

Systemsimulation in der Mechatronik – Einsatz beim Entwurf autonomer intelligenter Systeme

Peter Schneider, Sebastiaan Wielens, Peter Schwarz, Stefan König
Peter.Schneider@eas.iis.fraunhofer.de
Fraunhofer IIS, Außenstelle Entwurfsautomatisierung (EAS)
Zeunerstraße 38, 01069 Dresden

Am Beispiel autonomer intelligenter Systeme wird die Anwendung einer Modellierungsmethodik für mechatronische Systeme vorgestellt. Ausgehend von wesentlichen Systemeigenschaften wird der Modellierungsansatz beschrieben und seine Anwendung beim Entwurf eines autonomen Fahrzeugs demonstriert. Einen wesentlichen Schwerpunkt bildet dabei der Entwurf von Algorithmen zur Navigation in der Umgebung und zur Vermeidung von Kollisionen mit Hindernissen und anderen autonomen Fahrzeugen.

1 Einleitung

Für die Simulation mechatronischer Systeme existieren zahlreiche leistungsfähige Simulationsansätze und -werkzeuge [2],[3],[5]. Im Bereich autonomer intelligenter Systeme kommen zu den „üblichen“ Teilsystemen mechatronischer Systeme (Sensoren, Signalverarbeitung und Aktuatoren) noch Subsysteme für die Navigation in der Umgebung sowie für die Kommunikation und Interaktion autonomer Systeme untereinander hinzu (Bild 1).

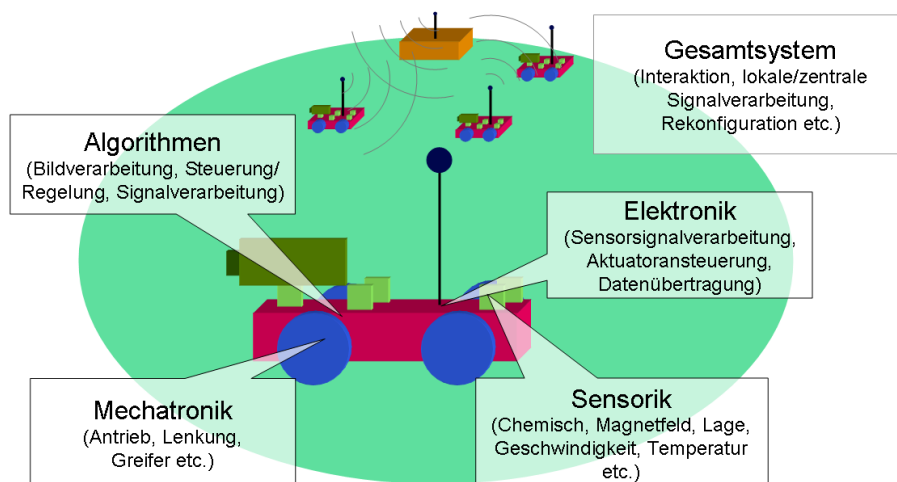


Bild 1: Entwurfsaufgaben bei autonomen, intelligenten Systemen

Gerade für diese Art der Umgebungsabbildung müssen die bestehenden Modellierungsansätze für mechatronische Systeme erweitert werden.

2 Modellierung mechatronischer Systeme

Für die Modellierung mechatronischer Systeme liegt aufgrund der im Komponentenentwurf verwendeten unterschiedlichen Beschreibungsmittel und Simulationwerkzeuge der

Gedanke nahe, für die Teilsystem-Modellierung das jeweils angepasste Beschreibungsmittel zu verwenden. Eine Gesamtsystemsimulation wäre dann durch Kopplung verschiedener Simulatoren realisierbar, was jedoch bei sehr komplexen heterogenen Systemen zu hohem rechentechnischen Aufwand für die Simulatorkopplung führt.

Da sich die physikalischen Sachverhalte in mechatronischen Systemen zum großen Teil durch gewöhnliche Differentialgleichungen (ODEs) bzw. differential-algebraische Gleichungen (DAEs) beschreiben lassen, wird im Folgenden ein Multi-Level-Modellierungsansatz [1],[7] auf Basis dieser mathematischen Beschreibung favorisiert, der allerdings auch die informationsverarbeitenden Teilsysteme der Steuerung und Regelung mit einbezieht. Verteilte Systeme können unter bestimmten Voraussetzungen nach räumlicher Diskretisierung ebenfalls erfasst werden.

Eine Vorgehensweise für die Modellierung mechatronischer Systeme lässt sich wie folgt beschreiben:

- Das Gesamtsystem wird für die Modellierung in der gleichen Weise in Teilsysteme zerlegt, wie es der Entwickler auf Grund der Kenntnis physikalischer Vorgänge im System als sinnvoll ansieht. Schnittstellen bzw. Modellgrenzen werden dabei so gelegt, dass diese einer experimentellen Untersuchung zugänglich sind. Dies vereinfacht die Validierung der Modelle anhand von Messungen.
- Die Teilsysteme tauschen untereinander „Signale“ aus – das können Energieflüsse, elektrische oder mechanische Größen, Signale im Regelungsteil usw. sein. Den Modellen der Teilsysteme stehen ebenfalls nur diese Signale am Rand (Interface) zur Verfügung. Die Modellbeschreibung erfolgt für unterschiedliche Detaillierungsstufen möglichst unter Beibehaltung der Schnittstellengrößen. Damit wird die Austauschbarkeit von Modellen unterschiedlicher Genauigkeit ermöglicht.
- Das Modell des Gesamtsystems entsteht durch Verbindung der Teilmodelle in der Struktur, wie auch das Gesamtsystem aus den Teilsystemen zusammengesetzt wird.
- Die Modelle der Teilsysteme bestehen wiederum aus der Verbindung elementarer Modelle (hierarchisches Vorgehen) oder aus gleichungsorientierten Verhaltensbeschreibungen; die Mischung von Struktur- und Verhaltensmodellen ist möglich.

Um die Modellierung für derartige Systeme zu vereinfachen, ist die Festlegung einer einheitlichen internen Struktur und des Interfaces aller Modelle sinnvoll. Bild 2 zeigt schematisch Klemmentypen und die verwendeten Größen.

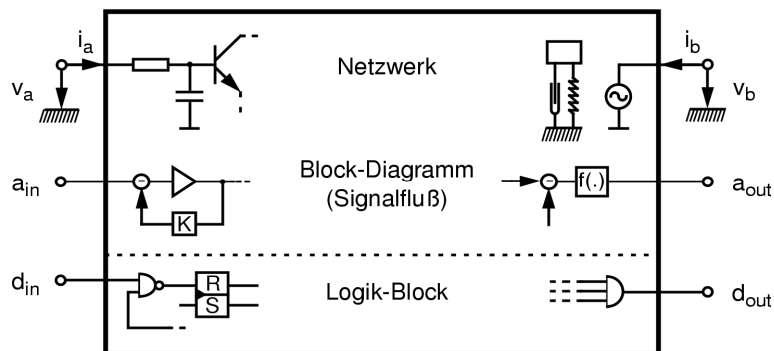


Bild 2: Klemmentypen und verwendete Größen

Weitere Details zu diesem Modellierungsansatz, einschließlich der einheitlichen internen mathematischen Beschreibung, sind in [1] beschrieben. Im Wesentlichen wird für die Modellierung zwischen den folgenden Größen an den Rändern eines Modells unterschieden:

- 1 Flussgrößen i_a , i_b und Differenzgrößen v_a , v_b – sie treten (z.B. in elektrischen Netzwerken) immer gemeinsam auf und dienen zur Beschreibung des Energieaustausches zwischen den Systemkomponenten (konservative Größen).
- 2 Eingangssignale a_{in} und Ausgangssignale a_{out} zur Modellierung der Ränder von Blöcken in Regelungssystemen und Signalflussdiagrammen (nichtkonservative Größen).
- 3 Digitale Signale d_{in} und d_{out} zur Modellierung digitaler Teilsysteme.

Die Klemmengrößen sind zeitabhängige Funktionen, so dass auch transiente Vorgänge untersucht werden können. Das interne Verhalten eines Blockes (Klemmenverhalten) wird durch DAE oder durch Netzwerkdarstellungen unter Verwendung von Grundelementen sowie auch Kombinationen dieser Beschreibungsformen beschrieben. Dieser Modellierungsansatz bietet somit die Möglichkeit, Modelle technischer Systeme in Struktur und mit wählbarer Genauigkeit zu erstellen. Dies unterstützt den Entwurfsprozess mechatronischer Systeme, da durch wechselseitige Abstraktion und Verfeinerung Lösungskonzepte am Simulationsmodell untersucht werden können.

Der hier beschriebene Modellierungsansatz wird durch, z.T. standardisierte, Modellbeschreibungssprachen wie MAST, VHDL-AMS, VERILOG-AMS und MODELICA sowie durch auf diese Sprachen ausgerichtete Simulationswerkzeuge unterstützt. Gleichungs- und signalflussorientierte Simulatoren wie MATLAB/Simulink sind ebenfalls für die Simulation der genannten Systeme geeignet, allerdings ist die Handhabung von konservativen Größen umständlicher, was sich insbesondere bei der Systemmodellierung mit Hilfe elektrischer oder mechanischer Netzwerke nachteilig auswirkt. Vorteile bieten diese Tools, wenn der Entwurf von Signalverarbeitungs- oder Regelalgorithmen im Vordergrund steht, da sie dafür zumeist über umfangreiche Toolboxen verfügen.

3 Anwendung beim Entwurf autonomer intelligenter Systeme

Für die Modellierung autonomer intelligenter Systeme [4], [6] ist der in Abschnitt 2 beschriebene Ansatz anwendbar, muss aber um folgende Elemente erweitert werden:

- Modelle für die Umgebungsabbildung und die gegenseitige Wechselwirkung von Fahrzeug und Umgebung, einschließlich anderer Fahrzeuge
- Leistungsfähige Visualisierungs- und Animationstechniken
- Modelle und Algorithmen zur Simulation von Kommunikationsprotokollen und der Ausbreitung von Funksignalen (einschließlich Dämpfung und Störung)
- Schnittstellen zur Ankopplung von realer Hardware an Simulationsmodelle

Nachfolgend wird anhand eines konkreten Beispiels auf einige dieser Aspekte näher eingegangen.

3.1 Beschreibung des Versuchsträgers

Als Versuchsträger wird ein Modellfahrzeug (Bild 3) eingesetzt, welches mit einer Steuerungs- und einer Sensorplattform ausgerüstet wurde. Algorithmen zur Regelung und

Navigation sind auf einem Hostrechner implementiert, der entweder auf dem Fahrzeug integriert ist oder über eine drahtlose Datenverbindung angekoppelt werden kann.

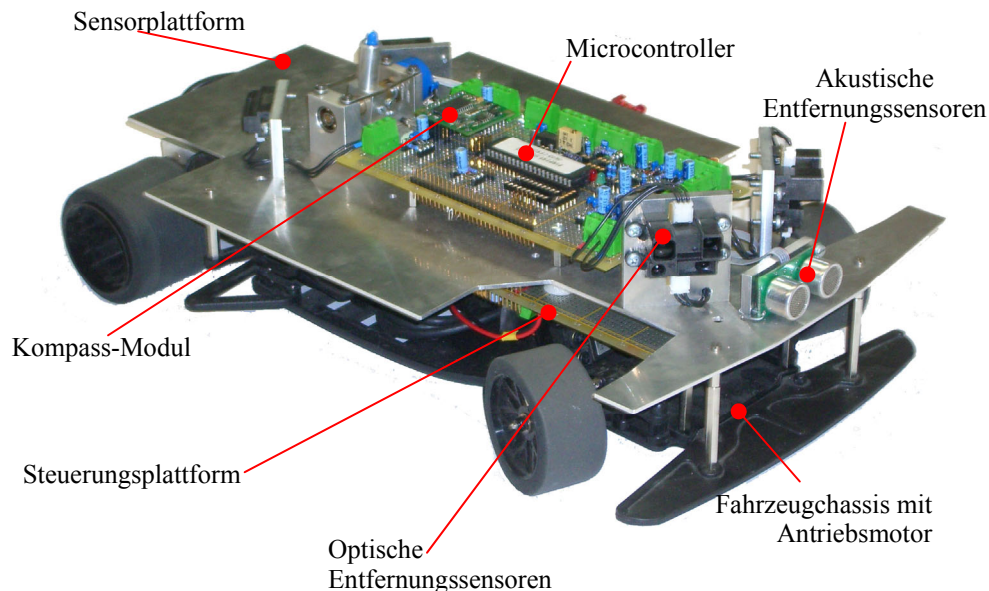


Bild 3: Versuchsträger mit Steuerungs- und Sensorplattform

Die Microcontroller-basierte Steuerungsplattform dient zur Realisierung der reinen Fahrfunktionalität. Dazu werden Informationen zu Fahrgeschwindigkeit und Fahrweg über einen Inkrementalgeber an der Antriebsachse des Fahrzeugs erfasst und an die Regelung weitergegeben. Die vom Regelalgorithmus empfangenen Stellsignale für Lenkung und Geschwindigkeit werden in PWM-Signale zur Ansteuerung des Motorreglers und Lenkservos gewandelt.

Auf der Sensorplattform sind vier Infrarot-Sensoren montiert – zwei schräg nach vorne, zwei schräg nach hinten orientiert. Sie dienen der Erfassung der Entfernung zu Hindernissen im direkten Umfeld des Fahrzeuges auf Basis des Triangulationsverfahrens. Insbesondere bei erhöhter Geschwindigkeit des Fahrzeuges ist es erforderlich, die Entfernung zu Hindernissen, die unmittelbar in Fahrtrichtung liegen, zu bestimmen. Aus diesem Grund wurde am Fahrzeug frontseitig ein Ultraschallsensor montiert, dessen Reichweite (3 m), verglichen mit Infrarot-Sensoren deutlich höher ist. Ein Problem an diesem Messprinzip sind die durch seitliche Reflektionen hervorgerufenen Messfehler. Anhand des zurückgelegten Weges (Drehgeber an der Hinterachse) ist noch keine eindeutige Positionsbestimmung im zweidimensionalen Raum möglich. Mit Hilfe eines Kompassmoduls wird deshalb zusätzlich die Ausrichtung des Fahrzeuges bezüglich des Erdmagnetfeldes ermittelt, wodurch eine näherungsweise Berechnung der Position möglich wird. Das Einlesen der Sensorwerte sowie eine erste Vorverarbeitung werden durch einen Microcontroller auf der Sensorplattform realisiert.

Neben diesen festgelegten Basisaufgaben können bei Bedarf bestimmte Aufgaben vom Hostrechner zu lokalen Microcontrollern transferiert werden. Wesentliche Kriterien für die Auswahl einer konkreten Konfiguration (Hostrechner auf dem Fahrzeug oder

drahtlose Ankopplung) und die Verteilung der Signalverarbeitungsaufgaben sind der numerische Aufwand für die Algorithmen und die Datenmenge, die zwischen Fahrzeug und Hostrechner übertragen werden muss. Auch das Echtzeitverhalten der einzelnen Teilsysteme ist ein wichtiger Aspekt. Für die nachfolgend beschriebene Applikation werden die Regelung zur Geschwindigkeit und zur Fahrtrichtung auf dem lokal eingesetzten Microcontroller der Steuerungsplattform implementiert. Beiden Regelungen fordern höhere Abstraten und müssen wegen der Nutzung zeitlicher Ableitungen echtzeitfähig realisiert werden. Die Ankopplung des Hostrechners erfolgte über eine drahtlose Verbindung.

3.2 Analyse des Fahrzeugantriebs und der Regelung

Für die detaillierte Analyse des Antriebssystems sowie der Lenkung bietet sich eine Modellierung unter Nutzung verallgemeinerter KIRCHOFFscher Netzwerke [8] an. Durch schrittweise Verfeinerung von Modellen für Getriebe, Motor, Antriebsachsen, Lenkgestängen etc. können das Zusammenwirken von verschiedenen, z.T. nichtlinearen Effekten, untersucht und konstruktive Modifikationen und die Auswahl besser geeigneter Baugruppen unterstützt werden.

Zur Absicherung des Entwurfs der Regelungsalgorithmen sowie zur Validierung der blockorientierten Modellierung wurden so Detailsimulationen des Antriebs- und Lenksystems durchgeführt. Bild 4 links zeigt unterschiedlich detaillierte Netzwerkmodelle des Fahrzeuggetriebes. In Bild 4 rechts ist das Ergebnis der Simulation des mechanischen Antriebs und der Fahrdynamik des Versuchsfahrzeugs mit dem später auch für die Gesamtsystemsimulation verwendeten Modell für Antrieb und Lenkung dargestellt. Dabei wurden u.a. der Einfluss des Lenkwinkels auf die Fahrzeuggeschwindigkeit sowie Haft-, Gleit- und viskose Reibung berücksichtigt.

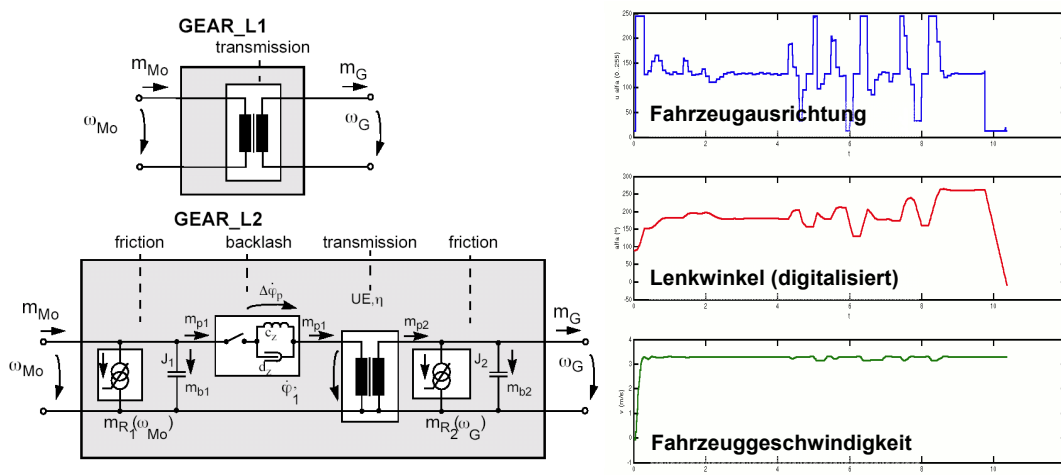


Bild 4: links - verschiedene Genauigkeitsvarianten für ein Getriebemodell, rechts - Ergebnisse der Simulation von Antriebs- und Lenksystem

3.3 Modellierung des Gesamtsystems

Wegen des hohen Anteils an Signalverarbeitungsalgorithmen und der späteren Ankopplung realer Hardware an die Simulation wurde für die Modellierung des Gesamtsystems MATLAB/SIMULINK eingesetzt. Für die Modellierung des Antriebsstrangs sowie der Fahrzeugdynamik musste dazu vom eigentlich angepassten netzwerkorientierten Ansatz auf die signalflossorientierte Modellierung übergegangen werden. Dieser Nachteil wird allerdings durch die einfachere Umsetzung der verschiedenen Algorithmen nach C-Code mittels „Real-Time-Workshop“ und somit die einfachere echtzeitfähige Implementierung auf einem Zielprozessor (Embedded Target) aufgewogen. Im Bild 5 ist die Struktur des Gesamtsystemmodells dargestellt.

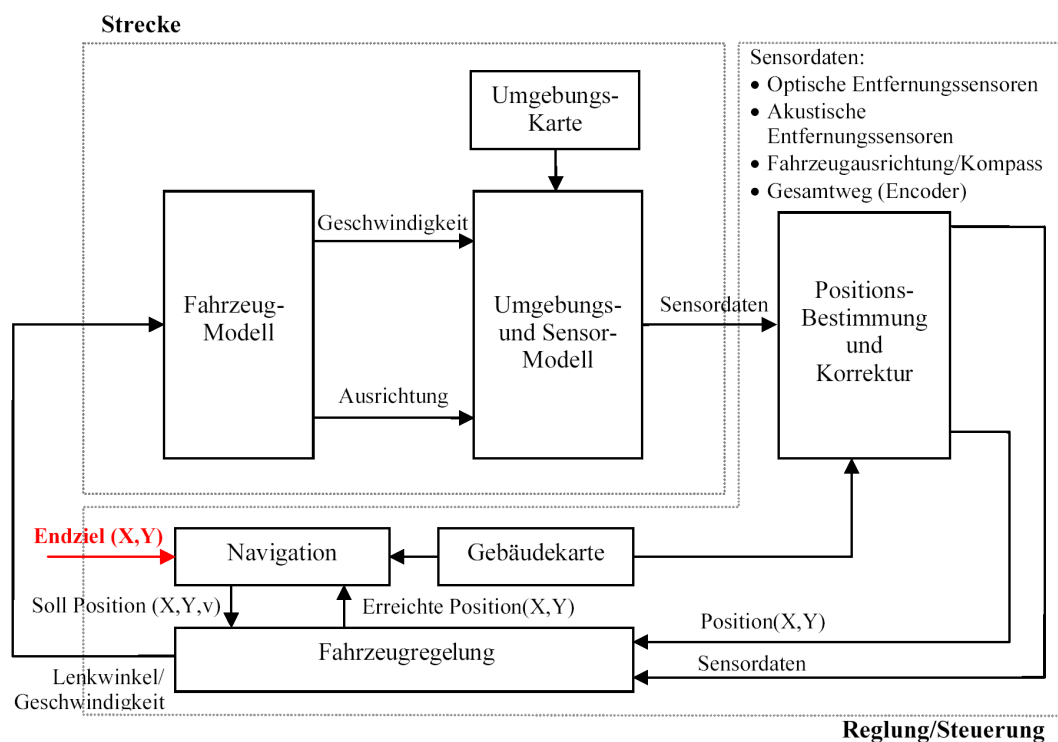


Bild 5: Gesamtsystemmodell

Umgebungsmodell

Neben dem in Abschnitt 3.2 beschriebenen Fahrzeugmodell wird ein Modell der Umgebung benötigt. Ziel der Modellierung war es, eine einfache Repräsentation der Begrenzungen des Fahrgebietes sowie evtl. Hindernisse zu finden, welche aber gleichzeitig noch die Grundlage für eine ausreichend genaue Simulation liefert. Ein Ansatzpunkt für die Vereinfachung war der eingeschränkte Wahrnehmungsbereich der Sensoren. Die Sensoren des autonomen Fahrzeugs messen die Entfernung der Objekte, auf die sie gerichtet sind. Hierbei wird nicht zwischen den Objekten unterschieden, also keine zusätzliche Information über Eigenschaften der erfassten Gegenstände ermittelt. Die Sensoren sind an-

nähernd auf der gleichen Höhe am Fahrzeug angebracht und tasten die Umgebung in horizontaler Richtung ab. Es entsteht somit ein zweidimensionales Abbild der Umgebung. Aus diesem Grund wurden im Simulationsmodell der räumlichen Umgebung des Fahrzeugs nur die Grundrisse der Objekte, wie Gegenstände und Raumbegrenzungen, abgebildet. Für die Beschreibung wurde die Darstellung als Punktmatrix favorisiert. Somit kann die Beschreibung der Umgebung durch die Erstellung einer einfachen Pixelgraphik erfolgen, die direkt in die Simulationsumgebung eingebunden werden kann.

Sensormodell

Für die Sensoren wurde ein einheitliches Sensormodell verwendet, welches unter Vorgabe der Position des Fahrzeugs, der Ausrichtung und Position des Sensors auf dem Fahrzeug sowie des Öffnungswinkels und der Reichweite die Entfernung des innerhalb des Sichtbereichs nächstgelegenen Punktes bestimmt.

3.4 Entwicklung von Algorithmen zur Navigation, Zielfindung und Kollisionsvermeidung

Positionsermittlung und -korrektur

Wegen der eingeschränkten Nutzbarkeit in Gebäuden wurde für die Positionserkennung kein GPS (Global Positioning System), sondern ein Algorithmus verwendet, der aus Fahrzeuggeschwindigkeit und Ausrichtung des Fahrzeuges näherungsweise die Position des Fahrzeugs ermittelt. Da die Genauigkeit dieser Berechnung stark von der benutzten Abtastrate abhängt und der Fehler kumulativ mit der Zeit steigt, wird eine Karte der Umgebung zur Positionskorrektur verwendet. Mittels der gemessenen Sensordaten wird eine interne Karte generiert, welche in einem bestimmten, vordefinierten Intervall mit der originalen Umgebungskarte korreliert und somit korrigiert wird.

Navigation und Fahrzeugregelung

Zur Steuerung des autonomen Fahrzeugs wird eine Komponente zur Berechnung der Route zum Ziel (Navigation) sowie eine zweite Komponente zur Regelung der Fahrzeugbewegung auf diesem Sollweg verwendet. Um die Weginformationen der Route nutzen zu können, wird diese in Zwischenpositionen zerlegt und in Abhängigkeit von der aktuellen Position an die Positionsregelung weitergegeben. Die Regelung des Fahrzeugs selbst besteht aus mehreren kombinierten Regelkreisen für Fahrzeugausrichtung, -position und geschwindigkeit.

Für die Planung der Route (Zielfindung) vor Fahrtantritt kommt der A*-Algorithmus [10] zum Einsatz, der auch häufig zur rechnergestützten Planung von Routen im Straßenverkehr verwendet wird. Der Algorithmus liefert den kürzesten Pfad zwischen einem Start- und einem Endpunkt. Die Berechnung erfolgt dabei auf Basis einer Rasterkarte der Fahrzeugumgebung, wobei der Algorithmus auf der Suche nach dem optimalen Weg Rasterpunkte der Karte nach einer Kostenfunktion bewertet und die Suche in Richtung der Punkte mit minimaler Wertung fortsetzt. Als Ergebnis der Streckenplanung entsteht eine Bahnkurve, die das Fahrzeug zur Zielposition führt. Zur Verwendung als Bahnkurve für die Fahrzeugpositionierung ist eine nachgelagerte Datenreduktion notwendig, die die Bahnkurve durch eine geringere Anzahl von Stützstellen approximiert. Während der Fahrt wird die Position des Fahrzeuges überwacht und der Positionsregelung jeweils der am nächsten liegenden Punkt der in Streckenabschnitte geteilten Bahnkurve übergeben. Die Positionsregelung berechnet einen Steuerkurs, der das Fahrzeug zur vorgegebenen

Zwischenposition führt. Die Regelung der Fahrzeugausrichtung schließlich errechnet aus der Geschwindigkeit sowie aus Soll- und Ist-Ausrichtung einen geeigneten Lenkwinkel, um das Fahrzeug möglichst zügig und ohne Überschwingen in die gewünschte Ausrichtung zu drehen. Für die Geschwindigkeitsregelung kommt ein PI-Regler zum Einsatz.

Ein Nachteil der kartenbasierten Navigation ist, dass die zur Positionierung des Fahrzeugs verwendete Karte immer auf einem aktuellen Stand sein muss. Hindernisse, die sich auf einer geplanten Bahnkurve befinden, führen zu Kollisionen. Dieser Umstand, der gerade bei der Interaktion mehrerer autonomer Fahrzeuge entscheidend ist, erfordert algorithmische Erweiterungen. Das Fahrzeug wurde daher mit einer Komponente zur reaktiven Kollisionsvermeidung ausgestattet. Diese berechnet aus den Sensordaten den Abstand des Fahrzeugs zu erkannten Hindernissen. Unterschreitet der Abstand einen bestimmten Wert, werden die Navigation und Positionsregelung deaktiviert, die Geschwindigkeit reduziert und das Fahrzeug um das Hindernis geführt. Dies wird erreicht, indem laufend ein Steuerkurs berechnet wird, der das Fahrzeug auf einem konstanten Abstand zu dem Hindernis hält. Wird die Bewegungsfreiheit in beiden Richtungen eingeschränkt, befindet sich also auf beiden Seiten ein Hindernis, wird ein Kurs berechnet, der das Fahrzeug möglichst mittig zwischen diesen Hindernissen hindurchführt.

Problematisch bei dieser Vorgehensweise ist, dass die Korrektur des Steuerkurses dazu führen kann, dass eine von der Navigation vorgegebene Zwischenposition überhaupt nicht erreicht wird. Dies trifft vor allem dann zu, wenn der geplante Weg, z.B. durch eine geschlossene Tür, vollständig blockiert ist. Hierfür ist ein Beobachter vorgesehen, der das Ausweichmanöver bewertet und es an geeigneter Stelle abbricht, um eine vollständig neue Berechnung der Route zu veranlassen.

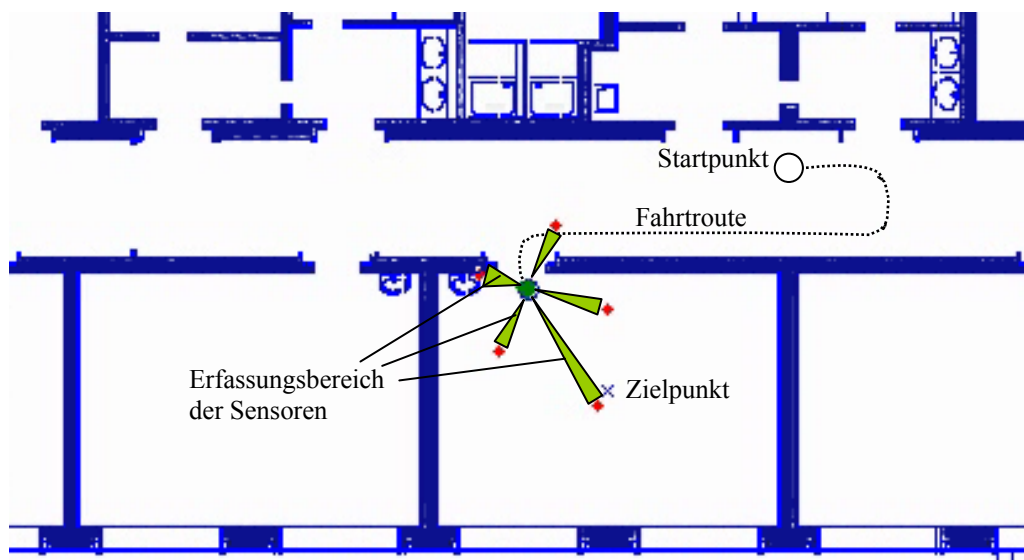


Bild 6: Visualisierung: Versuchsfahrzeug im Gebäude

4 Ergebnisse der Gesamtsystemsimulation und experimentelle Erprobung

Bild 6 zeigt eine Momentaufnahme aus der Visualisierung von Simulationen, die für den Entwurf der Navigationsalgorithmen verwendet wurde. Dazu wurde das in Bild 5 gezeigte Gesamtsystemmodell verwendet. Durch umfangreiche Simulationsexperimente konnten die in Abschnitt 3 beschriebenen Algorithmen entwickelt, erprobt und optimiert werden. Nachdem in den Simulationen für die verwendeten, aus realen Bauunterlagen entnommenen Gebäudegrundrisse keine Fehler mehr auftraten, wurde mit der Umsetzung der Algorithmen auf das reelle Fahrzeug begonnen.

Nach Analyse von Rechenzeit- und Speicherbedarf der Algorithmen wurde die Aufteilung auf die vorhandenen Zielprozessoren vorgenommen. Der Microcontroller auf der Sensorplattform übernahm die Erfassung und Vorverarbeitung der Sensorwerte, der Controller auf der Steuerungsplattform die Regelungen für Geschwindigkeit und Ausrichtung. Für beide Controller wurden die Algorithmen aus SIMULINK mittels „Real Time Workshop“ in C-Code umgesetzt.

Nach dieser Aufteilung der Regelungsalgorithmen wurden die Simulationsmodelle des Fahrzeuges und der Umgebung durch das reale Fahrzeug inklusive der Sensorplattform ersetzt. Navigation und Positionsreglung verblieben zunächst weiterhin als Simulationsmodell in SIMULINK, auf dem Hostrechner. Die Verbindung zum Fahrzeug erfolgt per Funk mittels einer bei Fraunhofer IIS/EAS Dresden entwickelten Schnittstelle [9] zur Anknüpfung realer Hardware an Simulationswerkzeuge.

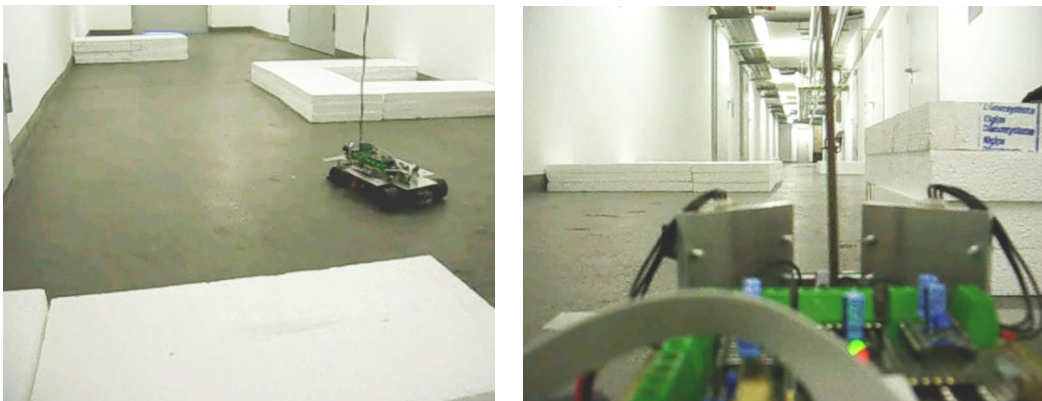


Bild 7: Experimentelle Validierung: links – Fahrzeug zwischen Hindernissen, rechts – Blick in Fahrtrichtung auf Hindernisse

Bild 7 zeigt das Fahrzeug in einer Umgebung zur praktischen Erprobung der Kollisionsvermeidungsalgorithmen.

5 Zusammenfassung und Ausblick

Anhand eines Beispiels aus dem Bereich autonomer intelligenter Systeme wurde der Einsatz von Modellierungsansätzen für mechatronische Systeme demonstriert. Dabei er-

wiesen sich diese Ansätze als prinzipiell praktikabel, wenngleich hinsichtlich der zu bewältigenden Datenmengen, z.B. für die Umgebungsabbildung, Vereinfachungen und Datenreduktion notwendig sind. Diese gilt für die Modellierung ebenso wie für die Implementierung von Algorithmen auf eingebetteten Zielprozessoren. Durch die Simulationsexperimente konnte der Entwurf von Regelungsalgorithmen sowie von Navigations- und Steuerungsstrategien wirksam unterstützt werden. Insbesondere die Analyse der algorithmischen Struktur mit dem Ziel einer optimalen Partitionierung war dabei eine wichtige Aufgabe. Durch die umfangreichen Simulationen verschiedener Betriebs- und Fehlerzustände des Systems konnten Nachbesserungen an den Algorithmen in der praktischen Erprobung weitgehend vermieden werden.

Das Hauptproblem im praktischen Einsatz und somit eine Aufgabe bei der Weiterentwicklung des Systems ist die Erhöhung der Zuverlässigkeit der relativ preiswerten Sensorik. Da Messfehler der Einzelsensoren oft zu falschen Reaktionen des Fahrzeuges, werden derzeit Konzepte zur redundanten, fehlertoleranten Sensorausrüstung des Fahrzeuges entwickelt und im Simulationsexperiment erprobt.

Literatur

- [1] Clauß, C.; Haase, J.; Kurth, G.; Schwarz, P.: Extended Admittance Description of Nonlinear n-Poles. Archiv für Elektronik u. Übertragungstechnik 40(1995)2, S. 91-97.
- [2] Kasper, R.; Koch, W.: Object-Oriented Behavioural Modelling of Mechatronic Systems. 3rd Conference on Mechatronics and Robotics '95, Paderborn, Oktober 1995.
- [3] Karnopp, D.; Margolis, D.; Rosenberg, R.: System Dynamics: Modeling and Simulation of Mechatronic Systems. Wiley, Chichester, 2000
- [4] Nিকেleit, I.: Verifikation mechatronischer Systeme mit μ Controller-Steuerung. Automotive 05-06/2004, Hanser
- [5] Pelz, G.: Mechatronic systems: modelling and simulation with HDLs. Wiley, Chichester, 2003
- [6] Schneider, P.; Schwarz, P.: Systemsimulation unter Einbeziehung Neuronaler Netze. ITG-Fachtagung "Mikroelektronik für die Informationstechnik", 21.-23. März 1994, Berlin, ITG-Fachbericht 127, S. 135-140
- [7] Schneider, P.; Huck, E.; Schwarz, P.: A Modeling Approach for Mechatronic Systems - Modeling and Simulation of an Elevator System. XI. Intern. Symposium in Theoretical Electrical Engineering, Linz, 19.-22. August 2001
- [8] Voigt, P.; Wachutka, G.: Electro-fluidic microsystem modeling based on Kirchhoffian network theory. Sensors and Actuators A 66 (1998) 1-3, S. 6-14.
- [9] Wielens, S.; Altmann, S.; Haufe, J.; Schneider, P.: Integration of Prototypes into the Design Flow of Digital Hardware for Applications in Mechatronics and Telecommunication. Model-Based Design Conference 2005, München, 8. Juni 2005
- [10] Hart, P. E.; Nilsson, N. J.; Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics SSC4 (2), 1968, S. 100-107.