

Entwicklung von Fahrerassistenzsystemen unter Verwendung von SIL- und HIL-Techniken im Verkehrsflusssimulationsprogramm PELOPS

Development of Driver Assistance Systems by Use of SIL and HIL Techniques in the Traffic Flow Simulation Program PELOPS

Dipl.-Ing. Karsten **Breuer**

Institut für Kraftfahrwesen der RWTH Aachen, ika

Dipl.-Ing. Frederic **Christen**

Forschungsgesellschaft Kraftfahrwesen mbH Aachen, fka

Zusammenfassung

Es werden Verfahren zur Software in the Loop (SIL) und Hardware in the Loop (HIL) Kopplung mit dem mikroskopischen Verkehrsflusssimulationsprogramm PELOPS vorgestellt. Anhand von Beispielen wird deren Anwendung dargestellt und die Möglichkeiten solcher gekoppelten Simulationen bei der Entwicklung von Fahrerassistenzsystemen veranschaulicht. Ein Ausblick eröffnet zudem weitere Anwendungsfelder der HIL-Kopplung im Bereich der Prüfstandssteuerung und Realfahrzeuguntersuchung im virtuellen Verkehr.

Summary

In this article methods are shown to use software in the loop (SIL) and hardware in the loop (HIL) techniques in connection with the microscopic traffic flow simulation program PELOPS. With some application examples the use of these methods are depicted. In this context the possibilities of such coupled simulation tools in development of driver assistance systems are presented. An outlook is given to future enhancements of the described HIL techniques to control for example test benches or assess real vehicles in virtual traffic.

1 Einführung

Die rapide Steigerung des Straßenverkehrsaufkommens in den letzten Jahren hat in Verbindung mit begrenzten Verkehrsraum zu einer immer stärkeren Belastung des Fahrers bei der Erfüllung seiner Fahraufgaben geführt. Aus diesem Grund werden seit einigen Jahren verstärkt sogenannte Fahrer-Assistenzsysteme entwickelt, die den Fahrer entlasten sollen und in Zukunft auch vermehrt zur Steigerung der Sicherheit auf der Straße beitragen sollen.

Bei der Entwicklung und Bewertung solcher Systeme wurde in der Vergangenheit bereits auf die computergestützte Simulation zurückgegriffen. Bisher gab es dabei nur die Wahl zwischen zwei Arten von Simulationssystemen, die beide nicht den Anforderungen nach einer einfachen Systemevaluation in einem realen Verkehrsumfeld gerecht werden konnten. Der eine Simulationstyp ist zwar in die heute vorhandenen Software-Werkzeuge der Assistenzsystementwickler integriert, erlaubt es jedoch nicht, komplexe und damit realistische Verkehrsszenarien zu berechnen. Die zweite Art der Simulation wurde ursprünglich zur Verkehrsflusssimulation entwickelt, bietet aber keine einfach zu handhabenden Möglichkeiten, im Sinne eines Rapid Prototyping schnell Systemmodelle oder Hardwareprototypen in die Simulation einzubinden und zu bewerten.

In diesem Artikel werden nun Verfahren vorgestellt, die diese Lücke schließen, indem sie das (sub-)mikroskopische Verkehrsflusssimulationsprogramm PELOPS mit den Möglichkeiten zur Software in the Loop und Hardware in the Loop Simulation erweitern und so Systementwicklern die Möglichkeit einer schnellen Funktionskontrolle und -bewertung neuer Systeme einräumen.

2 Das Simulationsprogramm PELOPS

Das Verkehrsflusssimulationsprogramm PELOPS (**P**rogrammsystem zur **E**ntwicklung **L**ängsdynamischer **mikr**oskopischer **V**erkehrs**P**rozesse in **S**ystemrelevanter **U**mggebung) wurde am Institut für Kraftfahrwesen Aachen in Zusammenarbeit mit der BMW AG entwickelt [1/ 2]. PELOPS stellt eine Kombination fahrzeug- und verkehrstechnischer Modelle dar. Dadurch besteht die Möglichkeit, alle Wechselwirkungen, die zwischen Fahrer, Fahrzeug und Verkehrsumgebung stattfinden, zu berücksichtigen. Den Kern des Programms bilden die drei wesentlichen Elemente des Verkehrs: Strecke / Umwelt, Fahrer und Fahrzeug (s. Abb. 1). Mit ihnen kann PELOPS den Verkehr in einer sehr hohen Genauigkeit simulieren. In einer modularen Programmstruktur werden die genannten Elemente abgebildet und durch Schnittstellen abgegrenzt.

Das Strecken- und Umweltmodell beinhaltet eine genaue Beschreibung der Straßentopologie hinsichtlich Höhenprofil, Kurvigkeit sowie Anzahl und Breite der Spuren. Zusätzlich werden Beschilderung und Umweltbedingungen wie etwa Sichtweiten und Fahrbahnreibwerte abgebildet. Mit diesem Modul kann ein breites Spektrum unter-

schiedlicher Straßen von der Autobahn bis zur innerstädtischen Kreuzung mit Ampeln simuliert werden. Um bestimmte Fahrsituationen nachbilden zu können, ist es möglich, ausgewählten Fahrzeugen einen fest vorgegebenen weg- oder zeitbasierten Geschwindigkeitszyklus aufzuprägen. Dies ist insbesondere dann notwendig, wenn Standardfahrzyklen wie 'EUDC' oder 'FTP-75' untersucht werden sollen.

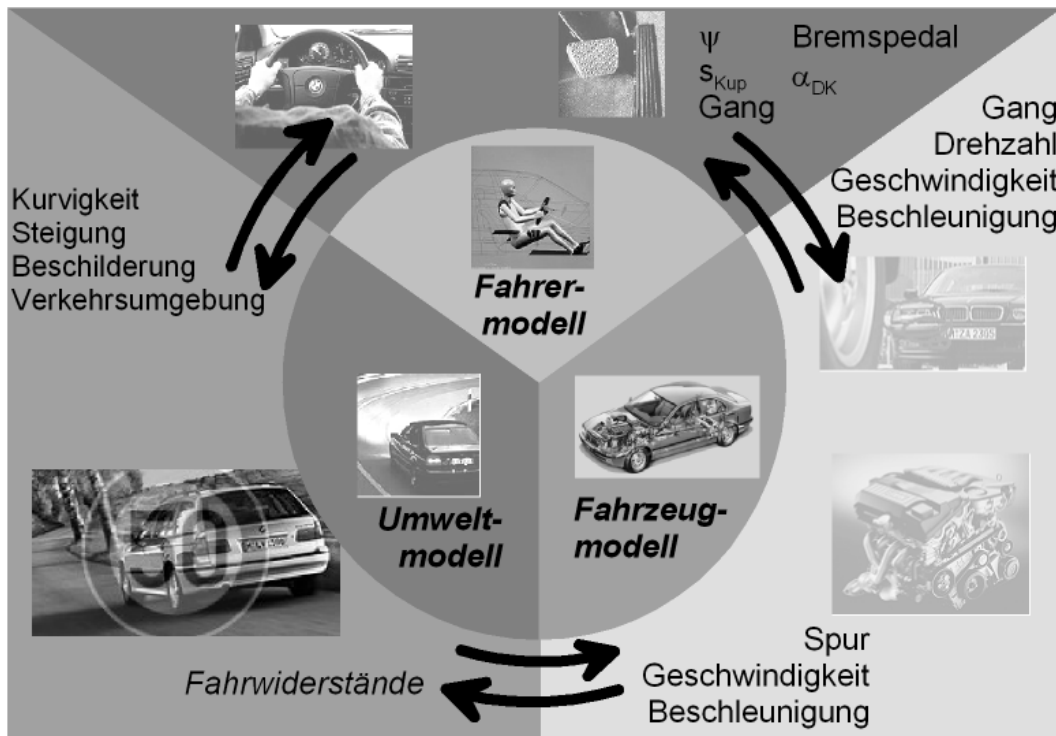


Abb. 1: Das Simulationsprogramm PELOPS /3/
elements of the simulation program PELOPS

Das Fahrermodell besteht aus einem Entscheidungs- und einem Handlungsteil. Im ersten Modul werden mit Hilfe eines psycho-physischen Verhaltensmodells auf Basis der aktuellen Verkehrssituation ein Beschleunigungs- und ein Spurwechselwunsch gebildet /4/. Im Handlungsmodell wird anschließend die Absicht des Fahrers in die entsprechenden Bedienelemente des Fahrzeugs wie etwa Gas- Brems- und gegebenenfalls Kupplungspedal, Blinker und Lenkradstellung umgesetzt.

Das Fahrzeugmodell basiert auf dem 'Ursache-Wirkungs-Prinzip', bei dem eine Berechnung der Antriebskraft ausgehend vom Motorbetriebspunkt über Kupplung, Getriebe und Differential zu den Rädern erfolgt, wo die Antriebskraft dann mit den Fahrwiderständen bilanziert wird. Der Betriebspunkt wird über die Änderung des Motormomentes (Ursache) gewechselt. Aus der dadurch verursachten Beschleunigung und Geschwindigkeitsänderung resultiert unter Berücksichtigung der Elemente des Antriebsstranges die Motordrehzahl (Wirkung). Als Getriebearten sind konventionelle Handschalt- sowie Automatikgetriebe implementiert. Für Nutzfahrzeuge kann zusätzlich ein Retarder im Antriebsstrang abgebildet werden /4/. Da das Fahrzeugmodell komponentenfein und damit sehr detailliert aufgebaut ist, können auch

Ergebnisse wie Kraftstoffverbrauch und Abgasemissionen hinreichend genau simuliert werden. Das Ursache-Wirkungs-Prinzip ermöglicht es ebenso, regelungstechnische Einrichtungen, wie etwa eine Getriebesteuerung zu untersuchen. Aber auch Fahrerassistenzsysteme können in PELOPS analysiert und auf ihre Tauglichkeit hin überprüft werden, um die Entwicklungszeiten solcher Systeme erheblich zu verkürzen.

Mit PELOPS können neben den schon erwähnten submikroskopischen Ergebnissen mit Hilfe von virtuellen Messschleifen auch mikroskopische (Verteilungen von Geschwindigkeiten, Zeitlücken etc.) und makroskopische Analysen durchgeführt werden. Mit den im Folgenden dargestellten Anwendungsbereichen kann die Vielseitigkeit des Programms verdeutlicht werden:

- Parametervariation in der Fahrzeugauslegung, Entwicklung von Konzepten zur Antriebsstrangregelung
- Konzeption, Entwicklung und Analyse von Fahrerassistenzsystemen zur längs- und querdynamischen Fahrzeugführung in verschiedenen Verkehrsumgebungen
- Analyse straßenseitiger Warn- und Informationssysteme und deren Interaktion mit Fahrzeugsystemen und Fahrern

3 Bisheriger Ansatz zur SIL-Simulation bei der Fahrerassistenzsystem-Entwicklung mit PELOPS

In verschiedenen Projekten wurden in der Vergangenheit schon Fahrerassistenzsysteme mit Hilfe von PELOPS simuliert und analysiert. Im Teilprojekt „ACC im Ballungsraum“ des nationalen Forschungsprogramms MoTiV wurden beispielsweise die verkehrlichen Auswirkungen des dort entwickelten ACC-Reglers untersucht und bewertet /5/. Als weiteres Beispiel kann in diesem Zusammenhang auch die Entwicklung des ika-eigenen S&G-ACC-Reglers angeführt werden, der für Funktionsprüfungen und Verkehrsuntersuchungen ebenfalls mit in die PELOPS-Simulation eingebunden wurde /6/.

Bei den genannten Untersuchungen wurde der zu analysierende Regler jeweils im Quellcode über eine proprietäre Schnittstelle innerhalb von PELOPS eingebunden (s. Abb. 2). Für diese Vorgehensweise ist sowohl auf Seiten des Regler- als auch des PELOPS-Entwicklers in bestimmten Bereichen Detailwissen über den Aufbau und die Funktionen des anderen Softwaremoduls notwendig, so dass hierbei keine Systementwicklung ohne Mitwirkung von Dritten möglich ist.

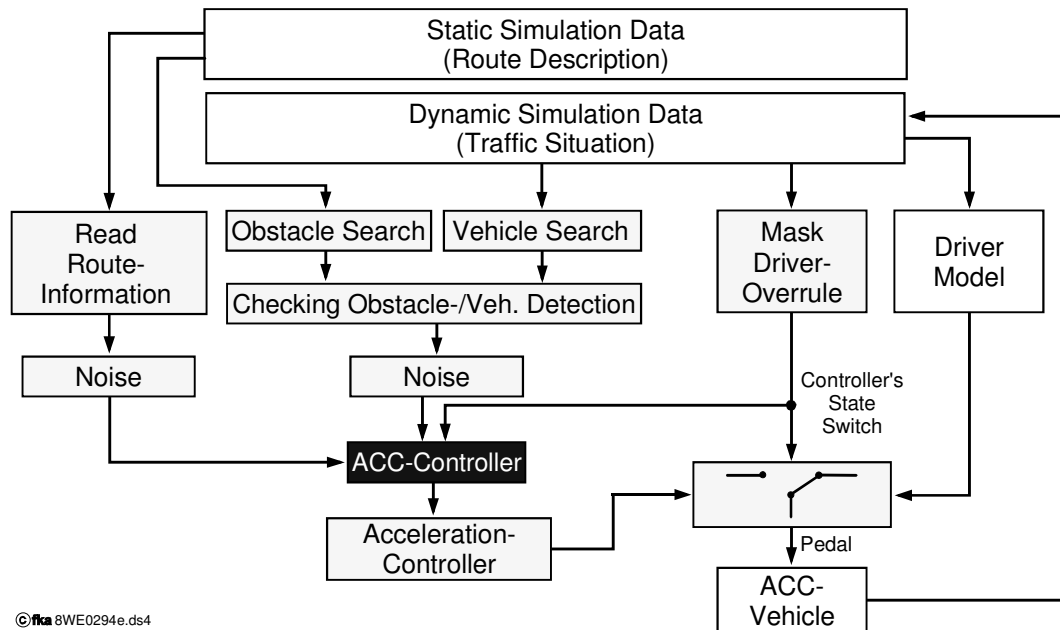


Abb. 2: bisherige Quellcodeeinbindung von ACC-Reglern in PELOPS /7/
integration of acc controller source code in PELOPS

4 Neuer SIL-Ansatz der gekoppelten Simulation am Beispiel MATLAB/Simulink

Mit Hilfe eines neuen Ansatzes ist es nun möglich, dass der im vorherigen Kapitel genannte Nachteil nicht mehr auftritt und gleichzeitig auch die von Systementwicklern normalerweise genutzten Werkzeuge eingesetzt werden können. Kern dieses Ansatzes ist die Kopplung verschiedener Simulationsprogramme. Das Ziel ist dabei die in sich geschlossene Abbildung eines virtuellen Fahrzeugs mit einem virtuellen Fahrer und einem virtuellen Assistenzsystem im virtuellen Verkehr, bzw. der sukzessive Austausch dieser einzelnen virtuellen Teilkomponenten durch reale Bauteile bei Aufrechterhaltung des gesamten Regelkreises. Abb. 3 zeigt beispielhaft, auf welche Weise hierbei die Modelle des Fahrerverhaltens und der Verkehrsumwelt mit einem Fahrzeugmodell in einem externen CAE-Tool verbunden werden können.

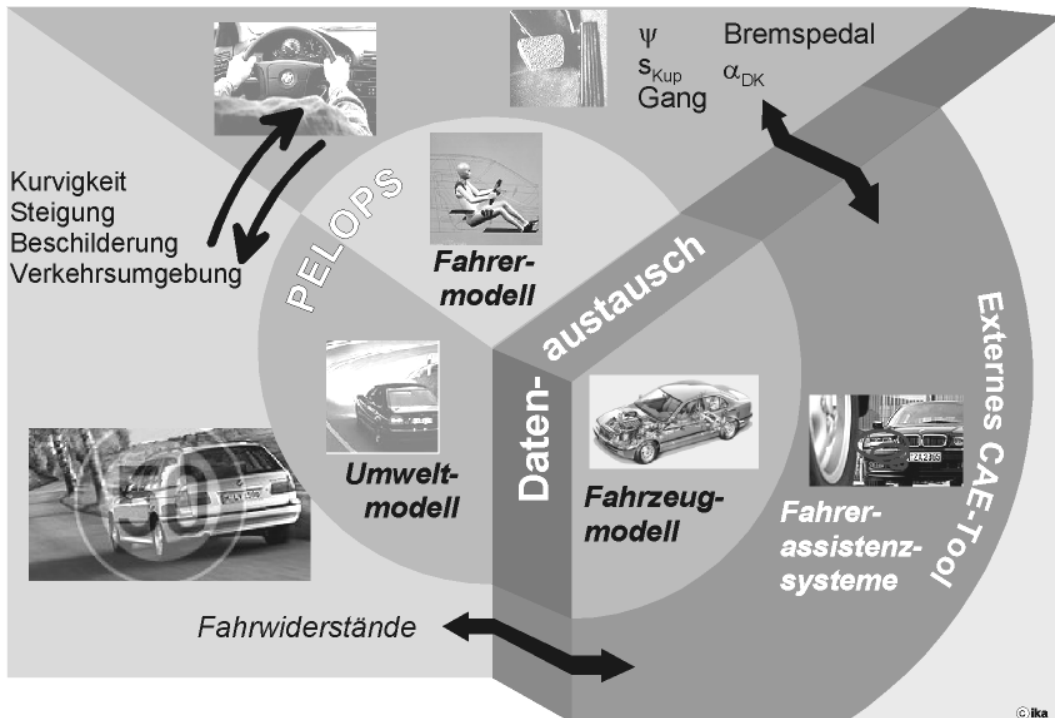


Abb. 3: Kopplung von Simulationstools über eine Online-Datenschnittstelle /8/
coupling of simulation tools via an online-data interface

Bei dieser Kopplung besteht sowohl die Möglichkeit, lediglich das zu entwickelnde Assistenzsystem als auch ein komplettes Vollfahrzeugmodell mit gegebenenfalls integriertem Assistenzsystem durch das gekoppelte CAE-Werkzeug abzubilden. Im ersten Fall stellt PELOPS zur Vervollständigung des betrachteten Fahrzeugs sein hochauflösendes Fahrzeugmodell zur Verfügung.

Nachfolgend wird die technische Umsetzung einer Online-Datenschnittstelle zwischen PELOPS und MATLAB/Simulink als Beispiel für das hier beschriebene Verfahren vorgestellt.

4.1 Funktionsweise des Datenaustausches zwischen MATLAB und PELOPS

Die zwischen den Teilmodellen anfallenden Daten in MATLAB und PELOPS werden über TCP/IP-Netzwerkverbindungen ausgetauscht. Diese Verbindungen können sowohl zwischen räumlich getrennten Computern als auch innerhalb eines einzigen Computers zwischen verschiedenen Prozessen erfolgen.

Prinzipiell gibt es zwei Möglichkeiten, solche Verbindungen herzustellen. Einerseits gibt es die Punkt-zu-Punkt-Variante, bei der sowohl dem sendenden Computer als auch der empfangenden Einheit die IP-Adresse / Port-Nummer des jeweils anderen bekannt sein muss. Diese Variante verschickt die Datenpakete immer nur an die dediziert angegebenen IP-Adressen, die auch außerhalb eines Subnetzes liegen können, sofern eine Firewall dies nicht verhindert. Dieses Verfahren wurde in einer ersten Version der PELOPS/MATLAB-Kopplung verwendet.

Eine zweite Verbindungsvariante realisiert den Datenaustausch durch die Verwendung sogenannter Multicast-Pakete. Dieses auch als „Broadcast“ bezeichnete Verfahren sendet die Pakete nicht an explizit bekannte IP-Adressen, sondern versieht die Nachrichten mit einer IP-Nummer aus dem Multicast-Adressraum, der sich von 224.0.0.0 ... 239.255.255.255 erstreckt. Diese Nachrichten verbreiten sich in der Regel aber nur innerhalb eines Klasse C Subnetzes (das letzte Byte der IP-Adresse ist variabel, d.h. die Subnetzmaske ist 255.255.255.0), was aber von der individuellen Konfiguration der jeweiligen Netzwerktechnik abhängig ist. Innerhalb dieses Subnetzes erhält jeder angeschlossene Rechner eine so versendete Nachricht und kann sie auf Wunsch verarbeiten. Dadurch ist es im Gegensatz zur ersten Variante auch möglich, Punkt-zu-Multipunkt-Verbindungen herzustellen, so dass z.B. ein PELOPS-Prozess mit mehreren MATLAB-Prozessen (auf einem oder mehreren Rechnern) kommunizieren kann, ohne dass ihm dafür die exakte Netzwerkkonfiguration der Gegenseite bekannt sein muss.

Während der Entwicklung und Nutzung hat sich die zweite Variante letztlich als die unproblematischere und leistungsfähigere herausgestellt und wird nun ausschließlich verwendet. Ein weiterer Vorteil liegt darin, dass so auch mehrere Benutzer auf einem Linux-Server mehrere gekoppelte PELOPS-Prozesse gleichzeitig rechnen können. Lediglich bei der Wahl der Multicast-Adressen muss dabei beachtet werden, dass die jeweils verwendete Netzwerktechnik (z.B. Router) unter Umständen einige Adressbereiche schon für administrative Prozesse im Netzwerk reserviert hat und diese daher für eine Kopplung nicht verwendet werden dürfen.

Die Synchronisation von PELOPS und MATLAB wird durch die Abwicklung der Kommunikation gewährleistet. Dabei werden in jedem Zeitschritt von PELOPS zunächst Daten an MATLAB verschickt und dann darauf gewartet, dass MATLAB für den aktuellen Zeitschritt seine Berechnungsergebnisse zurücksendet. Danach wartet MATLAB wiederum auf die nächste Nachricht von PELOPS. Durch diesen Mechanismus kann zwar keine (Echt-)zeitkontrolle bereitgestellt werden, aber die Datenkonsistenz zwischen den Simulationsprogrammen ist gegeben.

4.2 Realisierung der Kopplung mit PELOPS in einem MATLAB/Simulink-Modell

Zusätzlich zu den Simulink-Blöcken, die das eigentliche Fahrzeug- und/oder Reglermodell darstellen, werden weitere Funktionseinheiten benötigt, um den Datenaustausch mit PELOPS zu ermöglichen. Diese zusätzlichen Blöcke erfüllen mehrere Aufgaben. Ein Block ist für den Empfang der Daten und deren Bereitstellung in Datenstrukturen im Workspace notwendig, ein weiterer liest die Ergebnisse aus dem Workspace aus und schickt sie an PELOPS zurück. Außerdem werden zwei weitere Funktionsblöcke benötigt, die innerhalb des Modells zum Schreiben in den beziehungsweise zum Lesen aus dem Workspace benutzt werden müssen. Die beschriebenen Funktionsblöcke sind in Abb. 4 als Beispiel grau hinterlegt dargestellt.

Drei der oben angeführten Aufgaben werden von S-Functions übernommen, welche jeweils in einer höheren Programmiersprache (C / C++) programmiert sind und für

den Simulink-Nutzer nur als Block in Erscheinung treten. Das Lesen aus einer Workspace-Datenstruktur kann hingegen mit einer einfachen Matlab-Funktion erfolgen.

Die standardmäßige MATLAB/Simulink-Bibliothek enthält zwar einen *ToWorkspace*-Funktionsblock, der auf den ersten Blick für das Schreiben in den Workspace adäquat erscheint. Dieser ist jedoch für den Onlinezugriff nicht nutzbar, da die so geschriebenen Daten erst zugänglich wären, nachdem die jeweilige Simulation beendet oder angehalten wurde.

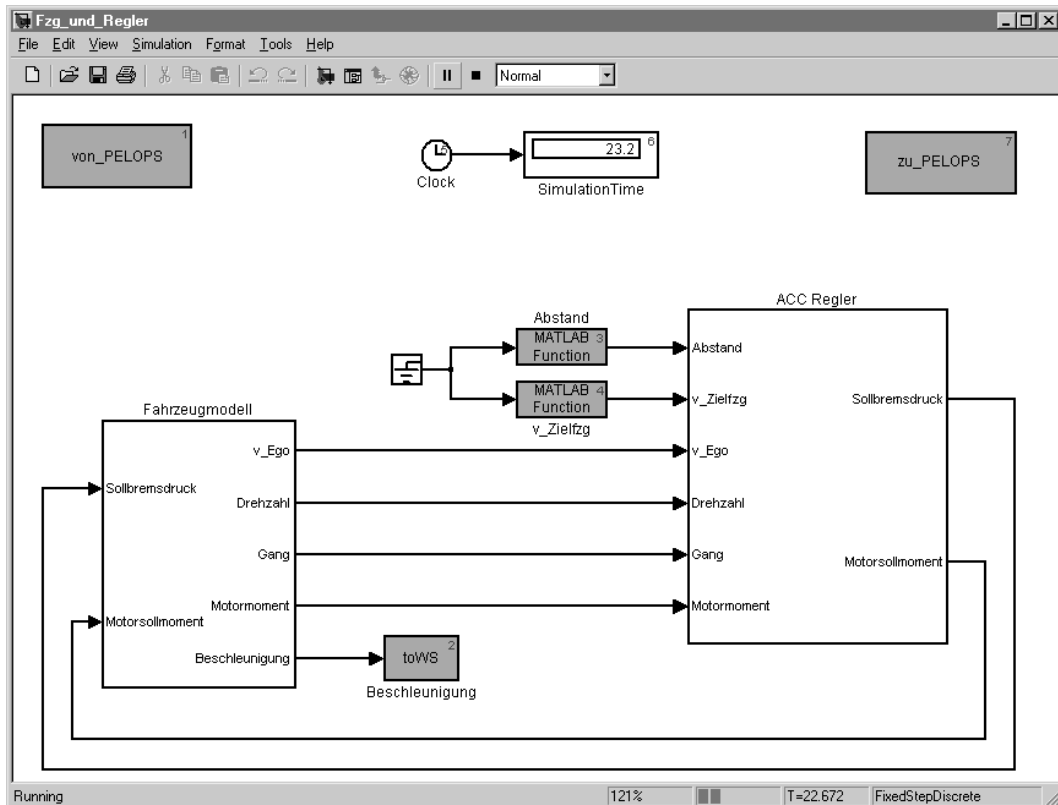


Abb. 4: beispielhafter Aufbau eines gekoppelten Modells in Simulink
example of a coupled model setup in Simulink

4.3 Anwendungsbeispiel

Das folgende Anwendungsbeispiel zeigt die Kopplung eines in MATLAB/Simulink realisierten Fahrzeugmodells mit integriertem ACC-Regler mit PELOPS (s. Abb. 5). In jedem Zeitschritt werden zu Beginn der Berechnung des externen Modells die hierfür benötigten Eingangsdaten über die TCP/IP-Verbindung von PELOPS nach MATLAB gesandt. Der ACC-Regler benötigt als Eingangsdaten von PELOPS den Abstand und die Relativgeschwindigkeit zum Zielfahrzeug. Anhand der von PELOPS empfangenen Daten und der Ergebnisse aus dem Simulink-Fahrzeugmodell (z.B. Eigengeschwindigkeit) errechnet der ACC-Regler einen Sollbremsdruck und ein Motorsollmoment. Diese Werte dienen wiederum dem Fahrzeugmodell zur Berechnung der aktuellen Beschleunigung, welche von MATLAB zurück nach PELOPS

geschickt wird, damit dort die neue Position des Fahrzeugs im Verkehrsraum berechnet werden kann. Anschließend kann PELOPS die Berechnung des externen Modells für diesen Zeitschritt beenden und die übrigen umgebenden Fahrer-Fahrzeug-Einheiten des Szenarios abarbeiten.

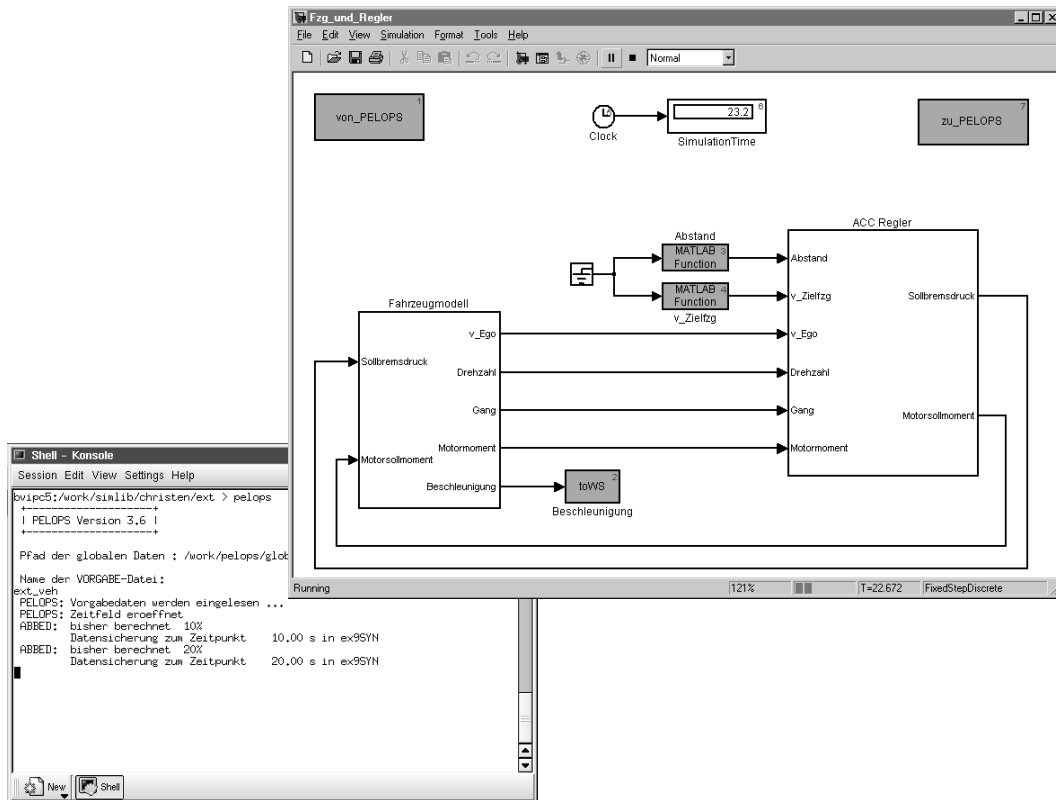


Abb. 5: gekoppelte Simulation PELOPS - MATLAB
coupled simulation PELOPS - MATLAB

In Abb. 6 und Abb. 7 sind Visualisierung und Simulationsergebnisse eines simulierten Spurwechsels dargestellt. Dabei fährt auf jeder Spur ein mögliches Zielfahrzeug mit konstanter Geschwindigkeit. Das ACC geregelte Fahrzeug nähert sich zu Beginn den beiden Fahrzeugen mit einer Differenzgeschwindigkeit von 20 km/h an.

In Abb. 7 ist zu erkennen, dass sich das geregelte Fahrzeug dem Zielfahrzeug auf der ersten Spur bis ca. 43 m annähert, bevor es einen Spurwechsel beginnt. Dabei kommt es zum Zielwechsel und der vom Sensor erfasste Abstand zum Zielfahrzeug springt auf etwa 64 m. Der ACC-Regler passt im weiteren Verlauf die Geschwindigkeit des Fahrzeugs der des neuen Zielfahrzeugs an und regelt wiederum einen (Ziel-)Abstand von 43 m ein. Um eine möglichst realitätsnahe Simulation zu erhalten, wird das Signalrauschen des Abstandssignals des simulierten Sensors durch eine Überlagerung des Grundsignals mit weißem Rauschen simuliert.

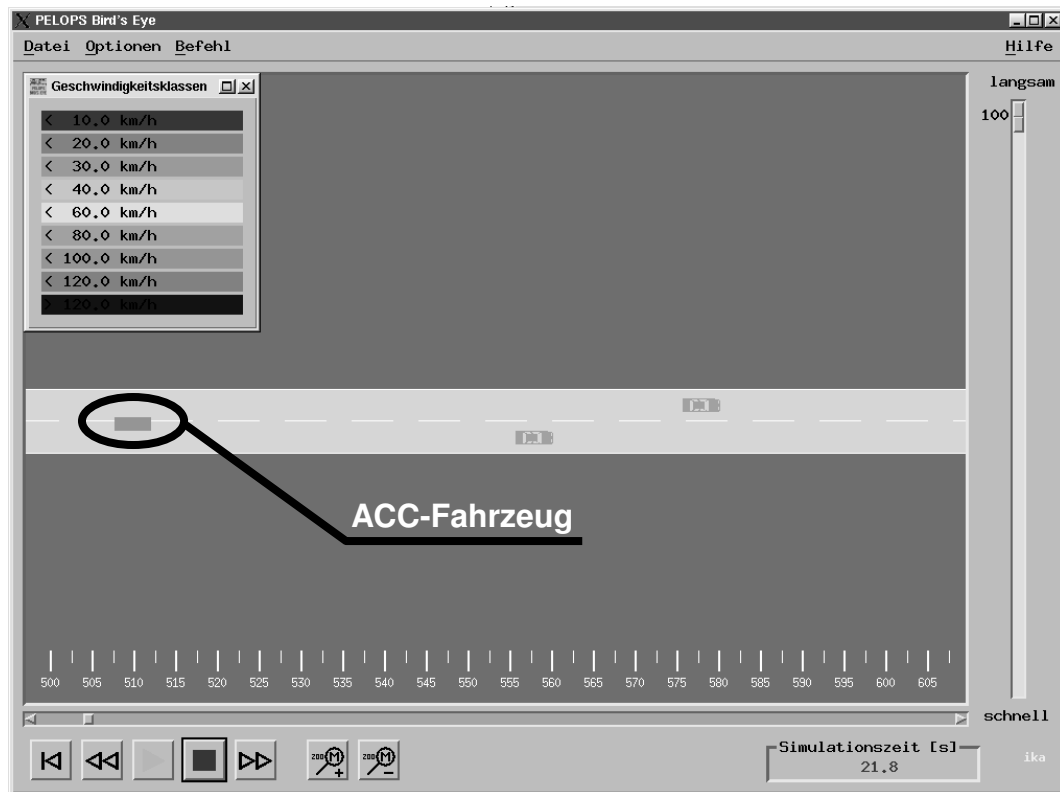


Abb. 6: Spurwechsel aus der Vogelperspektive
bird's eye view of the lane change situation

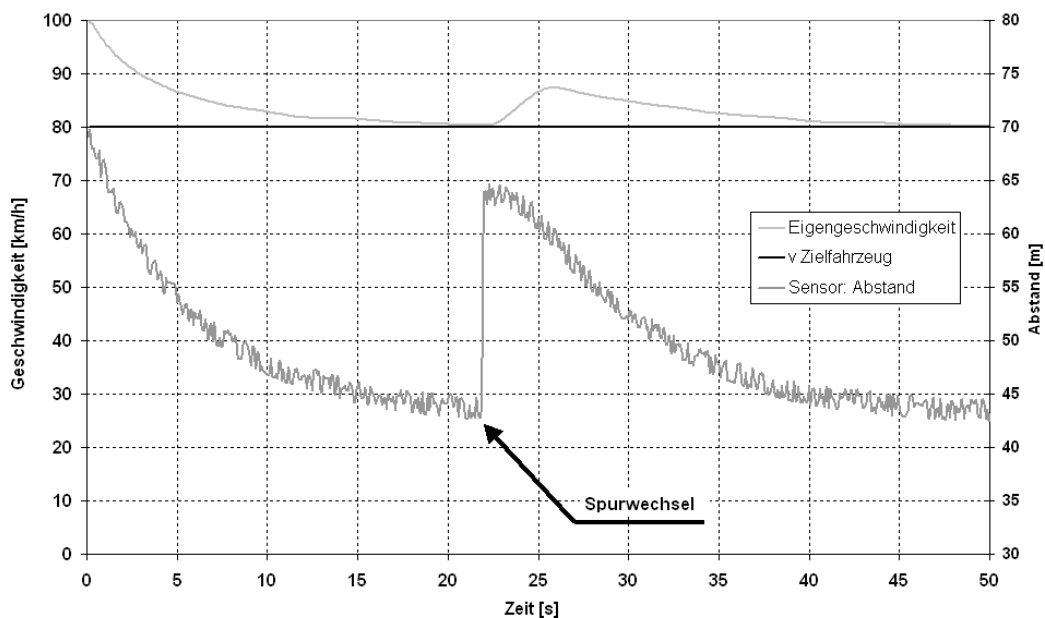


Abb. 7: Simulationsergebnisse (Abstand, Geschwindigkeit, Spur)
simulation results of a lane change situation (distance, velocity, lane)

5 Echtzeitkopplung von PELOPS mit Hardware als Weiterentwicklung des gekoppelten Ansatzes

Zusätzlich zu dem in den vorangegangenen Kapiteln beschriebenen Ansatz der SIL (Software in the Loop)-Kopplung von PELOPS, besteht seit kurzer Zeit auch eine erste Variante einer HIL (Hardware in the Loop)-Kopplung mit PELOPS, die nachfolgend näher beschrieben wird.

5.1 Funktionsweise

Während es bei einer SIL-Kopplung nicht notwendig ist, dass die Simulation in Echtzeit abläuft, muss diese Eigenschaft für eine HIL-Simulation gewährleistet werden. Da es sich bei dem auf der PELOPS-Seite verwendeten Betriebssystem Linux nicht um ein RT(real time)-Betriebssystem handelt, wird die Generierung der Zeitbasis einem dSpace-System übertragen, welches in diesem Fall mit einem digitalen Signalprozessor TMS320C40 von Texas Instruments ausgestattet ist. Dieses in der Industrie verbreitete System kann zudem gleichzeitig auch alle notwendigen Schnittstellen zur „realen“ Welt wie etwa A/D-Wandler oder CAN-Interfaces bereitstellen. Zwar kann diese Plattform selbst einen Prototyp eines Regelsystems darstellen. Aber der Hauptzweck dürfte im vorgestellten Fall der Kopplung mit PELOPS wohl in der Funktion der „Realweltschnittstelle“ für PELOPS liegen.

Die Kommunikation zwischen PELOPS und der dSpace-Box wird mit Hilfe einer seriellen RS422-Schnittstelle realisiert, die mit knapp 1 Mbit/s betrieben wird. Zu Beginn einer Simulation synchronisieren sich PELOPS und die Gegenseite zunächst und tauschen Datenprotokollparameter aus. Danach erfolgt der Datenaustausch in ähnlicher Weise wie bei der schon oben beschriebenen mit MATLAB. Innerhalb eines Simulationszeitschrittes schickt PELOPS zunächst Daten an die dSpace-Box und wartet anschließend auf eine Antwort.

Beim Versand werden die verwendeten Floating-Point Zahlen gleichzeitig entsprechend der auf den verschiedenen Prozessoren verwendeten Binärformate umkodiert und mit Zeitstempeln versehen. Beide Seiten kontrollieren bei jedem Datenempfang sowohl die Datenkonsistenz als auch die zeitliche Gültigkeit, so dass eine fehlerhaft laufende Simulation sofort erkannt werden kann. Damit dieser Fall nicht eintreten kann, muss die PELOPS-Simulation unbedingt schneller als Echtzeit rechnen können. Auf dem in diesem Fall verwendeten Linux-PC mit 1,2 GHz AMD Athlon Prozessor ist dies auch mit Szenarien möglich, in denen mehrere hundert Fahrzeuge berechnet werden müssen. Dadurch ist gewährleistet, dass schon mit dieser „einfachen“ PC-Ausstattung umfangreichere Szenarien mit der gekoppelten HIL-Plattform simuliert werden können.

5.2 Anwendungsbeispiel

Als ein Beispiel für die vorgestellte Methodik wird nachfolgend der S&G-ACC-Regler eines am ika aufgebauten Versuchsträgers (vgl. /9/) in HIL-Art an PELOPS gekoppelt. Bei dem verwendeten Steuergerät handelt es sich um ein am ika aufgebautes System auf Basis eines Infineon C167CS Mikrocontrollers, welcher normalerweise über eine CAN-Schnittstelle mit dem Versuchsfahrzeug verbunden ist. In den durchgeführten Simulationen ist das Gerät statt dessen über das CAN-Interface an die dSpace-Box angebunden (s. rechts unten in Abb. 8).

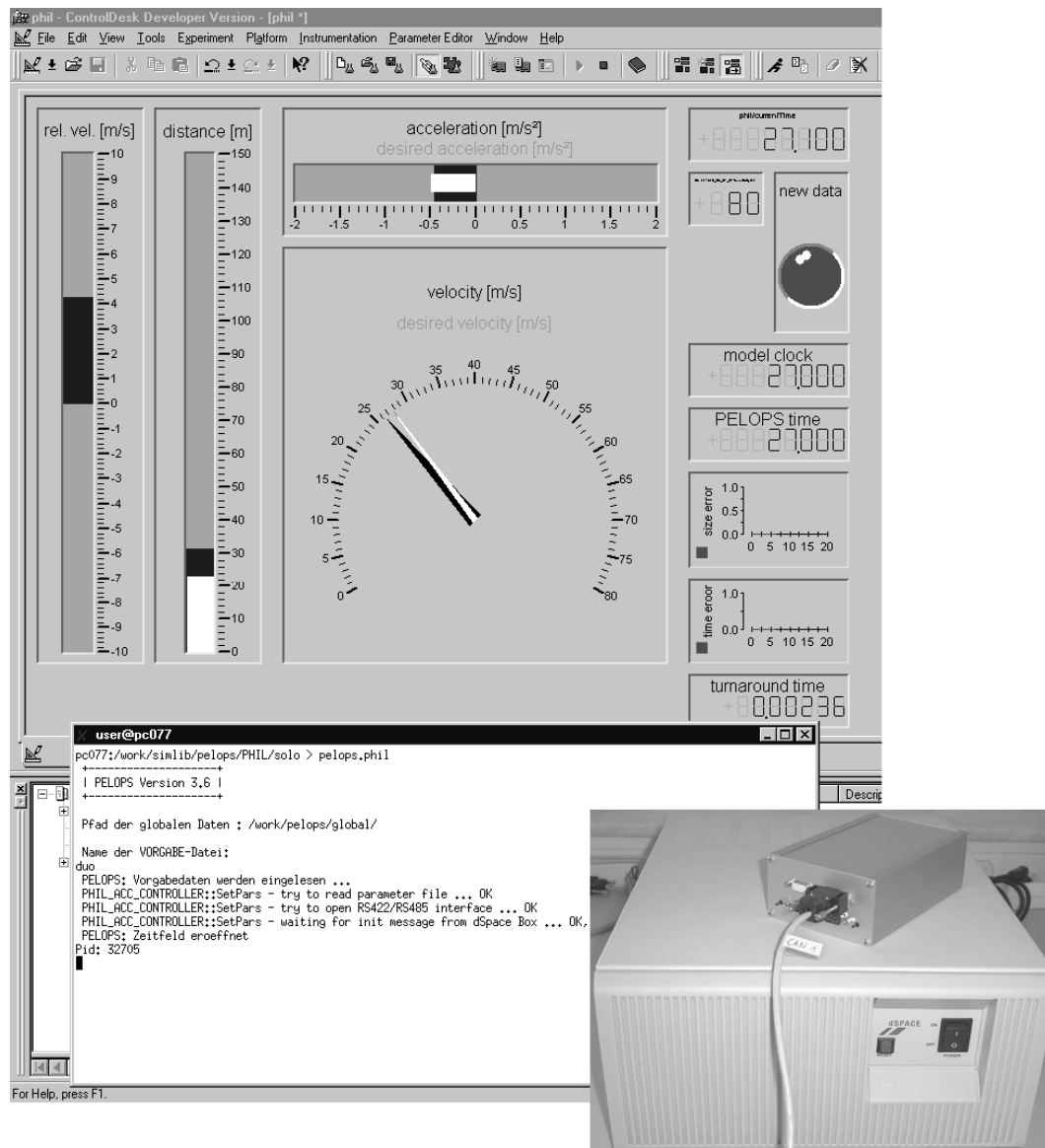


Abb. 8: HIL Simulation eines ACC-Steuergerätes mit PELOPS
HIL simulation of an ACC controller with PELOPS

Das in Abb. 9 gezeigte Simulink Modell, welches mit Hilfe des Real-Time Workshops und des zum DSP-Prozessor gehörigen TL-Compilers für die dSpace Plattform konvertiert wird, dient in diesem Fall lediglich als Schnittstelle für den Mikrocontroller zu PELOPS. Die von PELOPS empfangenen Fließkommazahlen werden entsprechend den Anforderungen des ACC-Reglers innerhalb der dSpace-Box in Integer-Werte umgesetzt und über CAN versendet, beziehungsweise die Antworten des Reglers vom CAN ausgelesen und nach entsprechender Konvertierung an PELOPS zurückgeschickt. Abb. 8 zeigt den Simulationsaufbau und die Ergebnis-anzeige während der Simulation eines Annäherungsvorganges an einen Lastkraftwagen.

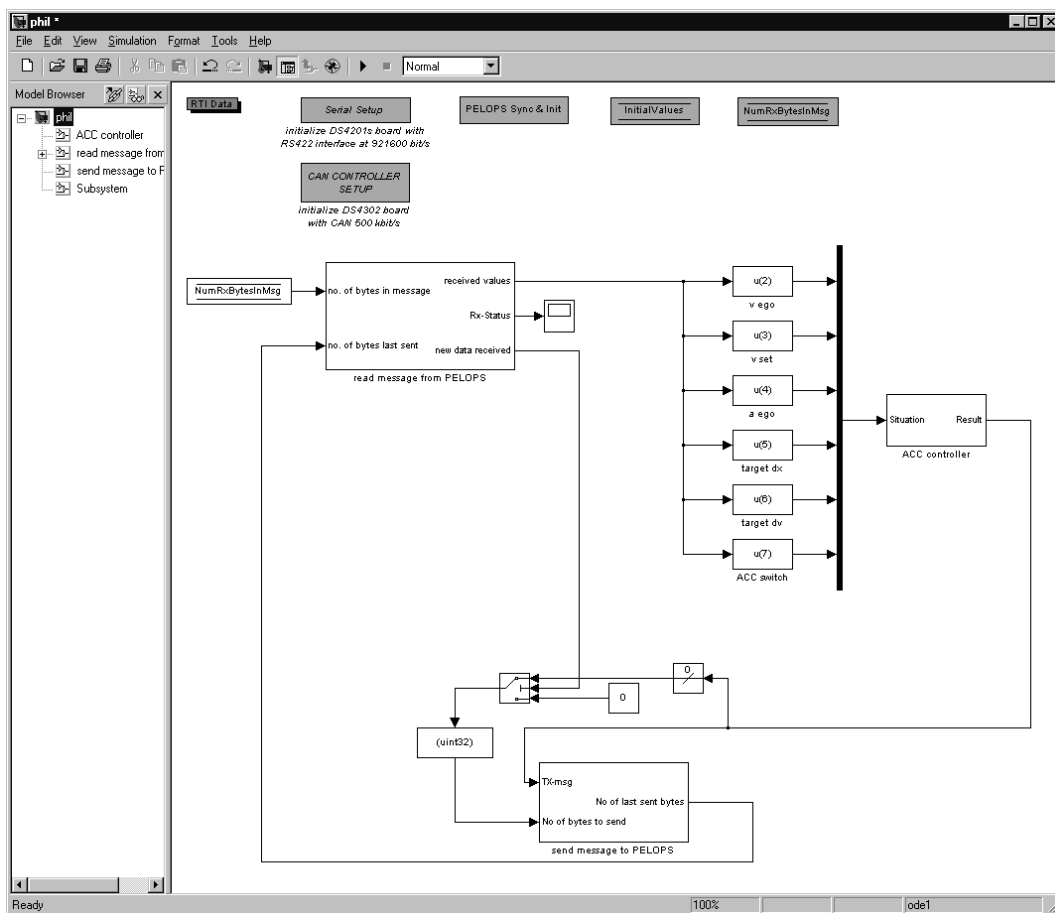


Abb. 9: Simulink Interface Modell für die HIL Simulation des ACC-Steuergerätes
Simulink interface model for the HIL simulation of the ACC controller

6 Ausblick

Mit der vorgestellten HIL-Kopplung von PELOPS wird es in Zukunft nicht nur möglich sein, einzelne Steuergeräte oder (Assistenz-)Systeme in einer realistischen Verkehrsumgebung zu testen und zu analysieren. Durch Erweiterungen der verwendeten Methodik kann die gekoppelte Simulation beispielsweise auch zur Ansteuerung von Antriebsstrangprüfständen genutzt werden. Anstelle von aufwendigen Messungen von Betriebszyklen im realen Verkehr, die danach als

Eingangsdaten für Prüfstandsuntersuchungen dienen, kann PELOPS direkt die benötigten Eingangsdaten an einen Prüfstand oder die dort zu testende Einheit liefern. Der Anwender wählt ein bestehendes Verkehrsszenario aus oder stellt sich ein neues nach seinen Wünschen zusammen. Durch diese Vorgehensweise sollte es möglich sein, die Vorbereitungszeiten für solche Untersuchungen zu reduzieren.

Darüber hinaus ergibt sich jedoch noch eine weitere Anwendung. Dabei wird die (sub-)mikroskopische Verkehrsflusssimulation direkt innerhalb eines realen Fahrzeugs eingesetzt. So wird es beispielsweise möglich sein, neuartige Assistenzsysteme ohne Gefährdung anderer Verkehrsteilnehmer im virtuellen, aber realistischen Verkehr auf einer leeren Teststrecke zu „erfahren“. Dies macht insbesondere auch eine Komfortbeurteilung möglich, die mit der reinen Simulation allein nur schwer erfolgen kann.

7 Literatur

- [1] LUDMANN, J.; DIEKAMP, R.; LERNER, G.
PELOPS Ein Programmsystem zur Untersuchung neuer
Längsdynamikkonzepte im Verkehrsfluß
VDI-Berichte Nr. 1007, Düsseldorf 1992
- [2] DIEKAMP, R.
Entwicklung eines fahrzeugorientierten Verkehrsflusssimulationsprogramms
Dissertation am Institut für Kraftfahrwesen der RWTH Aachen, 1995
- [3] EHMANN, D.
Simulationsmodell des menschlichen Spurwechselverhaltens
VDI/SAE/JSAE Gemeinschaftstagung „Der Fahrer im 21. Jahrhundert“
Berlin, 2001
- [4] HOCHSTÄDTER, A.; ZAHN, P.; BREUER, K.
Ein universelles Fahrermodell mit den Einsatzbeispielen Verkehrssimulation
und Fahrsimulator
9. Aachener Kolloquium für Fahrzeug- und Motorentechnik, 2000
- [5] WEILKES, M.; SCHREINER, F.; ONKEN, R.
Development and Assessment of a new ACC-Strategy for Urban Drive Control
1998 IEEE Conference on Intelligent Vehicles, Stuttgart 1998
- [6] WEILKES, M.
Auslegung und Analyse von Fahrerassistenzsystemen mittels Simulation
Dissertation am Institut für Kraftfahrwesen Aachen, 2000
- [7] BREUER, K. et al
PELOPS as a Tool for Development and Configuration
Euromotor-Course, Aachen, 1999

- [8] BENMIMOUN, A.; BREUER, K.; NEUNZIG, D.
Analyse der Potentiale von Assistenzsystemen mit Hilfe gekoppelter CAE-
Entwicklungswerkzeuge
19. VDI/VW Gemeinschaftstagung „Fahrzeugkonzepte für das 2. Jahrhundert
Automobiltechnik“, Wolfsburg, 2001

- [9] BREUER, K.
Implementierung eines intelligenten Tempomaten im Fahrzeug
Diplomarbeit am Institut für Kraftfahrwesen Aachen, 1998