

Schnelle Reglerauslegung in der Hydraulik durch Einsatz einer ganzheitlichen Entwicklungsumgebung

Rapid Control Design by Using a Total Development Environment

Thorsten Lang

Institut für Landmaschinen und Fluidtechnik, Technische Universität Braunschweig

Kurzfassung: Wie in anderen Bereichen ist auch in der Mobilhydraulik eine starke Zunahme von immer komplexeren, digital geregelten Systemen festzustellen. Die Ansprüche an die Regelgüte und integrierte Funktionsvielfalt entwickeln sich genauso rasch wie die Forderungen nach schneller Umsetzung von innovativen Ideen in Produkte. Um sich für diese Anforderungen zu rüsten, lohnt sich die Betrachtung und Diskussion moderner ganzheitlicher Entwicklungsumgebungen und numerischer Methoden. Neben der Vorstellung einer kommerziellen Umgebung wird in dem vorliegenden Aufsatz am Beispiel eines hydraulischen Oberlenkers gezeigt, dass der Einsatz einer numerischen Identifikation dem Regelungstechniker durchaus eine praktische Arbeitserleichterung bieten kann.

Deskriptoren: Schnelle Reglerauslegung, Identifikation, Automatische Codegenerierung

Abstract: Like other sectors, mobile hydraulic engineering is affected by a rapid rise in the number of ever more complex digitally controlled systems. The demands made on control quality and the variety of integrated functions are increasing just as quickly as the demands for swift translation of innovative ideas to the practical level. In order to prepare to meet these requirements, it is advisable to consider and discuss modern Total Development Environments (TDE) and numerical methods. Besides presenting a commercial environment, this paper illustrates by the example of a hydraulic upper link that the use of a numerical identification can certainly facilitate work practically for the control engineer.

Keywords: Rapid Control Prototyping, Hardware-In-The-Loop, system identification, control development

1 Einleitung

Betrachtet man den derzeitigen allgemeinen Entwicklungstrend bezüglich des Einsatzes von digital-elektronischen Steuerungen, so können unter anderem folgende Aussagen getroffen werden:

- 1) Nach Angaben des Verbandes Deutscher Maschinen- und Anlagenbau e.V. (VDMA) haben sich die Investitionskosten im Anlagenbau, wie in **Bild 1** dargestellt, in den letzten 30 Jahren in Mechanik von rund 90% auf 40% verringert, während sich die Investitionen in Elektronik von 7% auf 20% und in Software sogar von 4% auf rund 40% steigerten. Somit erlangen die beiden letztgenannten Bereiche mit einer Summe von rund 60% Investitionsvolumen eine erhebliche innerbetriebliche Bedeutung und können unter Umständen sogar über die Konkurrenzfähigkeit eines Produktsegmentes entscheiden.

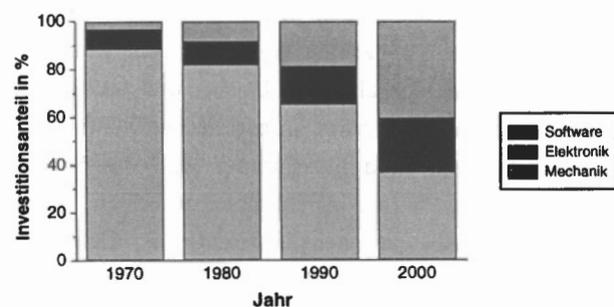
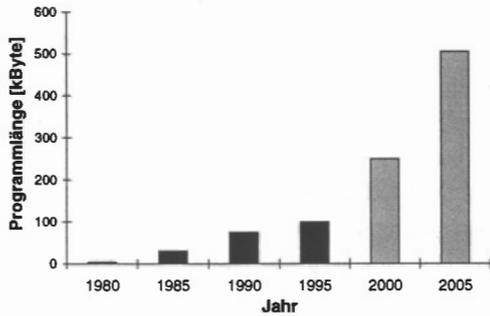


Bild 1: Investitionsanteile im Anlagenbau nach [1]
Fig. 1: Investment shares in plant construction according to [1]

- 2) Die Funktionsdichte in Steuerungen steigt stetig, gemeint ist die Anzahl der Funktionen pro Steuereinheit, zuzüglich eines immer größeren Anteils von Diagnosefunktionen. Damit einhergehend wächst die spezifische Programmlänge (**Bild 2**) und auch der Projektierungsaufwand. Die Zahlen können natürlich

je nach Anwendung und Branche variieren, zeigen aber einen klaren Trend. In der Automobilindustrie spricht man bereits vom Erreichen einer kritischen Grenze in der Entwicklung von Funktionalität, da immer mehr Ingenieure immer mehr Funktionen in immer kürzerer Zeit produzieren. Aus diesem Grund wächst der Bedarf an Werkzeugen zur Erleichterung des Software-Engineering.



Quelle: Automobiltechnische Zeitschrift ATZ 12/97
Hanselmann, Wohnhaas

Bild 2: Beispiel: Entwicklung der Programmlänge einer Motorsteuerung nach [11]
Fig. 2: Example: Development of the program code size for engine management acc. to [11]

- 3) Die technischen Anforderungen an Steuerungen bezüglich Dynamik und Güte wachsen stetig. Zusätzlich wird eine schnelle und flexible Reaktion auf veränderte Randbedingungen gefordert (z.B. auf den Wechsel eines Ventiltyps oder eines Zylinders). Ein hierdurch ausgelöster Neustart aufwendiger, manueller Entwicklungsschritte mit hohem personellem Aufwand wird selten bezahlt und sollte vermieden werden.
- 4) Es ist eine stärkere Vernetzung von Steuerungen festzustellen, worauf im Entwicklungsstadium bereits früh Rücksicht genommen werden sollte. Deswegen macht es unter Umständen immer weniger Sinn, isolierte Einzelentwicklungen zu betreiben, mit dem Risiko einer anschließenden Fehlfunktion im Gesamtnetz.

Um den genannten Anforderungen gewachsen zu sein, ist der Einsatz einer ganzheitlichen Umgebung zur Entwicklung von Regelungs- und Steuerungssoftware überlegenswert. Mit einem solchen univer-

sellen Werkzeug läßt sich der gesamte Entwicklungsprozeß mit geringem personellen Aufwand und entsprechend geringeren Reibungsverlusten im Informationsfluß durchführen - von der mathematischen Erstanalyse bis zum Festlegen logischer Steuerstrukturen und dynamischer Parameter. Nach Vorstellung einer solchen Umgebung im Vergleich zu einer konventionellen Projektierung soll anhand einer realen hydraulischen Regelstrecke ein praktisches Beispiel aufgezeigt werden.

2 Die konventionelle Projektierung

Betrachtet man eine konventionelle Projektierung einer Steuerung von der ersten Projektidee bis hin zu einer praktisch einsetzbaren Endversion, so kann dieser Vorgang beispielsweise nach Bild 3 strukturiert werden. Nach Auswahl einer geeigneten Hardware muß diese unabhängig von der später verwendeten Programmier-Software zusammengestellt und geprüft werden (linker Zweig). Parallel dazu wird eine mathematische Modell-Beschreibung der Regelstrecke zur Simulation des zu regelnden Gesamtsystems gesucht (rechter Zweig). Dieses kann sowohl mit analytischen als auch mit numerischen Methoden durchgeführt werden. Ziel ist eine geschlossene Simulation der entwickelten Programmstrukturen im Zusammenspiel mit dem (modellierten) System.

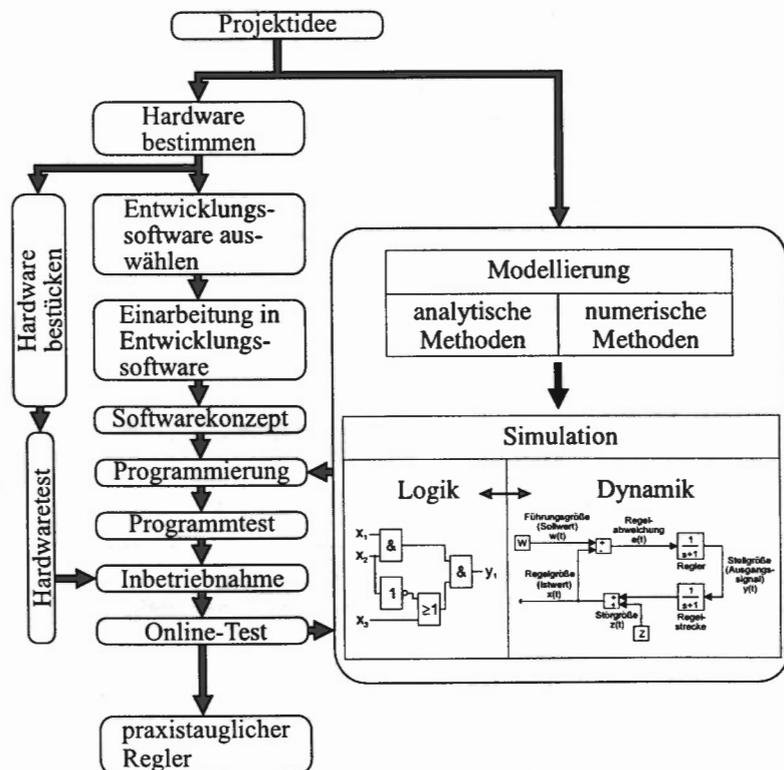


Bild 3: Klassisches Vorgehen bei der Projektierung einer Steuerung
Fig. 3: Classic method of control design

Meist besteht sowohl die Simulation als auch das Steuerungsprogramm aus zwei Teilen. Einem logischen Teil, der die Funktionalität und Sicherheitsroutinen bzw. Diagnosemöglichkeiten beinhaltet und einem Teil, der die dynamische Regelung der Regelstrecke übernimmt. Die Erkenntnisse aus der Simulation sind mit den eigentlichen Entwicklungsschritten (mittlerer Zweig) rekursiv zwischen Programmierung und Online-Test eingebunden. Der gesamte Prozeß fordert in der Regel einen hohen Personal-, Fachkenntnis- und Zeiteinsatz. Ändern sich Randbedingungen, muß unter Umständen der gesamte iterative Prozeß neu angestoßen werden. Das dargestellte Schema kann selbstverständlich in vielfältigen Variationen auftreten, ist aber geeignet, grundsätzlich den Entwicklungsprozeß zu beschreiben.

Um den Ablauf zu beschleunigen, können heute durch moderne ganzheitliche Entwicklungsumgebungen einige der gezeigten Schritte während der Entwicklungsphase zusammengefaßt werden bzw. vollständig entfallen.

3 Die ganzheitliche Entwicklungsumgebung

Heute erhältliche, handelsübliche, ganzheitliche Entwicklungsumgebungen (Total Development Environment, TDE) bestehen aus einem CASE-Tool (Computer Aided Software Engineering) in Verbindung mit einer flexiblen Hardware (Rapid Prototype, RP). Unter einem CASE-Tool versteht man eine Software zur meist grafischen Programmierung und der automatischen Erzeugung von Maschinencode ("Autocode"), meist mit einer Zwischenstufe in einer Hochsprache (z.B. "C"). Der Vorteil der automatischen Codegenerierung liegt in dem Ausschluß zeitraubender Syntax- bzw. Semantikfehler während der Entwicklungsphase gegenüber einer manuellen Programmierung. Die grafische Eingabe ist in der Regel leicht erlernbar und verkürzt die Einarbeitungszeiten erheblich. Als Produktbeispiele können unter anderem Statemate[®], MatrixX[®] oder Matlab/Simulink[®] genannt werden. Es werden neben Werkzeugen zur linearen und nichtlinearen Beschreibung dynamischer Systeme nach analytischen Methoden auch umfassende mathematische und numerische Funktionen zur Verfügung gestellt, so daß eine große

Bandbreite moderner Methoden zur Behandlung regelungstechnischer Aufgabenstellungen und zur Meßdatenverarbeitung in einem universellen System zusammengeführt sind.

Der Rapid Prototype seinerseits führt den erzeugten Autocode in Echtzeit aus und verbindet somit den gesamten Funktionsumfang der Software direkt mit einer realen Anlage.

3.1 Das CASE-Tool Matlab/Simulink[®]

Als Produktbeispiel wurde das Programmpaket Matlab/Simulink[®] ausgewählt, weil es zumindest an Hochschulen einen überdurchschnittlichen Verbreitungsgrad besitzt. Basis ist ein mathematischer Kern (Matlab), der mit rund 50 Modulen (sog. Toolboxen) für die jeweiligen Anwendungen erweitert werden kann. Für die Dimensionierung von Reglern stehen die klassischen Schulmethoden der linearen Regleranalyse und -synthese und umfangreiche numerische Verfahren zur Auswahl, beispielsweise die Auslegung von Fuzzy-Reglern oder die Optimierung einer Schar von Ungleichungen zur Beschreibung des geschlossenen Regelkreises (Gütevektoroptimierung). Bild 4 zeigt beispielhaft neben der Beschreibung eines dynamischen Systems durch Blockschaltbilder (Simulink), die Beschreibung von komplexen logischen Zustandsflüssen (Stateflow) und die Auslegung eines Reglers im geschlossenen Kreis anhand der Nonlinear-Control-Design-Toolbox durch numerische Parametervariation und Vorgabe eines gewünschten Antwortverhaltens im Zeitbereich. Im Grunde entspricht diese zuletzt genannte Methode dem Vorgehen eines Praktikers, der solange alle Reglerparameter verstellt, bis ein befriedigendes Antwortverhalten erreicht wird.

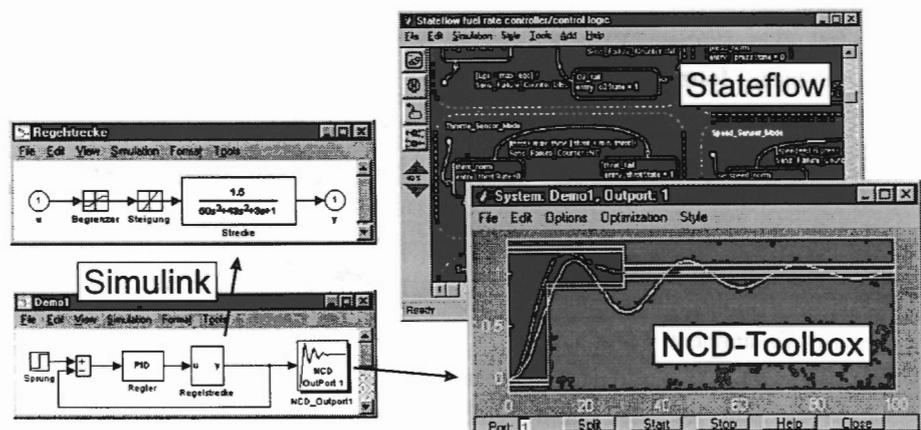


Bild 4: Das CASE-Tool Matlab/Simulink[®]
Fig. 4: The CASE tool Matlab/Simulink[®]

3.2 Der Rapid Prototype

Per Knopfdruck wird aus Matlab/Simulink[®] automatisch ein Echtzeit-C-Code erstellt, welcher durch einen speziellen Compiler sofort auf einen digitalen Signalprozessor (DSP) mit Fließkomma-basierter Arithmetik geladen wird. Da ein modularer und nicht optimierter Code vorliegt, muß die Hardware sehr viel leistungsfähiger als eine endgültige Serien-Zielplattform (meist Festkomma-basierte Mikrocontroller) sein. Ferner müssen ausreichend viele analoge und digitale Schnittstellen zur Verfügung stehen, um eine universelle, schnelle Anpassung an beliebige Randbedingungen zu gewährleisten. Die verwendete kommerzielle Hardware der Firma dSpace ist hauptsächlich auf Autocode von Matlab zugeschnitten. Hohe A/D-Auflösungen bis zu 16 Bit und schnelle Prozessoren, mit denen auch umfangreiche Algorithmen mit Abstraten von bis zu einigen kHz ausgeführt werden können, ermöglichen sehr genaue Untersuchungen der Regelstrecken. Über einen parallel und unabhängig arbeitenden PC ist die interaktive Steuerung, Analyse und Kontrolle sämtlicher Programmparameter während des Betriebes möglich. Bild 5 zeigt ein Schema der modular aufgebauten DSP-Hardware und ein Beispiel der zur Verfügung stehenden interaktiven und frei gestaltbaren Bedienersoftware.

3.3 Hardware-In-The-Loop

Eine weitere Funktion innerhalb des in Bild 6 dargestellten Einsatzbeispiels einer TDE ist der Hardware-In-The-Loop-Test von Steuergeräten. Wenn die endgültige Reglerstruktur durch die Verbindung der TDE mit der

realen Anlage gefunden ist (Rapid Control Prototyping, RCP) wird diese üblicherweise auf eine preiswerte Zielplattform eines Steuergerätes implementiert, mit anschließendem genauen Test der Hardware. Beim Hardware-In-The-Loop-Test wird nicht die Versuchsanlage, sondern das serienreife Steuergerät mit dem Rapid Prototype verbunden und mit der (aus der Entwicklungsphase bekannten) Simulation der Regelstrecke belastet. So lassen sich auch Tests in physikalischen Grenzbereichen und für Mensch oder Maschine gefährlichen Betriebszuständen durchführen. Für die Bearbeitung einer FMEA (Fehler-Möglichkeiten-und-Einfluß-Analyse) kann dieses von großer Hilfe sein.

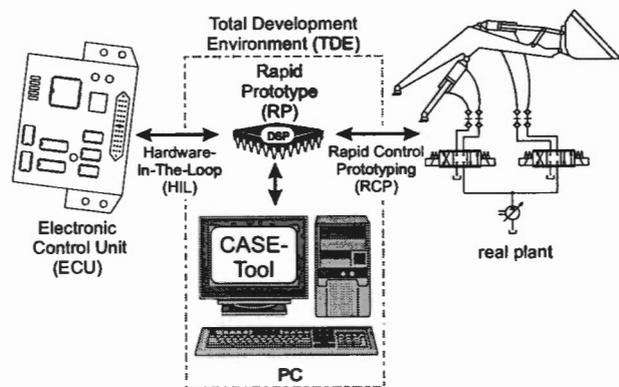
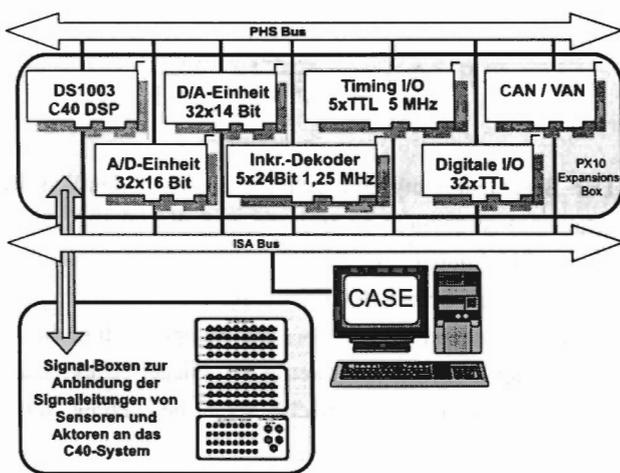
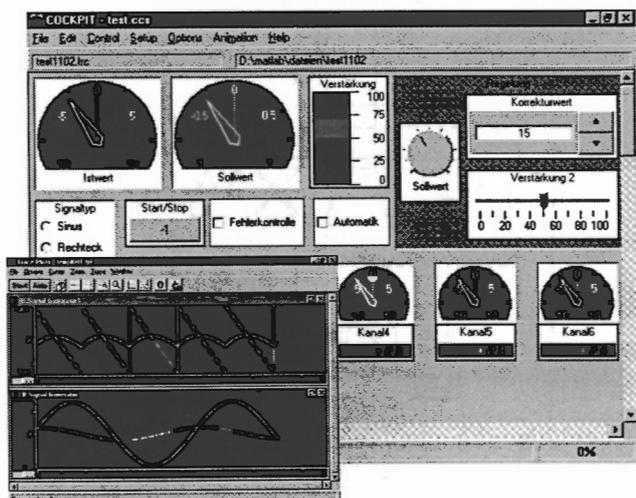


Bild 6: Beispiele von Einsatzmöglichkeiten mit einer ganzheitlichen Entwicklungsumgebung
 Fig. 6: Example of applications with a Total Development Environment



a) Hardwarestruktur



b) Bedienersoftware

Bild 5: Der Rapid Prototype
 Fig. 5: The Rapid Prototype

4 Beispiel einer Modellierung durch numerische Identifikation

Die klassische Methode der Reglerdimensionierung umfaßt bekanntlich im ersten Schritt die analytische Modellierung der Regelstrecke, wobei grundlegende physikalische Kenntnisse schrittweise zu einer linearisierten mathematischen Gesamtbeschreibung der Regelstrecke führen. In einem zweiten Schritt werden nach geeigneten Stabilitätskriterien mathematisch Reglerübertragungsfunktionen berechnet, die zu dem gewünschten Verhalten des geschlossenen Regelkreises führen sollen. Bei vielen realen Anlagen (insbesondere bei hydraulischen Systemen) sind allerdings durch viele Unstetigkeiten und Nichtlinearitäten (Anschläge, Abhängigkeiten höherer Ordnung usw.) linearisierte Beschreibungen schwer zu finden. Oft sind aber auch nur wenige oder gar keine Eigenschaften einzelner Komponenten bekannt, so daß ein analytisches Vorgehen unmöglich ist. Mit der oben beschriebenen Kopplung von Soft- und Hardware ist der Einsatz von Algorithmen zur linearen numerischen Identifikation von Regelstrecken praxisnah durchführbar. Hierzu wird die reale Anlage mit einem beliebigen Signal angeregt und die Reaktion aufgezeichnet. Aus den Datensätzen der Anregung und der zugehörigen Antwort läßt sich die Übertragungsfunktion numerisch bestimmen. Dies soll anhand des folgenden Versuchsaufbaus nach **Bild 7** gezeigt werden.

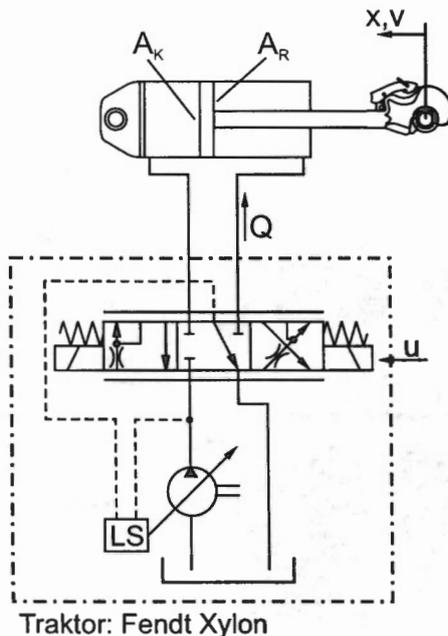


Bild 7: Beispiel: Eine hydraulische Anlage mit Pumpe, Wegeventil und hydraulischem Oberlenker

Fig. 7: Example: A hydraulic system with pump, valve and hydraulic upper link

An der serienmäßigen Load-Sensing-Hydraulik eines Traktors vom Typ Fendt Xylon 520 wird ein hydraulischer Oberlenker betrieben, der mit einer analogen Einrichtung zur Messung der Position x versehen ist. Zusätzlich wird der fließende Volumenstrom Q mit einem Inkrementalzähler gemessen. Da über das Hydrauliksystem des Traktors keine detaillierten Angaben vorliegen, soll mit Hilfe der Identifikation das dynamische Übertragungsverhalten der Oberlenkergeschwindigkeit v und der Position x festgestellt werden. Hierzu wird das Wegeventil mit einem Signalsprung von 30% beaufschlagt. Aus einer ersten Messung war zu erkennen, daß das Wegeventil deutlich langsamer öffnet als schließt, daher wird zunächst nur der Öffnungsvorgang betrachtet. Aus den beiden Signalen Q und x wird jeweils über die bekannten Zusammenhänge

$$v = \frac{dx}{dt} \quad (1)$$

und

$$v = \frac{Q}{A_R} \quad (2)$$

die Geschwindigkeit v berechnet und die Signale gefiltert (**Bild 8**). Deutlich zu erkennen ist die unterschiedliche Signalqualität aus der analogen Messung von x (Kurve c) und der digitalen Messung des Volumenstromes Q (Kurve b). Die Zeitverläufe für die Geschwindigkeit lassen ein PT_1 - oder PT_2 -Verhalten vermuten. Während der Durchführung der Identifikation zeigte sich, daß eine PT_1 -Funktion das Verhalten nur unzureichend wiedergeben kann. Die Übertragungsfunktionen der Zylindergeschwindigkeit wird daher angenommen mit:

$$F_v(s) = \frac{Z_v}{N_{v,2} s^2 + N_{v,1} s + 1} \quad (3)$$

Der Beschleunigungseinbruch (in Kurve c deutlich zu erkennen) ist dadurch zu erklären, daß die relativ langen Hydraulikleitungen ein Druckreservoir bieten, welches zunächst nach Öffnen des Ventils für die Beschleunigung sorgt, nach dessen Abbau aber die Pumpe noch nicht im geforderten Maß aufgeschwenkt ist. Dieser Effekt kann natürlich durch eine einfache PT_2 -Übertragung nicht wiedergegeben werden.

Für die Position x würde nach (3) eine Dynamik nach einem IT_2 -Verhalten vorliegen.

Da aber eine der beiden Zeitkonstanten klein zu sein scheint, wird zunächst ein vereinfachter IT_1 -Ansatz versucht, was auch durch den optischen Eindruck naheliegt:

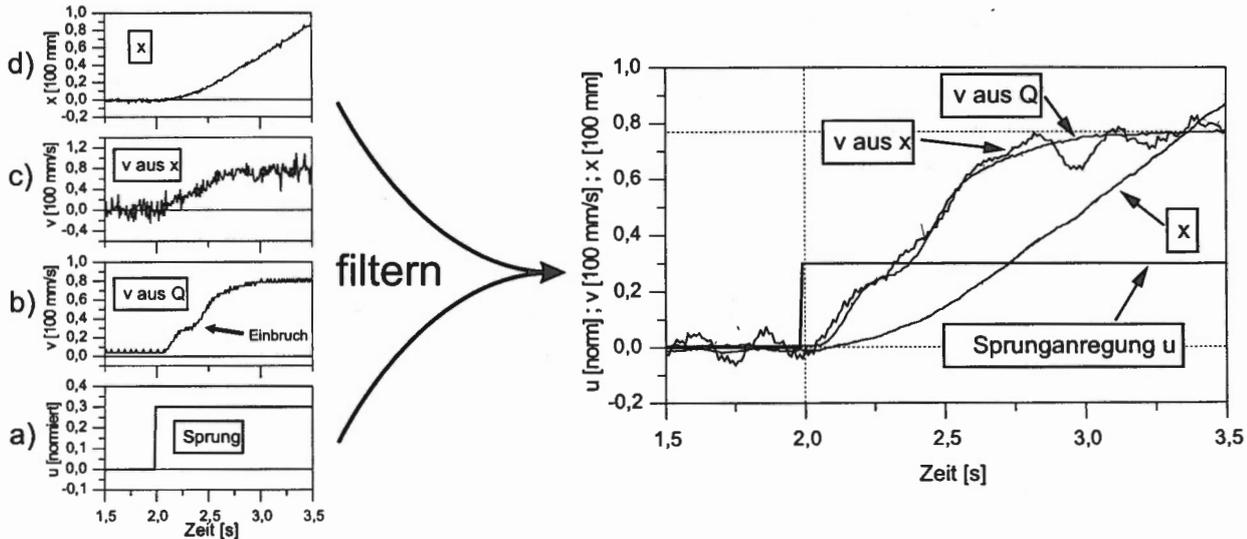


Bild 8: Gemessene und gefilterte Signale der Geschwindigkeit und Position
 Fig 8: Measured and filtered velocity and position signals

$$F_x(s) = \frac{Z_x}{(N_x s + 1) \cdot s} = \frac{Z_x}{N_x s^2 + s} \quad (4)$$

Nach erfolgreicher Identifikation der Streckenparameter wurden in einer Simulation alle identifizierten Übertragungsfunktionen mit einem Sprung angeregt und sowohl die Geschwindigkeiten als auch die Positionen (durch nachgeschaltete Integration der Geschwindigkeiten) verglichen (Bild 9).

Es zeigt sich, daß sowohl die beiden Sprungantworten der Geschwindigkeit an den Stellen 2 und 3 als auch die drei Sprungantworten der Position (Stellen 4, 5 und 6) jeweils gut übereinstimmen und somit die getroffenen

Approximationen bestätigt werden. Die Kurve 6 ist etwas "schneller" als die Kurven 4 und 5, was sich durch die oben beschriebene Vereinfachung von IT₂ auf IT₁ erklären läßt. Mit einer IT₂-Identifikation war allerdings kein befriedigendes Ergebnis zu erzielen, so daß diese Ungenauigkeit in Kauf genommen werden muß. Da als Quellen unterschiedliche physikalische Signale verwendet wurden, kann davon ausgegangen werden, daß für diesen Betriebspunkt tatsächlich das linearisierte dynamische Verhalten ausreichend gut beschrieben wird.

Nach selbem Muster wurde das Übertragungsverhalten für den Schließvorgang des Ventils aus der Positionsmessung bestimmt. Zur Qualitätsbewertung wurden anschließend über den Rapid Prototype gleichzeitig in Echtzeit die reale Strecke und die identifizierten Über-

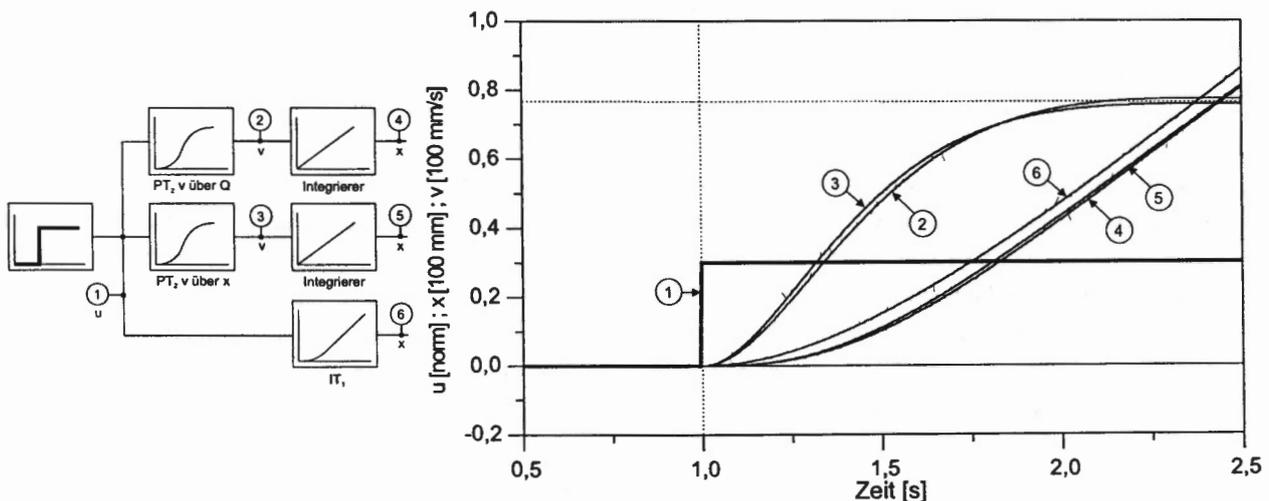


Bild 9: Simulierte Sprungantworten für Geschwindigkeit und Position
 Fig. 9: Simulated step responses of velocity and position

tragungsfunktionen angeregt, wobei in der Simulation eine logische Schaltung die Selektion der jeweils korrekten Funktion übernahm. Bild 10 zeigt die Verläufe der Messung an der realen Anlage (dünne Kurve) und die simulierte (dicke Kurve) Geschwindigkeit des Zylinders.

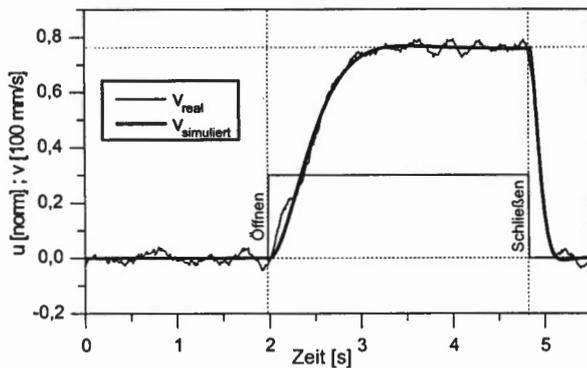


Bild 10: Reale und simulierte Oberlenkergeschwindigkeit für Öffnen und Schließen des Ventils

Fig. 10: Real and simulated velocity of the upper link for opening and closing of the valve

Auch wenn die Ergebnisse in Ihrer Qualität bestehen, sind bei dieser Vorgehensweise unbedingt folgende Einschränkungen zu beachten:

- 1) Die Ordnungen der gesuchten Polynome müssen für die Durchführung einer linearen Identifikation vorab qualitativ geeignet abgeschätzt werden, da nur Koeffizienten geliefert werden. Die vereinfachende Abschätzung von Gesamtübertragungsfunktionen niedriger Ordnungen, die ja eigentlich aus einer größeren Anzahl von Einzelfunktionen bestehen, ist nicht immer unkritisch. Wie oben gezeigt, kann dabei unter Umständen durchaus einer sinnvollen Vereinfachung gegenüber mathematischer Korrektheit der Vorzug gegeben werden.
- 2) Die gefundenen Funktionen hängen voll von den in den Meßsignalen enthaltenen Informationen ab. Da diese stochastischen Gesetzen unterliegenden, bilden sie nur eine Stichprobe. Die sorgfältige Erzeugung geeigneter Meßdaten ist somit außerordentlich wichtig. Durch geeignete Maßnahmen ist außerdem dafür zu sorgen, daß während der Messung alle wesentlichen Eigenschaften des Systems angeregt sind und damit repräsentative Daten vorliegen.
- 3) Die gefundene Funktion gilt nur in einem linearisierten Betriebspunkt, der in der Regel nicht das allgemeine Verhalten der Anlage repräsentiert.

- 4) Der Verlauf der Anregungsfunktion hat großen Einfluß auf die Gültigkeit der gefundenen Funktion im Zeit- und Frequenzbereich. Zu große Signaländerungen können unter Umständen durch Unstetigkeiten den Signalverlauf unbrauchbar machen (z.B. durch Anfahren mechanischer Anschläge) während bei zu kleinen Signaländerungen - in Verbindung mit zu großen Abtastraten - Singularitäten auftreten können, denen nicht jeder Rechenalgorithmus gewachsen ist. Wie in Bild 8b) durch den Beschleunigungseinbruch zu erkennen ist, zählt eine Sprungfunktion als Anregung nicht unbedingt zu den geeigneten Funktionen.
- 5) Wie oben gezeigt, ist ein Übertragungsverhalten oft auf bestimmte Betriebsbedingungen festgelegt (z. B. Aus- bzw. Einfahren eines Differentialzylinders, Öffnen bzw. Schließen eines Ventils). Hieraus folgt zum einen, daß das System in geeignete Teilsysteme zerlegt werden muß, die nicht durch mechanische Baugruppen (wie aus der Analytik gewohnt), sondern durch ihr Übertragungsverhalten bestimmt werden. Zum anderen wird deutlich, daß periodische Anregungsfunktionen (z.B. Sinusschwingungen) zur Identifikation ungeeignet sind, sobald zu befürchten ist, daß pro Zyklus unterschiedliche Betriebseigenschaften durchfahren werden können.

Der Entwicklungsstand leistungsfähiger Algorithmen in Verbindung mit einer TDE lassen heutzutage neben der gezeigten linearisierten Beschreibung auch Untersuchungen zur parametrischen oder nichtlinearen Identifikation bzw. zur Implementierung adaptiver Regler mit vertretbarem Zeitaufwand zu, so daß diese Verfahren in Zukunft durchaus dem praktikablen Einsatz näher gebracht werden können. Wie durch geeignete Maßnahmen aus von identifizierten Übertragungsfunktionen die allgemeine Beschreibung einer hydraulischen Anlage verbessert werden bzw. durch robuste Regler toleriert werden kann, ist Gegenstand der Untersuchungen am Institut für Landmaschinen und Fluidtechnik.

5 Wirtschaftlichkeit

Sowohl die beschriebene Hard- als auch Software entspricht dem neuesten Stand der Technik. Die Entwicklung dieser Werkzeuge basiert auf sehr hohem technischen Know-how und theoretischem Niveau, was sich in relativ hohen Preisen der Produkte widerspiegelt. Die Prüfung, ob sich die Investition in ein TDE-System lohnt, ist individuell und hängt von vielen Faktoren ab, wie Erfahrungen aus der Softwareindustrie zeigen. In [7] wird eine Abteilung von 10 Softwareentwicklern über

einen Zeitraum von 2 Jahren nach Einführung eines CASE-Tools analysiert und eine Kosten-Nutzen-Relation aufgestellt. Danach wird nach Einführung einer neuen Entwicklungsumgebung in den meisten Fällen zunächst ein Absinken der Produktivität zu beobachten sein, wie in Bild 11 dargestellt, welches sich dann aber in einen deutlichen Produktivitätsgewinn umkehrt. Entscheidend ist aber die Frage nach dem Eintritt in den „Return of Investment“, d.h. zu welchem Zeitpunkt t_x erstmals ein Produktivitätsgewinn zu verbuchen ist, um den Investitionsvorteil einzuleiten.

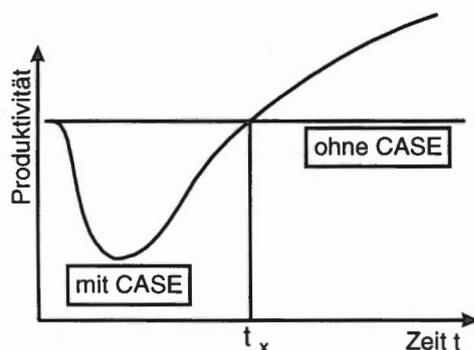


Bild 11: Produktivitätsverlauf nach Einführung eines CASE-Tools nach [7]

Fig. 11: Graph showing productivity after installation of a CASE tool acc. to [7]

Dieser Zeitpunkt, wie auch die Höhe der Produktivitätssteigerung, wird stark von dem Ausbildungsstand der Mitarbeiter und vom Anteil der mit CASE lösbaren Tagesgeschäfte beeinflusst. In [7] wird außerdem dargelegt, daß die Hauptinvestition nicht durch die direkten Anschaffungskosten verursacht wird, sondern durch Produktivitätsausfall während der Schulungs- und Umstellungsphase. Das gezeigte Beispiel ist aus dem Bereich der Informatik entliehen, läßt sich aber grundsätzlich übertragen. Bei kleineren und im Aufbau befindlichen Abteilungen hingegen kann die Bilanz aufgrund des geringeren Umstellungsaufwands deutlich verbessert werden. Hochschulen haben in der Regel neben meist günstigeren Preiskonditionen auch durch die junge Altersstruktur der Mitarbeiter deutliche Vorteile. Im eigenen Haus waren z. B. Studierende fast ohne Einarbeitung innerhalb weniger Arbeitsstunden in der Lage, Regelungen für Versuchsanlagen aufzubauen und aktiv an Forschungstätigkeiten mitzuwirken. In anderen Fällen wurden bestimmte Untersuchungen in einem vertretbaren Zeitaufwand erst aufgrund des effektiven Einsatzes einer TDE ermöglicht.

6 Zusammenfassung und Ausblick

Die Anforderungen bezüglich Flexibilität, Güte, Funktionsumfang und verkürzter Entwicklungszeit an digitalelektronische Steuerungen wachsen ebenso wie deren Einsatzgebiete. Moderne ganzheitliche Entwicklungsumgebungen werden in der Fahrzeugtechnik bereits seit einiger Zeit mit Erfolg eingesetzt und können die Entwicklungsphasen von Reglern oder Steuerungssoftware deutlich verkürzen und effektiver gestalten. Hierdurch steigen die Möglichkeiten erheblich, schneller mit innovativen Ideen in die Praxis zu kommen. Durch die Integration von numerischen Methoden können dem Praktiker zusätzlich wirksame Alternativen bei der Reglerauslegung bereit gestellt werden. Zur Zeit ist der modulare, automatisch erzeugte Code eines CASE-Tools noch zu wenig effektiv, um grundsätzlich auch auf einem preiswerten Microcontroller eingesetzt zu werden. Zum einen ist aber die Entwicklung auf dem Gebiet der Codegenerierung längst nicht abgeschlossen und zum anderen ist mit einer weiteren Leistungssteigerung elektronischer Bauteile bei gleichzeitigem Preisverfall zu rechnen.

Literatur

- [1] Bender, K.: Software - Technologietreiber für den Maschinenbau. VDMA Maschinenbau Nachrichten 1998, H. 9, S. 16-17
- [2] Hanselmann, H.: Beschleunigte Mechatronik-Entwicklung durch Rapid Control Prototyping und Hardware-in-the-Loop-Simulation. at Automatisierungstechnik 46 (1998), H. 3, S. 113-119
- [3] Hanselmann, H.: DSP in der Regelungstechnik: The Total Development Environment. Konferenzbeitrag: "Embedded Intelligence '96", 14.-16. Februar 1996, Stuttgart-Sindelfingen.
- [4] Kiffmeier, U.: Multi-DSP-Power auf Knopfdruck - Vom Simulink-Blockdiagramm zum Multiprozessorsystem. Elektronik 1995, H. 12, 108-117.
- [5] Lang, T.: Schnelle Reglerauslegung in der Hydraulik. 56. Internationale Tagung Landtechnik VDI/MEG 1998, Garching.
- [6] Schineis, W. u. Grund, C.: Der Hardware-in-the-Loop Prüfstand zur Erprobung von Radschlupfsystemen. Jahrbuch 1998, VDI Fahrzeug- und Verkehrstechnik FVT, S. 85-103, VDI-Verlag, Düsseldorf 1998.
- [7] Versteegen, G.: Die Alleskönner - Anforderungen an Software-Entwicklungsumgebungen. Software-Entwicklung 1 1996, H.7-8, S.10-15.
- [8] Versteegen, G.: Qual der Wahl - Individuelle Kriterien bestimmen das richtige CASE-Tool. iX 1992, H. 6, S. 86-89.
- [9] Weisser, M. u. Rüger, B.: Rapid Prototyping - Ein Weg, neue Funktionsideen schneller umzusetzen. ATZ Automobiltechnische Zeitschrift 57 (1996), S.108-115.
- [10] Weisser, M. u. Rüger, B.: Richtig in die Gänge kommen - Durch Rapid Control Prototyping schneller mit der Funktionsidee ins Auto. Elektronik 1995, H. 24, S.72-80.
- [11] Wohnhaas, A.: Rechnergestützte Steuergeräte-Softwareentwicklung. ATZ Automobiltechnische Zeitschrift 99 (1997), H.12, S. 744-754.