

Flexible Entwicklungsumgebung zur realitätsnahen Simulation multimodaler Bedienkonzepte

Projektleiter

Prof. Dr.-Ing.
Thomas Felderhoff

Kompetenzplattform

Kommunikationstechnik
und Angewandte
Signalverarbeitung

Zeitraum

2001–2004

Förderung

Bundesministerium für
Bildung und Forschung,
anwendungsorientierte
Forschung- und
Entwicklung an Fach-
hochschulen (aFuE),
Fachhochschule
Dortmund,
Forschungsbudget

Kontakt

Prof. Dr.-Ing.
Thomas Felderhoff,
Labor für Informations-,
Processor- und
Softwaretechnik,
Signalverarbeitung
– LIPS,
Fachbereich
Informations- und
Elektrotechnik,
Fachhochschule
Dortmund,
Sonnenstraße 96,
44139 Dortmund,
Telefon:
(0231) 9112-386,
E-Mail: felderhoff@
fh-dortmund.de

Abstract

In vielen Bereichen des alltäglichen Lebens kommt der Mensch mit moderner Technik in Berührung. Soll die jeweilige Technik für ihn auch tatsächlich nutzbringend eingesetzt werden, dann ist eine ergonomisch ausgelegte, möglichst intuitiv zu bedienende Schnittstelle zwischen der Maschine und dem Menschen (HMI: human machine interface) ein entscheidendes Qualitätsmerkmal des Produktes.

In verschiedenen Einsatzbereichen moderner Technik, z.B. im Kraftfahrzeug, muss man sich unterschiedlicher Ein- und Ausgabemedien und –mechanismen bedienen, um trotz stark unterschiedlicher kognitiver Belastung des Nutzers geeignete Mechanismen für eine sichere und zielgerichtete Bedienung der Maschine zu ermöglichen. Der hieraus erwachsende Anspruch nach geeigneten multimodalen Bedienkonzepten läuft parallel mit einer immer schnelleren Produktentwicklung, im Sinne des Rapid Prototyping. Dies führt nahezu zwangsläufig zu einer engen Verzahnung der Arbeitsprozesse und setzt den Einsatz geeigneter Entwicklungswerkzeuge voraus.

Ergebnisse unterschiedlicher Forschungsprojekte [4,5] versetzen das LIPS in die Lage, solche multimodalen Bedienkonzepte in Kooperation mit Partnerfirmen oder in deren Auftrag zu entwickeln bzw. zu simulieren. Realitätsnahe Simulationen multimodaler Bedienkonzepte sind die erzielbaren Ergebnisse; damit lassen sich von Vergleichen unterschiedlicher Bedienabläufe durch eine schnelle Evaluierung des jeweiligen Look&Feel-Eindrucks bis hin zur realitätsnahen Simulation eines vollständigen Bedienkonzeptes unter Integration realer Komponenten alle Aspekte einer Mensch-Maschine-Kommunikation kompetent untersuchen.

I. Einleitung

Allgemein ist die einfache, möglichst intuitive Bedienbarkeit von Produkten ein entscheidendes Verkaufsmerkmal. Weniger in den technischen Leistungsmerkmalen als in dem realisierten Bedienkonzept liegen die tatsächlichen Produktunterschiede.

Speziell die Entwicklung von Bedienkonzepten für Fahrerassistenzsysteme im Kraftfahrzeug unterliegt zusätzlich der Anforderung, dass die Hauptaufgabe des Nutzers, das ist in der Regel der Fahrer, die sichere Fortbewegung von einem Ort zu einem anderen ist. Diese Aufgabe ist zusätzlich mit weiteren Teilaufgaben, z.B. der Unterhaltung,

der Kommunikation oder der Assistenz während der Fahrt angereichert. Manche Teilaufgaben werden vom Fahrer selber initiiert, z.B. die Wahl des Radiosenders, andere Ereignisse treffen den Fahrer eher unvorbereitet, z.B. ein Telefonanruf. Abhängig von der Verkehrssituation und der Fitness des Fahrers führen solche Ereignisse schnell zu einer Stresssituation, die dann durchaus die Sicherheit im Straßenverkehr beeinträchtigt.

Bedienkonzepte setzen gerade hier in zunehmendem Maße neben einer taktilen Bedienung auf einen orthogonalen Kommunikationsweg des Menschen, die Sprache. Leistungsstarke, phonembasierte sprecherunabhängige Spracherkennung erlauben heute bereits in der gestörten Umgebung eines Kraftfahrzeuges eine robuste Erkennung des Gesagten. So lässt sich ein umfangreicher Wortschatz von Kommandoworten und Wortkombinationen für die einzelnen Anwendungsbereiche für den Anwender zusammenstellen.

Die Architektur moderner Fahrerassistenzsysteme basiert auf umfangreichen vernetzten Subsystemen. Durch die Vernetzung stehen somit Informationen eines Subsystems durchaus auch den restlichen Subsystemen zur weiteren Ver- und Bearbeitung zur Verfügung. Aufgrund seiner großen Verbreitung liegt der Fokus in diesem Forschungsbereich auf dem CAN-Datenbus (CAN: Controller Area Network). Reale Komponenten, die über den CAN-Datenbus mit der Außenwelt kommunizieren, können hierüber beobachtet und auch gesteuert werden.

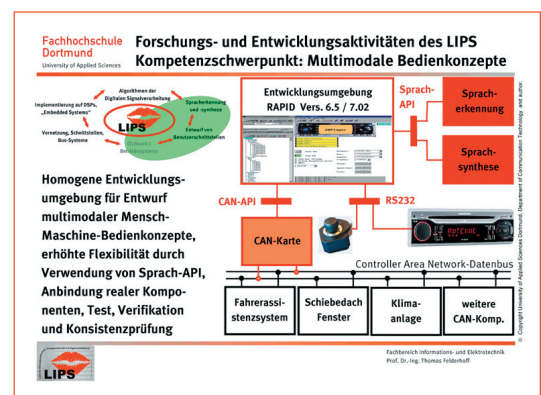


Abb. 1: Systemdarstellung der erweiterten Entwicklungsumgebung RAPID

Im LIPS ist eine kommerzielle Entwicklungsumgebung um Schnittstellen:

- zur Anbindung von Programmen zur Spracherkennung und zur Sprachsynthese (Sprach-API),
 - zur Anbindung von allen Komponenten mit einer Kommunikation über den CAN-Datenbus (CAN-API), sowie
 - zur Anbindung von Systemen, die eine Kommunikation über die serielle Schnittstelle (RS232) erlauben,
- erweitert worden (Abb. 1). Damit steht auf PC-Basis eine Entwicklungsumgebung zur Verfügung, mit der die logischen Zusammenhänge und zeitlichen Abläufe, die ereignisorientierte Steuerung und eben die Anbindung vieler externer Komponenten ermöglicht wird und so für die Entwicklung einer realitätsnahen Simulation von multimodalen Bedienkonzepten genutzt werden kann.

Unterschiedliche Demonstratoren bzw. durchgeführte Auftragsprojekte zeigen sowohl für die Sprachbedienung als auch für die Steuerung und Beobachtung realer Komponenten ein echtzeitfähiges Verhalten.

II. Entwicklungsumgebungen

Im Rahmen dieser Forschungsaktivität ist aus unterschiedlichen Entwicklungsumgebungen das Tool RAPID der Fa. e-Sim [3] ausgewählt worden. Besonderes Gewicht liegt auf einer flexiblen Anbindung weiterer Einheiten; dies sind sowohl Programme als auch PC-übliche Schnittstellen und industriennahe Schnittstellenkarten.

Komplexe Bedienabläufe sind auf einfache Weise beschreibbar. Für den Entwurf von Bedienabläufen benötigt der Entwickler keine Kenntnisse einer speziellen Programmiersprache, vielmehr ist es möglich, über eine entsprechend vorgegebene Entwicklungsoberfläche am PC die gewünschten Logikzusammenhänge und Beziehungen zwischen einzelnen Zuständen oberflächenorientiert einzugeben. RAPID verfügt über flexible Schnittstellen zur Anbindung externer Elemente oder auch externer Programme [3].

Durch die Möglichkeit benutzerdefinierte Objekte zu erstellen und zu kapseln, kann in einem Team von Entwicklern an dem gemeinsamen Ziel gearbeitet werden und jeder Entwickler hat eine separate, ihm zugeteilte Aufgabe.

III. Sprachein- und -ausgabe

Durch hochschuleigene Forschungsförderung wurde die Entwicklung der Sprach-API an RAPID [4] finanziert. Die Anforderungen an eine möglichst flexible Entwicklungsumgebung sollen nicht durch Leistungseinschränkungen bei der

Sprachbedienung reduziert werden. Deshalb ist die Spracherkennung und die Sprachsynthese der Fa. ScanSoft angebunden, eine umfangreiche und leistungsstarke Schnittstelle ist verfügbar [11]. Die Sprachprogramme laufen auf dem PC auf dem die Entwicklungsumgebung RAPID installiert ist.

Das Vokabular, das der Spracherkenner verstehen soll, kann anwendungsabhängig zusammengestellt werden. Besonders interessant ist die dynamische Anpassung dieses Vokabulars während der Laufzeit, d.h. während der Simulation. Somit können Listen, z.B. von gerade empfangbaren Radiosendern entsprechend ihrer ausgestrahlten RDS-Kennung, immer aktuell gehalten werden.

Die für reale Anwendungen im Kraftfahrzeug notwendige Vorverarbeitung des mit Störgeräuschen überlagerten Sprachbefehls wird im Bereich „Embedded Signalprocessing“ des LIPS ebenfalls bearbeitet. Erzielte Ergebnisse im Bereich der Geräuschreduktion, der (Stereo-)Echounterdrückung oder des adaptiven Beamforming sind in mehreren Projekt- und Diplomarbeiten gewonnen worden.

IV. CAN-Datenbusanbindung

Für die grundsätzlichen Zielsetzungen des Projektes boten sich zwei Kommunikationsstandards an. Dies sind der CAN-Datenbus und der MOST-Datenbus als optischer Datenbus mit hoher Datenrate.

Für eine realitätsnahe Simulation von Bedienabläufen bei modernen Car-Infotainment-Komponenten wäre die Anbindung des optischen Datenbusses interessant gewesen; die Verfügbarkeit von Komponenten, auch solchen, die dem LIPS als Sachspende oder mit offengelegter Schnittstellenbeschreibung zukamen, ist ungleich höher, wenn der CAN-Datenbus integriert wird. Alle prinzipiellen Erfahrungen bei der Anbindung eines Datenbusses [8] lassen sich auf andere Bussysteme übertragen.

Somit ist die Entwicklungsumgebung RAPID um eine Schnittstelle zum CAN-Datenbus erweitert worden. Es sind zwei standardisierte Protokolle nach CAN 2.0A oder nach CAN 2.0B implementiert. Ferner wird zwischen einem CAN-Low-speed- und einem CAN-High-speed-Datenbus für verschiedene Datenraten unterschieden. Mit der entwickelten Schnittstelle kann jede beliebige Kombination unterstützt werden, so dass keinerlei Einschränkungen in den Einsatzmöglichkeiten zu verzeichnen sind [5,8].

Auf PC-Ebene wird eine zweikanalige PCI-CAN-Karte verwendet. Die mitgelieferte DLL erlaubt

den vollständigen Zugriff auf alle Funktionen der Karte. Grundsätzlich sind diese Karten mit einem CAN-Highspeed-Controller ausgestattet. Für den Einsatz im CAN-Lowspeed-Modus ist zusätzlich ein entsprechendes Aufsteckmodul zu verwenden. Die Ansteuerung der Schnittstellenkarte ist davon unberührt. Es können also beide Datenratenbereiche vollständig bedient werden, ohne Änderungen in der softwaretechnischen Anbindung an die Entwicklungsumgebung berücksichtigen zu müssen.

Interessant wird solch ein Vorgehen, wenn nicht nur eine stationäre Lösung auf einem PC beabsichtigt wird, sondern wenn ein Laptop mit einer entsprechenden PCMCIA-Einsteckkarte benutzt wird. Dann ist es möglich, im mobilen Einsatz ein neues Bedienkonzept sogar im realen Umfeld eines Kraftfahrzeuges auszutesten. Im LIPS ist nun ein Schnittstellenprogramm verfügbar, das eine einheitliche Benutzung sowohl auf einem PC als auch auf einem Laptop erlaubt. Dieses eigenständige Schnittstellenprogramm ist die zentrale Einheit für die konkrete Anbindung einer CAN-Schnittstelle an die Entwicklungsumgebung RAPID [8].

Der prinzipielle Informationsaustausch zwischen der Entwicklungsumgebung RAPID und CAN-Komponenten ist in Abb. 2 dargestellt. Das Schnittstellenprogramm verwendet die mit der PCI-CAN-Karte gelieferte DLL, um Daten an die Karte zu senden bzw. um Daten von der Karte zu empfangen. Diese Daten werden umformatiert, so dass sie von RAPID gelesen werden können, bzw. die von RAPID gelieferten Daten an die CAN-Karte geschickt werden können.

RAPID selber gestattet eine Kommunikation mit diesem externen Programm über den APPLINK-Modus. Mit Hilfe eines von RAPID und dem Schnittstellenprogramm gemeinsam genutzten Speicherbereiches werden die Informationen ausgetauscht. Dieser Informationsaustausch ist ereignisgetrieben installiert, so dass nicht durch ständige Anfragen und Datenvergleiche auf Änderungen geschlossen werden muss, sondern es wird auf konkrete Änderungen zu dem Zeitpunkt der Änderung hingewiesen. Solche Meldungen treten immer dann auf, wenn neue Informationen über den CAN-Datenbus und der CAN-Karte vom Schnittstellenprogramm empfangen und aufbereitet wurden. Dieses ereignisgesteuerte Vorgehen gestattet eine optimale, d.h. möglichst geringe Rechenleistung des Schnittstellenprogramms auf dem PC.

Ein wesentlicher Punkt bei einer realitätsnahen Simulation ist die Realzeitfähigkeit des gesamten Systems, d.h. hier, speziell der Simulation auf

dem PC. Dabei ist anwendungsbezogen unter Realzeitfähigkeit zu verstehen, dass die Reaktionen auf externe Einflüsse so schnell geschehen, dass sich nicht mit der Zeit Befehle anhäufen, die nicht schnell genug abgearbeitet werden. Die Rechnerleistung in Kombination mit dem verfügbaren Speicherplatz sind wichtige Einflussgrößen, denn es ist neben der Behandlung der Schnittstellenkommunikation auch genügend Rechenleistung für die Entwicklungsumgebung RAPID zur Verfügung zu stellen. Eine eventuell angebundene Spracherkennungsmaschine benötigt im aktivierten Betrieb zusätzliche Rechenressourcen, wobei zu diesen aktiven Zeiten, davon ausgegangen werden kann, dass die Entwicklungsumgebung keine wesentlichen Anforderungen stellt, da sie auf Befehle zur Zustandsänderung wartet.

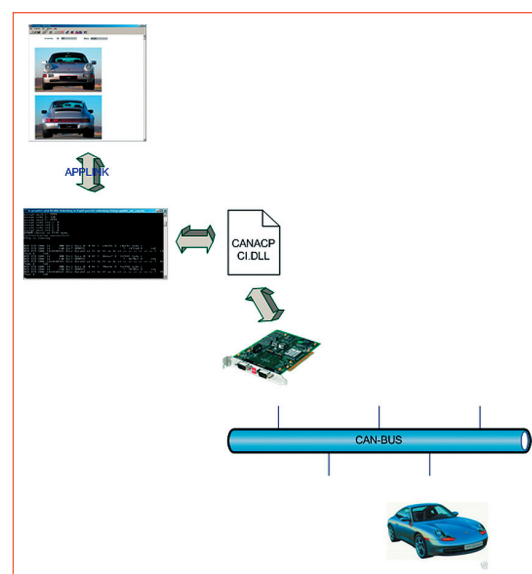


Abb. 2: Prinzipielle Kommunikation zwischen Entwicklungsumgebung RAPID und Komponenten am CAN-Datenbus

Der Einsatz von Echtzeitbetriebssystemen bei der Simulation ist unüblich und nicht weiter verfolgt worden. Ein produktfähiges Bedienkonzept wird sicherlich an ein Echtzeitbetriebssystem anzupassen sein.

V. Serielle Schnittstelle

Nicht alle Systeme sind mit einer Datenbusschnittstelle ausgestattet, da ältere Systeme meistens autonom arbeiten und somit gar kein Bedarf an einem Datenaustausch existiert. Für Prüfzwecke gibt es dann aber doch eine firmenspezifische Kommunikationsmöglichkeit über die serielle

Schnittstelle, RS232, zum PC. Naturgemäß sind solche Systeme nicht vollständig beobachtbar und meist nur eingeschränkt steuerbar.

Die Integration solcher Systeme in die Entwicklungsumgebung kann zur Steigerung der Realitätsnähe durchaus sinnvoll sein: Die Systemabhängigkeit einer eigenständigen Radioquelle anstelle einer Simulation durch Abspielen einer Wave-Datei auf dem PC ist sofort verständlich. Von besonderem Mehrwert für dieses Projekt ist die Anbindung eines haptischen Drehstellers über die serielle Schnittstelle [6]. Dies eröffnet neue Dimensionen in der taktilen Bedienung und damit in der Gestaltung aller Bedienabläufe. Abb. 1 zeigt, dass genau diese beiden Komponenten der Firmen Grundig bzw. Kostal integriert sind.

VI. Anwendungsbeispiele der multimodalen Entwicklungsumgebung

In Verbindung mit unterschiedlichen Entwicklungsschritten oder hervorgerufen durch konkrete Firmenkontakte zeigen verschiedene Demonstratoren die neue Leistungsfähigkeit der erweiterten Entwicklungsumgebung RAPID. Im Fokus stehen jeweils unterschiedliche Kombinationen von Aspekten bei der Betrachtung der Bedienabläufe bzw. beim Zusammenspiel einzelner Komponenten. Drei Anwendungsbeispiele seien exemplarisch vorgestellt:

1. Modellauto

Ein Modellauto vom Typ Porsche ist mit unterschiedlichen Leuchtdioden zusätzlich ausgerüstet, die von einem 68HC912BC-Entwicklungsboard aus getrieben werden. Neben Blinkern, Abblend- und Fernlicht kann das Bremslicht, die Innenbeleuchtung, oder ein Tür- und Temperatursensor abgefragt werden. Dieses Board kommuniziert über den CAN-Datenbus mit einem zweiten Board gleichen Typs, das wiederum mit einem PC verbunden ist. Nun besteht die Möglichkeit, sowohl vom PC über den CAN-Datenbus das Porsche-Modellauto zu steuern und zu beobachten [8] als auch über das am PC angebundene Entwicklungsboard diesen Datenaustausch vorzunehmen [1]. Beide Wege stehen grundsätzlich auch aus der Entwicklungsumgebung RAPID heraus zur Verfügung. Nun kann einerseits über eine simulierte taktil Bedienung das Modell beeinflusst werden oder andererseits über entsprechend definierte Sprachbefehle.

Abb. 3 zeigt das Interesse an solch einem Demonstrator, das weit über den Kreis der Studierenden hinausgeht.



Abb. 3: Interesse am Porsche-Modell

2. Drehsteller und Komfortautositz

Ein über die serielle Schnittstelle angebundener haptischer Drehsteller der Fa. Kostal [6] ist anders in ein Bedienkonzept zu integrieren als z.B. eine Kreuzwippe. Die einzelnen Menüs sind nun rotations-symmetrisch und weisen kaum noch eine rechteckige Anordnung auf. Von solch einer Simulation wird die Steuerung eines Komfortautositzes [12], den ebenfalls die Fa. Kostal zur Verfügung stellte, vorgenommen.

Aus den konkreten Untersuchungen lassen sich Erfahrungen sammeln, wie einerseits eine intuitive Bedienung durch eine geeignete Wahl der Bedienmasken erleichtert wird. Andererseits können die vielfältigen Bedienmöglichkeiten des haptischen Drehstellers mit den Erwartungen eines Nutzers zur Verstellung eines Autositzes evaluiert werden.

3. Routenführungssystem

Zu den wohl anspruchsvollsten Aufgaben einer multimodalen Bedienung im Umfeld der Kraftfahrzeuganwendungen gehört momentan die Zieleingabe bei einem Routenführungssystem. Eine Navigationseinheit der Fa. Blaupunkt kann über den CAN-Datenbus gesteuert und beobachtet werden.

Die Simulation eines möglichen Bedienkonzeptes mit der Entwicklungsumgebung RAPID erlaubt die Zieleingabe an diesem neuen Frontend sowohl über ein taktil Bedienungsfeld als auch per Sprache [2]. Ist die Intention des Nutzers, ein Routenführungsziel einzugeben, vollständig auf dem PC abgearbeitet, wird dieses Ziel an das reale Routenführungssystem über den CAN-Datenbus übermittelt und eine Zielführung kann gestartet werden.

Dieser Demonstrator erlaubt umfassende Untersuchungen für die geeignetste Dialoggestaltung, um effizient das Ziel eingeben zu können, wobei Mehrdeutigkeiten, Erkennungsfehler des Spracherkenners oder Bedienfehler des Nutzers ihre Berücksichtigung finden müssen. Wissenschaftliche Erkenntnisse [7,9,10] werden zusätzlich herangezogen, um die realisierten Bedienabläufe zu optimieren. Gerade in der Kombination einer taktilen Bedienung mit optionaler Ergänzung um eine Sprachbedienung mit einem Dialog zwischen Maschine und Mensch liegen vielfältige Variations- und Entwicklungsmöglichkeiten.

VII. Zusammenfassung

Im LIPS steht eine erweiterte Entwicklungsumgebung zur realitätsnahen Simulation multimodaler Bedienkonzepte unter Anbindung realer Komponenten zur Verfügung. In einer Sitzkiste werden alle untersuchten Komponenten integriert und über ein einheitliches Bedienkonzept angesteuert.

Das so aufgebaute Know-how erlaubt dem LIPS:

- die Simulation weiterer Bedienkonzeptideen, u.a. mit einem personalisierten oder adaptiven Verhalten;
- die Evaluierung unterschiedlicher Bedienkonzepte, speziell auch im Hinblick auf den Sicherheitsaspekt für den mobilen Einsatz im Kraftfahrzeug;
- die Beratung u.a. bei der Auswahl von Entwicklungsumgebungen, bei der Spezifikation von Schnittstellen oder bei der Entwicklung konsistenter Bedienabläufe;
- die Unterstützung von Firmen im Einsatz von Entwicklungsumgebungen und der Erweiterung um konkrete Schnittstellen;
- die Entwicklung spezieller Demonstratoren für den mobilen Einsatz durch Verwendung eines Laptops oder einer embedded PC-Plattform; oder
- die gezielte Erweiterung der eigenen Sitzkiste um weitere Komponenten.

Verankert sind diese Forschungs- und Entwicklungsaktivitäten auch in der Lehre, besonders zu nennen ist der Masterstudiengang „Informationstechnik“. Interessierte und motivierte Studierende werden in Zukunft weiterhin auf diesem Gebiet ihre Ideen und Innovationen beisteuern und gleichzeitig von der Kompetenz des LIPS profitieren.

Alle weiteren Forschungsaktivitäten des LIPS sowohl auf diesem Gebiet der „Multimodalen Bedienkonzepte“ als auch auf dem erwähnten Gebiet der „Embedded Signalprocessing“ werden in die Kompetenzplattform „Communications and Applied Signal Processing“ der Hochschule integriert.

VIII. Danksagung

Die Entwicklung der nun in vollem Umfang einsatzbereiten und erweiterten Entwicklungsumgebung wurde sowohl durch hochschulinterne Forschungsförderung als auch durch die Förderung im Rahmen des Programms angewandte Forschung und Entwicklung des Bundesministers für Bildung und Forschung, Förderkennziffer 170 22 02, ermöglicht. Darüber hinaus haben die Firmen Blaupunkt, e-Sim, Grundig, Kostal, ScanSoft, Telelogic und TerraTec durch Sachmittelspenden oder durch die Offenlegung detaillierter Produkt- und Schnittstelleninformationen maßgeblichen Anteil an der Überzeugungskraft der erstellten Demonstratoren.

IX. Literatur

- [1] Beulen, R., Anbindung des MC68HC912-Entwicklungsboards an die Simulationsumgebung RAPID, Diplomarbeit, FH Dortmund, 2000
- [2] Brockmann, M., Entwurf und realitätsnahe Simulation eines multimodalen Bedienkonzeptes zur Zieleingabe mit anschließender dynamischen Routenführung, Diplomarbeit, FH Dortmund, 2005
- [3] e-Sim, Rapid Plus 7.0 User Manual, e-Sim Ltd., 2002
- [4] Felderhoff, T. und Nennstiel, N., Erweiterung der Simulationsumgebung RAPID um Spracherkennung und Sprachsynthese zum homogenen Entwurf multimodaler Bedienkonzepte, Schlussbericht, FH Dortmund, 2001
- [5] Felderhoff, T., Erweiterung einer Entwicklungsumgebung um eine CAN-Kommunikation zum Entwurf realitätsnaher multimodaler Bedienkonzepte durch Anbindung realer Komponenten, Schlussbericht aFuE-Förderprojekt, FKZ 170 22 02, 2004
- [6] Grieger, D., Anbindung eines haptischen Drehstellers an die Simulationsumgebung RAPID, Projektarbeit, FH Dortmund, 2003
- [7] Hutchins, E.L. und andere, User-centered System Design: New Perspectives in Human-Machine-Interface, Lawrence Erlbaum, 1986
- [8] Nennstiel, N., Technischer Bericht zur Anbindung einer CAN-Schnittstelle an die Entwicklungsumgebung RAPID, LIPS-interner Bericht zum aFuE-Projekt FH Dortmund, 2004
- [9] O'Shaughnessy, D., Speech Communications-Human and Machine, 2. Aufl., IEEE Press, 2000
- [10] Shneiderman, B., Designing the User Interface, Addison-Wesley, 1997
- [11] ScanSoft, User Manuals for Automatic Speech Recognition and Speech Synthesis ScanSoft GmbH
- [12] Stahl, B., Anbinden eines Komfortautositzes über CAN-Datenbus an eine Entwicklungsumgebung zum Steuern und Beobachten, Diplomarbeit, FH Dortmund, 2004