Validierungsaktivitäten von ACC- und AEB-Systemen bereits früh im Produktentwicklungsprozess durchführen zu können. Wie in Kapitel 0 beschrieben, können durch eine frühzeitige Validierung Zeit und Kosten in der Produktentwicklung gespart werden.

4 Methodisches Vorgehen zum Aufbau der Validierungsumgebung

Aus den in Kapitel 0 beschriebenen Forschungsfragen und Zielen der Arbeit lassen sich zwei zentrale Anforderungen an die zu entwickelnde Validierungsumgebung ableiten:

- Die Validierungsumgebung soll einen Beitrag zur Automatisierung von Validierungsaktivitäten für die Fahrerassistenzsysteme ACC und AEB leisten
- In der entwickelten Validierungsumgebung durchgeführte Validierungsaktivitäten sollen einen erkennbaren Bezug zu real auftretenden Situationen im Straßenverkehr haben.

Für diese Arbeit wurde ein modellbasierter Ansatz gewählt (siehe Kapitel 2.2). Die entwickelte Validierungsumgebung ist nach dem IPEK-XiL-Ansatz aufgebaut. Neben dem zu validierenden System (ACC beziehungsweise AEB) werden somit Modelle des Restfahrzeugs, des Fahrers und der Fahrzeug-Umwelt mit in der Validierungsumgebung betrachtet und beschrieben.

Die entwickelte rein virtuelle Validierungsumgebung wurde verteilt über zwei Rechner aufgebaut. Auf dem ersten Rechner wurden verschiedene Modelle und Skripte zur Erzeugung der parametrisierten Testläufe verknüpft. Die einzelnen Testläufe können vom ersten Rechner aus automatisch nacheinander gestartet und die entsprechenden Testergebnisse in einer vorher definierten Form abgelegt werden. Der zweite Rechner dient zur Ausführung der parametrisierten Simulationen. Die Kommunikation zwischen den beiden Rechnern läuft über das „User Datagram Protocol“ - UDP (vgl. z.B. Schnabel 2004).

Für Bewertungsgrößen zur Validierung von Fahrerassistenzsystemen (beispielsweise Mindestabstand zweier Fahrzeuge) wurden relevante gesetzliche- und Hersteller-Vorgaben untersucht. Die Vorgaben wurden mit Hilfe des Elements *Requirement* (Anforderung) der MBSE-Modellierungssprache SysML (Object Management Group (OMG) 2017) in einem MBSE-Modell modelliert.

Zudem wurden die in der Validierungsumgebung erzeugten Testergebnisse mit den im MBSE-Modell modellierten Anforderungen verknüpft. Durch diese Verknüpfung wurde eine Möglichkeit geschaffen, die durchgeführte Validierungsaktivitäten mit MBSE-Modellen zu verbinden.

Zur Validierung des ACC- und AEB-Systems wurde die Funktionen dieser Systeme in Bezug auf die modellierten Anforderungen untersucht. Dazu wurden relevante Situationen im Straßenverkehr, in denen das ACC- und AEB-System Einsatz finden, identifiziert und in sogenannten (Fahr-) Manövern abstrahiert. Fahrmanöver beschreiben dabei eine Referenzanwendung eines Systems für seinen späteren Betrieb, in Bezug auf definierte Validierungsziele (vgl. Albers et al. 2016). Dies schließt unter anderem Nutzerinteraktion und weitere Randbedingung, wie Verkehrsbewegungen, Umweltbedingungen (Tageszeit, Wetter etc.) oder Straßenbeschaffenheit mit ein (vgl. Albers et al. 2016).

Zur Integration aller Modelle und Einstellung der automatisierten Testläufe wurde die Software „ModelCenter“ verwendet. ModelCenter bietet ein Framework zur Verknüpfung von Modellen, die in verschiedenen Software-Anwendungen erstellt wurden. Zusätzlich bietet ModelCenter Software-Unterstützung zur (statistischen) Versuchsplanung und Versuchsauswertung. (Phoenix Integration 2019)

5 Umsetzung des Konzepts

Die entwickelte Validierungsumgebung basiert auf dem IPEK-XiL-Ansatz (vgl. Albers et al. 2016) und enthält Modelle für das betrachtete System im Kontext des Restfahrzeugs, einen Fahrer und die Fahrzeug-Umwelt sowie Fahrmanöver und Testfälle. Die konkreten Ausprägungen der einzelnen Aspekte sind im Folgenden beschrieben:

5.1 Betrachtetes System und Restfahrzeug-Modell

Im vorliegenden Fall wurden die Funktionen des ACC- und AEB-Systems im Kontext des Gesamtfahrzeugs untersucht und als „X-in-the-Loop“ betrachtet. Das Fahrzeug wurde aus der Standard-Bibliothek der Simulationsumgebung „CarMaker“ (IPG Automotive GmbH 2019) als rein virtuelles System übernommen. Das untersuchte Fahrzeug wird im Folgenden als Ego-Fahrzeug bezeichnet.

Umwelt-Modell

Die Umwelt des Fahrzeugs wurde für die aufgebaute Validierungsumgebung ebenfalls virtuell simuliert. Wichtigster Aspekt des Umwelt-Modells sind Modelle des Verkehrs, bzw. im vorliegenden Fall eines dem Ego-Fahrzeug vorausfahrenden Fahrzeugs. Das vorausfahrende Fahrzeug wird im Folgenden als Target-Fahrzeug bezeichnet. Zusätzlich werden im Umwelt Modell der Straßenverlauf (unter anderem gerade, mit Kurven, steigend) und die Umgebungsbedingungen (unter anderem Tageszeit, Wetter) abgebildet.

Fahrer-Modell

Das in CarMaker abgebildete Fahrzeug wird von einem ebenfalls simulierten Fahrer gesteuert. Mit Hilfe der Fahrmanöver und Testfälle (siehe nächster Punkt) wird das Verhalten des Fahrers festgelegt.

Fahrmanöver und Testfälle

Für die Definition der Fahrmanöver und Testfälle steht in dieser Arbeit der qualitative Geschwindigkeitsverlauf des Target-Fahrzeugs im Fokus. Das Ego-Fahrzeug folgt in jedem Manöver dem Target-Fahrzeug mit aktivem ACC. Der Geschwindigkeitsverlauf des Target-Fahrzeugs soll dabei so abgebildet werden, dass ein erkennbarer Bezug zu Straßenverkehrssituationen, in denen das ACC- und AEB-System Anwendung finden, gegeben ist. Zusätzlich können verschiedene Umwelt-Parameter, wie der Straßenverlauf, eingestellt werden. Mit Hilfe stochastischer Verfahren (z.B. MonteCarlo-Analyse, vgl. z.B. Cruse 1997, oder Design of Experiments-DOE, vgl. z.B. Siebertz et al. 2017) können verschiedene Parameter der Fahrmanöver variiert und ein Testplan erzeugt werden. Die einzelnen Tests des Testplans können anschließend automatisiert nacheinander in der Validierungsumgebung durchgeführt werden.

Der Ablauf eines Tests in der Validierungsumgebung ist in Abbildung 1 zu erkennen. Das mit (1) gekennzeichnete Element dient zur Konfiguration der Parameter der Simulationsläufe nach stochastischen Verfahren. Das Skript bei (2) ermöglicht die automatische Benennung der Ergebnisse eines Testlaufs nach vorgegebenem Muster. Das Matlab-Skript (vgl. MathWorks 2019) bei (3) erzeugt aus den in (1) erzeugten Parametern das entsprechend parametrisierte Fahrmanöver. Das Matlab-Skript bei (4) überträgt das so erzeugte Fahrmanöver als Konfigurationsdatei über eine Netzwerkverbindung an die

Simulationsumgebung auf den zweiten Simulationsrechner. Das Skript bei (5) sorgt dafür, dass die in der Simulationsumgebung erzeugten Testergebnisse auf dem ersten Rechner wieder eingelesen werden können. Das Skript bei (6) schließlich speichert die Ergebnisse des Tests in einem vorgegebenen Format an einem vorher bestimmten Speicherort. Die Skripte in (2) bis (6) werden je nach Einstellung in (1) automatisch mehrfach hintereinander ausgeführt. Die in (6) gespeicherten Ergebnisse können nach Durchführung der Tests zu statistischen Untersuchung verwendet werden (z.B. Sensitivitätsanalysen einzelner Parameter) oder mit den vorher festgelegten Anforderungen abgeglichen werden.

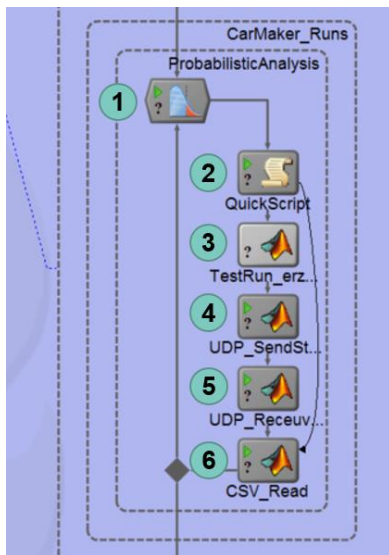


Abbildung 1: Darstellung der einzelnen Schritte zur Ausführung eines Tests in der entwickelten Validierungsumgebung – Ausschnitt aus ModelCenter

6 Anwendungsbeispiel

In diesem Kapitel wird beispielhaft auf einige in der Validierungsumgebung durchgeführte Testläufe eingegangen. Beispielhaftes Validierungsziel war die Überprüfung der Funktionen des ACC- und AEB-Systems eines Fahrzeugs. Die Testläufe basierten dabei jeweils auf einem festgelegten Fahrmanöver, welches einen Bezug zu einer realen Verkehrssituation hat. Zur Bewertung der Funktionen des ACC- und AEB-Systems wurde der Abstand zwischen

Ego-Fahrzeug und Target-Fahrzeug analysiert und mit geltenden Normen und Standards verglichen.

Wie in Kapitel 0 beschrieben steht bei der Definition der Fahrmanöver das Geschwindigkeitsprofil des Target-Fahrzeugs im Fokus. Zur Abbildung eines möglichst großen Umfangs relevanter Geschwindigkeitsprofile wurde, basierend auf häufig auftretenden Verkehrssituationen, eine Vielzahl von Fahrmanövern definiert und in einem Manöverkatalog abgebildet. Die definierten Manöver lassen sich dabei in drei Kategorien unterteilen:

- Statische Manöver: die stationäre Folgefahrt des Ego-Fahrzeugs hinter dem Target-Fahrzeug. Bei diesen Manövern ist das ACC-System aktiv.
- Dynamische Manöver: entweder als Annäherungsmanöver des Ego-Fahrzeugs auf das fahrende Target-Fahrzeug oder als Annäherungsmanöver des Ego-Fahrzeugs an das stehende Target-Fahrzeug. Auch bei diesen Manövern ist das ACC-System aktiv
- Hochdynamische Manöver: Notbremsung des Ego-Fahrzeugs durch das AEB-System

Alle identifizierten Manöver wurden in einem parametrisierbaren Geschwindigkeitsprofil des Target-Fahrzeugs (siehe Abbildung 2) integriert. Das Geschwindigkeitsprofil ist aufgeteilt in ein Annäherungsmanöver (1), eine stationäre Folgefahrt (2) und eine Folgefahrt mit Geschwindigkeitsänderung (3, 4), die den ACC-Bereich simulieren und eine Notbremsung (5), in welcher das AEB-System des Ego-Fahrzeugs testen soll.

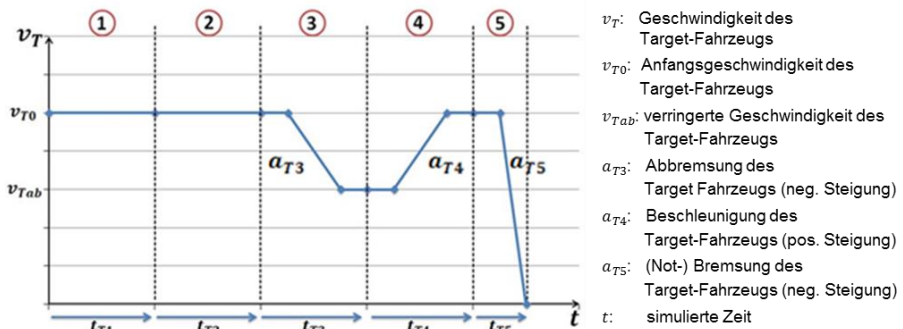


Abbildung 2: Parametrisierbares Geschwindigkeitsprofil des Target-Fahrzeugs zur Abbildung der Fahrmanöver aus dem Manöverkatalog

Durch die Variation der Eingangsgrößen des Target-Fahrzeugs, also der Anfangsgeschwindigkeit v_{T0} , der Stärke der Abbremsung a_{T3} , bzw. der Beschleunigung a_{T4} in den Bereichen 3 und 4 und der Stärke der Notbremsung a_{T5} im Bereich 5 sowie der zeitlichen Länge der einzelnen Bereiche t_{T1-5} , können alle Fahrmanöver des Manöverkatalogs abgebildet werden.

Wie in Kapitel 0 beschrieben können die Eingangsgrößen in ModelCenter mit Hilfe stochastischer Verfahren variiert werden (siehe Abschnitt *Fahrmanöver und Testfälle*). Zusätzlich können ebenfalls relevante Umweltparameter variiert und mit im Fahrmanöver hinterlegt werden. Für die vorliegenden Beispiele wurde dabei konstant eine gerade, ebene Fahrt bei Tag und Sonnenschein auf der Landstraße angenommen. Mit Hilfe des ausgewählten stochastischen Verfahren kann schließlich eine vorher festzulegende Anzahl verschieden parametrisierter Fahrmanöver erzeugt werden.

Nach dem in Abbildung 1 beschriebenen Vorgehen wurde jeweils ein konkretes Geschwindigkeitsprofil zusammen mit den Umweltparametern als Fahrmanöver in eine Konfigurationsdatei für die Simulationsumgebung überführt (siehe Schritt 3 in Abbildung 1). Anschließend wurde die Konfigurationsdatei per UDP an den Simulationsrechner übergeben und die Simulation für das entsprechende Fahrmanöver durchgeführt (Schritt 4-5 in Abbildung 1). Bei der Simulation wurden die Geschwindigkeits- und Beschleunigungs-Verläufe des Ego-Fahrzeugs aufgenommen. Die Ergebnisse der Simulation wurden wieder zurück an den ersten Rechner übermittelt und in einem auswertbaren Format abgelegt (Schritt 6 in Abbildung 1). Dieses Vorgehen wurde anschließend automatisch für das nächste erzeugte Fahrmanöver wiederholt so lange bis zu jedem erzeugten Fahrmanöver Ergebnisdateien abgelegt wurden. Zur Auswertung wurden die Testläufe in ACC-Bereich (Bereich 1-4 in Abbildung 2) und AEB-Bereich (Bereich 5 in Abbildung 2) eingeteilt und getrennt betrachtet.

In einem ersten Beispiel wurde a_{T3} schrittweise von -1m/s bis -5m/s erhöht, während die anderen Stellgrößen konstant gehalten wurden ($v_{T0} = 80\text{km/h}$, $a_{T4} = 4\text{m/s}$, $a_{T5} = -6\text{m/s}$). Abbildung 3 zeigt die Abweichung des tatsächlichen Fahrzeugabstands vom theoretischen Soll-Abstand für verschiedene Fahrmanöver im ACC-Bereich (dünne Linien). Der Soll-Abstand ist dabei durch die Gesetzgebung abhängig von der Fahrgeschwindigkeit vorgegeben (Soll-Abstand bei 80km/h ist ca. 22m (Verband für bürgernahe Verkehrspolitik e.V.)).

Im Beispiel ist zu beobachten, dass vor allem im Annäherungsmanöver (Bereich 3, ca. 130s-160s in Abbildung 3) der einzuhaltende Mindestabstand häufig unterschritten wird.

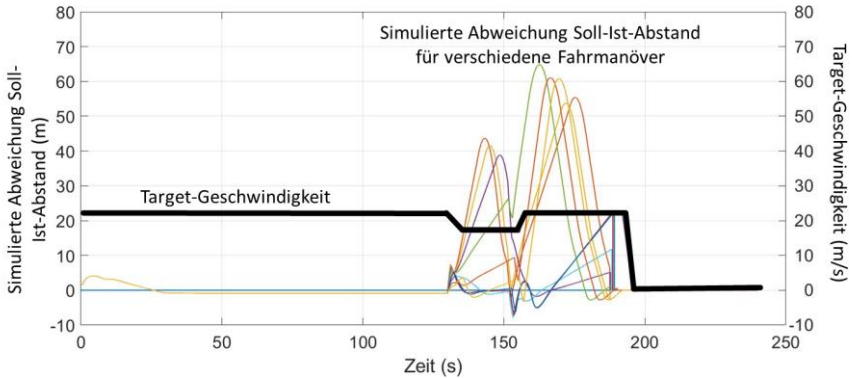


Abbildung 3: Abweichung des simulierten Ego-Fahrzeug-Target-Fahrzeug Abstands vom gesetzlich vorgegebenen Soll-Abstand

In einer weiteren Parameterstudie wurden v_{T0} , a_{T3} , und a_{T4} konstant gehalten ($v_{T0} = 80\text{km/h}$, $a_{T4} = 4\text{m/s}$, $a_{T4} = -6\text{m/s}$) während a_{T5} schrittweise verringert wurde. Für diese Parameterstudie wurde somit im AEB-Bereich der Abstand zwischen Ego- und Target-Fahrzeug (Bereich 5 in Abbildung 2) untersucht. Für den Abstand zwischen Ego-Fahrzeug und Target-Fahrzeug im AEB-Bereich ist entscheidend, ob dieser größer als null ist, es also nicht zu einer Kollision kommt. Daher zeigt Abbildung 4, im Gegensatz zu Abbildung 3, die simulierten Abstände zwischen Ego-Fahrzeug und vorausfahrendem Target-Fahrzeug für verschiedene Fahrmanöver und nicht die Abweichung des simulierten Abstands von einem gesetzlich vorgegebenen Soll-Abstand.

In Bezug auf den Abstand ist zu erkennen, dass es in keinem Testlauf zu einem Auffahren des Ego-Fahrzeugs auf das Target-Fahrzeug kommt. Das Ego-Fahrzeug kommt bei jedem Durchlauf ca. 3m-4m vor dem Target-Fahrzeug zum Stillstand. In den ersten Sekunden nach Beginn der Abbremsung liegt eine regelmäßige Streuung der Abstandskurven vor, bis sich die Kurven an einem Punkt bei ca. 18m aufteilen. Bei einigen Fahrten wird der Abstand von ca. 17m-18m konstant gehalten und nimmt anschließend noch einmal ab.

In Abbildung 5 ist zusätzlich die Auswertung der Ergebnisse eines Testlaufs (in der Spalte „New“) in Bezug auf die im MBSE-Modell hinterlegten Anforderungen Maximalbeschleunigung, minimale Abbremsung, Minimalabstand im Abbremsmanöver (AEB Eingriff) und Minimalabstand bei Folgefahrt zu sehen (bei aktivem ACC). Es ist zu erkennen, dass im analysierten Testlauf die Anforderungen für die Maximalbeschleunigung und maximale Abbremsung nicht eingehalten werden (rote Hinterlegung bei „Margin“ in Abbildung 5). Für die beiden Minimalabstände werden die Anforderungen hingegen erfüllt (grüne Hinterlegung bei „Margin“ in Abbildung 5).

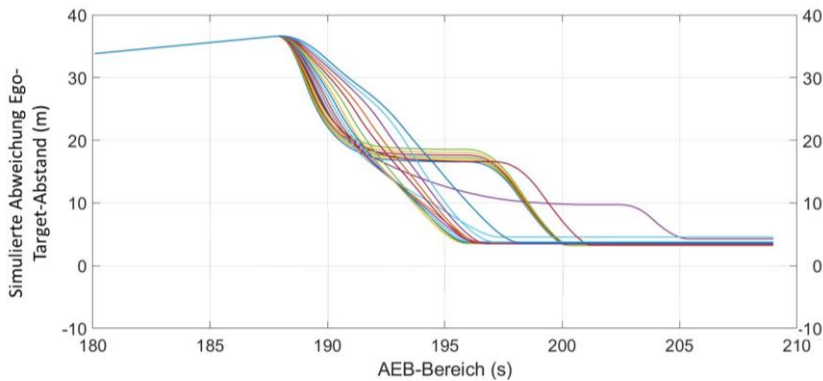


Abbildung4: Simulierter Abstand zwischen Ego- und Target-Fahrzeug im AEB-Bereich

Property	Units	Original	New	Margin
Fahrzeug				
TestDummy				
Maximalbeschleunigung	Real	0.0	4.9405398	0,94054...
Maximale Abbremsung	Real	0.0	-7.1786...	3,1787 ...
Minimalabstand Abbremsmar	Real	0.0	7.0344306	6,5344 ...
Minimalabstand Folgefahrt in Real	Real	0.0	35.674522	23,675 ...

Abbildung 5: Auswertung der Ergebnisse eines Testfalls (Spalte „New“) in Bezug auf die Abweichung zum im MBSE-Modell hinterlegten Anforderungen (Spalte „Margin“)

7 Diskussion

Die beschriebene Validierungsumgebung ermöglicht die automatische Durchführung verschiedener Testläufe als Beitrag zur Validierung von ACC- und AEB-Systemen. Wie in den vorangegangenen Kapiteln gezeigt, können mit Hilfe der Validierungsumgebung sowohl einfache Parameterstudien als auch komplexere, auf stochastischen Verfahren basierende, Testreihen durchgeführt werden.

Im Fokus dieser Arbeit stand der Aufbau der Validierungsumgebung zur Unterstützung automatisierter Tests im Rahmen der ACC- und AEB-Validierung. Die gezeigten Beispiele dienen zunächst dem Nachweis der Funktionsfähigkeit der Validierungsumgebung. Es bleibt zu untersuchen, inwieweit die durchgeführten Tests dem realen Verhalten von Fahrzeugen im Straßenverkehr entsprechen. Damit würde die Aussagekraft der in der Validierungsumgebung erzeugten Ergebnisse für die Validierung des ACC- und AEB-Systems eines Fahrzeugs verbessert.

Bestehende Forschung zeigt Möglichkeiten zur Einbindung physischer Prototypen und weiterer Systeme in die Simulationsumgebung CarMaker zum Aufbau einer gemischt-physisch virtuellen Validierungsumgebung (vgl. z.B. Lutz et al. 2017). Ziel der Einbringung physischer Prototypen ist die Erhöhung der Aussagekraft der Validierungsergebnisse mit fortschreitendem Reifegrad der Modelle im Produktentwicklungsprozess. Der Aufbau der Validierungsumgebung auf dem IPEK-XiL-Ansatz und die Wahl der Simulationsumgebung im vorgestellten Beitrag bietet die Möglichkeit die zunächst rein virtuelle Validierungsumgebung schrittweise durch physische Prototypen zu ergänzen. Wie genau physische Prototypen in die entwickelte Validierungsumgebung eingebracht werden können und welche Anforderung an Software- und Hardware-Schnittstellen sich daraus ergeben bleibt Gegenstand der Forschung.

Durch die Verteilung der Validierungsumgebung ist es möglich, die Konfiguration der Testläufe und die eigentliche Simulation an unterschiedlichen Standorten und über unterschiedlich performante Rechner durchzuführen. Dies wird insbesondere bei der Integration physischer Prototypen und –Prüfstände, mit gegebenenfalls an die Systeme gebunden Rechner, interessant.

In weiteren Arbeiten ist zudem zu untersuchen, wie sich die modellierten Anforderungen in weitere MBSE-Modelle und MBSE-Methoden integrieren lassen. Damit können beispielsweise Ansätze der modellbasierten mechatronischen Produktentwicklung wie FAS4X (Kleiner et al. 2017) erweitert werden.

8 Zusammenfassung und Ausblick

Dieser Beitrag beschreibt einen modellbasierten (MBE) Ansatz zum Aufbau einer virtuellen Validierungsumgebung für die Fahrerassistenzsysteme ACC und AEB. Durch die Möglichkeit eine Vielzahl von Fahrmanövern zu erstellen und automatisch nacheinander zu simulieren liefert die vorgestellte Validierungsumgebung einen Beitrag zur Automatisierung von Validierungsaktivitäten im Kontext von Fahrerassistenzsystemen. Die in der Validierungsumgebung durchgeführten Validierungsaktivitäten orientieren sich dabei an relevanten Gesetzen und Normen, um die Funktionalität des ACC- und AEB-Systems auf die Vorgaben hin zu untersuchen.

In zukünftigen Arbeiten wird die Nutzung von Schnittstellen und das Einbringen physischer Modelle in die virtuelle Validierungsumgebung untersucht und umgesetzt werden.

Das Einbringen physischer Modelle kann weiterhin einen Beitrag dazu leisten, die Aussagekraft der in der Validierungsumgebung erzeugten Ergebnisse zu überprüfen, um die Validierungsumgebung dahingehend optimieren zu können.

Die Verbindung der Validierungsergebnisse und MBSE-Modelle geschieht im vorliegenden Beitrag über die Verknüpfung der Testergebnisse und der im MBSE-Modell modellierten Anforderungen. In zukünftigen Arbeiten wird untersucht werden, wie sich die Erkenntnisse aus der Validierungsumgebung weiter in MBSE-Modellen nutzen lassen. Damit soll ein Beitrag zur Unterstützung modellbasierter Entwicklungsmethoden mechatronischer Systeme geleistet werden.

9 Literaturverzeichnis

Albers, Albert (2010): Five Hypotheses about Engineering Processes and their Consequences. In: Proceedings of the TMCE 2010.

- Albers, Albert; Behrendt, Matthias; Klingler, Simon; Matros, Kevin (2016): Verifikation und Validierung im Produktentstehungsprozess. Handbuch Produktentwicklung. In: Udo Lindemann (Hg.). München: Carl Hanser Verlag, S. 541–569.
- Beihoff, Bruce; Oster, Christopher; Friedenthal, Sanford; Paredis, Chris; Kemp, Duncan; Stoewer, Heinz et al. (2014): A World in Motion – Systems Engineering Vision 2025. Online verfügbar unter <https://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf>, zuletzt geprüft am 17.01.2019.
- Cruse, Thomas A. (Hg.) (1997): Reliability-based mechanical design. New York: M. Dekker (Mechanical engineering (Marcel Dekker, Inc.)), 108).
- Daimler AG (2018): PRE-SAFE®-Bremsen: Automatische Vollbremsung kann die Schwere eines Auffahrunfalls deutlich vermindern. Hg. v. Daimler AG. Online. Online verfügbar unter <https://media.daimler.com/marsMediaSite/de/instance/ko/PRE-SAFE-Bremsen-Automatische-Vollbremsung-kann-die-Schwere-eines-Auffahrunfalls-deutlich-vermindern.xhtml?oid=9361528>, zuletzt geprüft am 12.10.2018.
- Ehrlenspiel, K.; Meerkamm, H. (2013): Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit: Hanser.
- Eigner, Martin; Roubanov, Daniil; Zafirov, Radoslav (2014): Modellbasierte virtuelle Produktentwicklung. Berlin, Heidelberg: Springer Berlin Heidelberg.
- EuroNCAP (2015): European New Car Assessment Programme (Euro NCAP). Test Protocol - AEB systems. Version 1.1. Online verfügbar unter <https://cdn.euroncap.com/media/17719/euro-ncap-aeb-test-protocol-v11.pdf>, zuletzt geprüft am 11.03.2019.
- International Council on Systems Engineering (INCOSE) - Technical Operations (2007): INCOSE Systems Engineering Vision 2020. Hg. v. International Council on Systems Engineering. Online verfügbar unter http://www.cose.org/media/upload/SEVision2020_20071003_v2_03.pdf, zuletzt geprüft am 17.01.2019.
- IPG Automotive GmbH (2019): CarMaker. Online verfügbar unter <https://ipg-automotive.com/de/produkte-services/simulation-software/carmaker/>, zuletzt aktualisiert am 28.02.2019, zuletzt geprüft am 11.03.2019.
- Karl-Friedrich Kraiss (1998): Benutzergerechte Automatisierung – Grundlagen und Realisierungskonzepte. In: Ulrich Jumar (Hg.): at-Automatisierungstechnik. Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik, Band 46, Heft 10. Berlin, Boston: De Gruyter, S. 457–467. Online verfügbar unter <https://www.degruyter.com/view/j/auto.1998.46.issue-10/auto.1998.46.10.457/auto.1998.46.10.457.xml>, zuletzt geprüft am 11.03.2019.
- Kleiner, Sven; Husung, Stephan (2017): Model Based Systems Engineering. Prinzipien, Anwendung, Beispiele, Erfahrung und Nutzen aus Praxissicht. In: Sven-Olaf Schulze, Christian Tschirner und

- Rüdiger Kaffenberger (Hg.): Tag des Systems Engineering. Herzogenaurach, 25.-27. Oktober 2016. München: Carl Hanser Verlag GmbH & Co. KG, S. 13–22, zuletzt geprüft am 17.01.2019.
- Kleiner, Sven; Husung, Stephan; Mandel, Constantin; Albers, Albert; Behrendt, Matthias (2017): (Model-Based) Systems Engineering für die Digitalisierung der Produktentwicklung. In: Sven-Olaf Schulze, Christian Tschirner und Rüdiger Kaffenberger (Hg.): Tag des Systems Engineering. Paderborn, 8.-10. November 2017. München: Carl Hanser Verlag GmbH & Co. KG, 135-144, zuletzt geprüft am 11.03.2019.
- Lutz, S.; Albers, A.; Hettel, R.; Behrendt, M. (2017): Effizientere Validierung und Funktionsabsicherung von Fahrerassistenzsystemen auf dem Rollenprüfstand. In: Klaus Brökel, Karl-Heinrich Grote, Ralph Stelzer, Frank Rieg, Jörg Feldhusen, Norbert Müller und Peter Köhler (Hg.): Interdisziplinäre Produktentwicklung. 15. Gemeinsames Kolloquium Konstruktionstechnik 2017 : 5. und 6. Oktober in Duisburg : Modellbasierte Ansätze in der Produktentwicklung, konstruktionsbegleitende Berechnung und Simulation, Virtualität in der Produktentwicklung, Absicherung der Produktmodelldatenqualität, Wissensmanagement in Engineering-Prozessen, Bd. 15. Essen: Universität Duisburg-Essen Universitätsbibliothek.
- MathWorks (2019): MATLAB. Online verfügbar unter <https://de.mathworks.com/products/matlab.html>, zuletzt geprüft am 11.03.2019.
- Object Management Group (OMG) (2017): OMG Systems Modeling Language SysML 1.5. Online verfügbar unter <https://www.omg.org/spec/SysML/About-SysML/>, zuletzt geprüft am 07.03.2019.
- Paulweber, Michael; Lebert, Klaus; List, A. V.L.; Kiel, F. H. (2014): Mess- und Prüfstandstechnik. Wiesbaden: Springer Vieweg.
- Phoenix Integration (2019): Phoenix Integration ModelCenter. Online verfügbar unter <https://www.phoenix-int.com/>, zuletzt geprüft am 07.03.2019.
- PivotPoint Technology Corp. (2018): MBE Forum: Model-Based Engineering Visual Glossary. Online verfügbar unter <https://modelbasedengineering.com/glossary/>, zuletzt aktualisiert am 01.07.2018, zuletzt geprüft am 07.03.2019.
- Reinhard, G.; Lindemann, U.; Heinzl, J. (1996): Qualitätsmanagement. Ein Kurs für Studium und Praxis. Schnabel, Patrick (2004): Netzwerktechnik-Fibel. In: *Norderstedt: Books on Demand*.
- Siebertz, Karl; van Bebber, David; Hochkirchen, Thomas (2017): Statistische Versuchsplanung. Berlin, Heidelberg: Springer Berlin Heidelberg.
- VDI Norm 2206: VDI 2206 - Entwicklungsmethodik für mechatronische Systeme, zuletzt geprüft am 06.03.2019.
- Verband für bürgernahe Verkehrspolitik e.V.: Bußgeldkatalog 2018. Anhalteweg – Wie lange dauert der Bremsvorgang wirklich? Hg. v. Mathias Voigt (Vorsitzender). Online. Online verfügbar unter

https://www.bussgeldkatalog.org/anhalteweg/#bremsweg_8211_faustregel_fuer_normalen_br_emschweg_und_gefahrenbremsung, zuletzt geprüft am 06.10.2018.

- Winner, Hermann; Hakuli, Stephan; Lotz, Felix; Singer, Christina (2015): Handbuch Fahrerassistenzsysteme. Grundlagen Komponenten und Systeme für aktive Sicherheit und Komfort. 3., überarb. u. erg. Aufl. 2015. Wiesbaden: Springer Vieweg (SpringerLink : Bücher).
- Wolfgang Steichele (2016): Zahlen, Fakten, Wissen. Aktuelles aus dem Verkehr. Hg. v. ADAC e.V. Ressort Verkehr. Online (2831820/10.16/6'). Online verfügbar unter https://www.adac.de/_mmm/pdf/statistik_zahlen_fakten_wissen_1016_208844.pdf, zuletzt geprüft am 15.10.2018.

Kontakt

Constantin Mandel, M.Sc.
Constantin.Mandel@kit.edu

Sebastian Lutz, M.Sc.
Sebastian.Lutz@kit.edu

Olivia Rau, B.Sc.
Olivia.Rau@web.de

Dr.-Ing. Matthias Behrendt
Matthias.Behrendt@kit.edu

Univ.-Prof. Dr.-Ing. Dr. h. c. Albert Albers
IPEK-Institut für Produktentwicklung am KIT
Kaiserstraße 10
76131 Karlsruhe
sekretariat@ipek.kit.edu