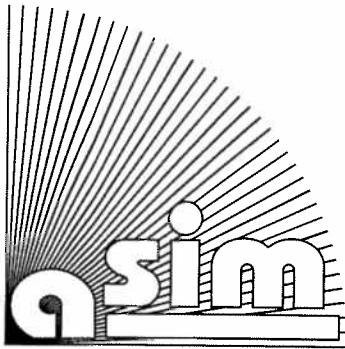




Posterband

12. ASIM - Symposium Simulationstechnik Zürich, September 1998

Max Engeli
Veronika Hrdliczka
(Hrsg.)

ARGESIM Report Nr. 13
ISBN 3-901608-13-3

13. Symposium Simulationstechnik

ASIM '99

<http://www.uni-weimar.de/~fiedler/ASIM/>

21.-24. September 1999
an der Bauhaus-Universität
Weimar



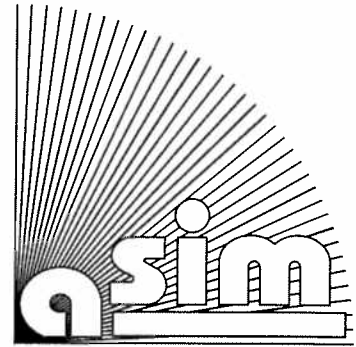
Arbeitsgemeinschaft Simulation
Fachausschuß 4.5



Gesellschaft für Informatik

Bauhaus-Universität Weimar
Wissenschaftszentrum für
Angewandte Informatik und
Mathematik
Fakultät Bauingenieurwesen

Technische Universität Ilmenau
Institut für Wirtschaftsinformatik



Posterband

**12. ASIM - Symposium
Simulationstechnik
Zürich, September 1998**

Max Engeli
Veronika Hrdliczka
(Hrsg.)

ARGESIM Report Nr. 13
ISBN 3-901608-13-3

© 1998 ARGESIM / ASIM

ISBN 3-901608-13-3

ARGESIM Report Nr. 13

ARGE Simulation News (ARGESIM)
c/o Technische Universität Wien
Wiedner Hauptstraße 8-10
A-1040 Wien, Österreich
Tel: +43-1-58801-11480, -11452, -11456
Fax: +43-1-58801-11499
Email: info@argesim.org
WWW: <http://www.argesim.org>

Modellierung und Konzepte

Dynamische Systeme, Modellbildung und Computersimulation

B. Arbenz 1

Theorie und Praxis einer automatischen Modellgenerierung im Rahmen eines integrierten Systems zur Modellierung, simulation und Analyse diskreter Produktionsprozesse

F. Eckardt, D. Palleduhn, P. Gmilkowsky 11

Prognose durch Verknüpfung von Fuzzy Inductive Reasoning mit einem kontinuierlichen Modell

E. Möbus, R. Hohmann 19

Simulation eines embedded Systems

W. Kastner 29

Modellbasierte Evolutionäre Designoptimierung am Beispiel eines optischen Mikrosystems

W. Süß, M. Gorges-Schleuter, W. Jakob, I. Sieber 37

Generieren, Präsentieren und Interagieren als integrale Bestandteile eines komplexen Modellbildungsansatzes

J. Perl 45

Mechatronic Processing Objects

C. Homburg 53

Simulation eines Common Rail Dieselmotors auf Basis eines ASCET Hardware in the Loop Systems

H. Kull, M. Münzenmay, S. Gohl 61

Anwendungen der Simulationstechnik

Möglichkeiten und Grenzen der Optimierung biologischer Abwasserreinigungsanlagen mit Hilfe der Simulation

J. Jungblut, D. P. F. Möller 67

Simulationsgestützte Prototypentwicklung der Regelung einer Großwindanlage

T. Krüger, M. Mevenkamp, J. Petschenka, M. Reichardt 75

Simulation menschlicher Handlungen in dynamischen Mensch-Maschine-Systemen

Th. Jürgensohn 83

Geschäftseinheiten planen und simulieren mit dem BB-Tool

U. La Roche, W. Heinzmann, M. Simon 91

Diskrete Simulation auf Stufe Auftragsabwicklung

V. Metzler 99

Bestandsentwicklung in einer Halbleiterfabrik nach dem Ausfall der EngpaßMaschinengruppe

O. Rose 107

Eine modulare Prozessumgebung zur Visualisierung, Simulation und Auswertung von Verkehrsmessdaten

Th. Riedel 115

Dynamische Systeme, Modellbildung und Computersimulation

Dr. B. Arbenz
e-mail: arbenz@unizh.ch
IFI Universität Zürich, Winterthurerstr.190, CH-8057 Zürich

Zusammenfassung

Dynamische Systeme kommen in der realen Welt an vielen Orten vor. Sie müssen als Modelle beschrieben werden. Sehr oft weisen sie Nichtlinearitäten auf. Dabei können nicht erwartete Verhaltensformen auftreten. Die Untersuchung des Verhaltens ist sehr wichtig, um Auswirkungen von Entwicklungen oder von Eingriffen in den Ablauf vorhersehen zu können. An Beispielen werden Probleme der Simulation und die Problematik der Modellbildung gezeigt. Es werden die Verhaltensformen von linearen Systemen und die zusätzlichen, andersartigen von nichtlinearen Systemen dargestellt. An Hand eines Beispiels wird gezeigt, wie durch geeignete Modellbildung numerische Probleme, verursacht durch extrem steife Differentialgleichungssysteme, vermieden werden können.

Dynamische Systeme existieren in den verschiedensten Fachgebieten [1][3][4][6][7][8][9]. Die realen dynamischen Systeme können als **Modelle** beschrieben werden. In einem Modell sollten alle wichtigen, entscheidenden, in der Wirklichkeit vorkommenden Beziehungen berücksichtigt und unwichtige, nebensächliche weggelassen werden. Es gibt, neben rein gedanklichen Modellen, physikalische und mathematische Modelle. Die physikalischen Modelle sind oft masstabsgetreue Verkleinerungen der Wirklichkeit oder es sind Analogiemodelle, bei denen ein physikalischer Vorgang den gleichen mathematischen Gesetzen gehorcht, wie die Wirklichkeit. Heute dominieren die mathematischen Modelle, denn sie lassen sich auf einem Computer abbilden.

1 Nichtlineare Phänomene.

Reale Systeme weisen in der Regel **Nichtlinearitäten** auf. Oft können die Nichtlinearitäten vernachlässigt werden, d.h. nichtlineare Zusammenhänge durch lineare angenähert werden. Sehr oft müssen jedoch die Nichtlinearitäten berücksichtigt werden, weil keine lineare Näherung existiert, welche die Zusammenhänge genügend genau wiedergibt. Nichtlineare dynamische Systeme können sich ähnlich verhalten, wie lineare, sie können aber auch ganz andere Verhaltensformen aufweisen, welche lineare Systeme gar nicht kennen. Zu nennen sind hier insbesondere die Phänomene **Grenzyklus** und **Chaos**.

Gerade in jüngster Zeit ist das Interesse an nichtlinearen Phänomenen gewaltig gewachsen. Dies hängt damit zusammen, dass nichtlineare Systeme durch nichtlineare Gleichungen, sehr oft nichtlineare Differentialgleichungen, beschrieben werden und nichtlineare Differentialgleichungen in der Regel nicht analytisch sondern nur numerisch gelöst werden können. Die numerische Lösung benötigt meistens leistungsfähige Computer. Die gewaltige Entwicklung im Computerbereich hat es ermöglicht, solche Systeme mit vertretbarem Aufwand zu untersuchen. Daher ist heute vor allem die Chaosforschung gross in Mode. Für die Lösung praktischer Probleme sind jedoch Anwendungen mit regulärem, nichtchaotischem Verhalten wichtiger. Das chaotische Verhalten ist sozusagen die Ausnahmesituation, die in der Regel zu vermeiden ist, da langzeitige Voraussagen der Bewegung dabei nicht möglich sind.

2 Verhaltensformen von dynamischen Systemen.

Im Folgenden beschäftigen wir uns etwas mit Verhaltensformen von dynamischen Systemen. **Lineare Systeme** zeigen entweder stabiles Verhalten oder periodisches Verhalten oder instabiles Verhalten. (Selbstverständlich gibt es auch noch die Verhaltensformen der Uebergangsbereiche, d.h. periodisch abklingendes Verhalten, das zum stabilen, und periodisch aufklingendes Verhalten, das zum instabilen Verhalten zählt.) Wir haben bei folgenden

Beispielen eine konstante Störung angebracht, die von der Zeit $t=1$ bis $t=4$ wirkt und danach wieder verschwindet (Bild 1a). Beim stabilen Verhalten strebt das Ausgangssignal des Systems nach einer Störung wieder zu einem festen Wert zurück (Bild 1b).

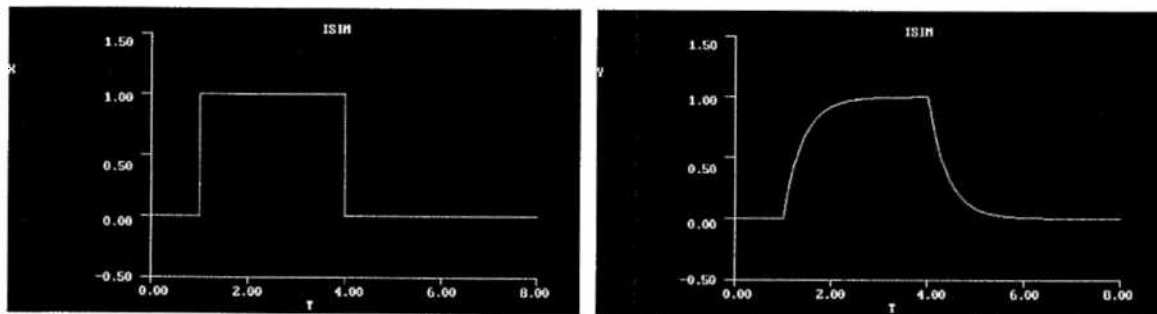


Bild 1a) Zeitlicher Verlauf der Störung (Eingangssignal)

Bild 1b) Zeitlicher Verlauf bei stabilem Verhalten des Systems (Ausgangssignal)

Beim periodischen Verhalten führt das System eine Dauerschwingung mit fester Amplitude durch (Bild 2a). Beim instabilen Verhalten wächst das Signal am Ausgang über alle Grenzen (Bild 2b).

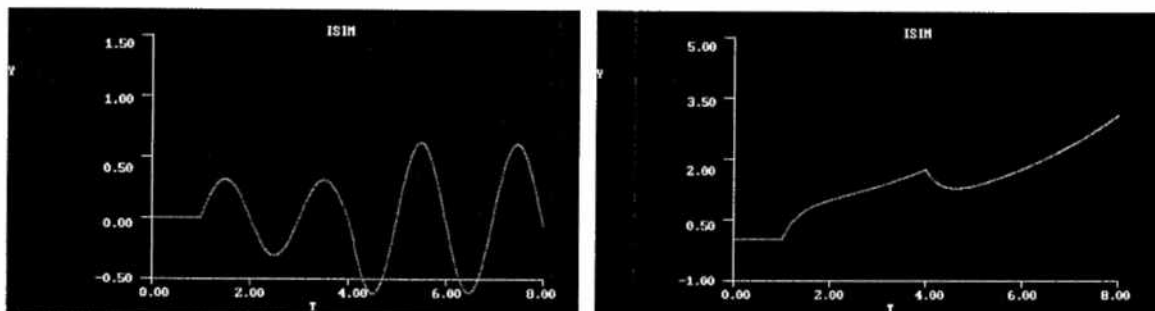


Bild 2a) Zeitlicher Verlauf bei periodischem Verhalten (Ausgangssignal)

Bild 2b) Zeitlicher Verlauf bei instabilem Verhalten des Systems (Ausgangssignal)

Eine Ausnahme bildet nur das System mit integralem Verhalten. Bei ihm wächst das Ausgangssignal solange die Störung andauert und bleibt dann auf dem erreichten Wert stehen. (Bild 3).

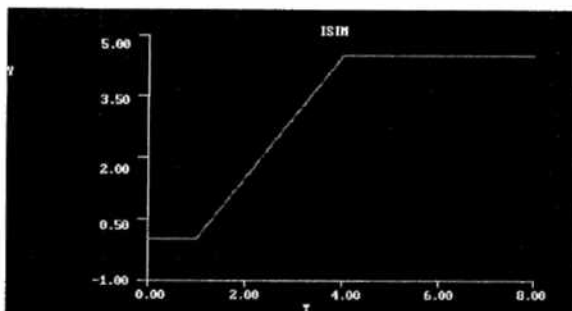


Bild 3) Zeitlicher Verlauf bei integralem Verhalten des Systems (Ausgangssignal)

Aus diesen Darstellungen ist ersichtlich, wie die verschiedenen Systeme reagieren, wenn ein anhaltendes Störsignal auftritt (Bereich $t=1$ bis $t=4$), und ebenso wie die Systeme sich verhalten, wenn das Störsignal wieder verschwunden ist (Bereich $t=4$ bis $t=8$). Bei Systemen, die nicht stabil sind, bleibt eine Nachwirkung der Störung. Nur bei absolut stabilen Systemen ist keine Nachwirkung vorhanden. Systeme mit stabilem Verhalten kehren zum ursprünglichen Zustand zurück. Systeme mit periodischem Verhalten dagegen führen auch nach dem Verschwinden der Störung eine periodische Schwingung aus und Systeme mit instabilem

Verhalten "laufen weg". Systeme mit integrelem Verhalten verharren auf dem neuen Wert.

Nichtlineare Systeme können weitere Verhaltensformen aufweisen. Als neues Phänomen können **Grenzzyklen** auftreten (Bild 4a,b). Diese sind entweder stabil, semistabil oder instabil. Bei stabilen Grenzzyklen führen die Trajektorien (Bahnkurven) von beiden Seiten gegen den Zyklus, bei semistabilen dagegen nur auf einer Seite und auf der andern führen sie vom Zyklus weg und schliesslich bei instabilen Grenzzyklen divergieren sie auf beiden Seiten vom Zyklus weg.

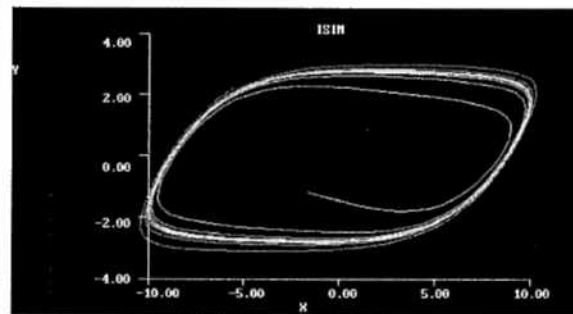
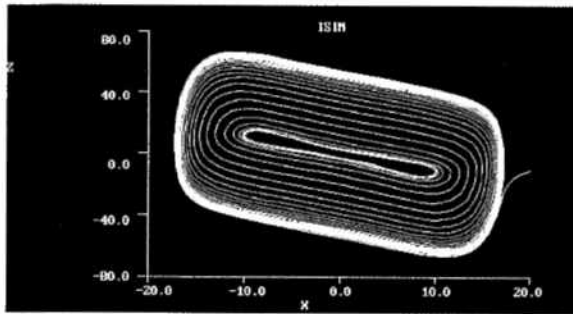


Bild 4a) Grenzzyklen in der Zustandsebene: Stabiler Grenzzyklus (eine Kurve führt von innen die andere Kurve von aussen zum Grenzzyklus)

Bild 4b) Grenzzyklen in der Zustandsebene dargestellt: Instabiler Grenzzyklus (eine Kurve führt nach innen, die andere Kurve nach aussen weg vom Grenzzyklus)

Des weitern kann bei kontinuierlichen Systemen mit mehr als 2 Freiheitsgraden auch **Chaos** auftreten. In Bild 5a sehen wir das Verhalten des chaotischen Pendels. Es gehorcht der Differentialgleichung

$$\frac{d^2x}{dt^2} = A \cos(\omega t) - \sin(x) - rx$$

und stellt physikalisch ein gedämpftes Pendel dar, das mit der Frequenz ω angeregt wird. Für bestimmte Werte von A und ω wird es chaotisch (z.B. $A=2$, $\omega=0.5$, $r=0.2$).

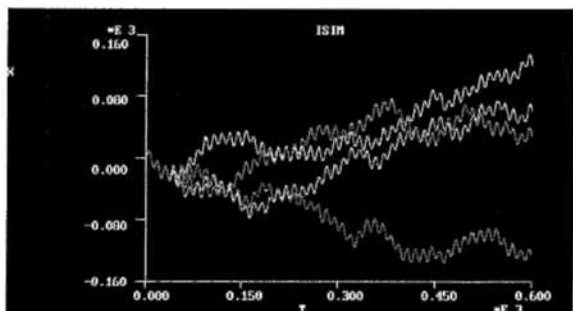
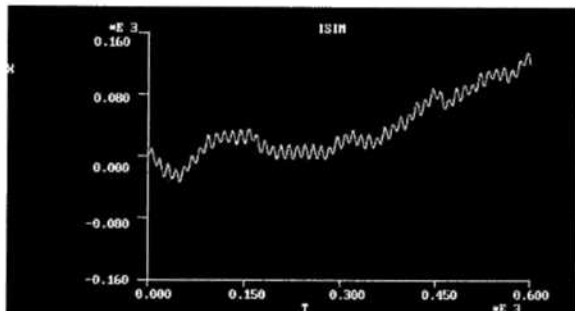


Bild 5a) Zeitlicher Verlauf des Auslenkwinkels beim chaotischen Pendel

Bild 5b) Zeitlicher Verlauf des Auslenkwinkels beim chaotischen Pendel, bei kleiner Variation der Anfangsbedingungen

Das **Chaos** kann ganz unterschiedliche Formen aufweisen. Allen chaotischen Verhalten gemeinsam ist nur, dass der Zustand langfristig nicht mehr voraussagbar ist. Das System ist extrem empfindlich auf kleinste Störungen bzw. kleinste Abweichungen in den Anfangsbedingungen. Variieren wir beim chaotischen Pendel leicht die Auslenkung x am Anfang, so erhalten wir das Bild 5b. Ganz am Anfang verhält sich das Pendel in allen diesen Fällen noch gleich, aber nach kurzer Zeit legt es, je nach Anfangswert, einen vollständig andern Weg zurück.

Punkte, Grenzzyklen oder Gebiete (bei chaotischem Verhalten), welche die Lösung an sich heranziehen bezeichnet man als **Attraktoren**. Die Gebiete, welche beim chaotischen Verhalten die Lösung an sich heranziehen, nennt man auch seltsame Attraktoren.

Nun ist es auch möglich, dass verschiedene Verhaltensformen bei einem System

nebeneinander vorkommen, d.h. je nach den Anfangswerten treten andere Verhaltensformen auf, welche dann ein bestimmtes Gebiet abdecken. Diese Gebiete werden durch Grenzzyklen voneinander getrennt.

3 Vom regulären zum chaotischen Verhalten.

Oft führt eine Parametervariation von stabilem zu chaotischem Verhalten. Für bestimmte Werte des Parameters zeigt das System reguläres Verhalten, durch Vergrössern des Wertes dieses Parameters tritt ab einem bestimmten Wert plötzlich periodisches Verhalten auf. Wird der Wert weiter vergrössert, so wird bei einem gewissen Parameterwert die Periode verdoppelt. Noch weiteres Vergrössern führt von Zeit zu Zeit zu nochmaligem verdoppeln der Periode, so dass der Reihe nach eine einfache, eine doppelte, eine vierfache, usw. Periode auftritt. Man spricht hier vom Bereich der Periodenverdoppelung. Bei einem bestimmten Wert geht das System über zum Chaos. Innerhalb des Chaosbereichs können aber plötzlich wieder periodische Verhalten auftreten, die dann erneut über Periodenverdoppelung ins Chaos führen. Dabei spricht man von Fenstern im Chaosbereich. Dieses Verhalten kann sowohl bei diskreten, wie auch bei kontinuierlichen nichtlinearen Systemen auftreten. Als Beispiel sei hier der Vorgang an Hand eines diskreten Systems, des logistischen Wachstums, das der Gleichung

$$x_{n+1} = px_n(1 - x_n)$$

gehört, für den Startwert $x=0.5$ und Werte des Parameters $p=0$ bis 4, aufgezeigt. Im Bifurkationsdiagramm Bild 6 sind die Iterationsschritte von $n=100$ bis 200 dargestellt, d.h. die ersten 100 Schritte wurden weggelassen, um das Verhalten im "eingeschwungenen Zustand" zu zeigen.

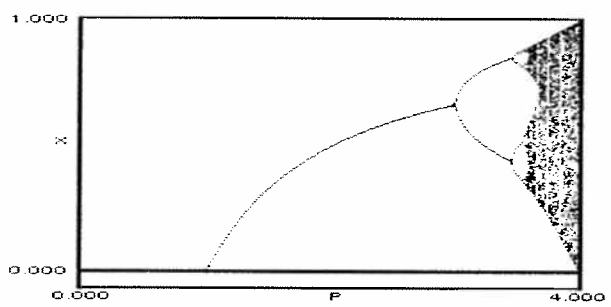


Bild 6) Bifurkationsdiagramm der logistischen Gleichung

Die gleichen Phänomene können auch auftreten, wenn bei der numerischen Integration zur Lösung eines kontinuierlichen Systems eine zu grosse Schrittweite gewählt wird. Dies ist nicht verwunderlich, erfolgt doch bei der numerischen Integration eine Diskretisierung des Systems. Wird z.B. das System

$$\frac{dy}{dt} = y - y^2$$

mit der Integrations-Methode von Gourlay (eine modifizierte Trapez-Methode) mit nach und nach grösser gewählter Integrationsschrittweite Δt im Bereich von 0.5 bis 4 für die Zeit $t=0$ bis 40 berechnet, so erhält man bei kleiner Schrittgrösse noch die richtige Lösung. Mit zunehmender Schrittweite beginnt das Resultat von der richtigen Lösung abzuweichen. Bei einem bestimmten Wert alterniert das Resultat zwischen 2 Werten, bei noch grösserer Schrittweite zwischen mehr Werten. Schliesslich beginnt die Lösung zwischen beliebigen Werten herumspringen, d.h. das Resultat wird chaotisch.

Den Weg über Periodenverdoppelung zum Chaos finden wir auch bei kontinuierlichen Systemen. Wir können dies z.B. beim "Rössler-Attraktor" beobachten. Das Rössler-Modell

$$\dot{x} = -y - z \quad \dot{y} = x + ay \quad \dot{z} = b + (x - c)z$$

zeigt für bestimmte Parameterwerte einen stabilen Grenzzyklus. Dieser wandelt sich beim Vergrössern des entsprechenden Parameters von einem einfachen zu einem verschlungenen Zyklus, wobei die Periode verdoppelt wird. Die Verdoppelung kann wiederholt stattfinden, so

dass eine Vervielfachung bzw. Verachtfachung der Periode vorliegen kann. Ab einem gewissen Wert des betreffenden Parameters treten chaotische Bänder auf. Diese verschmelzen langsam zu einem chaotischen Gebiet. Bei gewissen Werten wird das Chaos wieder durch Fenster mit nichtchaotischem Verhalten unterbrochen.

4 Reale Systeme.

Wir sehen also, dass das Verhalten von nichtlinearen Systemen viel komplizierter sein kann, als das von linearen. Dementsprechend wird heute viel über nichtlineare Dynamik geforscht. Neben diesen in der Regel sehr theoretischen Arbeiten, hat das Ganze auch praktische Bedeutung. Die Natur ist meistens nichtlinear, so dass die realen Systeme normalerweise eigentlich nichtlineare Systeme sind. Abgesehen von der Physik und der Technik, werden diese heute noch wenig untersucht. Die Computersimulation ist nun ein ideales Mittel solche Phänomene in den unterschiedlichsten Fachgebieten zu untersuchen. Sie erlaubt auf einfache Weise Parametervariationen durchzuführen, was bei experimentellen Untersuchungen sehr oft nur mühsam oder nur mit sehr grossem Aufwand zu bewerkstelligen ist. Um Computersimulationen durchzuführen, muss die Wirklichkeit als Modell erfasst werden.

Bei nichtlinearen Systemen können auch andere nicht erwartete Effekte auftreten. Als Beispiel sei hier das **Volterra-Prinzip** bei Räuber-Beute-Systemen erwähnt [2]. Das Volterra-Prinzip wurde auf Grund der Fischfangstatistik in Italien entdeckt. Der Biologe U.D'Ancona untersuchte die Verteilung verschiedener Fischarten im Meer. Dabei stiess er auf Daten über den Fischfang in den italienischen Mittelmeerhäfen, welche den prozentuellen Anteil von Speisefischen und ihren Räubern (Haie und verschiedene Rochenarten) am Gesamtfang aufzeigten. An Hand dieser Daten stellte er fest, dass von 1914 bis 1918, als der Fischfang wegen des 1. Weltkrieges stark eingeschränkt war, der Anteil der Räuberfische von 12% auf 36% anstieg und von 1919 bis 1923, als die Fischerei wieder intensiver betrieben wurde, kontinuierlich auf 11% zurück ging. Nachdem D'Ancona keine biologische Erklärung für dieses Phänomen finden konnte, bat er den Mathematiker V. Volterra ein mathematisches Modell für das Wachstum der beiden Fischarten aufzustellen. Volterra entwickelte ein Räuber-Beute-Modell, in welchem der Fischfang sowohl bei den Beutefischen (Population x) mit einer Rate $e x$, wie auch bei den Räuberfischen (Population y) mit einer Rate $e y$ berücksichtigt wird. Dies führt auf das Gleichungssystem

$$\frac{dx}{dt} = ax - cxy - ex \quad \frac{dy}{dt} = -by + dxy - ey$$

vernünftigerweise mit der Bedingung $a - e > 0$.

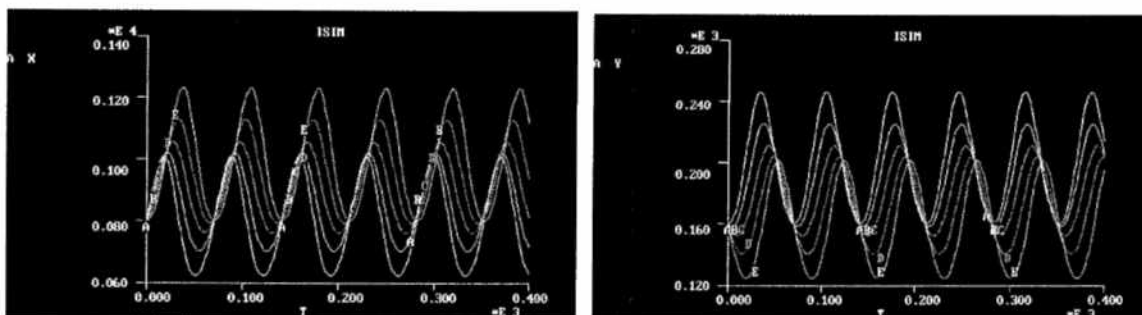


Bild 7) Volterra-Prinzip beim Räuber-Beute-System, Kurve A: Fangrate = 0, Kurve B: Fangrate = 0.005, Kurve C: Fangrate = 0.01, Kurve D: Fangrate = 0.015, Kurve E: Fangrate = 0.02

- Zeitlicher Verlauf der Beutepopulation X (unterste Kurve Fangrate = 0, oberste Kurve Fangrate = 0.02)
- Zeitlicher Verlauf der Räuberpopulation Y (oberste Kurve Fangrate = 0, unterste Kurve Fangrate = 0.02)

Dieses System zeigt bei geeigneten Parameterwerten (z.B. $a = 0.1$, $b = 0.08$, $c = 0.0005$,

$d = 0.0001$, $e = 0$ bis 0.02 in Schritten von 0.005 und den Anfangswerten $x(t=0) = 800$, $y(t=0) = 160$ mit steigender Fangrate e einen steigenden Bestand an Speisefischen (Population x) und einen sinkenden Bestand der Räuberfische (Population y). Bild 7 zeigt das Resultat mit den Fangraten 0 (unterste Kurve), bzw. 0.005 , bzw. 0.01 , bzw. 0.015 und 0.02 (oberste Kurve) für die Beutepopulation.

Das überraschende Verhalten dieser Lotka-Volterra-Gleichungen wird im Volterra-Prinzip festgehalten: *“Werden zwei in einem Räuber-Beute-Verhältnis stehende Arten durch eine äussere Einwirkung, z.B. jagen oder verwenden von Pestiziden durch den Menschen, in etwa gleichem prozentualen Ausmass reduziert, so wächst die Beutepopulation und die Räuberpopulation nimmt ab.”*

Dieses Prinzip zeigte sich auch beim Einsatz von Pestiziden. Das Baumwollschuppen-Insekt wurde 1868 aus Australien nach Amerika eingeschleppt und drohte dort die Zitrusfrüchte-Industrie zu vernichten. Man holte darauf seinen natürlichen Jäger “Thereupon”, ein Marienkäferchen aus Australien, welches die Schuppeninsekten auf ein niedriges Niveau reduzierte. Als das Insektenvertilgungsmittel DDT erfunden war, wollte man es damit noch weiter reduzieren. Doch wo DDT eingesetzt wurde, vermehrte sich das Insekt wieder stärker als vorher - ganz in Uebereinstimmung mit dem Volterra-Prinzip.

Aus dem Dargelegten geht deutlich hervor, dass das Verhalten von nichtlinearen Systemen nicht mehr so einfach durchschaubar ist, wie das von linearen. Demzufolge müssen nichtlineare Systeme einzeln untersucht werden und dies kann praktisch nur mit Hilfe von Simulationen erfolgen. Simulationen werden künftig in der Wissenschaft, insbesondere in der Forschung, in den verschiedensten Fachgebieten eine immer wichtigere Rolle spielen.

5 Modellbildung.

Ein Modell kann auch in einem beschränkten Bereich seine Gültigkeit haben. Dies sehen wir an Hand des folgenden Prozesses. Wir stellen ein chemisches Modell für einen Reaktionszyklus in der Atmosphäre auf, der die Entstehung von Ozon beschreibt.

Durch Lichtquanten wird das Molekül NO_2 aufgespalten in die Moleküle NO und O , wobei die Lichtquanten Energie zuführen: $\text{NO}_2 + h\nu \rightarrow \text{NO} + \text{O}$

Dieser Vorgang kann durch die Differentialgleichung

$$\frac{d[\text{O}]}{dt} = \frac{d[\text{NO}]}{dt} = -\frac{d[\text{NO}_2]}{dt} = J_{\text{NO}_2}[\text{NO}_2]$$

dargestellt werden. Dabei bedeutet

$[\text{NO}]$ = Konzentration an NO in Mol cm^{-3}

$[\text{NO}_2]$ = Konzentration an NO_2 in Mol cm^{-3}

J_{NO_2} = reziproke Zeitkonstante (abhängig von $h\nu$)

Da gleichzeitig noch andere Reaktionen ablaufen (z.B. $\text{O} + \text{O}_2 \rightarrow \text{O}_3$ und $\text{NO} + \text{O}_3 \rightarrow \text{NO}_2 + \text{O}_2$), beschreibt obenstehende Gleichung nur einen Teil des gesamten Prozesses.

Die in der Atmosphäre ablaufende Reaktion $\text{O}_2 + \text{O} \rightarrow \text{O}_3$ braucht einen Stosspartner (z.B. N_2), der den grössten Teil der frei werdenden Energie aufnimmt. Für sie gilt die Differentialgleichung

$$\frac{d[\text{O}_3]}{dt} = -\frac{d[\text{O}_2]}{dt} = -\frac{d[\text{O}]}{dt} = k[\text{O}][\text{O}_2]$$

Diese Reaktion läuft sehr schnell ab (Lebensdauer des O -Moleküls ca. 13 msec) und daher weist k einen grossen Wert auf.

Ähnliche Gesetze gelten auch bei bimolekularen Reaktionen, wie z.B.

$\text{NO} + \text{O}_3 \rightarrow \text{NO}_2 + \text{O}_2$. Hier gilt die Beziehung:

$$\frac{d[\text{NO}_2]}{dt} = \frac{d[\text{O}_2]}{dt} = -\frac{d[\text{NO}]}{dt} = -\frac{d[\text{O}_3]}{dt} = k_{bi}[\text{NO}][\text{O}_3]$$

Für einen realen Vorgang ergibt sich ein ganzes System von Differentialgleichungen. Da

die beiden Prozesse an der ein Stoff beteiligt ist, der bildende und der abbauende, gleichzeitig ablaufen, tritt in der Differentialgleichung für die Zunahme des Stoffes die Differenz der beiden auf. So erhalten wir hier für den ganzen chemischen Kreislauf folgendes Modell:

Differentialgleichungssystem

$$\begin{aligned}\frac{d[NO]}{dt} &= J_{NO_2}[NO_2] - k_{bi}[NO][O_3] & \frac{d[O]}{dt} &= J_{NO_2}[NO_2] - k[O][O_2] \\ \frac{d[O_3]}{dt} &= k[O][O_2] - k_{bi}[NO][O_3] & \frac{d[NO_2]}{dt} &= k_{bi}[NO][O_3] - J_{NO_2}[NO_2] \\ \frac{d[O_2]}{dt} &= k_{bi}[NO][O_3] - k[O][O_2]\end{aligned}$$

Werte der Koeffizienten

$$k = 10^{12} \text{ cm}^3 \text{ Mol}^{-1} \text{ sec}^{-1} \quad k_{bi} = 1.08 \cdot 10^{10} \text{ cm}^3 \text{ Mol}^{-1} \text{ sec}^{-1}$$

$$J_{NO_2} = 8 \cdot 10^{-3} \text{ sec}^{-1} \text{ (bei vollem Sonnenlicht)}$$

Anfangswerte der Stoffkonzentrationen

$$[NO] = 0.45 \cdot 10^{-11} \text{ Mol cm}^{-3} \quad [NO_2] = 0.45 \cdot 10^{-10} \text{ Mol cm}^{-3}$$

$$[O] = 0 \text{ Mol cm}^{-3} \quad [O_3] = 0.45 \cdot 10^{-10} \text{ Mol cm}^{-3}$$

$$[O_2] = 0.94 \cdot 10^{-5} \text{ Mol cm}^{-3}$$

Dieses System kann leicht in einer Simulationssprache (z.B. ISIM [3], welche dem CSSL-Standard folgt, oder in SIMULINK) programmiert werden.

ISIM-Programm:

```
:NOX NOX-Kreislauf, Chemische Reaktionen
: NO2 + hv --> NO + O , O2 + O --> O3 , NO + O3 --> NO2 + O2
CONSTANT TFIN = 1.0E-6
ALGO = 0 ; CINT = 1.0E-8 ; RERR = 0.00001
Jno2 = 8.0E-3 ; K = 1.0E12 ; Kbi = 1.08E10
SIM
INITIAL
NO = 0.45E-11 : NO2 = 0.45E-10
O = 0. ; O2 = 0.94E-5 ; O3 = 0.45E-10
DYNAMIC
KOO2 = K*O*O2 ; Jno2NO2 = Jno2*NO2 ; KbiNOO3 = Kbi*NO*O3
O' = Jno2NO2-KOO2
O3' = KOO2-KbiNOO3
NO' = Jno2NO2-KbiNOO3
NO2' = KbiNOO3-Jno2NO2
O2' = KbiNOO3-KOO2
OUTPUT T,NO,NO2,O,O2,O3
PLOT T,O,O,TFIN,0,1.E-19
PREPARE T,NO,NO2,O,O2,O3,NO',NO2',O',O2',O3'
```

Dieses ISIM-Programm liefert den Verlauf der Konzentration von O, aber keine brauchbaren Resultate für den Verlauf der andern Stoff-Konzentrationen, bei längern Zeiten.

Grund: Die Konzentrationsänderung von O läuft sehr viel schneller ab, als die der andern Stoffe, da O sofort mit O₂ reagiert und O₃ bildet.

Die Schwierigkeit kann umgangen werden, indem die extrem schnelle Reaktion O₂ + O ----> O₃ in die Reaktion NO₂ + hv ----> NO + O integriert wird, so als ob die Reaktion NO₂ + hv + O₂ -----> NO + O₃ heissen würde. Dies führt auf das vereinfachte Differentialgleichungssystem:

$$\frac{d[NO]}{dt} = J_{NO_2}[NO_2] - k_{bi}[NO][O_3]$$

$$\frac{d[O_3]}{dt} = J_{NO_2}[NO_2] - k_{bi}[NO][O_3] \quad \frac{d[NO_2]}{dt} = k_{bi}[NO][O_3] - J_{NO_2}[NO_2]$$

Im vollständigen Modell haben wir sehr unterschiedlich schnell ablaufende Reaktionen. Das führt dazu, dass das Differentialgleichungssystem extrem steif ist. Wird das vollständige Modell in SIMULINK erstellt (Bild 8), so kann es mit Hilfe einer Integrationsmethode für steife Differentialgleichungen gelöst werden (z.B. in SIMULINK 1.3 der Gear-Methode). Wir brauchen in diesem Fall kein vereinfachtes Modell zu erstellen. Andererseits schleppen wir einen Teil des Modells mit, der kaum von Interesse ist, da diese Reaktion in 10E-6 sec abläuft, uns aber den Rechenaufwand erheblich vergrößert. Ebenso können wir die Differentialgleichung für O₂ weglassen, da der Grundpegel an O₂ soviel grösser ist, als die Beiträge durch die chemische Reaktion, so dass die Konzentration von O₂ dadurch nicht spürbar verändert wird.

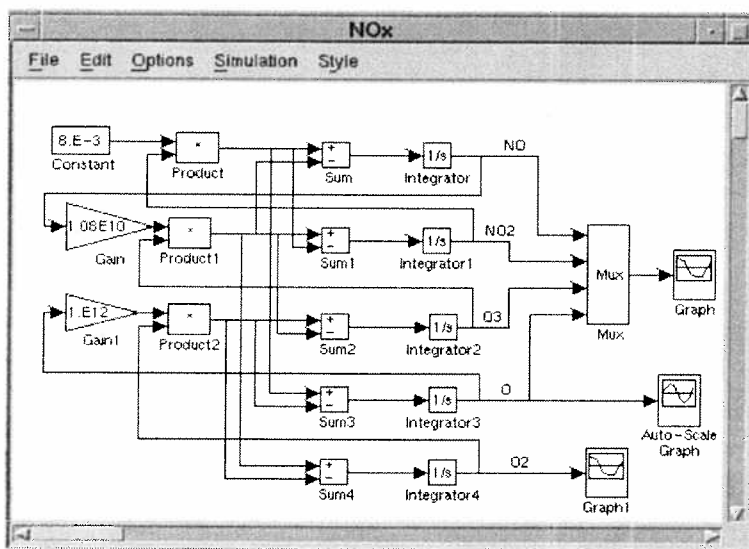


Bild 8) Modell für NOx-Kreislauf in SIMULINK

Man kann sich nun fragen, was geschieht, wenn die Abgase von Autos den natürlichen Reaktionsablauf stören. Wir wollen uns hier auf den Einfluss von Benzin-Autos beschränken. (Für Diesel-Autos ist das Vorgehen grundsätzlich gleich nur die Werte sind anders.)

Benzinautos stoßen rund 96% ihres NO_x-Ausstosses in Form von NO aus und rund 4% in Form von NO₂ [8]. Wird dieser Ausstoss in unserem Modell eingesetzt so lautet dieses, wenn wir das vereinfachte Differentialgleichungssystem zu Grunde legen:

$$\frac{d[NO]}{dt} = J_{NO_2}[NO_2] - k_{bi}[NO][O_3] + [NO]_{zu}$$

$$\frac{d[O_3]}{dt} = J_{NO_2}[NO_2] - k_{bi}[NO][O_3]$$

$$\frac{d[NO_2]}{dt} = k_{bi}[NO][O_3] - J_{NO_2}[NO_2] + [NO_2]_{zu}$$

In der Simulation lassen wir nur über einen beschränkten Zeitraum die Auto-Abgase auf die Reaktion einwirken (von 15 bis 20 sec), um zu sehen, welche Veränderung während der Einwirkung und welche danach besteht und variieren die ausgestossene Menge an Abgasen ([NO]_{zu} und [NO₂]_{zu}).

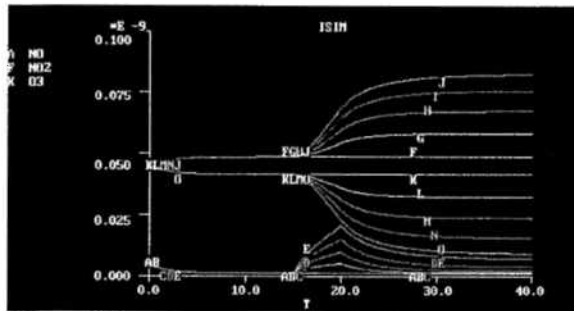


Bild 9) Verlauf der Konzentration von NO, NO₂ und O₃, wenn von T=15 bis 20 sec Abgase von Benzinautos zugeführt werden. Die ausgestossene Abgasmenge wurde von 0 bis 8.E-12 Mol/cm³ sec variiert in Schritten von 2.E-12 Mol/cm³ sec.

Das Resultat der Simulation zeigt (Bild 9) eine Zunahme an NO durch Autoabgase, welche nach beendeten Verkehrsemissionen wieder zurückgeht, aber nicht ganz auf den ursprünglichen Wert. Der NO₂-Pegel steigt an und verflacht sich nach Beendung der Verkehrsemissionen auf einem höheren Niveau. Im Gegensatz hierzu nehmen die O₃-Konzentrationen während den Verkehrsemissionen ab und stabilisieren sich danach auf einem tieferen Niveau.

Dieses überraschende Resultat deckt sich insofern mit Beobachtungen, als die tiefsten O₃-Werte entlang von vielbefahrenen Autobahnen gemessen werden und die höchsten O₃-Werte auf dem Lande (z.B. Bachtel). Das Modell gibt also in dieser Hinsicht richtige Resultate wieder. Es enthält aber nicht die weiteren, in der Natur auch vorkommenden, zum Teil noch nicht vollständig bekannten, anschliessenden, meist langsamer ablaufenden Reaktionen und kann daher auch keine Aussagen über langzeitiges Verhalten machen.

Wir sehen daraus, dass das was ein Modell aussagen soll wichtig ist, um zu entscheiden, welche Prozesse im Modell enthalten sein müssen und welche weggelassen werden können.

Literatur:

- [1] K.H.Becker, M.Dörfler: *Dynamische Systeme und Fraktale*, Vieweg Verlag 1989
- [2] M.Braun: *Differentialgleichungen und ihre Anwendungen*, Springer Verlag 1991
- [3] I.J.Dunn, E.Heinzle, J.E.Prenosil, J.Ingham: *Biological Reaction Engineering*, VCH Verlag 1992
- [4] G.Gandolfo: *Mathematical methods and models in economic dynamics*, North-Holland 1980
- [5] G.Jetschke: *Mathematik der Selbstorganisation*, Vieweg Verlag 1989
- [6] W.Metzler: *Dynamische Systeme in der Oekologie*, B.G.Teubner 1987
- [7] L.Merz, H.Jaschek: *Grundkurs der Regelungstechnik*, Oldenbourg Verlag 1990
- [8] N.Moussiopoulos, W.Oehler, K.Zeller: *Fahrzeugemissionen und Ozonbildung*, Springer Verlag 1989
- [9] H.G.Schuster: *Deterministisches Chaos*, Verlag Chemie 1994
- [10] H.R.Schwarz: *Numerische Mathematik*, B.G.Teubner 1993
- [11] R.Worg: *Deterministisches Chaos*, BI-Wissenschaftsverlag 1993

Theorie und Praxis einer automatischen Modellgenerierung
im Rahmen eines integrierten Systems
zur Modellierung, Simulation und Analyse diskreter Produktionsprozesse

F. Eckardt, D. Palleduhn, P. Gmilkowsky

frank.eckardt@wirtschaft.tu-ilmenau.de, dirk.palleduhn@wirtschaft.tu-ilmenau.de,

peter.gmilkowsky@wirtschaft.tu-ilmenau.de

Technische Universität Ilmenau, Fakultät für Wirtschaftswissenschaften,

Institut für Wirtschaftsinformatik, Fachgebiet Wirtschaftsinformatik I,

Postfach 10 05 65, D-98684 Ilmenau

Zusammenfassung

Auf der Grundlage einer Analyse der Kernprobleme des Einsatzes von Simulationstechniken in der kurz- und mittelfristigen Produktionsplanung und -steuerung werden zunächst die Grundzüge einer Theorie der automatischen Modellgenerierung dargelegt. Hiernach folgt die Beschreibung des realisierten Prototyps, der wissensbasierte Modellgenerierung und objekt-orientierte Simulation integriert. Dadurch kann der Prozeß der Modellbildung für einen definierten Problembereich weitestgehend automatisiert durchgeführt und der Einsatz der Simulation in der operativen Fertigungssteuerung entscheidend vereinfacht werden. Neben der grundlegenden Architektur des Systems wird insbesondere die Grobstruktur der entwickelten Klassenbibliothek vorgestellt.

1 Einführung

1.1 Aufgaben und Probleme der Werkstattsteuerung

In der betrieblichen Praxis ist seit längerer Zeit eine Trend zur verstärkten Integration von Simulationsanwendungen in bestehende PPS-Systeme zu beobachten. Diese Entwicklung resultiert primär aus den zahlreichen Schwachstellen der heute verfügbaren Softwareprodukte im Hinblick auf eine Unterstützung der Simultanplanung, der Mengen-, Termin- und Kapazitätsplanung sowie der echtzeitnahen Bereiche [5]. Ursächlich dafür ist die mangelnde Berücksichtigung einer gleichermaßen integrierten und simultanen Material-, Personal- und Betriebsmittelplanung unter zeitdynamischen Aspekten. Während diese Defizite in den längerfristigen Dispositionen noch keine signifikanten Auswirkungen auf die Gesamtvalidität der erzeugten Pläne haben, führen sie in den analytisch nicht mehr lösbaren taktischen und operativen Planungsläufen zu überwiegend unbrauchbaren Ergebnissen. Die Forderung nach einer

Behebung der genannten Schwachstellen erfordert demnach entweder einen neuen Typus von PPS-Systemen oder zumindest eine funktionale Erweiterung der vorhandenen Ansätze [5].

Eine Entwicklung, die aus dieser Forderung unmittelbar resultiert, ist die Auslagerung der Produktionssteuerung in einen unabhängigen Funktionskomplex. Dieser hat die operative Verwaltung, Planung, Steuerung und Kontrolle aller Fertigungsaufträge zum Gegenstand, deren erster Arbeitsgang kurz vor der Ausführung steht oder bereits gestartet wurde (mittelfristiger Planungshorizont) [1, 5]. Die Antizipation sowohl der Auswirkungen individueller Entscheidungen von Dispatchern und Meistern als auch der Konsequenzen von Störungen bleibt aber weiterhin problematisch. In diesem Kontext hat sich die Anwendung von Techniken der diskreten Simulation als effizient und sinnvoll erwiesen [6, 8].

1.2 Unterstützung der Werkstattsteuerung durch diskrete Simulation

Da eine simulative Abbildung von Produktionsprozessen zur Abschwächung und/oder Kompensation der oben erwähnten Schwachstellen klassischer PPS-Systeme im operativen Kapazitätsmanagement als integraler Bestandteil eines Fertigungslenkungssystems aufzufassen ist, wird nachfolgend ein möglicher Lösungsansatz in Form der Architektur eines wissensbasierten Entscheidungsunterstützungssystems zur Modellierung, Simulation und Analyse diskreter Fertigungs- bzw. Geschäfts- oder Unternehmensprozesse dargestellt [6].

Das System besteht aus drei elementaren Modulen, die intern über eine gemeinsame Daten- und Informationsbasis und extern über einen gemeinsamen Benutzerframework miteinander verbunden sind [vgl. Abbildung 1].

Modellierungskomponente

Bei der Durchführung von Simulationsuntersuchungen bildet die Anwendung der Modellierungskomponente den konstitutiven Ausgangspunkt, da sie die Schaffung der elementaren Voraussetzungen respektive die Unterstützung einer individuellen Simulationsmodellerstellung gewährleistet. Im Gegensatz zu klassischen und ergo meist statischen Modellierungsverfahren, bei denen durch einen menschlichen Experten zu einem gegebenen Zeitpunkt t_n ein Simulationsmodell des Produktionsprozesses manuell erstellt wird, generiert die Modellierungskomponente dynamische, granularitätsvariable und kontextsensitive Simulationsmodelle weitestgehend automatisch.

Aufgrund der speziellen Bedeutung der Modellierungskomponente für die Entwicklung des Gesamtsystems behandelt der folgende Abschnitt zunächst die Theorie einer automatisierten Modellgenerierung, um darauf aufbauend die prototypische Umsetzung der Komponente zu präzisieren.

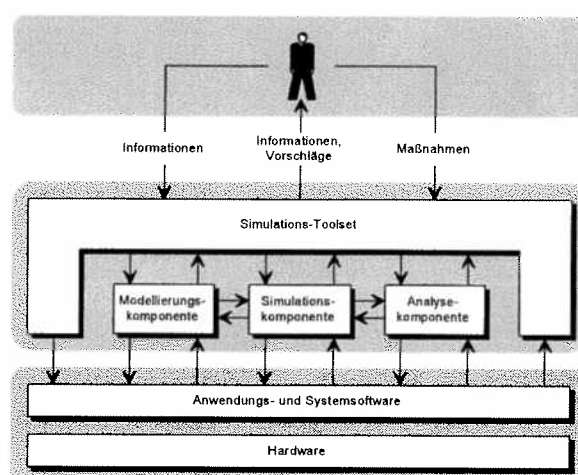


Abbildung 1: Architektur des Simulations-Toolset [7]

Simulationskomponente

Der Simulationskomponente obliegt im Systemkontext die eigentliche Durchführung der Simulationsexperimente im Sinne einer algorithmischen Abarbeitung der in der Modellierungskomponente erzeugten Modelle. Sie stellt mithin die originäre Simulationsfunktionalität zur Verfügung.

Analysekomponente

In der Analysekomponente erfolgt die betriebswirtschaftliche Untersuchung und Bewertung der Simulationsergebnisse, um Korrekturmaßnahmen abzuleiten, die entweder den Prozeßverantwortlichen für eine Evaluation zur Verfügung gestellt oder aber für eine automatisierte Prozeßregelung verwendet werden können [7].

2 Theorie einer automatisierten Modellgenerierung

2.1 Grundproblematik einer automatischen Modellgenerierung

Die Automatisierung einer Modellgenerierung kann terminologisch als die Schaffung und Anwendung solcher technischen und informationellen Mittel und Systeme abgegrenzt werden, die selbsttätig in der Lage sind, ein Simulationsmodell in Einheit mit dem zugehörigen Steuerungsmodell zu erzeugen [4]. In diesem Zusammenhang besteht das Grundproblem weniger in der originären Algorithmierung denn vielmehr in der vollständigen Abbildung eines ganzheitlichen Prozeßgeschehens auf einem beliebigen Granularitätsgrad. Diese Abbildung respektive das korrespondierende Simulationsmodell müssen zur Sicherstellung der Ganzheitlichkeit die folgenden Rahmenbedingungen eines realen Prozesses berücksichtigen:

- **Technologische Rahmenbedingungen**

Die technologischen Rahmenbedingungen umfassen alle Konditionen, die sich aus den eingesetzten Produktionstechniken ergeben. Als Beispiele seien hier technische Vorgaben und Restriktionen einzelner Produktionsvorgänge und –mittel hinsichtlich ihres Einsatzes bei spezifischen Prozeßkonstellationen genannt. Diese Bedingungen subsumieren dabei auch technische Maßtoleranzen und/oder Mindestgüten, die nur von spezifischen Produktionsmitteln innerhalb eines homogenen Funktionalpools erreicht werden können.

- **Organisatorische Rahmenbedingungen**

Die organisatorischen Rahmenbedingungen beinhalten alle Konditionen, Restriktionen und Vorgaben, welche die Organisation des Produktionsprozesses betreffen. Hierunter werden sowohl die personell-organisatorischen Bedingungen, wie beispielsweise Arbeitszeitmodelle, persönliche Arbeitsaufgaben und der Betriebskalender subsumiert als auch produktions-organisatorische Konditionen, wie etwa die Organisation des Transport- und Werkzeugversorgungssystems.

- **Strukturelle Rahmenbedingungen**

Strukturelle Rahmenbedingungen sind diejenigen Konditionen, Restriktionen und Vorgaben, die sich aus der räumlichen und zeitlichen Struktur des Produktionsprozesses ergeben. Beispielfhaft kann man hier die räumliche Aufstellung einzelner Betriebsmittel innerhalb der Fertigungsorganisation sowie räumliche Restriktionen bei Lager- und Pufferflächen nennen.

Daneben müssen das erzeugte Simulationsmodell und/oder das dieses Simulationsmodell generierende Modellierungssystem über Möglichkeiten verfügen, dynamische Veränderungen am zugrundeliegenden Realsystem selbständig zu erkennen und daraus automatisch Deduktionen hinsichtlich seiner eigenen Veränderung oder der Veränderung des erzeugten Modells abzuleiten respektive Strukturveränderungen und damit implizit Entitäts- und/oder Verhaltensänderungen des Realsystems in das Simulationsmodell zu integrieren [9].

2.2 Theoretische Lösungskonzeption

Das im Rahmen der Lösungskonzeption einer automatisierten Simulationsmodellgenerierung benötigte theoretische Grundkonzept basiert auf einer systemtheoretischen Betrachtung des Modellierungsprozesses. In Analogie dazu wird der Modellierungsprozeß als Systemtransformation auf der Grundlage eines Drei-Systeme-Modells verstanden, das aus einem modellierenden System S_M , einem zu modellierenden Bezugssystem S und einem automatisch

zu erstellenden Ersatzsystem $\tilde{S}$ als Zielsystem besteht. Bei der Umwandlung führt das modellierende System S_M unter der Verwendung von Wissen über das Bezugssystem S und das Zielsystem $\tilde{S}$ den Modellierungsvorgang durch. Das Zielsystem $\tilde{S}$ kann dabei die Systemrepräsentation in einer Simulationssprache oder einer anderen beliebigen und hinreichend mächtigen Repräsentationsform sein.

Die eigentliche Besonderheit des Transformationsprozesses liegt in dem Wissen, welches das modellierende System S_M über sein Bezugssystem S und sein korrespondierendes Zielsystem $\tilde{S}$ besitzt. Es umfaßt das problembereichsspezifische Wissen über die Verwendung, Zuordnung und Zusammenfassung der einzelnen Systemelemente des modellierenden Systems und ihren Referenzen auf diejenigen des Bezugs- und des Zielsystems. Dementsprechend beinhaltet es auch das Wissen eines menschlichen Modellierers über den betrachteten Problemereich und involviert mithin Wissen darüber, wie sich ein Element des modellierenden Systems in sein Bezugssystem einordnet (**Instanziierungswissen**) und integriert (**Aggregationswissen**). Ferner umfaßt es das Wissen hinsichtlich der Kommunikation eines Elementes mit seiner Systemumwelt (**Zuordnungswissen, Relationenwissen**) und der Erreichung einer bestimmten Modellhomogenität bei einer gegebenen Granularität. (**Granularitätswissen**). Mittels eines hier nicht näher spezifizierten, mehrstufigen Phasenmodells der Systemtransformation erfolgt die algorithmische Umsetzung des beschriebenen Transformationsprozesses [2]. Dieser Phasenprozeß basiert prinzipiell auf einer Inferenz der in einer gemeinsamen Wissensbasis gespeicherten Fakten, die aus unterschiedlichen produktionsbezogenen Informations- und Kommunikationssystemen gewonnen werden. Außerdem ist hierin klassenassoziiertes Wissen über die Elemente des Modellierungssystems enthalten. Dieses instanziierte Metamodell steht dann für eine abschließende Transformation in ein Simulationsmodell zur Verfügung.

3 Prototypische Realisierung der theoretischen Konzeption

Die prototypische Realisierung der theoretisch fundierten Lösungskonzeption erfolgt auf der Grundlage einer implementierungsbedingten, zweigeteilten Strukturierung des korrespondierenden Modellierungssystems S_M . Es ist einerseits in die Entwicklung des objektorientierten Laufzeitsystems **SimGen** und andererseits in die Entwicklung des Referenzklassensystems **MCL (Manufacturing Component Library)** eingeteilt [vgl. Abbildung 2]. Während das objektorientierte Laufzeitsystem SimGen die softwaretechnische Kernfunktionalität bereitstellt, um die Beschreibung von Elementen diskreter Simulationssysteme und deren Wissen vornehmen zu können, sind in der Sprache SimGen++ die eigentlichen Elemente des modellierenden Systems S_M und ihr spezifisches Wissen implementiert.

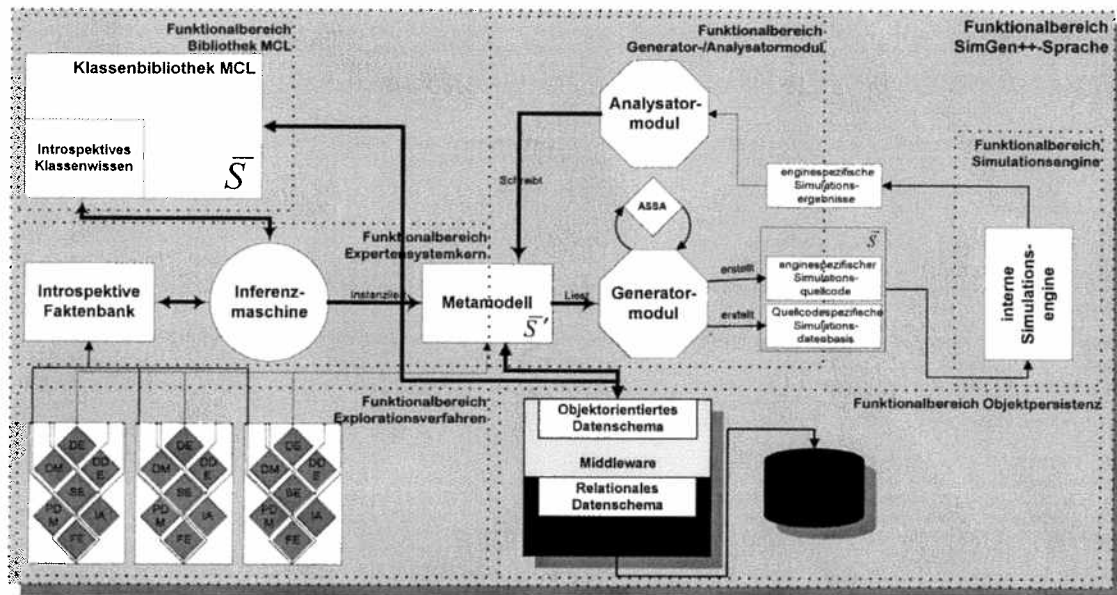


Abbildung 2: Funktionalbereichsorientierte Systemarchitektur

3.1 Entwicklung des Laufzeitsystems SimGen

Für die Entwicklung des Laufzeitsystems wurde eine sprachliche Repräsentationsform gewählt, die sich an die Programmiersprache C++ anlehnt, aber gleichzeitig als hybride Sprache die Verbindung zwischen einer objektorientierten Simulationssprache und einem integrierten Expertensystem herstellt. Sie ist im Rahmen des Modellgenerators speziell für die Anwendung der MCL konzipiert und darf nicht als allgemeine Programmiersprache der dritten Generation aufgefaßt werden. Während die prozeduralen Teile in einem Dialekt der Sprache C++ abgebildet sind, wird die Wissensrepräsentation durch die Implementierung von Frames und Produktionsregeln unterstützt. Somit weist die hybride Sprache SimGen folgende Spezifika auf:

- SimGen++ unterstützt die Integration von Wissensbasen in Klassen

Dieses Merkmal stellt sicher, daß keine globale Wissensbasis vorgehalten werden muß, in welcher die Fakten und Regeln zentral abgelegt werden. Es wird vielmehr der objektorientierte Ansatz der Simulationssprache in den Teil des wissensbasierten Systems überführt [3].

- SimGen++ unterstützt eine simulationsspezifische, diskrete Zeitsteuerung und multiple, quasi-parallele Prozesse

Die Möglichkeit der Definition aktiver Klassen, die einen eigenen Ausführungsthread besitzen, ermöglicht es, parallele Prozesse zu modellieren und diese interagieren zu lassen und/oder ihnen ein Zeitverhalten mitzugeben.

3.2 Entwicklung des MCL-Referenzklassensystems

Die implizite Entwicklung eines Referenzklassensystems für Produktionsprozesse einer werkstattorientierten Fertigungsorganisation erfolgt auf der Grundlage einer objektorientierten Analyse des Diskursbereichs und der objektorientierten Beschreibungsmöglichkeit von ereignisdiskreten Systemen in SimGen++. Ausgehend von der Produktionsprozeßbasisklasse wird eine Differenzierung in die Klasse der realen Objekte, deren Instanzen ein direktes physisches Korrelat im Produktionsprozeß besitzen, in die Klasse der informationellen Objekte, deren Instanzen für eine Aggregation der Instanzen der Klasse der realen Objekte zur Abbildung von Steuer- und Informationsflüssen zwischen diesen benötigt werden, und in die Klasse der wissensbasierten Objekte vorgenommen, deren Aufgabe in der Abbildung von informationellem Wissen nicht-deklarativer und nicht-prozeduraler Art besteht. Durch die umfangreichen Referenzklassen der Produktionsprozeßbasisklasse, die für eine Instanziierung zur Verfügung stehen, kann die angestrebte Strukturgleichheit zwischen dem Real- und dem abstrakten Modellsystem hergestellt werden. Damit ist gewährleistet, daß jederzeit eine Identifikation und Zuordnung der abstrakten Modellelemente zu den korrespondierenden realen oder informationellen Objekten stattfinden kann, um dadurch einerseits die Gesamttransparenz des Modells erhalten und andererseits dessen Validität überprüfen zu können.

4 Zusammenfassung

Bei der Entwicklung eines Systems zur automatischen Generierung von Simulationsmodellen sind sowohl die menschliche Abstraktionsfähigkeit als auch das Wissen der Modellierer über den Problembereich innerhalb eines technischen Systems nachzubilden. Während die automatisierte Parametrisierung vorhandener Grundmodelle, deren Parameterräume durch den Menschen bei ihrer Erstellung bereits festgelegt wurden, schon computerunterstützt verfügbar ist, wurde der Problembereich der Entwicklung flexibler Modellgeneratoren in der Wissenschaft bislang nur unzureichend behandelt. Dies liegt einerseits an der monolithischen Struktur von verfügbaren Simulationssystemen, die eine Umsetzung objektorientierter Weltansichten erheblich erschweren und andererseits vor allem an der komplexen Integration von wissensbasierten Systemen und deren spezifischen Repräsentationsformen mit klassischen Simulationssystemen. Eine allgemeingültige Lösung kann nur dadurch geschaffen werden, daß diese tendenziell konfliktären Bereiche transparent integriert werden, um sowohl einen Informations- und Systembruch zu vermeiden als auch die Wartbarkeit eines solchen komplexen Systems zu ermöglichen.

Das SimGen-System mit der Modellierungssprache SimGen++ stellt indes eine Möglichkeit dar, die skizzierten Restriktionen durch die zugrundeliegende objektorientierte Weltansicht und

die Integration wissensverarbeitender Funktionalitäten in einem System zu überwinden und den Einsatz simulativer Techniken im Bereich der kurz- und mittelfristigen Werkstattsteuerung vermehrt zu etablieren.

Literatur

- [1] Burger, C.: Produktionsregelung mit entscheidungsunterstützenden Informationssystemen, Berlin/Heidelberg/New York 1992
- [2] Gmilkowsky, P./Eckardt, F./Palleduhn, D.: Concept of an Integrated System for Modeling, Simulation and Analysis of Discrete Production Processes, in: Obaidat, M. S./Illgen, J. (Eds.): The Proceedings of the 1997 Summer Computer Simulation Conference - Simulation and Modeling Technology for the Twenty-First Century, July 13 - 17, Arlington/Virginia (USA) 1997, S. 329 - 334
- [3] Gmilkowsky, P./Eckardt, F./Palleduhn, D.: Einsatzmöglichkeiten wissensbasierter Technologien im Rahmen eines integrierten Systems zur Modellierung, Simulation und Analyse von diskreten Produktionsprozessen, Symposium über Operations Research 1997 (SOR 97), Jahrestagung der Deutschen Gesellschaft für Operations Research (DGOR) und der Gesellschaft für Mathematik, Ökonomie und Operations Research (GMÖOR), Friedrich-Schiller-Universität Jena, 3. - 5. September 1997, Jena 1997, S. 161 - 162
- [4] Känel, S. v.: Kybernetik: eine Einführung für Ökonomen, Berlin 1990
- [5] Kernler, H.: PPS der 3. Generation: Grundlagen, Methoden, Anregungen, 3. überarbeitete Auflage, Heidelberg 1995
- [6] Mattern, F./Mehl, H.: Diskrete Simulation – Prinzipien und Probleme der Effizienzsteigerung durch Parallelisierung, in: Informatik-Spektrum 12:4 1989, S. 198-210
- [7] Palleduhn, D. U.: Unterstützung einer kosten- und leistungsorientierten Fertigungssteuerung durch wissensbasierte Simulation, Dissertation, Ilmenau 1998
- [8] Weck, M.: Simulation in CIM, Berlin/Heidelberg 1991
- [9] Zeigler, B. P./Prähofer, H.: Systems Theory Challenges in the Simulation of Variable Structure and Intelligent Systems, in: Pichler, F./Moreno-Diaz, R. (Eds.): Computer Aided Systems Theory – EUROCAST '89, New York (USA) 1989, S. 41 - 51

Prognose durch Verknüpfung von Fuzzy Inductive Reasoning mit einem kontinuierlichen Modell

E. Möbus, R. Hohmann

{moebus,hohmann}@cs.uni-magdeburg.de

Universität Magdeburg, Institut für Simulation und Graphik, Postfach, 39016 Magdeburg

Zusammenfassung

Für die Prognose mit Simulationsmodellen (Decision Modelling) benötigt man auch die exogenen Größen im prädizierten Zeitabschnitt. Folglich müssen Verfahren zur Ermittlung solcher Größen bereitgestellt werden. Eine Methode ist das Fuzzy Inductive Reasoning (FIR), das anhand des Tools SAPS-II vorgestellt wird. Am Beispiel einer Kläranlage wird gezeigt, wie durch die Verknüpfung von FIR mit einem kontinuierlichen Modell eine prädiktive Regelung (Predictive Control) ermöglicht wird. Anhand eines Simulationsmodells von Ausgleichsbecken und prädiktiver Zulaufregelung einer Kläranlage wird das FIR-Verfahren mit anderen Prognosemethoden verglichen.

1 Einleitung

Die bei der Simulation technischer Prozesse und Anlagen benutzten Modelle sind oft sehr komplex - meist sind es umfangreiche Differentialgleichungssysteme - und erfordern eine hohe Rechenleistung. Um diese Komplexität nicht noch zu vergrößern, werden die Beziehungen des simulierten Systems zur Umwelt oft nur eingeschränkt berücksichtigt. Bei einer "rückschauenden" Simulation, also der Reproduktion beobachteter Prozesse mittels Simulation, stehen die tatsächlich beobachteten exogenen Einflußgrößen als Zeitfunktionen für die Simulation zur Verfügung, so daß die abgelaufenen Prozesse im allgemeinen sehr gut nachgebildet werden. Eine Möglichkeit, eine prognostische Simulation zu ermöglichen, besteht darin, das Simulationsmodell mit Prognoseverfahren für die exogenen Größen zu verknüpfen. Ein solches Prognoseverfahren stellt das Inductive Reasoning dar.

2 Inductive Reasoning am Beispiel von SAPS-II

2.1 Inductive Reasoning - ein qualitatives Simulationsverfahren

In der Simulation werden meist quantitative Simulationsverfahren eingesetzt, die durch genaue Zahlenwerte (Eingangs- und Simulationswerte) und Zeitpunkte gekennzeichnet sind. Die Eingangsgrößen müssen also exakt vorliegen, die Differentialgleichungen des Modells sind streng parametrisiert und die Simulation liefert ein (dem Modell entsprechendes) numerisches Ergebnis. Bei den qualitativen Simulationsverfahren wird auf diese exakten Größen verzichtet; es genügen die Größenordnungen (oder Qualitäten) der Werte. Im Inductive Reasoning

werden Werteintervalle als Qualitäten verwendet. Auf eine Modellbildung in der bei kontinuierlicher Simulation üblichen Form von Differentialgleichungen wird verzichtet; vielmehr versucht man, im beobachteten System ein Verhaltensmuster zu erkennen. Hierdurch werden Rückschlüsse aus dem beobachteten auf das zu erwartende Verhalten des Systems ermöglicht.

Für die Simulation eröffnen sich einige Vorteile: Die Bestimmung eines komplexen Differentialgleichungssystems ist nicht nötig. Das Finden eines Verhaltensmusters ist zudem im allgemeinen eine einfachere Aufgabe als die Strukturanalyse eines Modells. Und nicht zuletzt kann ein Mustererkennungsalgorithmus deutlich schneller sein, als ein "berechnender" Algorithmus, was oft einen benötigten Zeitgewinn bedeutet. Diese Vorteile werden durch Nachteile gegenüber Strukturmodellen erkauft: Die Verhaltenskomplexität ist viel größer als die Komplexität einer Struktur. Es müssen also wesentlich mehr Informationen über ein System verfügbar sein, wenn sein Verhalten erkennbar sein soll. Für eine allgemeine Systemanalyse sind diese Verfahren ungeeignet: Durch Verhaltensmustererkennung kann nur bestimmt werden, *wie* sich ein System verhält, *warum* es sich jedoch so verhält, ist nicht ablesbar.

2.2 Das Verfahren am Beispiel des Tools SAPS-II

Die Technik des Inductive Reasoning wurde in den siebziger Jahren entwickelt [1]. Eine erste Softwareimplementation dieses Simulationsansatzes war der System Approach Problem Solver - SAPS [2]. Als eine Erweiterung dieser Implementation wurde SAPS-II entwickelt [3], welches als MATLAB-Toolbox oder als Bibliothek für CTRL-C verfügbar ist.

2.2.1 Recoding

Wie in allen qualitativen Verfahren werden die kontinuierlichen Eingangsdaten im Inductive Reasoning diskretisiert. Im Unterschied zu anderen Verfahren ist die Anzahl der Bereiche in SAPS weder eingeschränkt noch vordefiniert. Jede dieser Regionen (oder *Level*) repräsentiert z.B. "kleine", "mittlere" oder "große" Werte. In der Praxis hat sich eine Zahl von fünf Levels bewährt [4], wobei diese in der Regel etwa die gleiche Datenmenge umfassen.

Die Diskretisierung der Werte in die Level wird *Recoding* genannt; eine Kombination von möglichen Levels aller Variablen heißt *State*. Ein *Episodical Behavior* ist eine zeitliche Abfolge von States, also das qualitative Gegenstück zum quantitativen Systemverhalten.

Um trotz der starren Levelgrenzen einen eher fließenden Übergang zwischen den Werten in benachbarten Levels zu ermöglichen, wurde Mitte der achtziger Jahre die Einführung eines Fuzzymaßes vorgeschlagen [5], die das "klassische" Inductive Reasoning zum Fuzzy Inductive Reasoning (FIR) erweitert. Zudem ermöglicht die Einführung der Fuzzy-Membership-Werte eine Umkehrung des Recoding. Die Membershipfunktion ist dabei so definiert, daß sie in der Mitte eines Levels einen Wert von 1 und an den Levelgrenzen einen Wert von 0.5 hat.

2.2.2 Masken und Verhaltensprädiktion

Das Episodical Behavior wird in einer Matrix festgehalten, in der jede Spalte eine der beobachteten Variablen und jede Zeile einen bestimmten Zeitpunkt repräsentiert. Für die Variablen in der Matrix sollen nun möglichst deterministische finite Automaten, also qualitative Funktionen, gefunden werden. Eine solche Funktion kann z.B. folgendermaßen aussehen:

$$a_1(t) = \varphi(a_2(t-2 \cdot \delta t), e_1(t-\delta t), a_1(t-\delta t), e_2(t))$$

Diese Funktion wird in einer Matrix, der sogenannten *Mask* bzw. *Maske*, repräsentiert:

$$\begin{array}{c} T \\ t - 2 \cdot \delta t \\ t - \delta t \\ t \end{array} \begin{pmatrix} e_1 & e_2 & a_1 & a_2 \\ 0 & 0 & 0 & -1 \\ -2 & 0 & -3 & 0 \\ 0 & -4 & +1 & 0 \end{pmatrix}$$

Die negativen Werte der Maske bezeichnen die Eingangsgrößen der Funktion, der positive Wert die Ausgangsgröße. Die Reihenfolge der Numerierung der Eingangsgrößen ist beliebig. Um ein optimales Inductive-Reasoning-Modell, d.h. die optimale Maske für ein gegebenes Episodical Behavior, zu finden, muß aus allen in einem gegebenen Rahmen möglichen Masken durch umfassende Suche jene ermittelt werden, für die die zugehörige *State Transition Matrix*, in der zu jeder Kombination von Eingangswerten die Wahrscheinlichkeit der möglichen Ausgangswerte abgelegt ist, die größte Determiniertheit besitzt. Der Rahmen möglicher Masken wird durch die sogenannte *Mask Candidate Matrix* vorgegeben:

$$\begin{array}{c} T \\ t - 2 \cdot \delta t \\ t - \delta t \\ t \end{array} \begin{pmatrix} e_1 & e_2 & a_1 & a_2 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & +1 & 0 \end{pmatrix}$$

Die Matrix enthält den Wert '-1', wo die Maske einen möglichen Eingangswert hat, der Ausgangswert ist mit '+1' gekennzeichnet. Die Elemente, die weder Ausgang sind noch Eingang werden können, sind auf '0' gesetzt.

Jede mögliche Maske führt bei einem gegebenen Episodical Behavior zu einer anderen State Transition Matrix, deren Determiniertheit aufgrund der Cross-Entropie als Gütemaß der Maske bestimmt wird. Als weiteres Gütekriterium wird eine *Observation Ratio* definiert, wodurch States mit geringer Beobachtungshäufigkeit schwächer gewichtet werden.

Mit der optimalen Maske kann nun die qualitative Simulation ausgeführt, d.h. ein zukünftiges Verhalten des Systems prädiziert werden: Für die Eingangszustände im Ein-/Ausgabemodell werden neue Ausgangswerte bestimmt, indem man aus der State Transition Matrix den für diese Kombination der Eingangswerte wahrscheinlichsten Ausgangswert ermittelt.

2.2.3 Maskenbestimmung und Prädiktion bei Nutzung der Fuzzy-Membershipfunktion

Zur Ermittlung der optimalen Maske kann der Fuzzy-Membershipwert benutzt werden. Dieser gibt an, wie verlässlich eine Variable zum jeweiligen Level gehört. Beim FIR wird zusätzlich zur reinen Beobachtungswahrscheinlichkeit eines Ausgangswertes bei einer Eingangswertekombination dieses Konfidenzmaß zur Ermittlung der State Transition Matrix benutzt.

Das Prädiktionsverfahren unterscheidet sich etwas vom ursprünglichen Algorithmus. Es wird nicht einfach der Ausgangswert mit der höchsten Verlässlichkeit gewählt, sondern der Outputwert der Eingangswerte, die dem aktuellen Input am nächsten liegen. Der Membershipwert für diesen Ausgang ist das distanzgewichtete Mittel der fünf Ausgangsmembershipwerte, deren Eingangsvektoren den geringsten Abstand zum aktuellen Inputvektor haben.

3 Modell eines prädiktiven Reglers in der Zulaufsteuerung einer Kläranlage

An die Reinigungsleistung von Abwasseranlagen werden hohe Anforderungen gestellt, was zu komplexen Steuerungs- und Regelsystemen führt. Von besonderem Interesse sind dabei prädiktive Regler, die aufgrund einer Systemverhaltensprognose regulierend in den laufenden Prozeß eingreifen können. In diesem Beitrag wurde der Zulaufbereichs einer Kläranlage mit der Modellbibliothek SIMBA[6] simuliert. Die Abwasserzulaufmenge ist recht ungleichmäßig über den Tag verteilt. Um eine gute Klärleistung zu erzielen, wird eine möglichst gleichmäßige Zuflußmenge angestrebt, weshalb man Ausgleichsbecken einsetzt. Um diese Glättung zu optimieren, kann ein prädiktiver Regler zur Steuerung des Füllstandes im Ausgleichsbecken eingesetzt werden. Für die nachstehenden Untersuchungen wurde ein solcher auf einem SIMBA-Modell beruhender Regler [7] benutzt.

3.1 Prädiktive Regelung

Klassische Regler vergleichen den Istzustand mit dem Sollzustand einer Anlage, um gegebenenfalls eine Stellgröße zu verändern. Bei den modellbasierten prädiktiven Reglern (Model-Based Predictive Control - MBPC) wird anhand eines Simulationsmodells der erwartete Verlauf der Störgröße ermittelt, so daß durch rechtzeitige Änderung der Stellgröße ein kritischer Zustand der Anlage vermieden werden kann. Welche Form das Prädiktionsmodell hat, ist für die Arbeit des Reglers unerheblich. Meist werden Differentialgleichungsmodelle verwendet, jedoch auch alternative Ansätze, insbesondere künstliche neuronale Netze. In dieser Arbeit wurde neben einem Fourier-Zeitreihenmodell und einer linearen Regression ein Fuzzy-Inductive-Reasoning-Modell für die Prädiktion genutzt. Für alle Ansätze sind dabei Vor- und Nachteile abzuwägen. So besitzen beispielsweise die neueren Ansätze meist erhebliche Vorteile gegenüber den Differentialgleichungsmodellen beim Modellerstellungsaufwand, jedoch ist u.U. ihre Extrapolierbarkeit unsicher. Letztlich hängt es meist von der jeweiligen Anwendung ab, welchen Modellansatz man bevorzugt.

3.2 SIMBA-Modell eines prädiktiven Reglers zur Zulaufmengenregulierung in einer Kläranlage

Die Regulierung des Zulauf einer Kläranlage durch Einsatz von Ausgleichsbecken ist für deren Reinigungsleistung notwendig. Im Bild 1 ist ein Simulationsmodell eines solchen Ausgleichbeckens – der Regelstrecke – dargestellt.

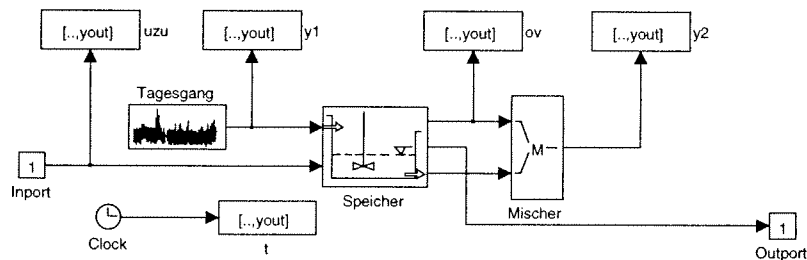


Bild 1. SIMBA-Modell für ein Ausgleichsbecken im Zufluß eines Klärwerks

Der Zulauf zur Kläranlage wird im Block Tagesgang beschrieben. Das Ausgleichsbecken wurde durch den Block Speicher realisiert. Der zweite Eingang dieses Blocks ist der Sollwert des aus dem Ausgleichsbecken in die Kläranlage zu pumpenden Volumenstromes, also die Stellgröße des Reglers. Der tatsächlich gepumpte Volumenstrom (Speicher, unterer Ausgang) sowie ein eventueller Überlauf (Speicher, oberer Ausgang) stellen den Zufluß in die Kläranlage dar. Als Regelgröße für den Zuflußregler wird der Füllstand des Speicherbeckens (Speicher, mittlerer Ausgang) gemessen und zum Reglermodell zurückgeführt.

Für eine prädiktive Regelung wird ein Modell dieser Regelstrecke benötigt (Bild 2). Der modellierte Zulauf, aus einer Datei "values.mat" ausgelesen, ist für das Streckenmodell also eine "Black-Box". So kann ohne eine Änderung des Modells der Regelstrecke das Zulaufmodell ausgetauscht werden, indem jeweils eine Datei mit den prognostizierten Zuflußmengen als Eingang übergeben wird.

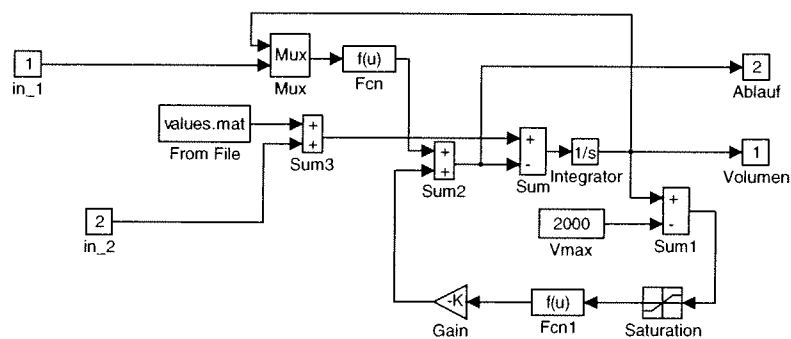


Bild 2. SIMULINK-Modell der Regelstrecke mit modelliertem Zulauf

Im Integrator-Block wird die Summe der zu- bzw. abfließenden Volumenströme integriert, so daß sein Ausgang den Füllstand des Ausgleichbeckens darstellt. Über die Blöcke Sum1,

Saturation, Fcn1 und Gain wird ein eventueller Überlauf (maximaler Füllstand des Ausgleichsbeckens $V_{\max} = 2000 \text{ m}^3$) berechnet und mittels der Blöcke Mux und Fcn der Ablauf aus dem Ausgleichsbecken begrenzt, wenn dieses nahezu leer ist. Der Eingang in_1 des Streckenmodells erhält den Sollwert des vom Ausgleichsbecken in die Kläranlage zu pumpenden Volumenstromes als Stellgröße, im zusätzlichen Eingang in_2 kann man dem Modell ein Signal zur Korrektur des Zulaufmodells übergeben.

Mit der Gütefunktion, die zur Ermittlung einer optimalen Stellgrößenfolge dient, werden Füllstände nahe des Überlaufs bzw. Leerlaufs sowie Stellbewegungen, also Änderungen des gepumpten Volumenstroms, stark bestraft, wodurch der angestrebte, möglichst gleichmäßige Zulauf zur Kläranlage erreicht wird.

4 Modelle zur Prädiktion der Eingangsdaten für ein kontinuierliche Modell

Aus den Zulaufdaten wurden mit verschiedenen Prognoseverfahren Verhaltensprädiktionen getroffen: Zu jedem Zeitpunkt des vorhandenen Datensatzes wurde aus dem bis dahin beobachteten Zulauf das Zuflußverhalten der nächsten zwölf Stunden prognostiziert. Diese prädizierten Daten wurden dann als Eingang für den Regler benutzt. Je besser also die Zuflußprognose war, um so besser wird das Regelverhalten sein. Im Idealfall wäre der vom Becken in die Kläranlage fließende Volumenstrom konstant. Die für die Zulaufprädiktion benutzten Modelle sind nachfolgend beschrieben. Ein Vergleich der Ergebnisse erfolgt im abschließenden Abschnitt.

4.1 Fuzzy Inductive Reasoning

Beim Fuzzy Inductive Reasoning besteht die Modellbildung lediglich im Festlegen und Recoding der Eingangsgrößen sowie in der Vorgabe einer Mask Candidate Matrix. Neben dem Volumenstrom wurde der aktuelle Wochentag als zweite Eingangsgröße berücksichtigt. Dies war notwendig, da sich der mittlere Zufluß zur Kläranlage während des Wochenendes vom Zulauf während der normalen Werktage unterscheidet. Die Bestimmung einer Maske, die eine gesamte Woche überdeckt, würde eine zu hohe Rechenzeit erfordern. Die Maskengröße war so konzipiert, daß für die Prädiktion der aktuelle Wochentag sowie der Zulauf der letzten 24 Stunden für die Eingabe berücksichtigt werden konnten. Die Mask Candidate Matrix besaß also 24 Zeilen und zwei Spalten. Die Wochentage waren dabei in sieben Level recodet, die Zulaufvolumina in fünf mit jeweils etwa der gleichen Anzahl von Werten.

4.2 Lineare Regression

Eine recht einfache, schnelle Methode, Daten zu prognostizieren, ist eine Extrapolation aus Meßwerten der Vergangenheit mittels linearer Regression. Obwohl eine solche Prognose bei periodischen Funktionen mit kurzer Periodenlänge relativ ungeeignet ist, wurde dieses Ver-

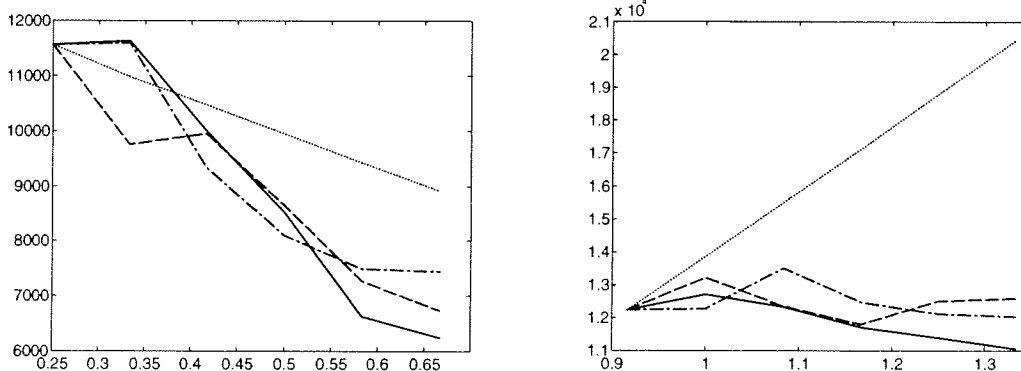
fahren hier zum Vergleich angewandt. Von den 12 Meßwerten eines Tages wurden wegen dieser kurzen Periode nur fünf in die Regressionsanalyse einbezogen. Diese Prognose lieferte kaum verlässliche Werte, entsprechend schlecht ist die erreichte Regelung.

4.3 Zeitreihenanalyse mittels Fourierreihen

In der klassische Zeitreihenanalyse wird die der Reihe zugrundeliegende Funktion aus einer Trend- und einer Saisonkomponente zusammengesetzt. Bei den hier verwendeten Meßdaten war zu erkennen, daß der mittlere beobachtete Wert längerfristig auf etwa gleichem Niveau blieb. So reduzierte sich die Zeitreihenanalyse auf die Fourieranalyse. Als Periodenlänge wurden zwei verschiedene Möglichkeiten in Betracht gezogen: Eine Periodenlänge von einem Tag und eine Periodenlänge von einer Woche. Bei der Analyse wurden jeweils die vom aktuellen Zeitpunkt letzten drei Perioden betrachtet und die Fourierkoeffizienten a_k und b_k bis $k = 28$ ermittelt. Mit dieser Fourierreihe ließen sich dann die benötigten Prognosen treffen.

5 Vergleich der Ergebnisse bei den verwendeten Prognoseverfahren

Für jeden Stellzeitpunkt wurde eine Verlaufsprognose für den nächsten halben Tag getroffen. In den Bildern 3 und 4 sind die mit den verwendeten Verfahren erzielten Ergebnisse zu zwei verschiedenen Zeitpunkten einander gegenübergestellt (Vollinie: tatsächlicher Verlauf; Strichpunktlinie: Fuzzy Inductive Reasoning; gestrichelt: Zeitreihenanalyse mit Fourierreihen; gepunktet: lineare Regression).



Bilder 3 und 4. Tatsächliche und prognostizierte Zulaufvolumenströme

Mit den verschiedenen Methoden konnten unterschiedlich gute Regelungsleistungen erzielt werden. Um für die graphische Gegenüberstellung einen weiteren Vergleichsmaßstab zu haben, wurde das bei "perfekter Prädiktion" erreichbare Ergebnis ermittelt. So ist erkennbar, ob Abweichungen unvermeidbar sind, oder ob ein ideales Ergebnis bei exakter Prognose erreichbar ist. Eine "perfekte Prognose" der Zulaufwerte wurde mit den realen Zulaufdaten erzielt. Damit erhält man den Meßwertverlauf in Bild 5.

Neben der graphischen Gegenüberstellung in den Bildern 5 bis 9 läßt sich die $\mathcal{L}_2$ -Norm der Differenz der Abflußmenge bei Verwendung des jeweiligen Prädiktionsverfahrens und der Abflußmenge bei exakter Prognose als Gütemaß verwenden:

Zulaufprognoseverfahren	$\mathcal{L}_2$ -Norm der Abflußmengendifferenz
Fuzzy Inductive Reasoning	237,8
Lineare Regression	2303,2
Fourieranalyse, ein Tag	130,3
Fourieranalyse, eine Woche	67,0

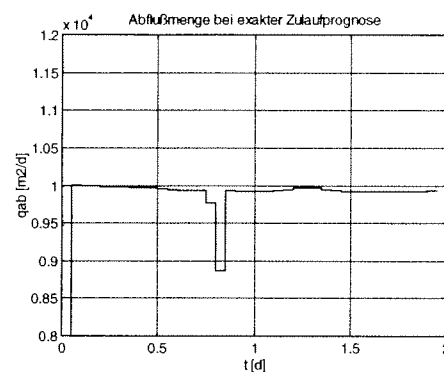


Bild 5. Abflußmenge bei exakter "Prognose"

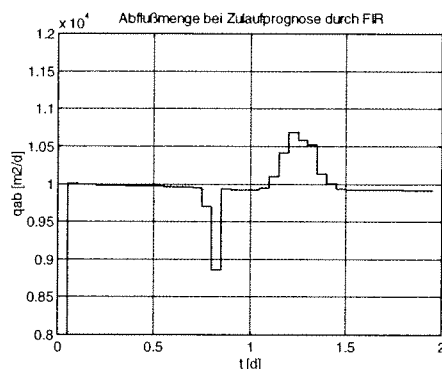


Bild 6. Abflußmenge bei Prognose mittels Fuzzy Inductive Reasoning

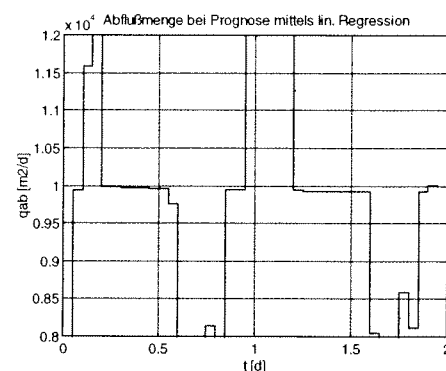


Bild 7. Abflußmenge bei Prognose mittels linearer Regression

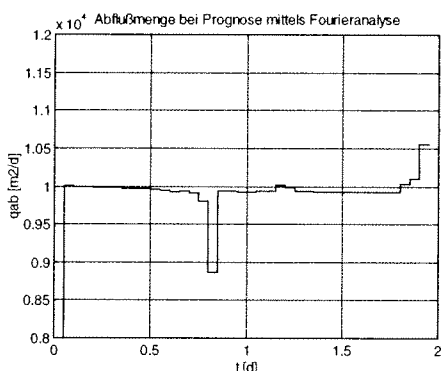


Bild 8. Abflußmenge bei Prognose mittels Fourieranalyse, ein Tag

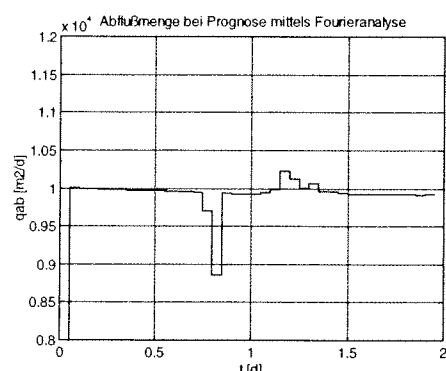


Bild 9. Abflußmenge bei Prognose mittels Fourieranalyse, eine Woche

Es ist also deutlich erkennbar, daß in diesem konkreten Anwendungsfall die Fourieranalyse die besten Resultate liefert. Dies war jedoch zu erwarten, da der Zuflußvolumenstrom eine nahezu periodische Funktion darstellt, und diese durch Fourierreihen fast ideal angenähert werden kann. Die Nichteignung der linearen Regression in diesem Fall war schon festgestellt worden; jedoch hat diese Methode, wie auch die Regression durch quadratische oder kubische Polynome, in Anwendungen mit relativ glatter Eingangsfunktion durchaus ihre Berechtigung, zumal sie sich durch ihre geringe Rechenzeit auszeichnet. Die Prognose mittels Fuzzy Inductive Reasoning liefert gute Ergebnisse. Sie besitzt nicht ganz die Qualität der Fourierreihen, wenn sich die Eingangsfunktion wie hier durch ausgeprägte Periodizität auszeichnet. Es ist anzunehmen, daß bei Eingangsfunktionen, die diese Eigenschaft nicht besitzen, die Prognose mittels FIR im Vergleich an Güte gewinnt. Außerdem zeichnet sich diese Methode dadurch aus, daß sie in Anwendungsfällen, die mehrere Eingangsgrößen besitzen, ebenfalls angewandt werden kann, während die Fourieranalyse auf eine Eingangsgröße beschränkt ist.

Literatur

- [1] Klir, G.J.: *Architecture of Systems Problem Solving*, Plenum Press New York, 1985
- [2] Uyttenhove, H.J.: *SAPS - System Approach Problem Solver*, PhD dissertation, State University of New York at Binghamton, 1979
- [3] Cellier, F.E., Yandell, D.: SAPS-II. A New Implementation of the Systems Approach Problem Solver, *International Journal of General Systems*, 13(4), 1987, S. 307-322
- [4] Cellier, F.E.: *Continuous Systems Modeling*, Springer New York/Berlin/Heidelberg, 1991
- [5] Zadeh, L.A.: Syllogistic Reasoning in Fuzzy Logic and its Application to Usuality and Reasoning with Depositions, *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15(6), 1985, S. 754-763
- [6] Alex, J., Tschepezki, R.: *SIMBA 3.0⁺. Simulation der biologischen Abwasserreinigung*, Benutzerhandbuch, ifak Barleben, 1995
- [7] Alex, J., Tschepezki, R.: *Modellanwendungen zur Regelung von Belebungsanlagen*, ifak Barleben, 1997
- [8] Möbus, E.: *Prognose durch Verknüpfung von Inductive Reasoning mit einem kontinuierlichen Modell*, Diplomarbeit am Institut für Simulation und Graphik der Otto-von-Guericke-Universität Magdeburg, 1997

Simulation eines embedded Systems

W. Kastner

k@auto.tuwien.ac.at

Technische Universität Wien, Treitlstraße 1, A-1040 Wien

Zusammenfassung

Aufgabe der Laborübung *Prozeßautomatisierung*, die für Absolventen der Studienrichtung Informatik an der Technischen Universität Wien angeboten wird, ist es, einige Teilgebiete des in der zugehörigen Vorlesung vorgestellten Stoffes durch praktische Übungen zu vertiefen. Den Studenten sollen dabei Grundlagen des Rechnereinsatzes in der Automatisierungstechnik, vor allem die Themenbereiche der hardwarenahen Programmierung von Prozeßperipherie-Komponenten, Multitasking unter Echtzeitbetriebssystemen sowie gängige Sensoren und Aktuatoren vorgestellt bzw. nähergebracht werden. Die Lehrveranstaltung wird pro Jahr von ca. 200 Informatikern besucht. Die hohe Teilnehmerzahl und das stark unterschiedliche technische Wissen der Studenten macht die Konzeption einer qualitativ hochwertigen Laborübung sehr schwierig. Das unter Berücksichtigung dieser Randbedingungen ausgearbeitete Übungskonzept ist auf die Programmierung eines klassischen *embedded Systems*, also auf die Programmierung von Prozeßperipherie-Elementen eines Targetsystems abgestimmt. Um auch die Debugging-Phase kurz zu halten und potentielle Gefahren an technischen Prozessen auszuschließen, wurde ein Targetsimulator konzipiert und implementiert. Der Simulator wird an die Studenten weitergegeben und erlaubt ein transparentes Entwickeln von Programmen, die letztlich auf einem Targetsystem exekutierbar sind und einzelne (Modell-)Prozesse ansteuern. Somit ist es für die Studenten möglich einen großen Teil der Laborübung zu Hause vorzubereiten und erst letzte Abnahmetests an der realen Hardware durchzuführen.

1 Einleitung

Aufgabe der Laborübung *Prozeßautomatisierung*, die begleitend zur gleichnamigen Vorlesung angeboten wird, ist es, einige Teilgebiete des dort vorgestellten Stoffes (siehe, [6]) durch praktische Übungen (siehe, [5]) zu vertiefen. Unter anderem sollen die Absolventen dieser Lehrveranstaltung einige Grundlagen der Themenbereiche

- Hardware-nahe Programmierung von Peripherie-Komponenten wie z.B. digitalen und analogen Interfaces,
- Multitasking unter dem Betriebssystem-Kernel pSOS⁺ und
- Sensor- und Aktuator-Systeme

kennenlernen. Dazu werden reale technische (Modell-)Prozesse zur Verfügung gestellt, die mittels moderner Sensor/Aktuator-Systeme und Prozeß-Interfaces an ein spezielles Mikroprozessorsystem (das *Targetsystem*) angekoppelt sind. Mit Hilfe einer Cross-Entwicklungsumgebung haben die Studenten die Möglichkeit, Software für Automatisierungssysteme

in einer realistischen Umgebung zu entwickeln und zu testen. Das stark unterschiedliche technische Wissen der Studenten macht die Konzeption einer qualitativ hochwertigen Laborübung sehr schwierig (siehe, [6]). Verschärft wird die Situation dadurch, daß die Lehrveranstaltung pro Jahr von ca. 200 Informatikern besucht wird. Die Übungsräumlichkeiten umfassen daher mehrere identische Entwicklungszellen bestehend aus Modell-Prozessen, Targetsystem samt Prozeßperipherie und einigen Workstations. Alle Rechner sind durch ein Netzwerk untereinander und mit einigen File- und Computing-Servern verbunden. Auf diese Weise können alle Schritte der Software-Entwicklung - außer dem Debugging-Vorgang! - gleichzeitig und auf jeder beliebigen Workstation durchgeführt werden.

Um auch die Debugging-Phase kurz zu halten sowie potentielle Gefahren an den Modellen auszuschließen - schließlich sind die ersten Programmerversuche in Hardware-naher Umgebung nur in den seltensten Fällen frei von Fehlern -, wurde ein Targetsimulator konzipiert und implementiert. Der Simulator, in der momentanen Version lauffähig unter den gängigen Betriebssystemen Windows95 und WindowsNT, wird an die Studenten weitergegeben und erlaubt ein transparentes Entwickeln von Programmen, die letztlich auf dem Target-Rechensystem exekutierbar sind und die Modell-Prozesse ansteuern. Auf dem Simulator können somit die Programme von den Studenten entwickelt und vorweg „probiert“ werden, bevor sie auf das Targetsystem geladen und dort nochmals an der richtigen Hardware, also den eigentlichen technischen Prozessen, getestet werden. Somit ist es für die Studenten möglich einen großen Teil der Laborübung zu Hause vorzubereiten und erst letzte Abnahmetests an der realen Hardware durchzuführen.

Im ersten Teil dieses Artikels werden - um den Leser einen Einblick in die realisierte Simulationsumgebung zu geben - die realen Modelle und das zu programmierende Targetsystem kurz vorgestellt. Danach wird der Simulator, der objektorientiert entworfen und entwickelt wurde, und seine interessantesten Implementierungsdetails diskutiert und gezeigt, wie einfach seine Verwendung ist.

2 Targetsystem

Bei der Laborübung *Prozeßautomatisierung* werden embedded Systems auf der Basis von Motorola M68030 Mikroprozessoren eingesetzt. Die Systeme sind aus VMEbus Standard-Komponenten aufgebaut. Die Ankopplung der technischen Prozesse erfolgt mittels eines standardisierten Interface-Bussystems, genannt *M-Modul* (siehe, [4]). Mittlerweile bieten sehr viele Hersteller eine breite Palette von M-Modulen an, die von einfachen digitalen und analogen Ein- und Ausgängen, über CAN oder Profibus-Interfaces bis hin zu Fuzzy-Prozessoren reichen. Die weitere Kopplung an das embedded System wird unter Zuhilfenahme der Carrier-Boards, die maximal vier M-Module aufnehmen können, erreicht.

Die Programmierung der Targetsysteme, die unter dem Echtzeit-Betriebssystem pSOS⁺ laufen, erfolgt mittels Cross-Entwicklungsumgebung in der Programmiersprache C. Als Entwicklungsrechner werden pro Targetsystem je vier durch Ethernet eingebundene PCs unter DOS eingesetzt; ergänzt durch einen Drucker sei eine solche Einheit als *Entwicklungszelle* bezeichnet (Bild 1).

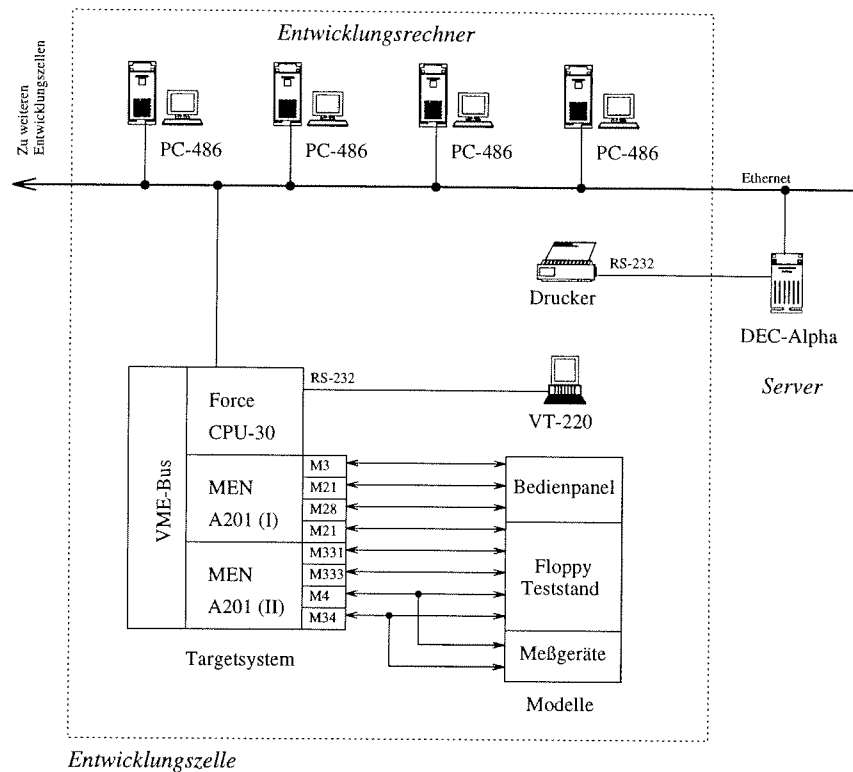


Bild 1: Entwicklungszelle

Die Kopplung zwischen den PCs und dem Targetsysteem erfolgt ebenfalls über Ethernet. Eine tatsächliche Verbindung ist nur während des Programmtests (Debugging) nötig, nicht aber bei der übrigen Entwicklungsarbeit. Man beachte, daß ein Targetsysteem zur gleichen Zeit nur von *einem* User (also von einem PC aus) verwendet werden kann; simultane Aufrufe des Debuggers von unterschiedlichen Entwicklungsrechnern aus werden in einer Warteschlange (durch einen Lock-Manager realisiert) gereiht.

Aus den an das Targetsysteem angeschlossenen Modellen ergibt sich eine thematische Teilung von Übungsbeispielen (siehe, [2]), die vor allem mit der Programmierung von Prozeß-Interfaces zu tun haben.

Zum Einstieg in die Programmierung digitaler I/O Komponenten werden den Studenten Möglichkeiten zur Ansteuerung verschiedener In- und Output Devices vorgestellt. Diese „low-level“ Human-Interfaces sind auf einem *Bedienpanel* angebracht, das an einige digitale Ein- und Ausgänge der Prozeßperipherie angeschlossen ist. Besonderes Augenmerk im Sinne didaktischer Transparenz wurde beim Printentwurf des Bedienpanels auf dessen einfachen und für Einsteiger überschaubaren Aufbau gelegt. So sind alle wesentlichen Leiterbahnen für die Übungsteilnehmer leicht ersichtlich und verständlich auf der Platinen-Oberseite geführt. Mit Unterstützung der in begleitenden Unterlagen enthaltenen Schaltpläne ist es nicht weiter schwierig, die unterschiedlichsten Devices mittels *memory-mapped I/O* Zugriffs (durch simple Bit-Manipulationen) anzusteuern und somit die gestellten Übungsaufgaben, wie das Einlesen eines BCD-Schalters (BCD), verschiedener anderer Schalter- (Scha) und Tasterstellungen (Tast), das Aktivieren eines Summers (Summ), diverser Leuchtdioden (LEDs) und der Siebensegmentanzeige (7Seg), zu lösen.

Frequenzzähler zur Messung der Frequenz einer Wechselspannung.

Abschließend sei die für den Simulator relevante Besonderheit des Zugriffs auf die Prozeßperipherie in VMEbus-Systemen in aller Kürze vorgestellt. Die Aufgabe der Carrier-Boards ist es, den relativ einfachen Interface-Bus der M-Module an den sehr komplexen VMEbus zu koppeln. Dabei werden die hauptsächlich auf den M-Modulen befindlichen Register in den VMEbus-Adreßraum gemappt, sodaß sie mittels normaler VMEbus-Schreib/Lesezyklen angesprochen werden können (*memory-mapped I/O*). Durch diese Gleichartigkeit von Speicher- und Peripheriezugriffen können Register einer Karte nun auch logisch als „Speicherbereich“ angesehen werden, der an einer fixen Adresse liegt. Der logische Adreßraum der M-Module wird in das entsprechende große Fenster im VMEbus-Adreßraum eingeblendet; Schreib/Lese-Zugriffe der CPU (also vom VMEbus aus) werden somit transparent in entsprechende Zugriffe auf das jeweilige M-Modul umgesetzt.

3 Simulator

Im Simulator sind sämtliche einfache und komplexere M-Module und die Funktionsweise darauf befindlicher Bausteine (z.B. Interruptlogik, Z8536 CIO-Chip) in Software nachgebildet. Weiters wurden das Bedienpanel und die Labormeißgeräte graphisch dargestellt und die entsprechenden Ein- und Ausgabeelemente mit den Software-Modulen - so wie es die Schaltpläne vorschreiben - verbunden. Dadurch werden alle Ausgaben auf den Modellen nun auf „virtuellen“ Modellen am Bildschirm zur Anzeige gebracht, Eingaben anstatt auf realen Tastern, Schaltern etc. an den entsprechenden Stellen der graphischen Benutzeroberfläche durch Betätigen der Maus durchgeführt.

3.1 Design und Implementierung

Der Targetsimulator wurde vollständig objekt-orientiert entworfen und gliedert sich in mehrere logische Teile:

1. *Modell-Objekte*. Vertreter sind Bedienpanel, Oszilloskop, Universalmeßgerät, VT220-Terminal, sowie Eingabe-Elemente (z.B. Schiebeschalter, Radiotasten) und Ausgabe-Elemente (z.B. LCD-Displays, Segment-Anzeigen). Diese Objekte sind graphische Aufbereitungen der realen Gegebenheiten und können animiert werden, bzw. Eingaben vom Benutzer entgegennehmen.
2. *VMEbus-Hardware-Objekte*. Diese Objekte simulieren das Verhalten der Carrier-Boards und M-Module. Ein- und Ausgänge können invertieren, das Modul z.B. von sich aus Entprellung durchführen oder einen Change-of-state Interrupt an das Carrier-Board weiterreichen. Auch Eigenheiten analoger Komponenten, wie z.B. Multiplexing und Pipelining bei Analog/Digital-Karten werden simuliert.
3. *Simulations-Objekte*. pSOS⁺ ist ein echtzeitfähiges Multitasking-Betriebssystem für Ein- oder Mehrprozessoranwendungen. In der aktuellen Version des Targetsimulators wurden nur einige wenige pSOS⁺-Systemcalls auf vergleichbare Systemaufrufe des Hostbetriebsystems abgebildet.

4. *Hauptapplikation.* Sie nimmt die Verknüpfung der Modell-Objekte und der VMEbus-Hardware-Objekte vor, legt die virtuelle Memory-Map fest und enthält das eigentliche Kernstück des Targetsimulators, den Disassembler.

Bei der Realisierung des Kernstücks des Targetsimulators standen mehrere Möglichkeiten offen. Prinzipiell hätte die Simulation eines memory-mapped I/O basierenden Rechnersystems unter Zuhilfenahme der Überladung des Zuweisungsoperators mit den sprachlichen Mitteln der Programmiersprache C++ realisiert werden können. Auch das C++-Exception Handling hätte ein Abfangen von Zugriffen auf Hardwareadressen mit nachfolgendem „Umbiegen“ der Speicheradressen unterstützt. Konkret realisiert wurde der Targetsimulator allerdings mit dem Exception Handling der Betriebssysteme Windows95 bzw. WindowsNT. Diese Form funktioniert nach folgendem Muster:

```
void TargetMainWrapper()
{ __try {main();}
  __except(CheckException(GetExceptionCode(),
                        GetExceptionInformation())) {}
}
```

Tritt in der `main()`-Funktion des Studenten eine Exception auf, werden alle CPU-Register in einer Struktur, dem sogenannten `ExceptionRecord`, gespeichert. Außerdem stellt diese Struktur noch weitere Informationen zur Verfügung, z.B. Adresse der Exception, Zugriffs-Adresse der Exception, war der Zugriff schreibend oder lesend. Danach wird die interne Funktion `CheckException` des Targetsimulators ausgeführt. Zu Beginn dieser Funktion wird überprüft, ob es sich um eine Speicherzugriffsverletzung handelt. War dies der Fall, wird der Befehl, der die Exception hervorgerufen hat, durch einen Disassembler analysiert; andere Exceptions werden an das Betriebssystem weitergereicht.

War der verursachende Befehl ein Lesezugriff, wird überprüft, ob die gelesene Speicheradresse ein Carrier-Board oder ein M-Modul des Targetsystems betroffen hat. Falls auf eine gänzlich falsche Speicheradresse zugegriffen wurde, gibt der Simulator eine Warnung aus und beendet die Applikation des Studenten. Anderenfalls wird der entsprechende Wert vom Simulationsobjekt gelesen und in eine (statische) Variable übernommen, sodann der eigentliche Lesebefehl dahingehend modifiziert. Bei einem Schreibzugriff muß ebenso festgestellt werden, welches „virtuelle“ Carrier-Board bzw. M-Modul angesprochen wurde. Auch hier muß das Adressierungs-Register auf die Adresse der internen (statischen) Variable umgebogen werden. Zu beachten ist weiters, daß der im Targetsystem vorhandene Rechner ein big-endian Prozessor ist. Das höchstwertige Byte eines multi-byte Wortes liegt somit an der niedrigsten Speicheradresse. Da jene Prozessoren, auf denen der Targetsimulator läuft, hingegen little-endian Prozessoren sind, muß der Disassembler zudem bei jedem Zugriff überprüfen, ob eine Umwandlung von big-endian auf little-endian vorzunehmen ist.

Nach Beendigung der Funktion `CheckException` wird das Programm an der Stelle weiterverarbeitet, an der die Exception aufgetreten ist, genauer gesagt wird sogar nochmals

der (Maschinen)-Befehl ausgeführt, der die Exception ursprünglich ausgelöst hat - zuvor werden jedoch noch die eventuell vom Disassembler geänderten CPU-Register aus der Struktur wiederhergestellt.

Wird im Zuge der Programmausführung nun auf einen Hardware-Baustein zugegriffen (d.h. die Hardware des Targetsystems wurde via memory-mapped I/O angesprochen), so rufen die entsprechenden Zeilen im Programm eine Exception hervor. Der Disassembler analysiert die Befehlszeilen und wandelt sie in korrespondierende Aufrufe für die Targetsimulation um. Richtig programmierte Lesebefehle liefern somit den Zustand der Eingabemodule (und damit z.B. aktuelle Schalterstellungen), richtig programmierte Schreibbefehle modifizieren die Register der „virtuellen“ Hardware (und damit z.B. Ausgabeelemente wie die Siebensegmentanzeige).

3.2 Programmierung und Bedienung

Sämtliche Funktionen des Targetsimulators sind in einer Programmbibliothek untergebracht. Nachdem der Student seine Applikation programmiert hat, muß er diese nur noch mit der Library linken und das resultierende Programm starten. Die Programmoberfläche ist entsprechend der realen Hardware in vier Fenster unterteilt:

1. *Hauptfenster*. Vom Hauptfenster aus können die Subfenster für das Bedienpanel, für das Oszilloskop und für den Universalmeßplatz geöffnet werden. Verwendet ein Student zu Testzwecken `printf`-Befehle, so werden diese in einem Terminal (Debug-Fenster) ausgegeben. In der realen Umgebung würden diese Ausgaben auf dem vorhandenen VT220-Terminal dargestellt werden. Der Simulator kann mit einem Druck auf den Beenden-Button beendet werden.
2. *Bedienpanel*. Alle für die Laborübung wichtigen Bedien- und Ausgabeelemente wurden nachgebildet. Die Taster und das Keypad können komfortabel mit der Maus bedient werden. Soll eine Taste für längere Zeit gedrückt bleiben oder möchte man das gleichzeitige Drücken mehrerer Tasten simulieren, so müssen die Tasten nur bei gedrückter Maustaste vom entsprechenden Element gezogen werden. Die Schalter wurden animiert wiedergegeben. Jeder Klick bewegt den Schalter in eine andere Richtung, wobei auch unangenehme Zwischenstellungen beim BCD-Schalter berücksichtigt wurden. Die Siebensegmentanzeigen werden mit unterschiedlichen Helligkeiten dargestellt, sodaß der Benutzer - wie bei der realen Hardware - ein leichtes Nachleuchten feststellen kann.
3. *Oszilloskop*. Jene Komponenten, die von den Studenten auch im Übungsbetrieb nicht verstellt werden sollen, können auch im Targetsimulator nicht bewegt (angeklickt) werden. Nach Kalibrierung des Oszilloskops und Auswahl des Kanals, einer gewünschten y-Eingangsabschwächung und Zeitauflösung werden sodann die anliegenden Spannungen periodisch neu gezeichnet.
4. *Universalmeßplatz*. Am Universalmeßplatz kann mit der Maustaste ein regelbares Netzgerät und ein Funktionsgenerator, ausgestattet mit Radio-Buttons und Kipp-

schalten, bedient werden. Das Digitalvoltmeter und der Frequenzzähler liefern die an den Eingängen anliegenden aktuellen Werte.

Bild 3 zeigt abschließend eine Applikation eines Studenten, exekutiert am Targetsimulator.

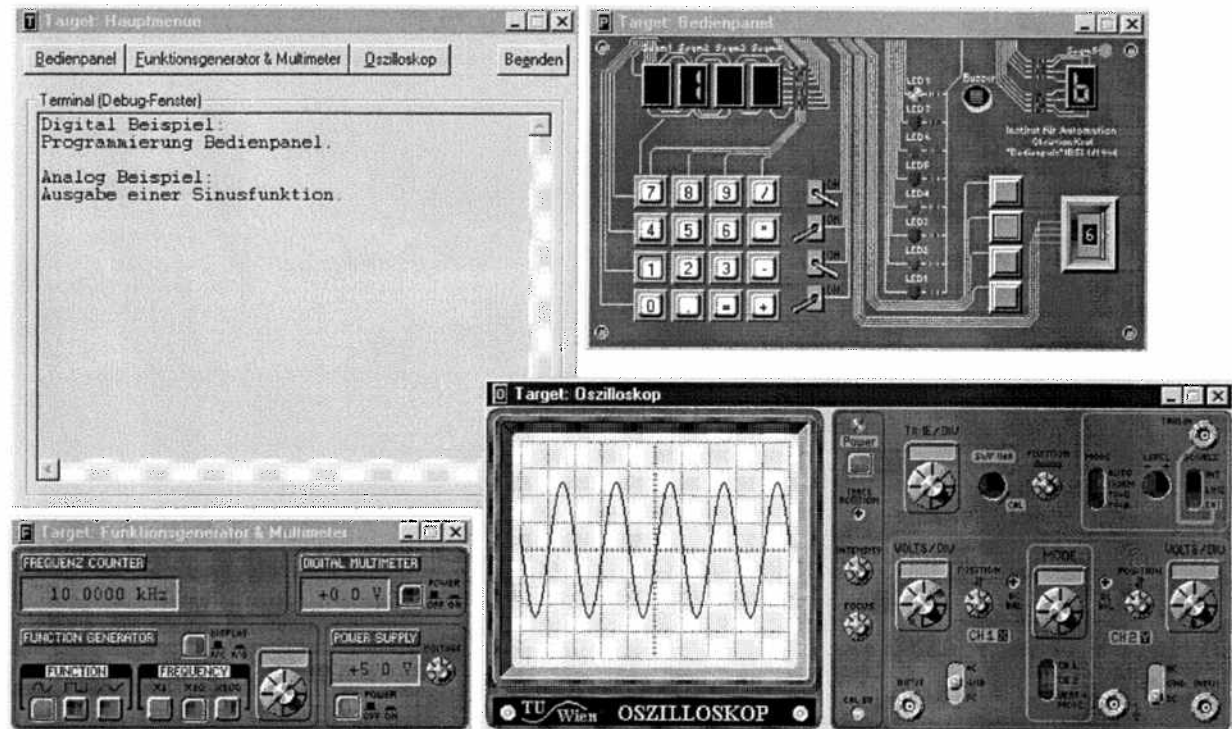


Bild 3: Targetsimulator

Danksagung

An dieser Stelle sei Günter Gridling für die Erstellung des ersten Prototypen, Michael Mayer für die Programmierung des Disassemblers und Franz Breunig für die letztlich entstandene Endversion des Targetsimulators gedankt.

Literatur

- [1] F. Breunig. *Wartungs-/Benutzerhandbuch Simulator*, Studienarbeit TU Wien 1998
- [2] W. Kastner, U. Schmid, B. Weiss. *Beispielsammlung Laborübung Prozeßautomatisierung*, Eigenverlag 1998
- [3] W. Kastner, U. Schmid. Process Control Engineering for Computer Science Students, *Proceedings on Advances in Control Education IFAC ACE'97*, 1997, Seiten 289-295
- [4] MUMM Manufacturers and Users of M-Modules. *M-Module Standard 2.2*, 1993
- [5] U. Schmid, W. Kastner, J. Klasek, C. Kral. *Übungsskriptum Laborübung Prozeßautomatisierung*, Eigenverlag 1998
- [6] G.-H. Schildt, W. Kastner. *Prozeßautomatisierung*, Springer-Verlag 1998

Modellbasierte Evolutionäre Designoptimierung am Beispiel eines optischen Mikrosystems

Wolfgang Süß, Martina Gorges-Schleuter, Wilfried Jakob, Ingo Sieber

e-mail: {suess,gorges,jakob,sieber}@iai.fzk.de

Forschungszentrum Karlsruhe, Institut für Angewandte Informatik

Postfach 3640, D-76021 Karlsruhe, Germany

Zusammenfassung

Dieser Artikel beschreibt einen Evolutionären Algorithmus zur Designoptimierung anhand des Beispiels eines mikrooptischen Kollimationssystems. Nach einer Einführung in die Problematik der Optimierung erfolgt eine kurze Vorstellung des mikrooptischen Kollimationssystems. Danach werden die wesentlichen Charakteristika des verwendeten Evolutionären Optimierungsverfahrens beschrieben. Die Ergebnisse der evolutionären Optimierung werden mit einem Hill-Climbing-Verfahren in Bezug auf Konvergenzgeschwindigkeit und -sicherheit verglichen. Hierbei zeigen sich die Vorteile dieser neuartigen Vorgehensweise für die Erstellung eines fertigungstoleranten Designs.

1 Einführung

Evolutionäre Algorithmen basieren auf der Modellierung von Prinzipien der Populationsgenetik und Darwin'scher Evolution. Die Methodik züchtet Problemlösungen ohne dabei die Problemstruktur genau kennen zu müssen. Im Extremfall genügt dem Verfahren die Möglichkeit Lösungen präsentieren zu können und für diese eine Bewertung, qualitativer oder quantitativer Art, zu erhalten. Der Einsatzbereich dieser Klasse von Suchverfahren reicht von der Lösung schwieriger Such- und Optimierungsprobleme bis zu Verfahren zum Maschinellen Lernen.

Viele Probleme in Wissenschaft und Technik, Wirtschaft und Finanzen, aber auch im Bereich der Managements können als Optimierungsaufgaben formuliert werden. Die gewünschte Lösung kann die Minimierung einer oder mehrerer Kostenfunktionen mit widersprüchlichen Zielsetzungen sein, die Maximierung des Profits oder als Beispiel für konkrete Anwendungen die kostenoptimale Struktur eines Stabtragwerks (Brücke) [1], die optimale Terminierung der Aufträge bei der Ablaufplanung durch geeignete Maschinenbelegung, die Erzeugung von Portraits aufgrund von Zeugenaussagen [2] und Prognosemodelle für Handel mit Währungen [3].

Solche Systeme, wie sie in der Praxis behandelt werden, sind häufig so komplex, daß eine analytische Vorgehensweise entweder gar nicht oder nur mit nicht vertretbarem Aufwand möglich ist. Insbesondere setzt der Einsatz numerischer Lösungsverfahren eine explizite Formulierung der Zielfunktion und deren Differenzierbarkeit voraus. Tatsächlich haben reale Probleme oft eine große Anzahl von Parametern, die zusätzlich nichtlinear gekoppelt sind und der Suchraum ist häufig multimodal und auf Grund von Nebenbedingungen nicht zusammenhängend. Des weiteren können Rauschen und zeitabhängiges Verhalten eine effiziente Suche erschweren.

Können keine Verfahren eingesetzt werden, die das Optimum berechnen, so bieten heuristische Methoden die Möglichkeit „gute“ Lösungen mit akzeptablem Zeitaufwand zu finden. Gut heißt

in diesem Zusammenhang, daß auf das Auffinden des mathematischen Optimums verzichtet wird und man sich stattdessen mit einer hinreichend guten Lösung, die aber mit vertretbarem Aufwand erreicht werden kann, zufrieden gibt. Die geforderte Qualität der Lösungen und die geforderte Sicherheit mit der diese gefunden werden sollen, schlägt sich dabei im benötigten Aufwand nieder.

Hier wird der Einsatz unseres evolutionären Verfahrens für die Optimierung eines mikrooptischen Kollimationssystems vorgestellt. Ziel dieser Optimierung ist es, die Auswirkungen der fertigungstechnischen Einfügetoleranzen auf die Güte des Kollimationssystems zu minimieren. Diese Vorgehensweise stellt eine neue Anwendung für unseren evolutionären Ansatz dar. Der Erfolg dieser Vorgehensweise wird in einem Vergleich gegenüber den Optimierungsergebnissen einer Gauß-Seidel-Strategie verdeutlicht.

2 Das mikrooptische Kollimationssystems

Für viele mikrooptische Anwendungen ist es notwendig, die Abstrahlcharakteristik der Quellen (i.a. Laser oder Lichtfasern) zu verändern, das heißt, sie zu kollimieren oder zu fokussieren [4]. Dazu dienen mikrooptische Linsen oder Kombinationen verschiedener Linsen. Das hier beschriebene System benützt zwei Kugellinsen, um die Abstrahlung einer Single-Mode-Faser (SMF) zu kollimieren und die kollimierte Strahltaile auf eine Photodiode abzubilden (Bild 1).

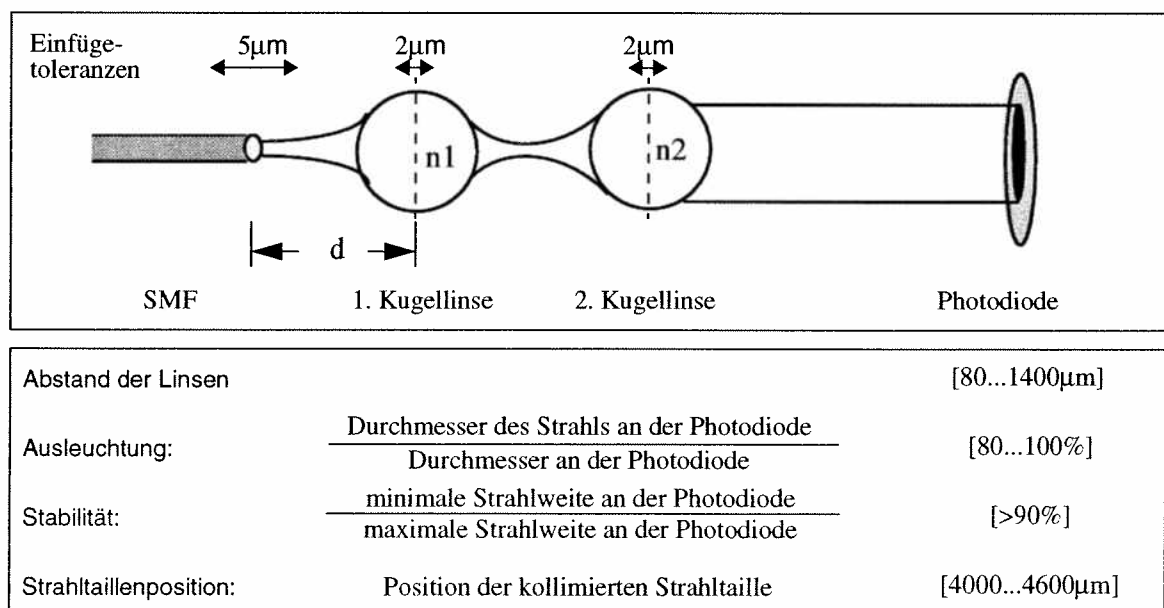


Bild 1: Aufbauskeizze des Kollimationssystems und Fertigungstoleranzen (oben).

Die Systemparameter sind n_1 , n_2 und d .

Der Kasten unten zeigt die Definition der Optimierungskriterien und den Bereich akzeptabler Werte.

Im Idealfall der geometrischen Optik kann unter bestimmten Bedingungen zu jeder Kugellinse mit bestimmter Brennweite eine zweite Kugellinse gefunden werden, so daß eine beliebige Abstrahlung erzeugt werden kann. Abweichend von diesem Idealfall sollen nun aber auch Tole-

ranzeffekte, die durch das Einfügen der optischen Elemente in bereits vorgefertigte LIGA-Strukturen [5] entstehen, berücksichtigt werden. Diese Einfügetoleranzen haben Auswirkungen auf die Strahlweite am Ort der Photodiode und auf die Lage der Strahltaille. Es soll nun ein Kollimationssystem bestimmt werden, das möglichst unempfindlich gegenüber den erwarteten Einfügetoleranzen der einzelnen Elemente ist.

Die Systemparameter, die bei den einzelnen Optimierungsberechnungen variiert werden können, sind die Brechungsindices der beiden Kugellinsen (n_1 und n_2) und der Abstand der Faser, die das Licht in das System einspeist, zu der ersten Kugellinse (d). Als Optimierungskriterien werden Ausleuchtung, Stabilität, Strahltaillenposition und Linsenabstand definiert:

- **Ausleuchtung:** die Ausleuchtung wird als Quotient aus der maximalen Strahlweite an der Position der Photodiode und dem Durchmesser des photosensitiven Bereichs der Photodiode definiert. Als Optimierungsziel wird eine Ausleuchtung von 90% gewählt, um bei lateralem Strahlversatz eine Überstrahlung der Photodiode zu verhindern.
- **Stabilität:** die Stabilität wird als der Quotient aus minimaler und maximaler Strahlweite in Abhängigkeit von den Toleranzeffekten definiert. Das Optimierungsziel bezüglich der Stabilität besteht darin, einen Wert nahe bei 100% zu finden, um so Schwankungen der Ausleuchtung der Photodiode in Abhängigkeit von Toleranzeffekten gering zu halten.
- **Strahltaillenposition:** der optimale Wert für diesen Parameter liegt bei 4300 μm . Dies entspricht dem Abstand der zweiten Kugellinse zu der Photodiode. Dieses Optimierungskriterium sorgt dafür, daß die kollimierte Strahltaille auf die Photodiode abgebildet wird.
- **Linsenabstand:** das Kriterium Linsenabstand wurde eingeführt, um eine Grenze für die Ausdehnung des Gesamtsystems festzulegen. Die angestrebten Werte sollen im Bereich zwischen 80 μm und 1400 μm liegen.

3 Das Optimierungsverfahren

Am Institut für Angewandte Informatik des Forschungszentrums Karlsruhe wurde die Werkzeugumgebung SIMOT (Simulation and Tool Environment) entwickelt. SIMOT besteht aus zwei Hauptkomponenten: den Simulations- und den Optimierungswerkzeugen. Für die Simulatoren werden Standardwerkzeuge eingesetzt. Die beiden Optimierungswerkzeuge GAMA (Genetischer Algorithmus für die Modelladaptation) und GADO (Genetischer Algorithmus für die Designoptimierung), die auf Evolutionären Algorithmen basieren, sind Eigenentwicklungen. GAMA wird eingesetzt, um die Qualität eines parametrisierten analytischen Makromodells zu verbessern, wobei zur Bewertung der Qualität des Makromodells auf die Ergebnisse von FEM-Simulationen zurückgegriffen wird. Das verbesserte und hinreichend genaue Makromodell bildet dann die Basis für den Optimierungsprozess des Systemdesigns mittels GADO. Die in SIMOT eingesetzten Werkzeuge GAMA und GADO [6] basieren auf dem evolutionären Algorithmus GLEAM (Genetic LEarning Algorithm and Method) [7]. Diesem Algorithmus liegen sowohl die genetischen Algorithmen, die J. Holland durch sein Studium adaptiver Systeme begründet hat [8], als auch die Evolutionsstrategien, deren Begründer I. Rechenberg [9] und H. P. Schwefel [10] waren, zugrunde.

Ein *evolutionärer Prozeß* ist beispielsweise die stammesgeschichtliche Entwicklung der Lebewesen von niederen zu höheren Formen. Dieser Entwicklungsprozeß beruht auf *evolutionären Mechanismen*. Bei der formalen Beschreibung dieser Entwicklungsvorgänge werden die Lebewesen als *Individuen* bezeichnet. Die Gesamtheit der Individuen einer Art in einem bestimmten geographischen Raum wird eine *Population* genannt. Genetische Vorgänge wie *Mutation* und *Rekombination* tragen zur Veränderung der Individuen bei. Bei einer Mutation erfolgt eine spontane Veränderung im Erbmateriale. Als Rekombination wird die Neukombination von Erbfaktoren bezeichnet, d.h. der Austausch von Erbfaktoren zweier Individuen. Die dabei entstehenden neuen Nachkommen unterliegen einem *Selektionsprozeß*. *Evolutionäre Algorithmen* sind eine Abstraktion dieser grundlegenden evolutionären Prinzipien.

Bei den *genetischen Algorithmen* entspricht ein Individuum einer Designvarianten, d.h. alle relevanten Merkmale eines Systems (Systemparameter) werden in diesem Individuum mit einer bestimmten Datenstruktur beschrieben. Bei einer Mutation wird beispielsweise durch einen Zufallsgenerator ein Merkmal verändert. Bei einer Rekombination findet der Austausch einiger Merkmale (Daten) zweier Individuen statt, wobei Individuen mit höherer Güte eine größere Chance besitzen, als Eltern ausgewählt zu werden. Durch die Selektion geben Individuen mit guten Eigenschaften diese mit einer höheren Wahrscheinlichkeit an die Nachkommen weiter. So kann die Güte der Individuen von Generation zu Generation ansteigen.

Auch bei den *Evolutionstrategien* wird von einer Population von Individuen ausgegangen. Ein Individuum enthält aber neben den Designeigenschaften zusätzliche Informationen über die sogenannten Mutationsschrittweiten, die als Strategieparameter wirksam sind. Die Mutationsschrittweiten spezifizieren die Standardabweichung einer normalverteilten Zufallsvariablen mit dem Erwartungswert Null, so daß die Wahrscheinlichkeit kleiner Änderungen größer ist. Eine Rekombination zweier zufällig gewählter Elternteile kann entweder bei den Designeigenschaften durch zufällige Mischung einiger Elternteilstücke oder aber bei den Mutationsschrittweiten, z.B. durch Mittelwertbildung der beiden Elternteilstücke gebildet werden. Bei der anschließenden Selektion werden je nach Strategie die besten Individuen entweder aus der Eltern-/Nachkommen-Generation oder nur aus der Nachkommengeneration ausgewählt und bilden die nächste Elterngeneration.

Die evolutionären Algorithmen sind allgemeine Planungs- und Optimierungsverfahren und ihr Einsatz ist dann von Vorteil, wenn sehr viele Parameter zu verarbeiten sind, lokale Optima und eventuell zusätzliche Restriktionen und Unstetigkeiten vorliegen. Gerade bei einer nichtlinearen, multimodalen und diskontinuierlichen Optimierungsfunktion können traditionelle Verfahren an ihre Grenzen stoßen. Der Einsatz evolutionärer Algorithmen ist insbesondere dann sehr sinnvoll, wenn über das konkrete Problem (Bewertungsraum) kein Vorwissen vorhanden ist. Verfügbares Vorwissen kann jedoch eingebracht werden, was zu einer Reduzierung der Optimierungszeit führt. Diese Eigenschaften der evolutionären Algorithmen können sich als Vorteile gegenüber traditionellen Optimierungsverfahren erweisen, da traditionelle Verfahren wie z.B. das Gradientenverfahren mathematische Kenntnisse über das Optimierungsproblem voraussetzen.

Evolutionäre Algorithmen gehen im Gegensatz zu den traditionellen Verfahren nicht nur von einer Lösung aus, sondern von einer ganzen Population, die mehrere Individuen (Lösungen) enthält. Wir betrachten hier eine räumlich strukturierte Population. Das Modell besteht aus Individuen, die in dieser Anwendung linear angeordnet sind. Die Interaktion der Individuen, insbesondere Selektion der Partner zur Rekombination und die Selektion von Nachkommen entsprechend der Überlebensregel, ist auf die geographisch nahen Individuen beschränkt. Diese lokalen Reproduktionsgemeinschaften bilden sich überlappende Nachbarschaften. Die Vorteile für den evolutionären Lernprozeß sind zum einen eine robustere Suche und zum anderen durch ausschließlich lokale Interaktion und lokale Ersetzungsregeln die explizite Parallelisierbarkeit des evolutionären Verfahrens.

Die Vorgehensweise in GLEAM kann in zwei Schritte untergliedert werden (siehe auch Bild 2):

- Initialisierung und Bewertung der Startpopulation
- Erzeugen neuer Folgegenerationen unter Anwendung evolutionärer Mechanismen (Mutation, Selektion, Rekombination...).

Der zweite Schritt wird solange ausgeführt, bis ein vom Benutzer definiertes Abbruchkriterium erreicht wird.

Die hohe Simulationsanzahl, die bei diesem heuristischen Suchverfahren jedoch notwendig ist, ließ bisher dieses Verfahren gegenüber anderen Optimierungsverfahren aufgrund der bei manchen Aufgaben auftretenden geringen Konvergenzgeschwindigkeiten in den Hintergrund tre-

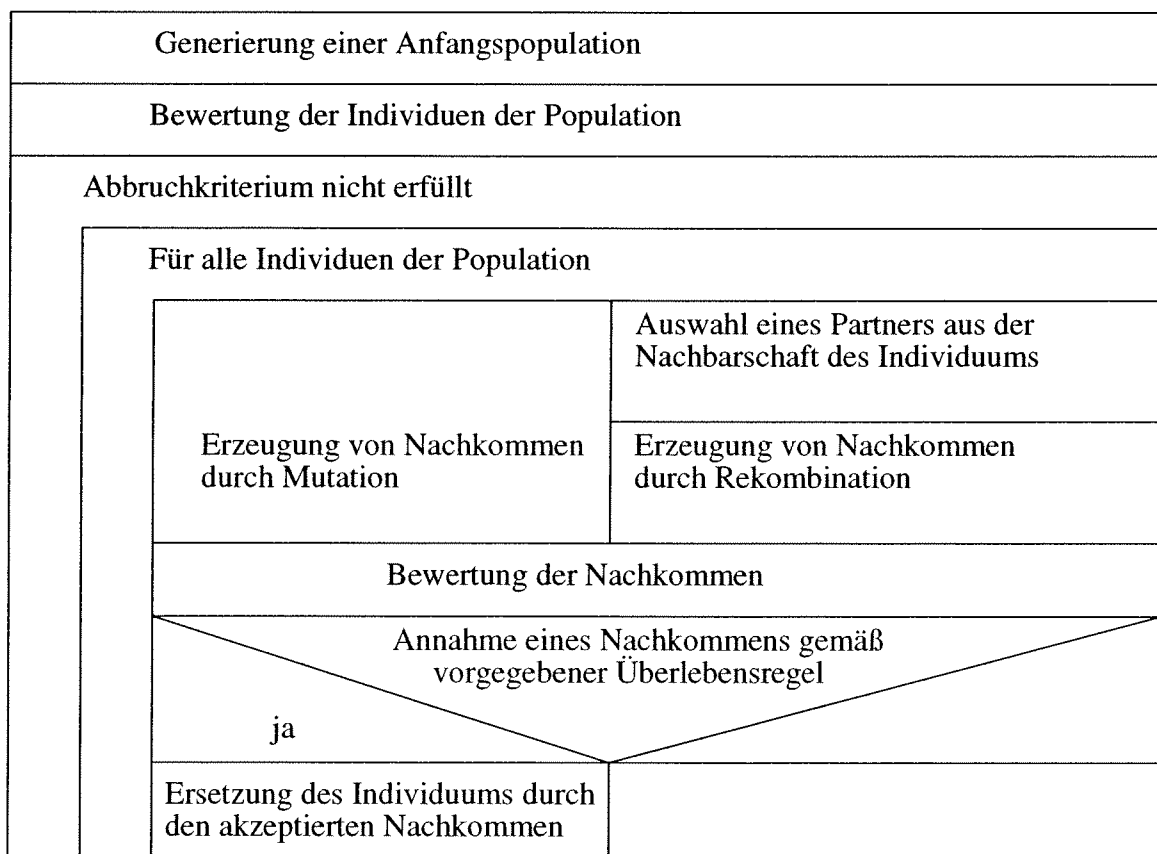


Bild 2: Ablaufschema des evolutionären Verfahrens GLEAM

ten. Durch eine Parallelisierung, d.h. durch Ausnutzung mehrerer Rechner in einem Rechnernetz, kann die Optimierungszeit jedoch drastisch reduziert werden. Auch die voranschreitende Leistungssteigerung der Rechner führt zu sinkenden Optimierungszeiten und damit zum vermehrten Einsatz dieser Verfahren.

Die hohe Konvergenzsicherheit, die die evolutionären Verfahren jedoch aufweisen, da sie nicht nur von einer, sondern von einer Population von mehreren Lösungen ausgehen, spricht für den Einsatz dieser Verfahren gerade auch bei Problemen, bei denen traditionelle Verfahren an ihre Grenzen stoßen. Am Beispiel des „symmetrischen und asymmetrischen Handlungsreisenden Problems“ sind hierzu genauere Untersuchungen durchgeführt worden [11].

4 Ergebnisse

Das 2-Linsen-System zeigt, daß auch Systeme mit nur wenigen Systemparametern, hier die Brechungsindices der beiden Kugellinsen und der Abstand zwischen SMF und erster Kugellinse, zu sehr komplexen multi-modalen Suchräumen führen können. Die multi-kriterielle Systembewertung bzgl. Ausleuchtung, Stabilität, Strahltaillenposition und Linsenabstand und die Empfindlichkeit dieser Bewertung gegenüber kleinen Änderungen der Systemparameter erschweren die Optimierung zusätzlich.

Das Kollimationssystem wird mit Mathematica von Wolfram Research Inc. simuliert, wobei die Extremwerte der Fertigungstoleranzen benutzt werden, um die Stabilität zu bestimmen. Aus dem Simulationsergebnis werden die absoluten Werte der Optimierungskriterien berechnet und auf jeden Wert eine vorgebene Bewertungsfunktion angewendet. So kann erreicht werden, daß z.B. Designs mit einer Ausleuchtung von unter 90% und über 95% exponentiell proportional mit dem Betrag der Abweichung von 90% abgewertet werden. Für die Qualitätsbewertung eines Designs, die Notenberechnung, werden diese Werte gewichtet und aufsummiert. In unserem Optimierungswerkzeug GADO liegt die Notenskala zwischen 0 und 100000. Bild 3 zeigt die Bewertungsfunktionen für die Optimierungskriterien Ausleuchtung und Stabilität.

Unsere ersten Simulationen des Kollimationssystems zeigten, daß es unmöglich ist zufriedenstellende Designwerte durch einen einfachen Monte-Carlo-Ansatz zu erhalten. Scatterplots der Ergebnisse von Optimierungsläufen, die nur jeweils ein Optimierungskriterium betrachteten, zeigten, daß die Optimierungskriterien im Widerspruch zueinander stehen, und daß der gemein-

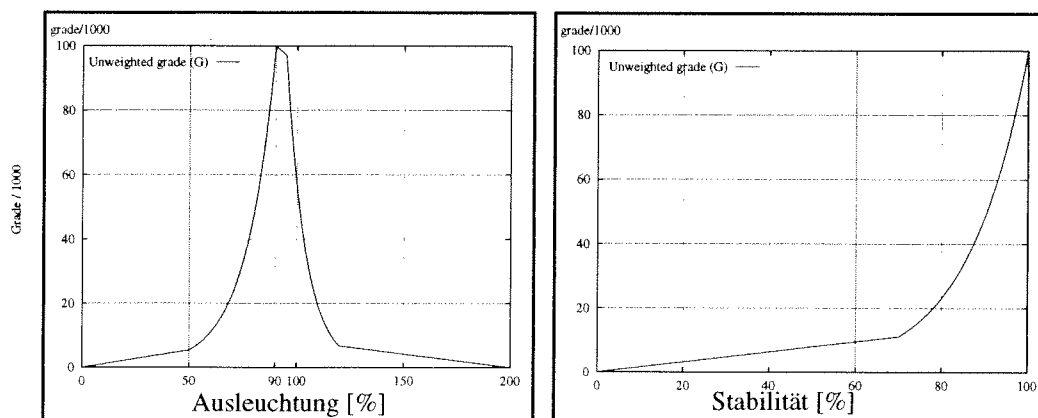


Bild 3: Bewertungsfunktionen für Ausleuchtung (links) und Stabilität (rechts) .

same Suchraum aller Optimierungskriterien sehr wahrscheinlich fraktaler Natur ist. Speziell die Forderung nach Stabilität ist sehr schwer zu erfüllen.

Als nächstes wendeten wir ein Hill-Climbing Verfahren (Gauß-Seidel) an. Ausgehend von einem zufällig generierten Startwert wurde dabei ein Parameter ausgewählt und dieser optimiert, bis keine Verbesserung des Parameters mehr möglich war. Anschliessend wurde der nächste Parameter gewählt und wiederum optimiert, bis keine Verbesserung mehr erreicht werden konnte. Dies wurde für alle Optimierungsparameter wiederholt. Die Anzahl der Evaluierungen, die benötigt werden, bis diese Strategie konvergiert variiert von 2000 bis 42000. Die erreichbare Qualität ist abhängig von dem zufällig gewählten Startwert. Bei unterschiedlichen Startwerten erhält man bei dieser Aufgabe in der Regel unterschiedliche Lösungen. In Bild 4 sind die Ergebnisse von sechs Läufen (HC1 bis HC6) des Hill-Climbing-Verfahrens dargestellt. Man erkennt an diesem Bild die dabei erreichten, unterschiedlichen Optimierungsergebnisse. Die beste Lösung, die mit dem Hill-Climbing-Verfahren erzielt werden konnte, lieferte eine Stabilität von 90.3%, eine Ausleuchtung von 90.7% und eine Strahltaillenposition von 4294.3 μm . Dies ergibt eine Note von 79068.

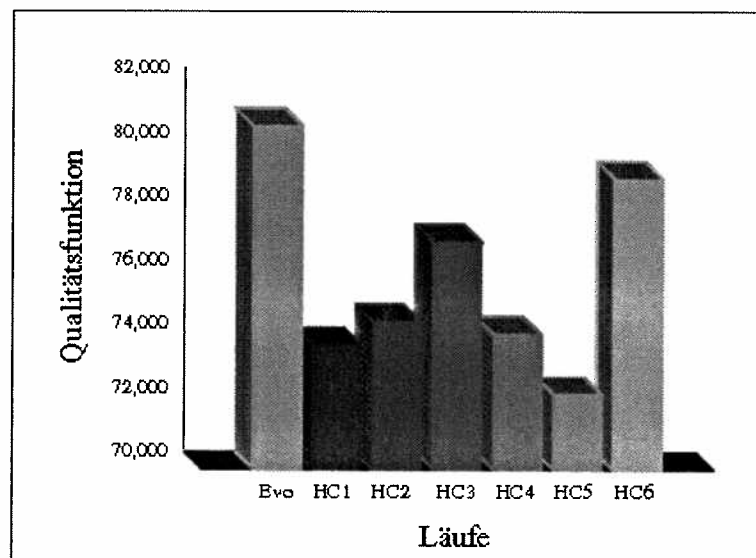


Bild 4: Vergleich zweier Optimierungsverfahren

Die Frage war nun, welche Ergebnisse mit einem „intelligenten“ Verfahren erreicht werden können. Wir konnten mit unserem evolutionären Verfahren GADO bessere Ergebnisse mit durchschnittlich nur 4200 Simulationen erzielen. Im Bild 4 wird deutlich, daß die mit Evo gekennzeichnete Säule eine wesentlich bessere Lösung des Optimierungsproblems liefert als die Hill-Climbing-Verfahren. Die beste Lösung von GADO lieferte eine Stabilität von 91.22%, eine Ausleuchtung von 90.0% und eine Strahltaillenposition von 4300.1 μm . Dies ergibt eine Note von 81031. Verglichen mit den Ergebnissen des Hill-Climbing-Verfahrens ergab sich bei GADO eine völlig andere Lösung; d.h. es ergaben sich völlig andere Brechungsindices ($n_{1\text{HC}} = 2.0$, $n_{1\text{Evo}} = 1.60$, $n_{2\text{HC}} = 1.58$, $n_{2\text{Evo}} = 1.55$) für die beiden Linsen. Dadurch ist die Verwendung unterschiedlicher Materialien für die Linsen notwendig, woraus deshalb völlig verschiedene optische Systeme resultieren.

5 Fazit

Das vorgestellte Beispiel zeigte die Einsatzmöglichkeit unseres evolutionären Verfahrens. Der Vorteil evolutionärer Verfahren liegt dabei in ihrer Möglichkeit, im Unterschied zu anderen Heuristiken, nicht nur einen Lösungsansatz zu verfolgen, sondern eine Vielzahl von Lösungen gleichzeitig zu betrachten. Die über den aktuellen Erfolg oder Mißerfolg der anderen Lösungspfade zur Verfügung stehende Information wird für den weiteren Suchvorgang genutzt. Hierdurch kann das Verfahren gut zwischen Breiten- und Tiefensuche balancieren und es kann ein robustes Optimierungsverhalten erreicht werden. Im Gegensatz zu vielen mathematischen Verfahren liefern Evolutionäre Verfahren kontinuierlich Zwischenlösungen und nicht erst ein Ergebnis nach der Terminierung des Verfahrens. In dem aktuellen Pool von Lösungen wird das durch den Lösungsfindungsprozeß akkumulierte Wissen über die Struktur des Suchraumes abgelegt.

Das Verfahren ist von allgemeiner Natur und kann für viele Optimierungsprobleme in Wissenschaft und Technik, Wirtschaft und Finanzen, sowie im Bereich des Managements angewendet werden. Es hat vor allem bei komplexen Problemen, bei denen eine analytische Vorgehensweise entweder gar nicht oder nur mit nicht vertretbarem Aufwand möglich ist, seine Stärken. Insbesondere setzt der Einsatz Evolutionärer Verfahren keine explizite Formulierung der Zielfunktion und deren Differenzierbarkeit voraus.

6 Literaturverzeichnis

- [1] G. Rudolph: *Globale Optimierung mit parallelen Evolutionstrategien*, Diplomarbeit, Universität Dortmund, Fachbereich Informatik 1990
- [2] C. Caldwell, V.S. Johnston: *Tracking criminal suspect through face-space with a genetic algorithm*, in Proceedings of the Forth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Mateo, 1991
- [3] N. Packard: *A genetic learning algorithm for the analysis of complex data*, Complex Systems 4 (1990), S. 573 - 586
- [4] D. C. O'Shea: *Elements of Modern Optical Design*, Wiley, New York, 1985
- [5] W. Ehrfeld, E. W. Becker: *Das LIGA-Verfahren zur Herstellung von Mikrostrukturkörpern mit großem Aspektverhältnis und großer Strukturhöhe*, KfK-Nachrichten (1987), 4, S. 167-179
- [6] W. Jakob, S. Meinzer, A. Quinte, W. Süß, M. Gorges-Schleuter, H. Eggert: *Partial Automated Design Optimization Based on Adaptive Search Techniques*, Proceedings of ACEDC'96, PEDC, University of Plymouth, 1996
- [7] C. Blume: *GLEAM - A System for Simulated Intuitive Learning*, Proc. of the 1st Int. Workshop on Parallel Problem Solving from Nature, Dortmund 1990
- [8] J. H. Holland: *Adaption in Natural and Artificial Systems*, University of Michigan press, Ann Arbor 1975
- [9] I. Rechenberg: *Evolutionsstrategie '94*, Frommann-Verlag, Stuttgart, 1994
- [10] H. P. Schwefel: *Evolution and Optimum Seeking*, Wiley, New York, 1995
- [11] M. Gorges-Schleuter: *On the power of evolutionary optimization at the example of ATSP and large TSP Problems*, European Conference on Artificial Life, Brighton, July 1997

Generieren, Präsentieren und Interagieren als integrale Bestandteile eines komplexen Modellbildungsansatzes

Jürgen Perl
Universität Mainz
Institut für Informatik

Zusammenfassung

Obwohl Tools und Entwicklungsumgebungen im Bereich Modellbildung inzwischen ein Entwicklungs- und Anwendungsfeld eröffnen, das weit über das Übersetzen und Auswerten von Formeln hinausgeht, werden gerade im Bereich der wissenschaftlichen Modellbildung die verfügbaren Möglichkeiten der Computerunterstützung selten ausgeschöpft. Kapazitäten, Ressourcen und Geschwindigkeit werden häufig nur genutzt, um systemseitige Verbesserungen der Modelle zu bewirken, wie etwa schnellere Verarbeitung größerer Datenmengen. Die Interaktion zwischen System und Nutzer ist dagegen kaum komfortabler geworden, auch wenn standardisierte, graphische Benutzungsoberflächen mit Ereignisorientierung das suggerieren. Abhilfe können hier Konzepte schaffen, die es erlauben, Modelle interaktiv zu generieren und zu modifizieren, die suggestive und informative Präsentationen liefern, und die schließlich auf allen Ebenen komfortable Interaktionen ermöglichen. Ziel sollte es dabei sein, nur aus der Problemwelt heraus und möglichst ohne Orientierung am Kontext der unterstützenden Hard- und Software das Modell "modellierend" zu entwickeln und anzupassen.

Im folgenden sollen Konzepte und exemplarische Modellformen angesprochen werden, mit deren Hilfe Ideen für Modellgenerierung, Präsentation und Interaktion demonstriert und verdeutlicht werden können. Den Bezugsrahmen hierfür bilden unsere Modellbildungs-Aktivitäten im Bereich Sport.

1 Ansätze und Werkzeuge

In der Phase der Modellentwicklung sind Modellgeneratoren oder Metamodelle wichtig, die dem Nutzer das interaktive Entwerfen der Modellstruktur sowie das interaktive Spezifizieren, Zusammenfügen und Verfeinern der Modellkomponenten ermöglichen, ohne ein technisches Verständnis des Modellbildungswerkzeuges vorauszusetzen. Für Präsentation und Interaktion sind on line arbeitende metaheuristische Verfahren sowie interaktive visuelle Oberflächen von Bedeutung.

Speziell die Aspekte Phänomenanalyse und Kalibrierung von Modellen durch Parametervariation mit Hilfe von interaktiven Bedienfeldern (Kapitel 2, 3 und 4), Analyse und optimierende Realisierung von Wirkungsprofilen mit Genetischen Algorithmen und Fuzzy-Neuronalen Netzen (Kapitel 2 und 3) sowie interaktive Generierung und Modifizierung von Modellen (Kapitel 2 und 3) werden im folgenden exemplarisch angesprochen.

2 Das Leistungspotential-Metamodell

2.1 Exkurs: Leistungspotential

Geht man davon aus, daß Leistung durch Belastung trainierbar ist, dann hat eine Belastungsrate auf das zu trainierende System zwei gegenläufige Effekte (vgl. Bild 1, Struktur-Modell zum Leistungspotential): Zum einen baut sie ein Entwicklungspotential auf, das das Leistungspotential positiv beeinflusst. Zum anderen baut sie ein Belastungspotential auf, das negativ auf das Leistungspotential wirkt. Abhängig davon, ob die Verzögerung VB, mit der das Belastungspotential wirksam wird, größer ist als die Verzögerung VE, mit der das Entwicklungspotential wirksam wird, ergeben sich typische Verhaltensformen, z.B. die sog. Superkompensation im Fall $VB < VE$.

Ein weiterer vertrauter und in konkreten Systemen auch meßbarer Effekt ist der des Übertrainings: Bei einer Steigerung der Belastungsrate wird zunächst das Leistungsverhalten verbessert; wird die Trainingsbelastung jedoch zu hoch, dann treten starke gegenläufige, leistungsmindernde Effekte auf. Dieses Phänomen wird in Bild 1 durch eine Überlaufrate im Belastungsbereich modelliert, die mit einer kleinen Verzögerung VÜB negativ auf das Leistungspotential wirkt. (Ein analoger Überlauf im Entwicklungsbereich ist angedeutet, wird hier aber nicht thematisiert.) Verdeutlicht wird der Effekt durch die "Reserve" $R = \max BP - BP$ mit maxBP als maximaler Kapazität des Belastungspotentials BP.

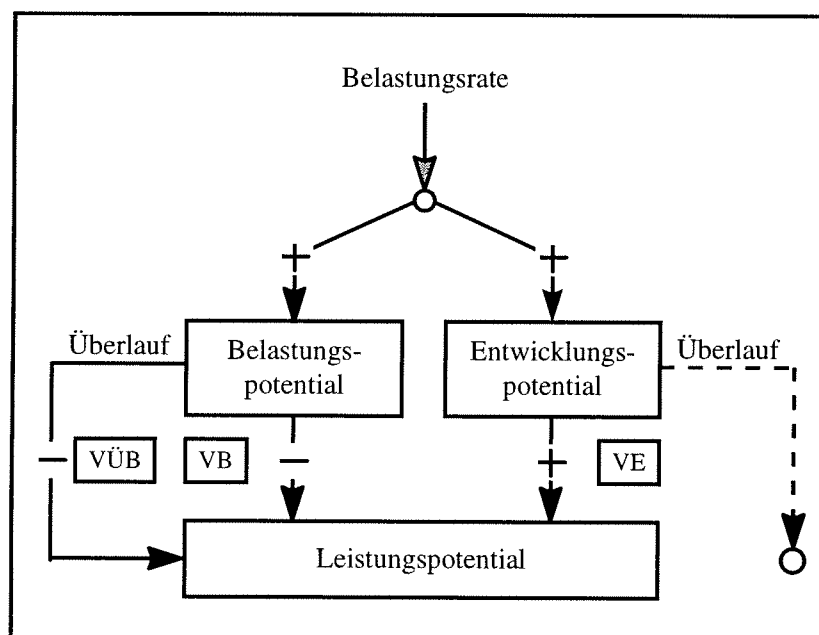


Bild 1 : Strukturmodell zum Leistungspotential

Im Unterschied zu Leistungs- oder Wirkungsmodellen, wie sie in den Bereichen Sportmedizin oder Physiologie für spezielle (Mikro-)Systeme entwickelt worden sind (vgl. speziell z.B. die Arbeiten von MADER), erhebt das hier vorgestellte Metamodell keinerlei Anspruch auf konkrete Gültigkeit oder unmittelbare Umsetzbarkeit für konkrete Systeme. Es soll dazu dienen, grundlegende Phänomene klarer erkennen und einfacher analysieren zu können.

2.2 Interaktives Simulationsmodell zum Leistungspotential

Wechselwirkungsmodelle, in denen wie beim Leistungspotential-Modell Flüsse eine wesentliche Rolle spielen, können mit Hilfe von Differentialgleichungen kontinuierlich oder mit Hilfe von Level-Raten-Gleichungen diskret modelliert werden. Simulation ermöglicht die Analyse des Modellverhaltens abhängig von den Modellparametern und verfolgt damit mehrere Ziele:

Erstens kann das qualitative Verhalten des Modells untersucht werden. Dazu gehören hier die Phänomene der Superkompensation und des Übertrainings. Zweitens kann das quantitative Verhalten des Modells überprüft werden. Beim Modell des Leistungspotential wären das z.B. die Zusammenhänge zwischen Startwerten, Wirkungsverzögerungen und Wirkungseffekten. Wünschenswert für diese beiden ersten Ziele ist einerseits ein on line-feed back des Modellverhaltens auf Parameteränderungen, um typische Phänomene und fehlerhafte Ansätze schnell erkennen zu können. Drittens schließlich muß das qualitativ und quantitativ verifizierte Modell an die konkrete Problemsituation angepaßt, d.h. kalibriert werden. Auch hierfür ist eine interaktive Parameteränderung mit on line-feed back des Modellverhaltens sehr nützlich.

Der hier vorgestellte, speziell für das Leistungspotential-Metamodell entwickelte Prototyp versucht, durch gekoppelte analoge Dateneingabe (Scrollbar) und digitale Dateneingabe (Edit-Fenster), durch interaktive Datenmodellierung (Scrollbar-Felder, on event-Datenerfassung) und durch unmittelbare Präsentationsreaktionen auf Dateneingaben (Ereignissteuerung) analog dem Konzept des "wysiwyg" den Nutzungskomfort komplexer Simulationsmodelle zu verbessern.

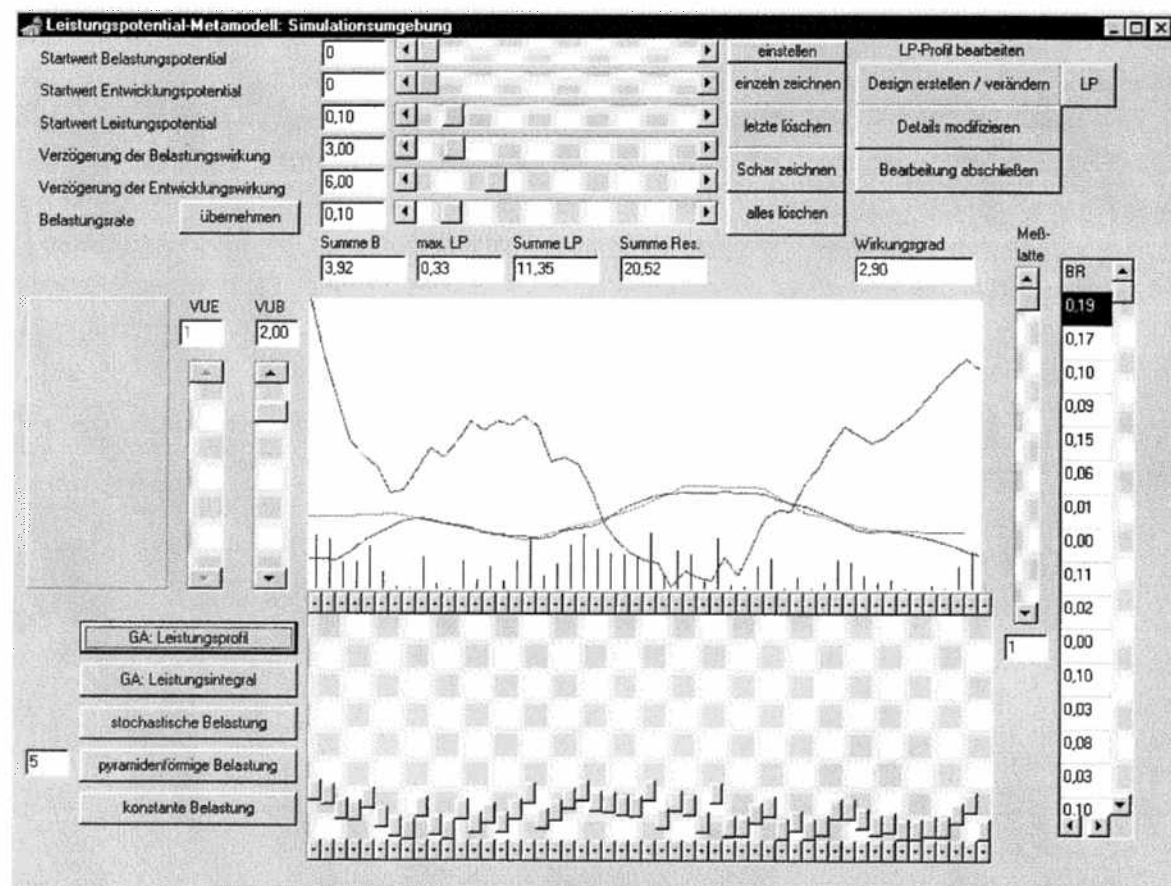


Bild 2 : Bedienoberfläche des Leistungspotential-Modells

Im vorgestellten Ansatz können Modell-Parameter und Eingabedaten mit Scrollbars oder Edit-Feldern eingegeben werden und resultieren unmittelbar in eine Änderung der graphischen Ergebnisausgabe. So kann das Modell simulativ und interaktiv an optimale Lösungen "herangefahren" werden. Zusätzlich berechnet ein Genetischer Algorithmus ("GA") in Quasi-Echtzeit optimale, an interaktiv eingebbaren Problemprofilen (Ziel-Leistungsprofile) orientierte Lösungen (Belastungsprofile). Bild 2 zeigt die GA-Optimierung, die vom Ziel-Leistungsprofil im wesentlichen für die ersten sieben Schritte wegen des Effektes der Superkompensation abweicht. Die darüberliegende stark schwankende Kurve stellt die Reserve dar, die in dem Sinne optimiert ist, daß sie nicht auf Null sinkt, so daß also kein Belastungsüberlauf ausgelöst wird. Das berechnete Belastungsprofil selbst ist durch die senkrechten Linien im Graphikfeld dargestellt, die sich auch in den Knopfpositionen im darunterliegenden Scrollbarfeld widerspiegeln. Auch manuelle Eingaben in diesem Feld bewirken unmittelbare Reaktionen im graphischen Ausgabefeld und erlauben so unmittelbare Einblicke in das Wirkungsverhalten des Modells.

Ziel der gegenwärtigen Projektarbeit ist es, diese Form interaktiver Benutzungsoberflächen so in den bereits entwickelten Modellgenerator einzubeziehen, daß in der interaktiven Phase von Modell-Entwicklung, -Validierung und -Revision sowohl Parameterbelegungen, Wertverläufe und Profile als auch Modellstrukturen interaktiv und on line manipuliert werden können.

3 Das Prozeßmuster-Modell

3.1 Neuronale Netze zur Klassifikation von Verhaltensprozessen

Verhaltensprozesse, wie sie exemplarisch in Mannschaftsspielen im Sport in Erscheinung treten, sind gekennzeichnet durch eine große Zahl unabhängiger Kategorien und Attribute. Dementsprechend ist die Zahl der unterscheidbaren Prozesse groß und die Aussagekraft statistischer Analysen gering. Neuronale Netze (hier speziell: Kohonen Feature Maps) können eingesetzt werden, um durch Clusterung zu Mustern und Typen die Prozeßvielfalt auf ein verwendbares Maß zu reduzieren.

Nach der im Training erfolgten Zuordnung der Objekte zu den Neuronen besteht die Verwendbarkeit des Netzes anschließend darin, aktuelle (d.h. speziell: *nicht* gelernte) Objekte den passenden Neuronen zuzuordnen, sie also zu identifizieren (vgl. Bild 3). Über die Zuordnung zu den Objekt-Clustern werden sie dabei zugleich typisiert. Die in Bild 3 exemplarisch dargestellten, den Neuronen zugeordneten Prozeßmuster können etwa als Angriffe interpretiert werden, die durch das Positionen-Raster <rechts,halbrechts,mitte,halblinks,links> charakterisiert sind. Neuronen eines Clusters definieren Prozeßtypen, in denen ähnliche Prozeßmuster zusammengefaßt sind.

Vorteile ergeben sich aus der Verwendung von Neuronalen Netzen aber nicht nur unter dem Aspekt der Datenreduktion. Ein weiterer wesentlicher Vorteil ist, daß die Ergebnisdaten unmittelbar weiterverwendbar sind und ohne aufwendige Transferarbeiten für die Verwaltung in Datenbanken oder für die Manipulation in Expertensystemen zur Verfügung stehen. Und schließlich ermöglichen die schnelle Trainierbarkeit des Netzes und der direkte Zugriff auf die Prozeßmuster die interaktive Untersuchung und Bewertung spezifischer Phänomene sowohl in

der Entwicklungsphase als auch im Einsatz. Zu diesem Zweck sind nahezu alle Bereiche der Netzkonfiguration, des Netztrainings, der Mustereingabe, der Analyse und der Identifikation interaktiv mit on line-feed back manipulierbar.

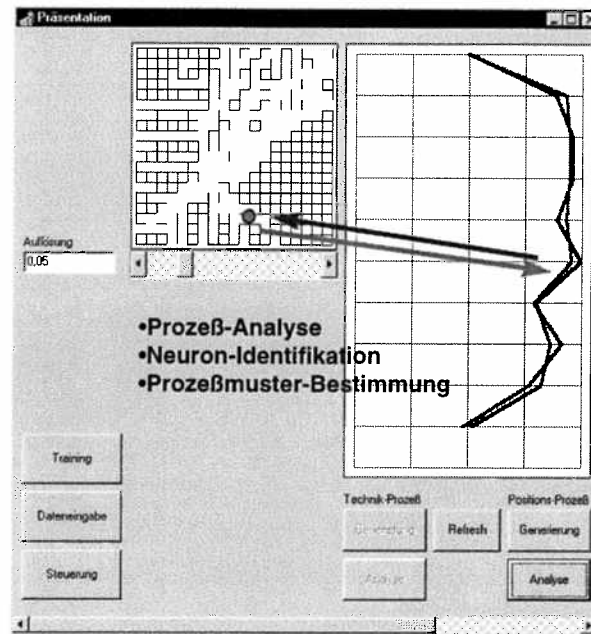


Bild 3 : Bedienoberfläche für die Identifikationskontrolle bei der Prozeßmuster-Analyse

Hilfreich wäre weiterhin eine Manipulations-Metaebene, die im Sinne eines Generators helfen würde, das benutzte Modell - d.h. hier: das durch Training generierte Netz - den Erfordernissen optimierend anzupassen. Inhalt der gegenwärtigen Projektarbeit ist es daher, wie in Bild 4 gezeigt, die Netzstruktur hinsichtlich Form und Verteilung der Cluster interaktiv zu modifizieren, um so die Typisierung der Prozesse z.B. an das Verständnis des Trainers zu adaptieren.

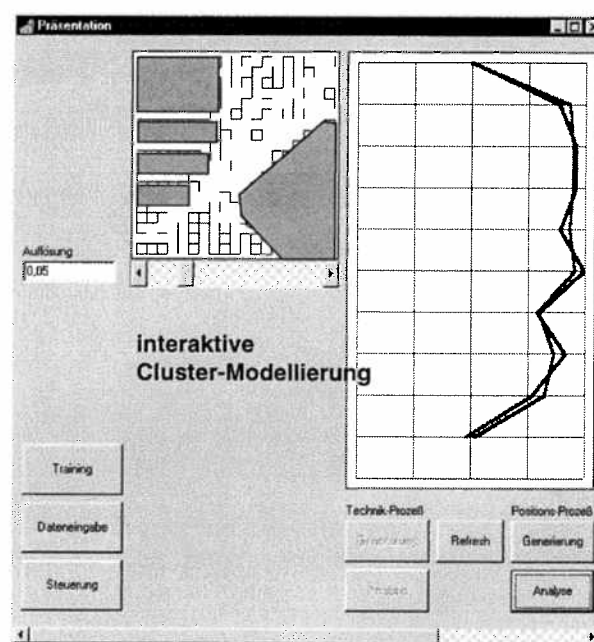


Bild 4 : Interaktive Cluster-Modellierung

3.2 Einsatz der netzgestützten Prozeßanalyse in der strategischen Spielvorbereitung

Der Einsatz des Neuronalen Netzes soll im hier angesprochenen Szenario darin bestehen, die gegnerischen Prozesse hinsichtlich ihrer strategischen Bedeutung zu identifizieren und den Typ der optimalen eigenen Gegenstrategien zu bestimmen.

Durch Abfragen an die Datenbank können Prozeß-Typen quantitativ und qualitativ selektiert werden. Denkbar ist dabei u.a., daß zunächst signifikant häufige Prozeß-Typen aufgespürt werden, deren spezielle Ausprägungen dann in graphischer oder anderer Form präsentiert werden. Für das Beispiel in Bild 5 wäre z.B. denkbar, daß eine quantitative Analyse die signifikante Häufigkeit von erfolgreichen Prozessen von Meyer und erfolglosen Prozessen von Müller ergeben hat, die nun Anlaß für eine qualitative Analyse sein sollen.

Ziel könnte es hier sein, in Interaktion mit dem System typische erfolgreiche oder erfolglose Prozeß-Strukturen hypothetisch zu benennen und - eventuell unter Einschaltung eines Expertensystems - bestätigen oder ablehnen zu lassen.

Die essentielle Intention dieser Vorgehensweise besteht darin, daß der Nutzer von aufwendigen Routinearbeiten wie Codieren, Klassifizieren oder Bewerten entlastet wird und sich in Interaktion mit dem System verstärkt qualitativen Aufgaben wie Diagnose und Ursachenanalysen zuwenden kann.

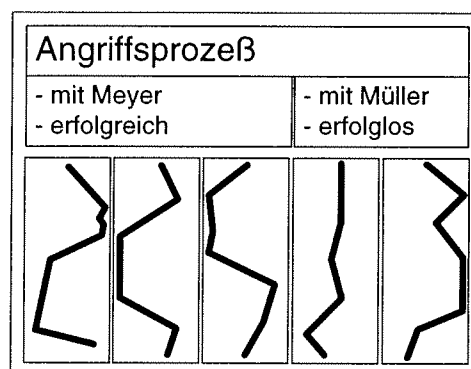


Bild 5 : Erkennung und Bewertung von Prozeßtypen

4 Das Squash-Ballweg-Modell

Zum Einsatz von Modellen im Sport gehört, wie in Kapitel 3 diskutiert, die Simulation von Prozessen. Während dort der Verhaltensprozeß im Vordergrund stand, ist hier der technische Prozeß der Ballbewegung von Interesse: Für das sog. "Lesen" des Spiels und die Antizipation seiner Abläufe ist es essentiell, daß die Ballweg-Muster bekannt sind und den Aktionen der Spieler zugeordnet werden können. Beim Squash ergibt sich dabei das Problem der Reflexion. Mit jeder Wandberührung des Balles vergrößert sich der von den Anfangsparametern abhängige mögliche Streubereich des Ballverlaufs und damit die Unschärfe in der Relation von Aktion und Resultat. Die Bilder 6 und 7 verdeutlichen dies für die Situation, daß bei minimaler Variation des Ball-Treffpunktes die jeweiligen Ballwege in extremer Weise voneinander abweichen.

Dieses natürlich aus anderen Anwendungen und auch aus der Chaosforschung bekannte Phänomen hat (nicht nur) für den Sportbereich eine wesentliche Konsequenz:

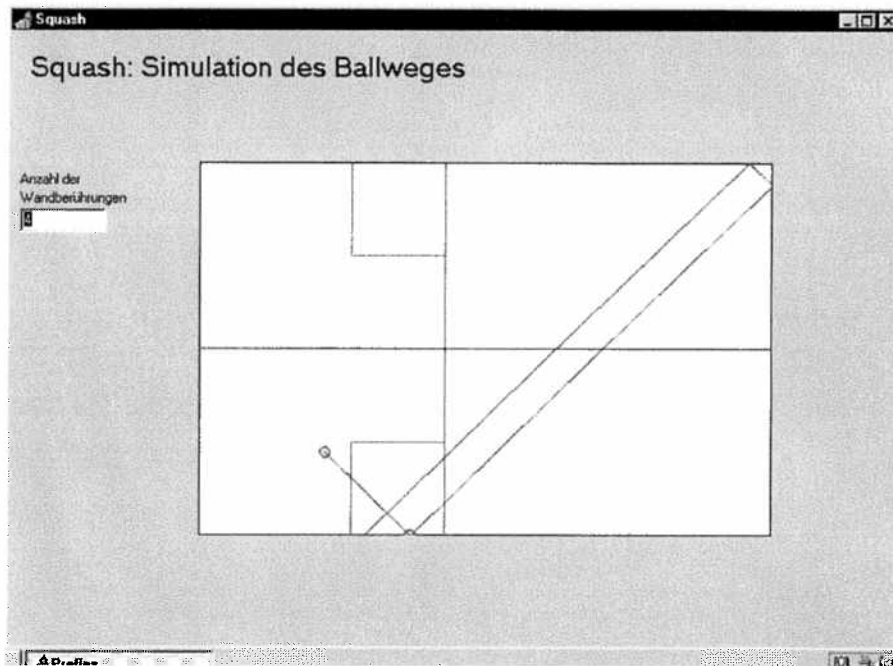


Bild 6 : Squash-Ballweg abhängig vom Treffpunkt (a)

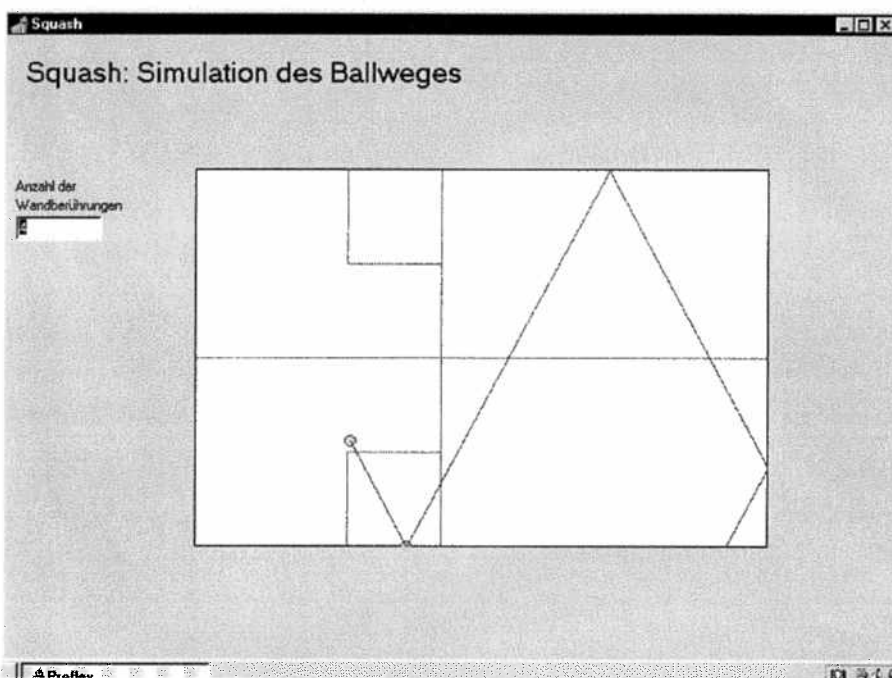


Bild 7 : Squash-Ballweg abhängig vom Treffpunkt (b)

Um den Effekt von Handlungen in einem so sensiblen Szenario deutlich und trainierbar zu machen, sind "technische" Modelle und Simulationen ungeeignet. Simulationsmodelle, die zunächst erfordern, über mehrere Fenster Parameter einzustellen, um dann in einem weiteren Fenster eine Ergebniskurve auszugeben, machen zwar den Zusammenhang zwischen Ursache und Wirkung, nicht aber den zwischen Ursachenänderung und Wirkungsänderung deutlich. Da es aber weder im Spiel noch im Training statische Situationen noch unverändert wiederholbare Handlungen gibt, sind Änderung und Variation die handlungsprägenden Aspekte, die somit auch über die Simulation verdeutlicht werden müssen.

Die im Ballweg-Modell umgesetzte Konsequenz besteht darin, daß, unter bewußtem Verzicht auf 3-dimensionale Darstellungen, die Eingaben mit Balltreffpunkt und Wandtreffpunkt auf das absolute Minimum reduziert wurden. Beide Eingaben können durch Mausklick (Positionseingabe) und durch Mausziehen (Positionsveränderung) analog ausgeführt werden und führen zu einem unmittelbaren feed back. Aus ergonomischen Gründen wurde dabei die Verwendung von Scrollbars vermieden: Da es zumindest in der von uns benutzten Entwicklungsumgebung DELPHI keine zyklischen Scrollbars gibt, hätte für jede Spielfeldseite ein separates Scrollbar verwendet werden müssen. Das hätte zu einer Segmentierung der Bedienung und zu "toten" Bereichen geführt. Stattdessen wurden die Maus-Koordinaten innerhalb eines definierten Rahmens um das Spielfeld für die Steuerung eines entsprechenden move-Ereignisses verwendet. Mit der gleichen Technik wird die kontinuierliche Veränderung der Schlagposition realisiert. Zur Fixierung definierter Positionen für Schlag und Wandberührung werden on click-Ereignisse der Maus verwendet.

Die Ergebnisse dieses Simulationsmodells haben im Kollegenkreis in der Sportwissenschaft großes Interesse ausgelöst und alte Diskussionen neu belebt: Wie kann eine Aktion, die so sensibel von Parametern abhängt, die ihrerseits in der Dynamik des Spiels überdies kaum bewußt wahrzunehmen oder zu steuern sind, in der Praxis so präzise ausgeführt werden. Ein reines Analyse-Synthese-Modell, wie etwa für das Training in technischen Disziplinen, scheint hierfür völlig unzureichend. Naheliegender erscheint die Vermutung, die auf die Prozeßmuster aus Kapitel 3 zurückführt, daß nämlich durch die häufige Ausführung von Aktionen beim Athleten komplexe Muster und Typen angelegt werden, die nicht nur durch einen Satz meßbarer Parameter, sondern durch den gesamten situativen Kontext der Aktion definiert und geprägt sind.

5 Zusammenfassung

Der langjährige Forschungskontakt mit Sportwissenschaftlern und -praktikern hat gezeigt, daß häufig nicht die Entwicklung eines Modells und die Berechnung einer Simulation die zentralen Probleme sind, sondern daß oft Akzeptanzprobleme bei der Verwendung und Unzulänglichkeiten bei der Nutzbarkeit von Simulationsmodellen die eigentlichen inhaltlichen Probleme überlagern.

Die Verfügbarkeit leistungsfähiger Entwicklungsumgebungen einerseits und die interdisziplinäre Auseinandersetzung mit Anforderungen und Realisierungsmöglichkeiten andererseits zeigen Ansätze auf, früher eher weniger beachtete Aspekte von Präsentation und Interaktion zukünftig stärker in den Vordergrund zu stellen.

Literatur

- [1] MADER, A., ULLMER, S.: Biologische Grundlagen der Belastungsanpassung der Muskelzelle - ein Beitrag zum Mechanismus der Trainingswirkung. 1993, Meyer & Meyer Verlag, Edition Sport & Wissenschaft.
- [2] PERL, J.: Möglichkeiten und Probleme der computerunterstützten Interaktionsanalyse am Beispiel Handball. *Berichte und Materialien des Bundesinstituts für Sportwissenschaft I* (1997), S. 74-89.

Mechatronic Processing Objects

C. Homburg

Carsten.Homburg@mlap.uni-paderborn.de

Universität-GH Paderborn - MLaP, Pohlweg 55, D-33098 Paderborn

Zusammenfassung

Ein Mechatronic Processing Object (MPO) ist eine autonome, direkt ausführbare verarbeitungsorientierte Modellrepräsentation für mechatronische Systeme. Ein MPO kann über den ganzen Entwicklungsprozeß hinweg als effiziente, auf die Bedürfnisse der rechentechnischen Verarbeitung wie nichtlinearer Simulation, linearer Analyse und Synthese, Optimierung, und Hardware-in-the-Loop-Anwendungen abgestimmte Modellbasis verwendet werden. Es erlaubt die Abbildung dynamisch vernetzter mechatronischer Systeme mit sowohl kontinuierlichem als auch diskretem Systemverhalten. Insbesondere wird verteilte Informationsverarbeitung berücksichtigt, sowohl in der Modellierung als auch in der Implementation.

1 Einleitung

Mechatronische Systeme sind komplexe technische Systeme, die sich aus Komponenten unterschiedlichster Disziplinen wie z. B. der Mechanik, Hydraulik, Elektrotechnik und Informationsverarbeitung zusammensetzen. Die einzelnen Komponenten werden in einer jeweils fachspezifischen Form modelliert und topologisch zu einem Gesamtsystem verknüpft. Über die Transformation in eine normierende, mathematische Darstellung, bei der die spezifizizierte hierarchische Systemstruktur erhalten bleibt, ist es sinnvoll, das Systemmodell in eine verarbeitungsorientierte Repräsentation zu überführen. Diese verarbeitungsorientierte Modellrepräsentation kann dann über den gesamten Entwicklungsprozeß [1] eines mechatronischen Systems hinweg als effiziente, auf die Bedürfnisse der rechentechnischen Verarbeitung abgestimmte Modellbasis verwendet werden. Auch auf dieser Ebene bleibt die hierarchische Modellstruktur und die eindeutige Zuordnung zur originalen Modellbeschreibung erhalten.

Als verarbeitungsorientierte Modellrepräsentation wurde am Mechatronik Laboratorium Paderborn (MLaP) die Beschreibungsform DSC (Dynamic System Code) [2] eingeführt, schwerpunktmäßig zur Behandlung blockorientierter Systeme in nichtlinearer Zustandsraumdarstellung 1. Ordnung. Zur adäquaten Behandlung von übergeordneten, intelligenten Steuerungsmechanismen wie z. B. des Verkehrsmangagements an einer Kreuzung mit Kolonnenbildung unter Berücksichtigung der Längsdynamik der einzelnen Fahrzeuge wurde dieser Ansatz um diskrete Modellierungsaspekte und dynamische, verteilt hierarchische Systemstrukturen erweitert. Grundlage dieses erweiterten Ansatzes sind die *Mechatronic Processing Objects* (MPO), die in diesem Beitrag vorgestellt werden. Als unterste, verarbeitungsorientierte Modellebene im Objektmodell für den Mechatronikentwurf [3] bilden sie die Basis für eine durchgängige, offene Entwicklungsumgebung.

Die Verwendung von MPOs bietet gegenüber derzeitig verfügbaren Entwurfswerkzeugen für mechatronische Systeme mehrere Vorteile. Jedes Teilsystem kann als eigenständiges, an den Gesamtsystemkontext sich adaptierendes, ablauffähiges Programmobjekt bereitgestellt werden. Die einzelnen Modellobjekte sind auch ohne Verfügbarkeit des Modellierungswerkzeugs direkt

ausführbar. Mechanismen zur Rückannotation ermöglichen jederzeit eine transparente Anbindung an die Modellierungsumgebung, z. B. zum Modell-Debugging. Eine schlanke, effiziente Implementierung ermöglicht den Einsatz von MPOs auch in Echtzeitanwendungen wie z. B. einer Hardware-in-the-Loop-Simulation. Die Software-Architektur der MPOs ist hochgradig portabel, einzige Voraussetzungen sind ein C-Compiler und für einige Anwendungen zusätzlich das Laden von Code zur Laufzeit z. B. in Form von dynamischen Bibliotheken (dynamic link libraries, shared libraries). Zur Unterstützung des Entwurfs einer verteilten Informationsverarbeitung ist die Erzeugung und Reorganisation hierarchisch verteilter Strukturen unter Verwendung vorhandener MPOs problemlos möglich.

2 Konzept

Ein Mechatronic Processing Object (MPO) ist die verarbeitungsorientierte Repräsentation eines autonomen, dynamischen Systems unter besonderer Berücksichtigung der Randbedingungen auf dem Gebiet der Mechatronik. Als eigenständig ablauffähiges, nebenläufiges Programmobjekt repräsentiert es ein Teilmodell eines mechatronischen Systems in Form einer ausführbaren Systemspezifikation. Dabei lassen sich sowohl kontinuierliche als auch diskrete Aspekte abbilden. Im Gegensatz zu vielen auf Spezialformen hybrider Automaten [4] basierenden Werkzeugen wie z. B. HYTECH werden die kontinuierlichen Anteile nicht in ein diskretes Abbild umgewandelt, sondern bleiben als ODE erhalten.

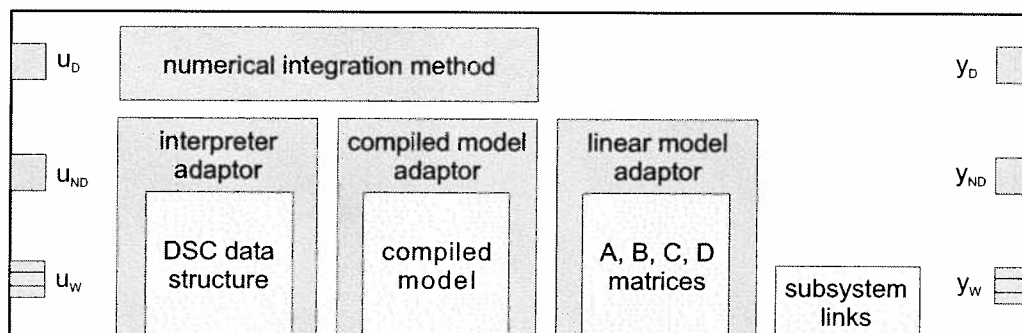


Bild 1: Mechatronic Processing Object

Ein MPO kann als Block mit gerichtetem Ein-/Ausgangsverhalten betrachtet werden (siehe Bild 1). Die ausführbare Spezifikation des Systemverhaltens kann dabei intern in den folgenden drei Repräsentationen vorliegen:

1. als DSC-Datenstruktur

Diese Repräsentation enthält alle Informationen der erweiterten DSC-Notation des entsprechenden Teilsystems inkl. der Rückannotation auf die ursprüngliche Beschreibung auf mathematischer bzw. topologischer Modellierungsebene. Diese DSC-Datenstruktur kann vom MPO-Kern interpretativ abgearbeitet oder auch modifiziert werden. Diese Repräsentation eignet sich insbesondere zum Modell-Debugging und zur Modellpartitionierung. Desweiteren bildet sie den Ausgangspunkt für die beiden weiteren Repräsentationsformen.

2. in linearer bzw. linearisierter Form als Matrizen A , B , C , D

Zur linearen Systemanalyse und zur Reglersynthese wird das Systemverhalten in linearer

bzw. linearisierter Form benötigt. Die Ermittlung dieser Matrizen kann durch symbolische Linearisierung auf Gleichungsebene und anschließende numerische Verkopplung oder durch numerische Linearisierung erfolgen. Auf diese einmal gewonnenen Darstellungen können dann sehr effizient Methoden der linearen Systemanalyse und -synthese angewendet werden.

3. *als kompilierter Code (DLL)*

Für Systemsimulationen mit langen Simulationsintervallen, Parameterstudien und insbesondere Echtzeitanwendungen ist die Laufzeitperformanz des Modells von höchster Bedeutung. Dazu bietet jedes MPO die Einbindung des Systemverhaltens als kompilierter Code an. Dieser kann zur Laufzeit geladen, entfernt oder ausgetauscht werden. Dies erfordert besonders im Hinblick auf die Vernetzung mehrerer MPOs eine hohe Flexibilität des Codes. Auch die kompilierte Repräsentation eines MPO kann sich an den Gesamtsystemkontext adaptieren, so kann z. B. zur Laufzeit die Auswertereihenfolge der Modellgleichungen verändert bzw. neu ermittelt werden.

Jedes MPO kann wahlweise eine oder auch mehrere dieser drei Repräsentationen annehmen, zur Laufzeit wechseln, hinzufügen oder auch entfernen. Zum Zugriff auf diese Repräsentationen steht jeweils ein Adapter zur Verfügung, der ebenfalls entfernt werden kann, falls eine Repräsentationsform nicht verwendet wird.

Über Koppelpunkte kann sich ein MPO mit anderen MPOs verbinden und somit dynamisch zur Laufzeit ein beliebig komplexes, strukturvariantes System aufbauen. Nach erfolgter Verkopplung kann in einem Initialisierungsschritt die Auswertereihenfolge auf Gleichungs- oder auch höherer Ebene über das Netzwerk hinweg von den MPOs ermittelt werden.

Betrachtet man ein System aus mehreren autonomen MPOs, so muß eine Infrastruktur bereitgestellt werden, über die ein MPO eine Verbindung zu weiteren MPOs aufnehmen kann. Dazu wird ein Namensdienst, der *MPONameServer* eingeführt. Jedes MPO hat einen eindeutigen Namen, üblicherweise der Name des modellierten Systems. Über eine Anfrage an den *MPONameServer* kann es die übrigen MPOs dann eindeutig adressieren.

Im folgenden werden die einzelnen Interaktionen zwischen MPOs schrittweise vorgestellt (s. Bild 2). Gegeben seien zunächst drei Systemtypen G , A , und B , die mit jeweils fachspezifischen, evtl. unterschiedlichen Modellierungswerkzeugen erzeugt wurden. Das System G sei dabei ein hierarchisches System, das ein Subsystem A_1 vom Typ A und ein Subsystem B_1 vom Typ B enthalte. Entsprechende Frontends erzeugen für jeden der drei Systemtypen mindestens eine erweiterte DSC-Beschreibung. Jeder DSC-Beschreibung kann nun ein MPO zugeordnet werden, das den entsprechenden Systemtyp repräsentiert. Dazu werden drei MPOs erzeugt, die durch Ausführen der Methode *readDSC* den jeweiligen Systemtyp laden. Danach erfolgt über *registerMPO* die Registrierung beim *MPONameServer*, der Name des MPO entspricht dem jeweiligen Systemnamen. Die Repräsentation der Subsysteme A_1 und B_1 enthält zunächst nur die Angabe, daß es sich um Instanzen der Systemtypen A bzw. B handelt, Struktur und Verhalten dieser Subsysteme sind dem MPO G noch nicht bekannt.

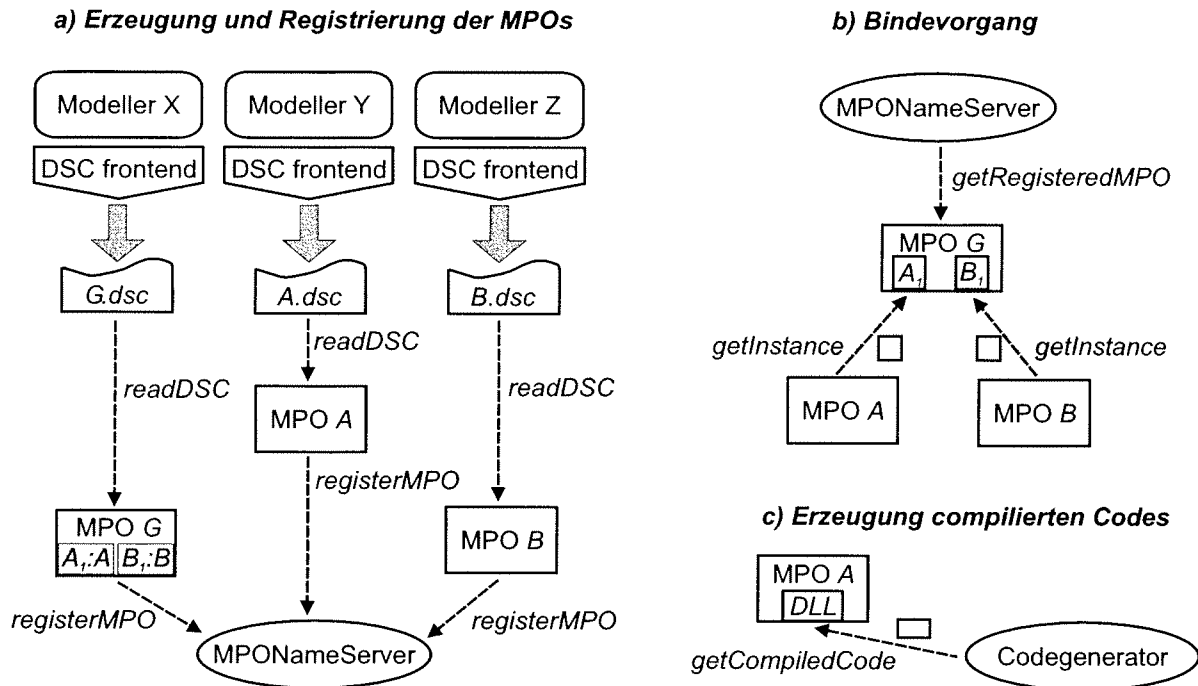


Bild 2: Interaktionen zwischen MPOs

Die Auflösung der externen Referenzen erfolgt in einem separaten Schritt, dem Bindevorgang. Die zugehörige Methode *bind* hat folgenden Ablauf: Über eine Anfrage an den MPONameServer stellt das MPO des Gesamtsystems *G* eine Verbindung zu den MPOs der Systemtypen *A* und *B* her und erhält als Kopien die Instanzen *A₁* und *B₁*, die es in seine innere Struktur einbaut. Enthält ein System mehrere Instanzen desselben Systemtyps, so werden nur die instanzspezifischen Datenbereiche mehrfach angelegt. Die Instanzen können sowohl als erweiterte DSC-Datenstruktur, als kompilierter Code als auch als lineare Systemmatrizen eingebaut werden. Die im originalen Systemtyp abgelegten Repräsentationen können dazu unverändert übernommen werden. Eine erneute Erzeugung des kompilierten Codes für das gesamte System ist nicht mehr erforderlich, kompilierte Codestücke können modular ausgetauscht werden. Dies verkürzt besonders bei größeren Systemen die Turn-around-Zeiten bei Veränderungen an Teilsystemen erheblich.

Die Erzeugung von kompiliertem Maschinencode zur Abbildung des Systemverhaltens wird durch den Dienst *Codegenerator* bereitgestellt. Dieser steht zentral (oder auch verteilt) allen MPOs zur Verfügung. Mit der Anfrage *getCompiledCode* nimmt ein MPO Verbindung zu diesem Dienst auf und stellt ihm seine erweiterte DSC-Datenstruktur bzw. seine lineare Repräsentation zur Verfügung. Der Codegenerator erzeugt daraus C-Programmcode, der mit Hilfe handelsüblicher Compiler übersetzt und als dynamisch ladbares Objekt (z. B. DLL, shared library) dem MPO zur Verfügung gestellt wird. Der erzeugte Code ist modular und kann sich an den Gesamtsystemkontext adaptieren. Dies ist insbesondere für Simulationsanwendungen von großer Bedeutung.

3 Nichtlineare Simulation

Blockorientierte Modellierungsformen beschreiben das Systemverhalten in Form von gerichteten Ein-/Ausgangsbeziehungen. Bei der Systemsimulation ist dafür zu sorgen, daß keine al-

gebraischen Schleifen auftreten, die gerichteten Beziehungen über alle Systemverkopplungen hinweg also zyklensfrei sind, anderenfalls gibt es keine korrekte Auswertereihenfolge der Systemgleichungen. Je nach Verkopplung der einzelnen Teilsysteme ergeben sich unterschiedliche Abhängigkeiten und damit auch jeweils andere Auswertereihenfolgen. Hieraus läßt sich für eine wirklich modulare Repräsentation eines Teilsystems die Anforderung ableiten, daß die innere Auswertereihenfolge nicht starr codiert sein darf, sondern in Abhängigkeit von der Struktur des Gesamtsystems anpaßbar sein muß. Diese Anforderung wird von den Codegeneratoren derzeit verfügbarer Werkzeuge wie z. B. MATRIX/X und Matlab/Simulink nicht erfüllt. Soll bei ihnen für ein Teilsystem separat Code erzeugt werden, so entsteht ein großer Block (*black box*) mit jeweils einem Eingangs- und einem Ausgangsvektor, der pro Simulationsschritt durch einmaliges Einlesen des gesamten Eingangsvektors und einmaliges Ausgeben des kompletten Ausgangsvektors ausgewertet werden muß. Eine durch den Verkopplungskontext evtl. erforderliche sequentielle Auswertung einzelner, interner Übertragungspfade ist durch diesen monolithischen Zugang nicht möglich, obwohl es nach der inneren Struktur des Modells zulässig wäre. Als Ausweg müssen dann entweder Verzögerungsblöcke eingeführt werden, die einen zusätzlichen, kritischen Phasenabfall bewirken können, oder es müssen viele feingranulare Blöcke erzeugt und die Verkopplungen nachgetragen werden.

Um die Anforderungen einer modularen, verkopplungssensitiven, aber dennoch laufzeiteffizienten Simulationsumgebung zu erfüllen, unterteilt ein MPO seine Ein- und Ausgangsgrößen in drei Kategorien:

Nichtdurchgriffsgrößen (u_{ND}, y_{ND}) Zu dieser Kategorie zählen alle Ausgangsgrößen, die nicht direkt von Eingangsgrößen abhängen, das heißt, ihr Wert zu einem Zeitpunkt t läßt sich allein aus den Zustandswerten zum Zeitpunkt t ermitteln. Eine Veränderung der Eingangsgrößen greift hier nicht direkt auf den Systemausgang durch. Die zugehörigen Gleichungen können zu Beginn eines jeden Simulationsschritts sofort ausgewertet und die Nichtdurchgriffsausgänge unmittelbar ausgegeben werden. Der diesen Größen zugeordnete Teil eines MPOs ist für die Gewährleistung einer korrekten Auswertereihenfolge unproblematisch.

Durchgriffsgrößen (u_D, y_D) Zu dieser Kategorie gehören alle Eingangsgrößen, die ohne Zeitverzug direkt auf mindestens eine Ausgangsgröße wirken. Die jeweiligen Ausgangsgrößen sind ebenfalls hier einzuordnen. Für die korrekte Abarbeitung der zugehörigen Gleichungen muß eine Auswertereihenfolge ermittelt werden. Dzu stößt ein MPO die Methode *calcEvaluationOrder* an, über die miteinander vernetzten MPOs wird eine verteilte, umgekehrte topologische Sortierung gestartet. Pro MPO wird dabei der gesamte den Durchgriffsgrößen zugeordnete Code als ein Block betrachtet. Die Durchgriffsgrößen des MPOs werden also pro Auswertung einmal als ganzer Vektor eingelesen bzw. ausgegeben. Wie die Implementation von DSL [5] zeigt, läßt sich mit der Zweiteilung des Codes zur Bestimmung der Ausgangsgrößen in einen Durchgriffs- und einen Nichtdurchgriffsteil bei einer Vielzahl von Systemen eine korrekte Auswertereihenfolge ermitteln.

White-box-Größen (u_W, y_W) Repräsentiert ein MPO jedoch ein komplexes Teilsystem mit vielen Subsystemen, so läßt sich mit der vektoriellen Zusammenfassung aller Durchgriffs-

eingänge bzw. -ausgänge pro MPO keine Auswertereihenfolge mehr ermitteln. Für solche Fälle bietet ein MPO die White-box-Größen an. Diese Größen können jederzeit „zweischendurch“ kommunizieren. Für problematische Durchgriffsgrößen kann ein Wechsel in diese Kategorie vorgenommen werden, so daß sich spätestens auf dieser Ebene eine Auswertereihenfolge findet läßt.

Die freie Wahl einer beliebigen Auswertereihenfolge stellt im besonderen an den compilierten Code einige Anforderungen. Einzelne Berechnungspfade innerhalb eines MPO müssen getrennt voneinander ausführbar sein. Aus Effizienzgründen, insbesondere auch für Echtzeitanwendungen, sollte die Auswertereihenfolge für jedes MPO während der Simulation statisch bestimmt sein und nicht bei jedem Simulationsschritt erst datengetrieben ermittelt werden. Die MPO-Repräsentation des Systemverhaltens als kompilierter Code stellt dazu die einzelnen Berechnungspfade über einen Sprungverteiler zur Verfügung, so daß ein sehr schneller Zugriff auf den auszuführenden Code gewährleistet ist. In einem Initialisierungsschritt vor dem Start der Simulation wird entsprechend der durch *calcEvaluationOrder* ermittelten Auswertereihenfolge für jedes MPO eine lokale Schedule-Tabelle aufgebaut, die auf die einzelnen Einträge im Sprungverteiler verweist. Diese Tabelle kann sehr effizient abgearbeitet werden, der Performanzverlust gegenüber einer festcodierten Auswertereihenfolge ist sehr gering.

Die zur Lösung der in Form von Differentialgleichungen beschriebenen Aspekte des Systemverhaltens erforderlichen numerischen Löser lädt ein MPO ebenfalls dynamisch. Dazu existiert eine einheitliche, modellunabhängige Schnittstelle, so daß die Löser beliebig ausgetauscht werden können. MPOs gestatten sowohl Zentralintegration (auf einem MPO oder verteilt) als auch Multi-Rate- und Multi-Method-Verfahren.

4 Entwurf verteilter Informationsverarbeitung

Komplexe, mechatronische Systeme enthalten eine Vielzahl von lokalen Reglern, die jeweils für die korrekte Funktion einer Teilkomponente sorgen. Durch hierarchische Organisation dieser Regler und zusätzliche, übergeordnete Reglerstrukturen wird die gewünschte Funktionalität des Gesamtsystems hergestellt. Dazu wurde am MLaP eine informationstechnische Klassifizierung von Systemen in MFM (Mechatronisches Funktionsmodul), AMS (Autonomes Mechatronisches System) und VMS (Vernetztes Mechatronisches System) eingeführt [6]. Zur Unterstützung der verteilten Informationsverarbeitung in Entwurf und Realisierung bieten MPOs das Ordnungsprinzip der Cluster.

Beim Entwurf eines mechatronischen Systems wird zunächst unter Anwendung der Subsystemtechnik ein hierarchisches Modell aus einzelnen bereits im Rahmen eines Modellkatalogs verfügbaren oder auch neu zu entwerfenden Komponenten erstellt. Diese Subsystemhierarchie ist im MPO in einer Datenstruktur *SpecificationTree* abgelegt. Zusätzlich zur spezifizierten Subsystemhierarchie können zur Abbildung der tatsächlichen informationstechnischen Struktur hierarchische Cluster gebildet werden. Dazu wird eine Struktur *ClusterNode* eingeführt: Zunächst ist ein Clusterknoten einfach eine Ansammlung von Subsystemen, die über eine gemeinsame informationstechnische Anbindung mit den restlichen Komponenten des Systems verbunden sind. Die einzelnen Clusterknoten können wieder zu neuen Clustern zusammenge-

faßt werden. Es ergibt sich eine hierarchische Struktur, die die Informationsverarbeitung widerspiegelt. Die einzelnen Hierarchieebenen können dann als Prozesse, Prozessoren, MFMs, AMS bzw. VMS interpretiert werden, die Verwendung weiterer Hierarchieebenen ist ebenfalls möglich.

Die Clusterhierarchie und die in der Modellbildung spezifizierte Systemhierarchie sind zunächst voneinander völlig unabhängig. So können z. B. einige Komponenten eines Teilsystems einem anderen Cluster zugeordnet sein. Dies erlaubt eine hohe Flexibilität beim Entwurf einer geeigneten informationstechnischen Struktur. Eine Veränderung der Informationsverarbeitung erfordert zunächst keine Modifikationen am ursprünglichen Modell, damit lassen sich ohne großen Aufwand unterschiedlichste informationstechnische Strukturen modellieren. Hat man eine geeignete, modulare Struktur der Informationsverarbeitung gefunden, so stellt diese auch einen Anhaltspunkt für ein sinnvollen Aufbau der Subsystemhierarchie dar. Aufgrund dieser Erkenntnisse sollte dann auch die Subsystemhierarchie entsprechend angepaßt werden, sinnvollerweise mit entsprechender Werkzeugunterstützung.

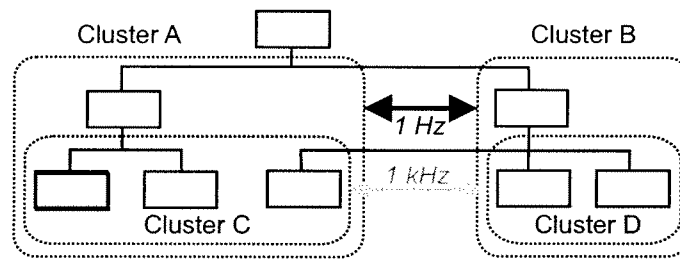


Bild 3: Strukturierung der Informationsverarbeitung mit Hilfe von Clustern

In Bild 3 ist eine Clusterbildung beispielhaft dargestellt. Die hierarchisch organisierten Cluster können unterschiedlichen *Kommunikationsebenen* zugeordnet sein. Eine Kommunikationsebene wird durch die Frequenz charakterisiert, mit der der Datenaustausch zwischen den Clustern auf dieser Ebene durchgeführt wird. So befinden sich Cluster C und D z. B. auf einer Kommunikationsebene mit der Frequenz 1 kHz, d. h. zwischen C und D werden im Takt von 1 ms Daten ausgetauscht. Auf der übergeordneten Ebene mit den Clustern A und B findet nur ein Datenaustausch im Sekundentakt statt. Die Ein- bzw. Ausgangsgrößen der Subcluster C und D sind auf dieser Ebene als White-box-Größen herausgeführt.

5 Anwendungen und Ausblick

Das hier vorgestellte Konzept der Mechatronic Processing Objects wurde für das Beispiel eines intelligenten Kreuzungsmanagements [7] auf einer Implementierung unter Windows NT angewendet. Für verteilte Echtzeitanwendungen ist eine Anpassung der Laufzeitplattform IPANEMA [8] in Arbeit. Die dynamische Erzeugung neuer MPOs unter harten Echtzeitbedingungen ist allerdings problematisch, hier besteht noch Forschungsbedarf. Zur optimalen Werkzeugunterstützung zur Bildung von Clustern und der Restrukturierung der Informationsverarbeitung ist ein auf MPOs angepaßtes, graphisches Werkzeug für Partitionierungsaufgaben in Entwicklung. Damit steht dem Anwender dann eine leistungsfähige Entwicklungsumgebung für komplexe informationstechnische Strukturen vernetzter mechatronische Systeme zur Verfügung.

Literatur

- [1] M. Hahn, J. Lückel, G. Witter. Eine Entwurfsmethodik für mechatronische Systeme. *Magdeburger Maschinenbau-Tage — Entwicklungsmethoden und Entwicklungsprozesse im Maschinenbau*, 1997.
- [2] J. Richert, C. Homburg, A. Engelke. DSC (Dynamic System Code) — Eine abstrakte blockorientierte Beschreibungssprache für mechatronische Systeme. 8. *Symposium Simulationstechnik*, S. 223–228, Berlin, September 1993.
- [3] M. Hahn, J. Lückel, R. Naumann, R. Rasche. Ein Objektmodell für den Mechatronikentwurf. 12. *Symposium Simulationstechnik*, 1998.
- [4] R. Alur, C. Courcoubetis, T.A. Henzinger, P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. *Hybrid Systems I*, Lecture Notes in Computer Science 736, S. 209–229. Springer, 1993.
- [5] J. Schröder. *Eine Modellbeschreibungssprache zur Unterstützung der Simulation und Optimierung von nichtlinearen und linearisierten hierarchischen Systemen*. Fortschr.-Ber. VDI Reihe 20, Nr. 128. VDI-Verlag, Düsseldorf, 1994.
- [6] U. Honekamp, R. Stolpe, R. Naumann, J. Lückel. Structuring Approach for Complex Mechatronic Systems. *30th International Symposium on Automotive Technology & Automation — Mechatronics/Automotive Electronics*, S. 165–172, Florence, Italy, 16th–19th June 1997.
- [7] J. Lückel, R. Rasche, R. Naumann. Design of the Mechatronic System Of Interconnected Traffic Vehicles. *Control and Configuration aspects of mechatronics*, S. 63–76, TU Ilmenau, 21.-26. Sept. 1997.
- [8] U. Honekamp. IPANEMA - Verteilte Echtzeit-Informationsverarbeitung in mechatronischen Systemen. Fortschr.-Ber. VDI Reihe 20, Nr. 267. VDI-Verlag, Düsseldorf, 1998.

Simulation eines Common Rail Dieselmotors auf Basis eines ASCET Hardware in the Loop Systems.

Prof. Dr.-Ing. Hermann Kull,
Dipl.-Ing. (FH) Micha Münzenmay, Dipl.-Ing. (FH) Stephan Gohl
TZ System- und Simulationstechnik / Automotive
FHT Esslingen, Flandernstr. 101, 73732 Esslingen

Zusammenfassung

Ziel dieses Projektes war der Aufbau eines Systems, das einen Dieselmotor mit einem Common Rail Einspritzsystem (=CR) simuliert. Die Simulation wurde auf der speziellen Hardware in the Loop Umgebung ASCET-RS von der Firma ETAS implementiert. Es werden im Wesentlichen zwei Regelkreise, der Drehzahlregelkreis und der Hochdruckregelkreis geschlossen. Das bedeutet, das System läßt sich direkt an konventionelle CR-Motorsteuergeräte anschließen.

Die Genauigkeit des Modells läßt die Auslegung des Raildruckreglers und auch des Leerlaufreglers zu.

Diese Hardware in the Loop Simulation wird in Seminaren verwendet, deren Ausbildungsinhalt vornehmlich die elektronische Dieselregelung (EDC) ist [2]. Eine didaktisch wichtige Voraussetzung für den Ablauf der Übungen ist die Möglichkeit, mit einem geschlossenen Regelsystem, wie es in einem Fahrzeug vorhanden ist, arbeiten zu können.

1 Das Common Rail System

Der Hauptvorteil des Common Rail Systems ist die Möglichkeit, innerhalb der gegebenen Systemgrenzen den Einspritzdruck unabhängig von Drehzahl und Last frei wählen zu können. So kann der Druckbedarf für eine optimale Gemischbildung angepaßt werden.

Zu den Vorteilen der freien Raildruck- und Spritzbeginnwahl kommt noch die Möglichkeit der Vor- und Nacheinspritzung hinzu. Dadurch lassen sich die Emissionen (Abgas wie auch Schall) nachhaltig verbessern.

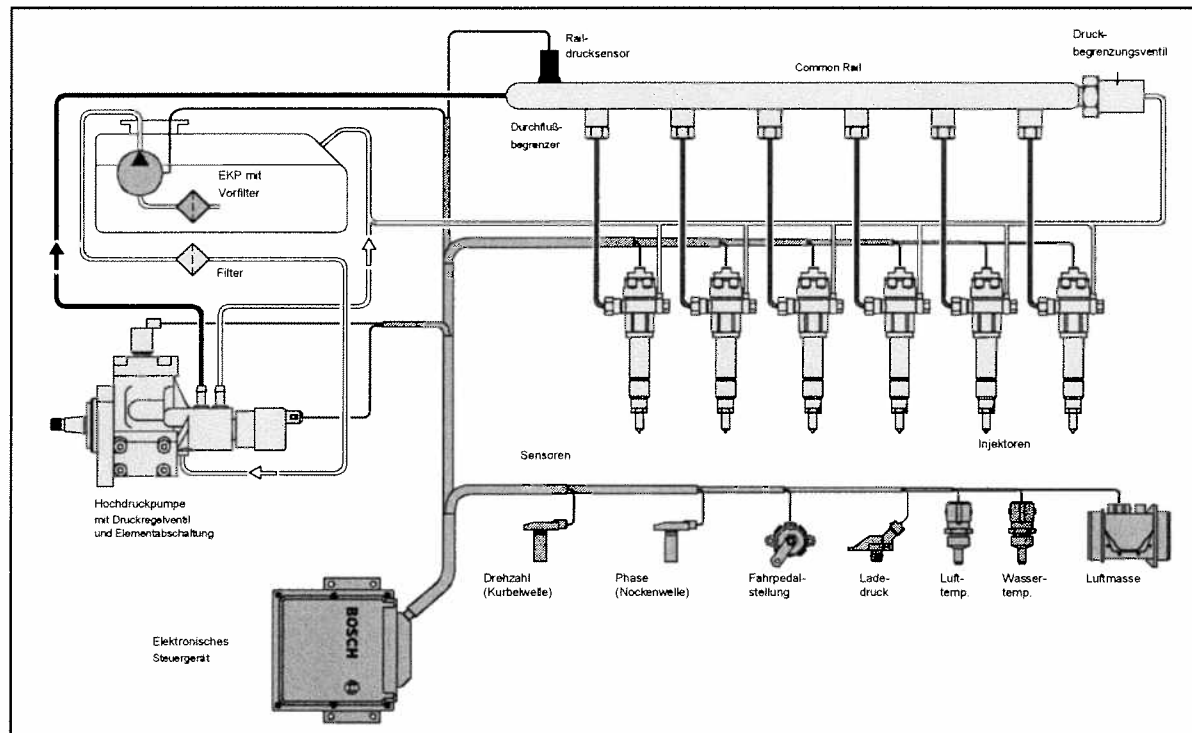


Bild 1: CR-System für einen Sechszylindermotor

Das obige Systemschaubild zeigt ein Common Rail System mit der Hochdruckpumpe CP1. Dieses System wurde für die Simulation zu Grunde gelegt.

2 Das Simulationswerkzeug ASCET-RS

„Simulation ist die Nachbildung eines dynamischen Prozesses in einem Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“ [VDI-Richtlinie 3633]

ASCET-RS steht für **A**dvanced **S**imulation and **C**ontrol **E**ngineering **T**ool-**R**ealtime **S**imulator. Dies ist eine integrierte Entwurfs- und Rapid-Prototyping-Umgebung für nichtlineare Systeme. Es können damit dynamische Systeme grafisch modelliert und in Echtzeit simuliert werden. Die Parameter sind während der Simulation veränderbar.

Kern von ASCET ist das sogenannte Multi-Transputersystem (MTS), daß mit einer Transputerkarte, die mit mehreren Transputern bestückt ist, arbeitet. Das MTS unterstützt das parallele Berechnen der in dem Modell verwendeten Blöcke; deren Parallelität und Verteilung auf die Transputer ist maßgebend für die Rechenzeit.

Die Modellbildung in ASCET erfolgt blockorientiert in einer eigenen Blockbeschreibungssprache. Außer den in der Standardbibliothek vorhandenen Blöcken können mit dem Blockschaltbildeditor eigene Blöcke erstellt werden. Zur besseren Übersichtlichkeit bei komplexen Modellen können Blöcke hierarchisch in Subsysteme zusammengefaßt werden. Mit einer

einheitlichen Schnittstellendefinition können so auch mehrere Blöcke aus unterschiedlichen Modellen zu neuen Systemen zusammengefügt werden, was dem heutigen Trend zum modularen Aufbau von Systemen entspricht.

3 Das Hardware in the Loop System

Übersicht Laborarbeitsplatz:

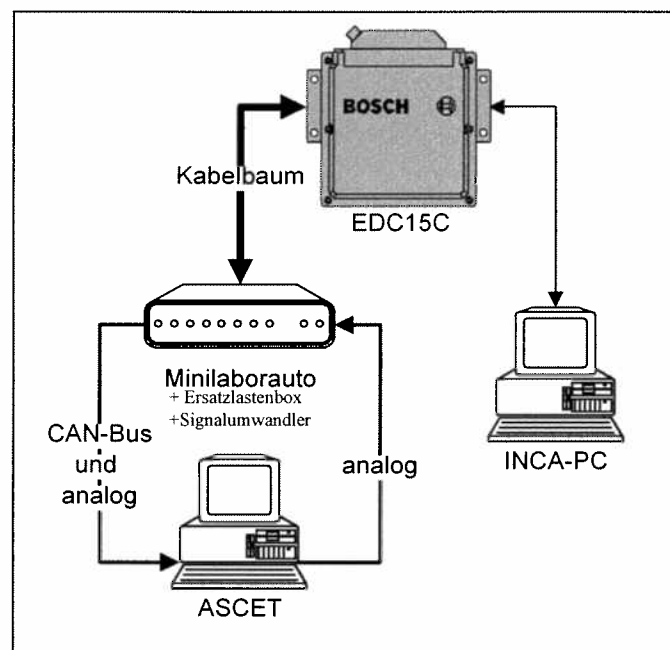


Bild 2: Aufbau eines Laborarbeitsplatzes

Das Bild zeigt einen kompletten Laborarbeitsplatz, wie er in den Seminaren verwendet wird. Der ASCET Rechner kommuniziert mit dem Steuergerät über analoge Signale (z.B. Raildruck) und auch über CAN Messages (Sollmoment, Reibmoment, etc.). Im Minilaborauto werden die wichtigsten Sensoren wie Wassertemperaturfühler oder Pedalwertgeber in Hardware nachgebildet. Dort kann über Potentiometer der Messwert des entsprechenden Sensors verändert werden. Der sog. INCA-PC dient zur Änderung und Messung von steuergeräteinternen Daten wie beispielsweise der Leerlaufsolldrehzahl. Hierzu wird eine spezielle Applikations Software (INCA) verwendet.

3.1 Drehzahlregelkreis

Eine physikalisch orientierte Nachbildung der Drehzahlregelstrecke ist sehr aufwendig. Daher wurde die Nachbildung dieses Regelkreises als Kennfeld realisiert. Das Kennfeld wurde durch am realen Fahrzeug durchgeführte Messungen parametrisiert.

Da die zur Identifikation notwendigen Sprünge im Fahrzeug am einfachsten durch

Manipulation des Parameters Einspritzmenge durchgeführt werden, wird diese als Eingangsgröße verwendet und nicht das Drehmoment.

3.2 Druckregelkreis

Die Druckregelstrecke wurde mit einem PT1-Glied für das dynamische und einer Kennlinie für den stationären Endwert modelliert. Die Parameter für das Kennfeld wurden durch eine Reihe von Drucksprüngen gewonnen.

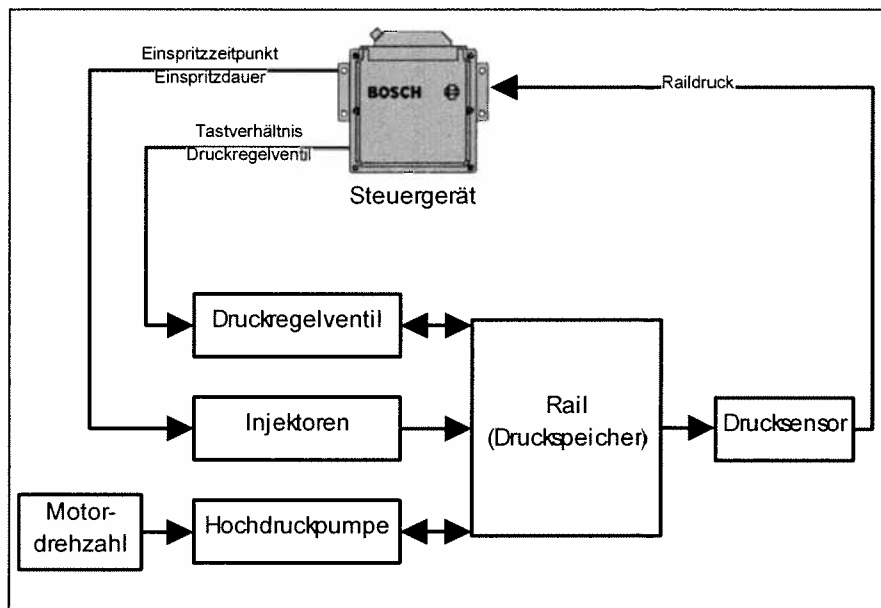


Abbildung 3: Druckregelkreis im Fahrzeug

Die Beschreibung der Vorgehensweise zur Parametrierung der Modelle ist [1] zu entnehmen.

3.3 Die Bedienoberfläche



Bild 4: Die Bedienoberfläche des Modells

Die Hauptbestandteile des Modells sind Eingang, Drehzahlregelstrecke, Druckregelstrecke, Antriebstrang und Ausgang. Im Eingangsblock sind alle Funktionen zum Einlesen der notwendigen Signale enthalten. Dieser Block enthält zwei Bereiche. Erstens den CAN-Block in welchem die über den CAN-Bus übertragenen Signale eingelesen werden. Zweitens den Analog-Block in dem die Signale, die über die Analog-In-Karte eingelesen werden, enthalten sind. Mit der Analog-Karte wird das Tastverhältnis zur Ansteuerung des Druckventils eingelesen. Dieses Signal liegt am Eingang als analoges Spannungssignal vor.

Mit dem Schalter Rail_broken läßt sich ein defektes Rail simulieren. Es wird ein starker Druckverlust simuliert, der durch z. B. eine plötzlich auftretende Leckage entstehen würde. Mit Gear und Clutch können Schaltvorgänge des Antriebstranges simuliert werden.

Zusammenfassung

Insgesamt wird das dynamische Verhalten des realen Motors für diesen Verwendungszweck ausreichend genau dargestellt. Die Struktur des Modells hat den Vorteil, daß es sehr einfach auf einen anderen Motor, z. B. einen Nutzfahrzeugmotor abgeändert werden kann. Dazu wären lediglich die Drehzahl- und Druckregelstrecke zu identifizieren und diese neuen Parameter im Modell zu implementieren.

Literatur

- [1] Günter Hay, Entwicklung, Parametrisierung und Validierung eines CR Hardware in the Loop Systems, Diplomarbeit 1998
- [2] EDC-CR Seminar der Firma Robert Bosch GmbH.

Möglichkeiten und Grenzen der Optimierung biologischer Abwasserreinigungsanlagen mit Hilfe der Simulation

J. Jungblut, D.P.F. Möller

{jungblut, moeller}@informatik.tu-clausthal.de

Institut für Informatik, Technische Universität Clausthal,

Julius-Albert-Str. 4, D-38678 Clausthal-Zellerfeld

Zusammenfassung

Die Simulation biologischer Abwasserreinigungsanlagen ist in den letzten Jahren speziell für Anlagen zur Behandlung kommunaler Abwässer zu einem wichtigen Werkzeug in Planung und Betriebsoptimierung herangereift. Dies gilt insbesondere für die Untersuchung dynamischer Aspekte, die sich aus der Variabilität des Kläranlagenzulaufs einerseits und der Anpassungsfähigkeit der Anlagenbiologie andererseits ergeben. Trotz derartiger, einen gleichmäßigen Anlagenbetrieb störenden Faktoren ist eine möglichst sichere Einhaltung der gesetzlich geforderten Grenzwerte anzustreben, welche das primäre Ziel einer Optimierung ist. Die Möglichkeiten und Grenzen der Optimierung biologischer Abwasserreinigungsanlagen mit Hilfe der Simulation werden im Beitrag am Beispiel eines dreistufigen Abwasserreinigungsverfahrens aufgezeigt.

1 Einleitung

Abwasser unterliegt ständigen Schwankungen, die im Tages-, Wochen- und Jahresrhythmus zu beobachten sind. Zusammen mit Stoßbelastungen führt dies zu einer immer wieder wechselnden Zusammensetzung des Abwassers. Um dennoch die steigenden Anforderungen an die Qualität des gereinigten Wassers erfüllen zu können, wurden neuartige, oft mehrstufige Abwasserreinigungsverfahren entwickelt, die durch komplexe Interaktionen zwischen biologischen, chemischen und physikalischen Prozessen gekennzeichnet sind.

Als nützliche Werkzeuge für den Umgang mit dieser Komplexität haben sich gerade in den letzten Jahren Modellbildung und Simulation herausgestellt. Insbesondere bietet die Simulation über die in der Auslegung von Kläranlagen bislang vielfach verwendeten statischen Berechnungsansätze hinaus die Möglichkeit, die dynamischen Aspekte der Abwasserreinigung zu untersuchen. Auf diese Weise können sowohl in Planung und Betrieb Kosten gespart werden, ohne die Anforderungen an die Betriebsstabilität des Verfahrens zu vernachlässigen.

Das Grundprinzip der biologischen Abwasserreinigung besteht darin, daß die im Abwasser enthaltenen Schadstoffe von Bakterien als Nährstoffe genutzt und in umweltverträgliche Sub-

stanzen umgewandelt werden. Als wichtigstes Verfahren der biologischen Abwasserreinigung gilt das sogenannte Belebtschlammverfahren, das eine technische Umsetzung des natürlichen Selbstreinigungsprozesses darstellt und dessen schematischen Aufbau Bild 1 zeigt.

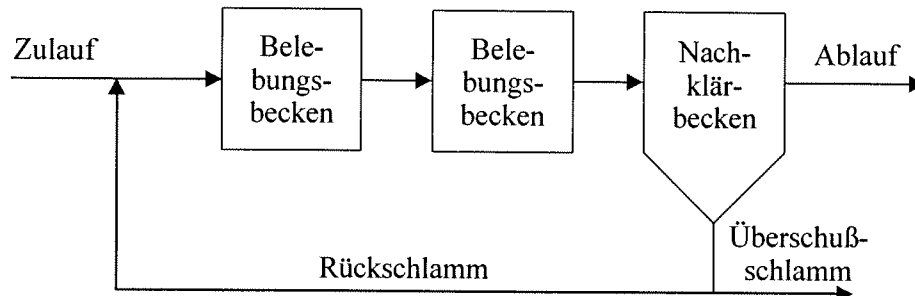


Bild 1: Schema des Belebtschlammverfahrens

In den teilweise belüfteten oder mit Rührvorrichtungen versehenen Belebungsbecken kommt das Abwasser mit dem sogenannten Belebtschlamm, der überwiegend aus Bakterien besteht, in Kontakt, so daß die angesprochenen Umwandlungsprozesse ablaufen können. Ein nachgeschaltetes Nachklärbecken sorgt unter Ausnutzung von Sedimentationsvorgängen für die Trennung von gereinigtem Wasser und belebtem Schlamm. Letzterer wird zum überwiegenden Teil in die Belebungsbecken zurückgeführt, um die Schlammkonzentration zu erhöhen. Da unter normalen Betriebsbedingungen ein Bakterienwachstum zu beobachten ist, muß von Zeit zu Zeit oder kontinuierlich Überschussschlamm aus der Anlage entfernt werden.

2 Modellierung

Als international anerkannter Standard für die Modellierung des Belebtschlammverfahrens kann das von einer Arbeitsgruppe der IAWQ (International Association on Water Quality) entwickelte Activated Sludge Model No.1 (ASM1) angesehen werden [1]. Das ASM1 ist auf die Modellierung kommunaler Abwasserreinigungsanlagen ausgerichtet, berücksichtigt den Abbau von Kohlenstoffverbindungen sowie die Stickstoffelimination mit Nitrifikation und Denitrifikation und unterliegt einer ständigen Weiterentwicklung [2, 3, 4]. Unter Nitrifikation wird hier die Umwandlung von Ammoniumstickstoff in Nitratstickstoff verstanden, welcher im Prozeß der Denitrifikation in gasförmigen Stickstoff umgewandelt wird.

Da es aus Gründen der Handhabbarkeit nicht sinnvoll ist, alle chemisch nachweisbaren Abwasserinhaltsstoffe im Modell zu berücksichtigen, werden nur wenige Größen betrachtet, die sich zum Teil aus der Aggregation verschiedener Substanzen mit ähnlichen Eigenschaften ergeben. Auf diese Weise enthalten Modelle biologischer Abwasserreinigungsanlagen normalerweise die folgenden drei Gruppen von Modellgrößen:

- Bakterien: Die Bakterien werden in der Regel entsprechend ihrem Stoffwechsel in autotro-

phe und heterotrophe Bakterien unterteilt, das heißt, die Vielzahl unterschiedlicher Bakterienarten wird in lediglich zwei Modellgrößen zusammengefaßt.

- organische Verbindungen: Hierunter fallen überwiegend Kohlenstoffverbindungen, die in biologisch leicht abbaubare, biologisch schwer abbaubare und inerte, das heißt, biologisch nicht abbaubare Stoffe unterteilt werden. Der Grad der Abbaubarkeit bezieht sich dabei auf die Aufenthaltsdauer der Substanzen in der Anlage.
- Stickstoffverbindungen: Als wesentliche Stoffe sind hier Ammonium- und Nitratstickstoff zu nennen, detailliertere Modelle berücksichtigen darüber hinaus noch Nitritstickstoff und gasförmigen Stickstoff.

Die Modellgrößen aller drei Gruppen werden als Konzentrationen aufgefaßt. Konzentrationsänderungen kommen im Fall der Bakterien durch Wachstums- und Zerfallsprozesse und bei den anderen Substanzen durch Umwandlungsprozesse zustande. Die Abhängigkeit der Prozeßgeschwindigkeiten von Einflußgrößen wie Substratkonzentration und Konzentration an gelöstem Sauerstoff wird in der Regel durch die auf Monod zurückgehende einfache hyperbolische Beziehung $c/(K_s+c)$ zum Ausdruck gebracht, wobei c für die entsprechende Konzentration steht und K_s als Sättigungskoeffizient bezeichnet wird [5]. Wird bei den Belebungsbecken vom Typ des idealen Rührkessels ausgegangen, das heißt, wird eine vollständige Durchmischung angenommen, so lassen sich die Konzentrationsänderungen durch gewöhnliche Differentialgleichungen beschreiben.

Die ASM-Modelle dienen als Basis für die Modellbildung eines dreistufigen biologischen Abwasserreinigungsprozesses, dessen schematischen Aufbau Bild 2 zeigt. Das dreistufige Verfahren entspricht hinsichtlich der Verfahrensführung einer nachgeschalteten Denitrifikation und wurde zur Behandlung kommunaler Abwässer eingesetzt [6].

In der ersten Stufe erfolgt eine weitgehende Kohlenstoffelimination nach dem Belebtschlammverfahren. Eine reduzierte Belüftung führt dazu, daß ein Teil der im Zulauf vorhandenen gelösten organischen Kohlenstoffverbindungen an die Bakterien adsorbiert wird. In der zweiten Stufe findet eine Nitrifikation in einem Festbettreaktor statt, während in der dritten Stufe eine Denitrifikation nach dem Belebtschlammverfahren durchgeführt wird.

Die Besonderheit dieses Verfahrens ist der Bypass von der ersten zur dritten Stufe, mit dessen Hilfe die adsorbierten Kohlenstoffverbindungen für die Denitrifikation bereitgestellt werden. Der Bypassvolumenstrom sollte einerseits möglichst klein sein, damit möglichst wenig Ammoniumstickstoff an der Nitrifikationsstufe vorbei in den Ablauf gelangt. Andererseits ist aber eine für eine weitgehende Denitrifikation ausreichende Menge Kohlenstoff in die dritte Stufe zu transportieren.

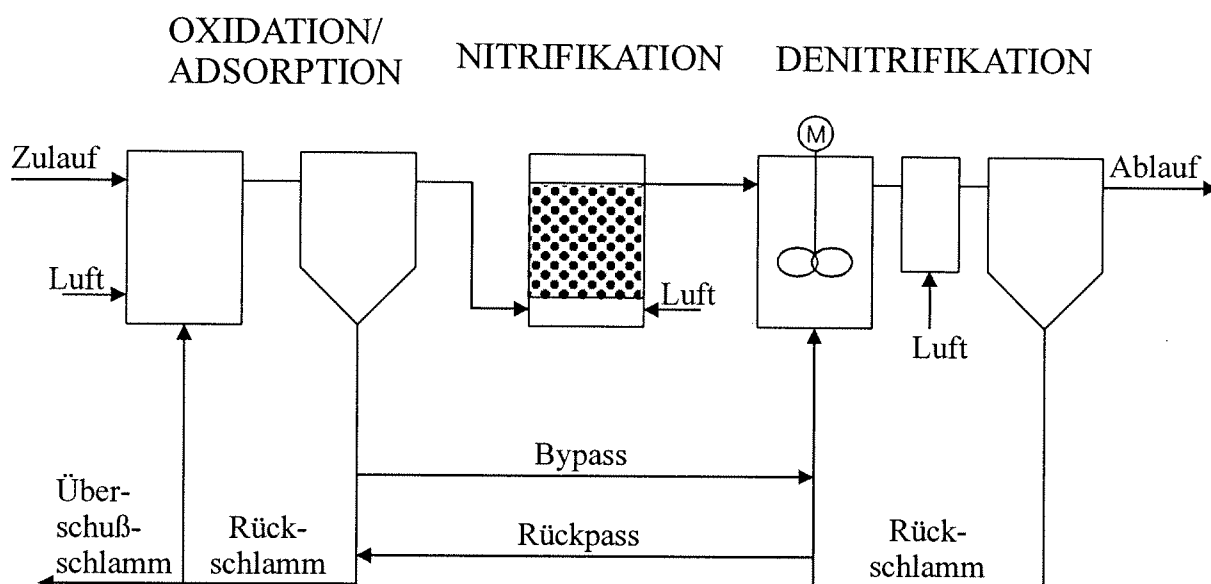


Bild 2: Schema des dreistufigen Verfahrens

Der speziellen Charakteristik des betrachteten Verfahrens wurde in der Modellbildung unter anderem dadurch entsprochen, daß eine Modellgröße für adsorbierte Stoffe und dazugehörige Umwandlungsprozesse in das Modell aufgenommen wurden.

3 Simulation

Grundvoraussetzung für aussagekräftige Simulationen ist eine sorgfältige Kalibrierung des Modells. Unter Kalibrierung wird hier die Anpassung des Modells an eine reale Anlage auf der Basis von Messungen verstanden. Diese Aufgabe umfaßt im wesentlichen die Erfassung der Zusammensetzung des Abwassers im Anlagenzulauf sowie die Bestimmung der Parameter zur Beschreibung der Prozeßkinetik, was nur zum Teil mit Hilfe von Messungen und gezielten Experimenten erfolgen kann. Modellparameter, die keine direkte Entsprechung in der Realität haben, können nur aufgrund von Simulationen mit dem Modell ermittelt werden.

Die bei der Kalibrierung des Modells des dreistufigen Verfahrens berücksichtigten Meßwerte wurden mit Hilfe einer Pilotanlage ermittelt, die über eine umfangreiche Meßausstattung verfügt [7, 8]. Insbesondere erlauben die verwendeten online-Prozeßanalysatoren eine Erfassung von Meßwerten in regelmäßigen zeitlichen Anständen von wenigen Minuten, so daß die Veränderungen des Prozeßzustands hinreichend genau dokumentiert werden können.

Bild 3 zeigt die Ergebnisse einer zur Kalibrierung des Modells des dreistufigen Verfahrens herangezogenen 25-tägigen Versuchsphase. Auf der linken Seite sind die Meßwerte für den Zulaufvolumenstrom und die Ammoniumstickstoffkonzentration aufgetragen. Hier sind aus Gründen der Übersichtlichkeit nur die Tagesmittelwerte dargestellt, in der Simulation wurden

aber die vollständigen Meßwerte der Pilotanlage verwendet, die im Tagesverlauf gewisse Schwankungen aufweisen.

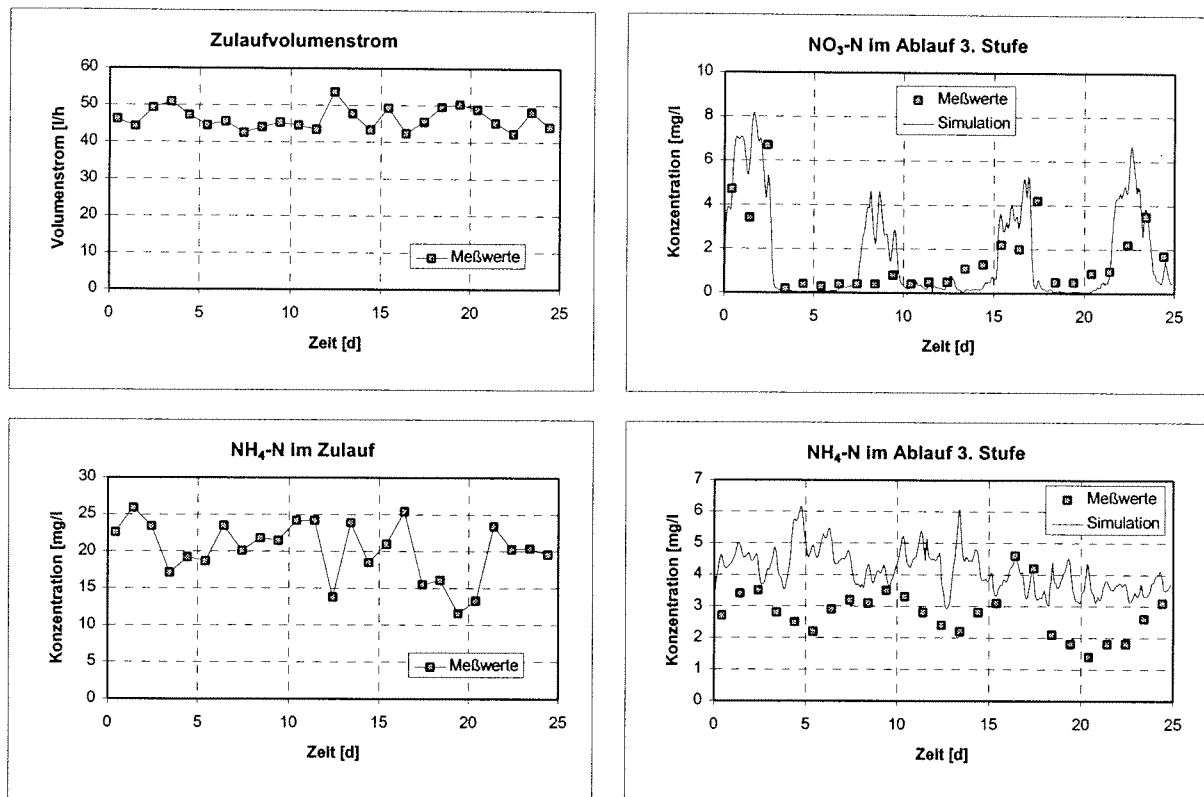


Bild 3: Simulationsergebnisse

Auf der rechten Seite findet sich eine Gegenüberstellung von Meßwerten und Simulationsergebnissen für Ammoniumstickstoff und Nitratstickstoff im Ablauf. Für die Nitratstickstoffkonzentration ist eine weitgehende Übereinstimmung von Messung und Simulation zu beobachten. Insbesondere können die sich im wöchentlichen Rhythmus wiederholenden Anstiege der Konzentration, die aus einer geänderten Abwasserzusammensetzung an Wochenenden resultieren, gut wiedergegeben werden. Die im Vergleich zu den Meßwerten geringfügig größeren Simulationswerte für Ammoniumstickstoff sind auf eine Besonderheit der Modellierung zurückzuführen.

4 Optimierung

Mit Hilfe des kalibrierten Modells besteht nun die Möglichkeit, unterschiedliche Regelungsstrategien zu testen und so eine Optimierung des Abwasserreinigungsverfahrens vorzunehmen. In diesem Zusammenhang ist der Einsatz der Simulation vorteilhaft, da die Optimierung ohne Gefahr für Anlage und Umwelt durchgeführt werden kann. Dies ist insbesondere für Belebtschlammanlagen von Bedeutung, da ein Prozeßfehler zum Absterben von Bakterien führen kann, was eine unter Umständen mehrere Wochen lange Erholungsphase nach sich

zieht, ehe die Mikroorganismen wieder in ausreichender Zahl und Leistungsfähigkeit verfügbar sind.

Ein Beispiel für eine Optimierung wird in Bild 4 dargestellt, wobei die Simulationen mit dem kalibrierten Modell des dreistufigen Verfahrens durchgeführt wurden. Das erste Diagramm zeigt den berücksichtigten Volumenstrom sowie die Ammoniumstickstoffkonzentration im Zulauf für einen Zeitraum von 72 Stunden. Für beide Zulaufgrößen wurden Sinuskurven verwendet, um in etwa den beobachtbaren täglichen Schwankungen zu entsprechen. Der Kurvenverlauf muß daher als idealisiert eingestuft werden, die Minima und Maxima können in bezug auf das betrachtete Abwasser und die eingesetzte Versuchsanlage dagegen als realistisch angesehen werden.

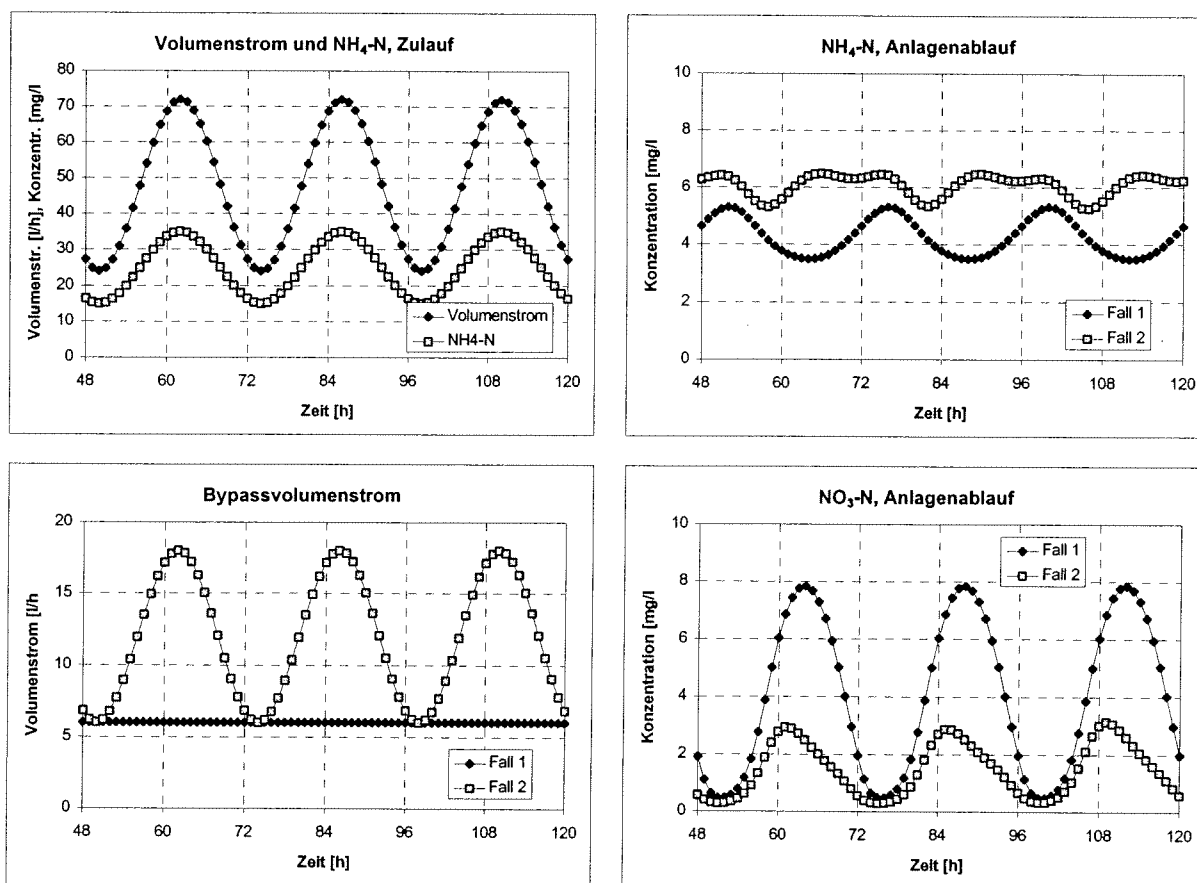


Bild 4: Beispiel einer Optimierung

Im unteren linken Diagramm sind zwei verschiedene Einstellungen des Bypassvolumenstroms abgebildet. Im ersten Fall wurde ein konstanter Volumenstrom verwendet, im zweiten Fall ein Volumenstrom, der dem Verlauf des Zulaufvolumenstroms folgt. Die beiden rechten Diagramme zeigen die sich mit den beiden Bypassvolumenströmen ergebenden Ablaufkonzentrationen an Ammonium- und Nitratstickstoff. Für den Ammoniumstickstoff stellt sich mit dem größeren sinusförmigen Bypassvolumenstrom eine geringfügige Verschlechterung ein,

die in einer etwas höheren Ablaufkonzentration zum Ausdruck kommt. Gleichzeitig sind wesentlich niedrigere und damit bessere Ablaufwerte für Nitratstickstoff erreichbar. Dieses Ergebnis ist im wesentlichen darauf zurückzuführen, daß mit dem größeren Bypassvolumenstrom mehr adsorbierte Kohlenstoffverbindungen in die dritte Stufe gelangen und dort zu einer Verbesserung der Denitrifikation führen.

Das betrachtete Beispiel deutet ansatzweise die Möglichkeiten der Simulation für die Optimierung von Abwasserreinigungsanlagen an. Über die hier untersuchten Stickstoffkonzentrationen hinaus müssen aber normalerweise auch die Ablaufkonzentrationen der organischen Verbindungen berücksichtigt werden. Weitergehende Untersuchungen dürfen zudem nicht nur auf eine Verbesserung der Wasserqualität im Ablauf zielen, sondern müssen auch die für einen optimierten Anlagenbetrieb anfallenden Kosten ins Kalkül ziehen. Liegt ein Modell vor, daß auf der Basis von Meßwerten verschiedenartiger Betriebssituationen kalibriert wurde, so sind durch den Einsatz der Simulation auf schnelle und damit preiswerte Art und Weise Vorhersagen über das Anlagenverhalten möglich.

Die Grenzen der Optimierung biologischer Abwasserreinigungsanlagen mit Hilfe der Simulation sind dann erreicht, wenn das Modell in einem Anwendungsbereich betrieben wird, der über den Bereich der Kalibrierung hinausgeht. Insbesondere in derartigen Fällen sollte sinnvollerweise eine Überprüfung der erzielten Ergebnisse auf einer kleineren Versuchsanlage erfolgen, bevor die Regelungsstrategien auf eine Abwasserreinigungsanlage im großtechnischen Maßstab übertragen werden.

Literatur

- [1] M. Henze, C.P.L. Grady, W. Gujer, G.v.R. Marais and T. Matsuo: *Activated Sludge Model No.1*, IAWPRC Scientific and Technical Reports No.1, IAWPRC, London 1987
- [2] M. Henze, W. Gujer, T. Mino, T. Matsuo, M.C. Wentzel, G.v.R. Marais: *Activated Sludge Model No.2*, IAWQ Scientific and Technical Reports No.3, IAWQ, London 1994
- [3] M. Henze, W. Gujer, T. Mino, T. Matsuo, M.C. Wentzel, G.v.R. Marais, M.C.M. Van Loosdrecht: *Activated Sludge Model No. 2d (Outline)*, *Proceedings of the IAWQ Seminar on Modelling and Microbiology of Activated Sludge Processes*, Copenhagen, Denmark, 16-18 March 1998
- [4] W. Gujer, M. Henze, M. Takashi, M. van Loosdrecht: *Activated Sludge Model No. 3*, *Proceedings of the IAWQ Seminar on Modelling and Microbiology of Activated Sludge Processes*, Copenhagen, Denmark, 16-18 March 1998

- [5] W. Gujer: *Ein dynamisches Modell für die Simulation von komplexen Belebtschlamm-anlagen*, Habilitationsschrift, ETH Zürich, 1985
- [6] J. Jungblut, M. Sievers, A. Vogelpohl, B.R. Bracio, D.P.F. Möller: Simulation eines komplexen biologischen Abwasserreinigungsverfahrens, in: *W. Krug (Hrsg.): Simulationstechnik - 10. Symposium in Dresden*, September 1996, Vieweg Verlag, 1996, S. 393-398
- [7] M. Sievers: *Entwicklung und Inbetriebnahme einer Pilotanlage zur Behandlung komplexer Abwässer*, Cuvillier Verlag, Göttingen, 1993
- [8] M. Sievers, B. Morawe, S.-U. Geißen, A. Vogelpohl, G. Anger, F. Jürschick, M. Högl: Neues Verfahren zur Stickstoffelimination: Ergebnisse eines Pilotversuches auf der KA Landshut, *Korrespondenz Abwasser* 42, 8 (1995)

Simulationsgestützte Prototypentwicklung der Regelung einer Großwindkraftanlage¹

T. Krüger, M. Mevenkamp, J. Petschenka, M. Reichardt

dce@iset.uni-kassel.de

Institut für Solare Energieversorgungstechnik, ISET e.V.

Abteilung Regelungstechnik, Königstor 59, D-34119 Kassel

Zusammenfassung

Im Bereich der Entwicklung von Anlagensteuerungen und -regelungen ist die Simulation in den letzten Jahren zu einem der wichtigsten Entwurfswerkzeuge geworden. In den aufeinander aufbauenden Phasen einer Entwicklung sind dabei verschiedene Arten des Einsatzes der Simulation zu unterscheiden, die sich als *Entwurfssimulation*, *"Software-in-the-Loop"*- und *"Hardware-in-the-Loop"-Simulation* charakterisieren lassen.

Der Beitrag stellt eine Simulationsumgebung auf der Basis einer VME-Bus-Workstation (HP 9000/748i) vor, die alle Phasen des Controllerentwurfs – Modellbildung, Entwicklung von Algorithmen, Implementierung und Labortest von Prototypen – unter weitgehend einheitlicher Hard- und Software unterstützt. Der Einsatz des Systems in den verschiedenen Entwurfsphasen wird am Beispiel der Entwicklung einer Regelung für eine moderne Großwindkraftanlage dargestellt.

1 Unterstützung gerätetechnischer Entwicklungen durch Simulation

In der modernen Geräteentwicklung, speziell im Bereich der Steuergeräte und des Reglerentwurfs, ist die Verkürzung der Entwicklungszeit mehr und mehr zu einem entscheidenden Faktor für die erfolgreiche Markteinführung eines Systems geworden. Dabei besteht das Hauptproblem darin, daß üblicherweise in allen Entwicklungsphasen – Entwurf, Prototypentwicklung, Vorseriengerät – immer wieder neue Design-Ideen und Modifikationen einzuarbeiten und zu bewerten sind. In diesem Prozeß ist die Simulationstechnik heute zu einem der wichtigsten Entwicklungswerkzeuge geworden. Sie bietet vielseitige Entwurfsunterstützung und ermöglicht die Fehlerminimierung und Qualitätssicherung in der Entwicklung durch einfache, reproduzierbare Testroutinen, so daß einsatzfähige Prototypen schneller verfügbar werden.

Simulationsbasierte Entwicklungsumgebungen müssen flexibel, leistungsfähig und einfach handhabbar sein. Dabei ergeben sich in den verschiedenen Phasen der Geräteentwicklung

¹ Die vorgestellten Arbeiten wurden gefördert durch den Bundesminister für Bildung, Wissenschaft, Forschung und Technologie sowie das Hessische Ministerium für Umwelt, Energie, Jugend, Familie und Gesundheit.

jeweils spezifische Anforderungen. Für den theoretischen Entwurf und die Methodenentwicklung ist eine möglichst nahtlose Verbindung von Simulationsumgebung, mathematisch-regelungstechnischen Werkzeugen sowie Optimierungsroutinen wesentlich. Eine enge Verzahnung des Simulationssystems mit einer Software-Entwicklungsumgebung steht bei der Umsetzung des Entwurfs in Hochsprachenprogramme im Vordergrund, um entwickelte Softwaremodule in die Simulationsumgebung einbinden und testen zu können ("Software-in-the-Loop" – Simulation [3]).

Für den Test von Prototypen soll das System darüber hinaus "Hardware-in-the-Loop" (HIL) – Simulationen unterstützen. Außerdem werden in allen Entwicklungsphasen leistungsfähige Analyse- und Visualisierungswerkzeuge benötigt. Nicht selten sieht sich der Entwickler aufgrund dieser unterschiedlichen Anforderungen im Verlauf eines Projektes mit einer Vielzahl von Softwarepaketen und Hardwareplattformen konfrontiert.

Um zeitraubende Einarbeitung und Bedienungsprobleme zu vermeiden, sollte eine Simulationsumgebung aus möglichst einheitlichen Bausteinen bestehen. Dies gilt für die Hardware (möglichst gleiche Plattform für die Modellentwicklung und die eigentliche Echtzeitsimulation) wie für die Software (gleicher Code im simulationsbasierten Test wie auf dem Zielsystem etc.).

Der vorliegende Beitrag beschreibt eine HIL-Simulationsumgebung, die in dieser Hinsicht gegenüber alternativen Systemen Vorteile bietet und insbesondere für die Entwicklung VME-basierter Controller weitgehend einheitliche Hard- und Software aufweist. Sie basiert auf einer VMEbus-Workstation unter UNIX, bei der die I/O-Schnittstellen direkt in das Simulationssystem (Simulink) integriert sind. Die Vorteile werden am Beispiel der Entwicklung einer Windkraftanlagenregelung deutlich.

2 Regelung einer Großwindkraftanlage

Die Windenergienutzung hat in Europa seit Ende der achtziger Jahre eine boomartige Entwicklung genommen. In Deutschland wurden in den letzten 10 Jahren rund 5000 Anlagen mit einer installierten Leistung von inzwischen deutlich über 2000 MW errichtet. Dabei hat sich auch die Konstruktion und die Technik der Anlagen mit einer enormen Dynamik entwickelt. Bild 1 zeigt die Entwicklung der Baugrößen der am Markt verfügbaren Anlagen in den neunziger Jahren. Die heutigen Anlagen der Megawattklasse, wie die im Bild dargestellte TW 1.5 der Tacke Windenergie GmbH, haben bei etwa dreifachem Rotordurchmesser eine um den Faktor zehn höhere Leistung als die vor 10 Jahren verfügbaren. Dabei wurden zum Teil völlig neue Triebstrangkonzeppte entwickelt und am Markt etabliert. Diese äußerst dynamische Entwicklung der Anlagentechnik wurde u. a. durch den breiten Einsatz von Simulationswerkzeugen beim Entwurf von Konstruktion und Steuerung der Anlagen ermöglicht.

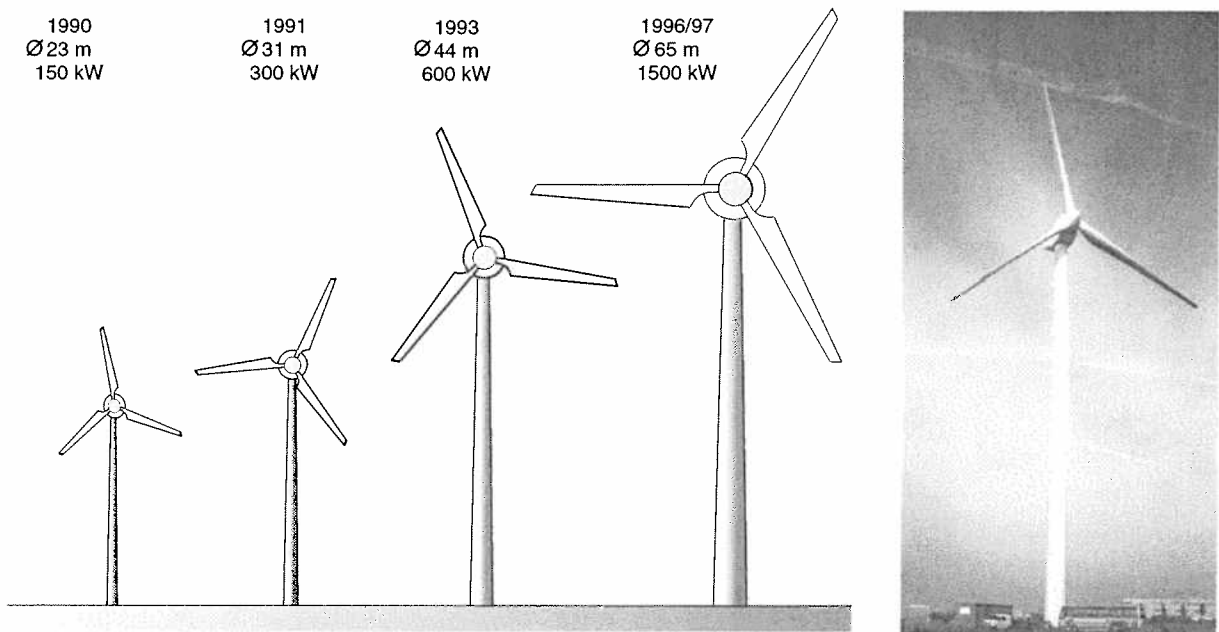


Bild 1: Größenentwicklung bei modernen Windkraftanlagen – Megawattanlage TW 1.5

2.1 Regelungstechnik moderner Windkraftanlagen

Neben der Konstruktionstechnik spielt für die weitere Entwicklung im Windkraftbereich die Regelungstechnik eine große Rolle. Dies gilt vor allem für die Anlagen der Megawattklasse, bei denen man zunehmend von Konzepten der rein aerodynamischen Leistungsbegrenzung (Stall-Effekt) zu einer Anlagentechnik übergeht, die durch aktive Blattverstellung und drehzahlvariable Generatorsysteme sehr weitgehende regelungstechnische Eingriffsmöglichkeiten eröffnet.

Aus der Sicht der Regelungstechnik stellen sich Anlagen dieser Art als nichtlineare Mehrgrößensysteme dar (Bild 2). Der Entwurf hat dabei unter Berücksichtigung des Kurzzeitverhaltens der unterlagerten Regelkreise verschiedenste Optimierungskriterien bezüglich des Betriebsverhaltens der Anlage zu erfüllen, wie etwa

- die Maximierung der Energielieferung,
- die Begrenzung der Leistung im Vollastbereich (Überlastschutz des Generatorsystems),
- die Begrenzung der Drehzahl (Auswirkung auf aerodynamische Kräfte und Geräusch),
- die Reduktion der mechanischen Beanspruchungen.

Aufgabe der Regelung ist die entsprechend dieser Kriterien optimierte Sollwertvorgabe für die Blattwinkelverstellung und das Generatormoment sowie die Überwachung diverser Sicherheitskriterien in den verschiedenen Betriebszuständen. Hinzu kommt die Umsetzung externer Vorgaben, wie Leistungsbegrenzung u. ä. In modernen Windkraftanlagen werden für diese Regelungsaufgaben Mikrocontroller eingesetzt.

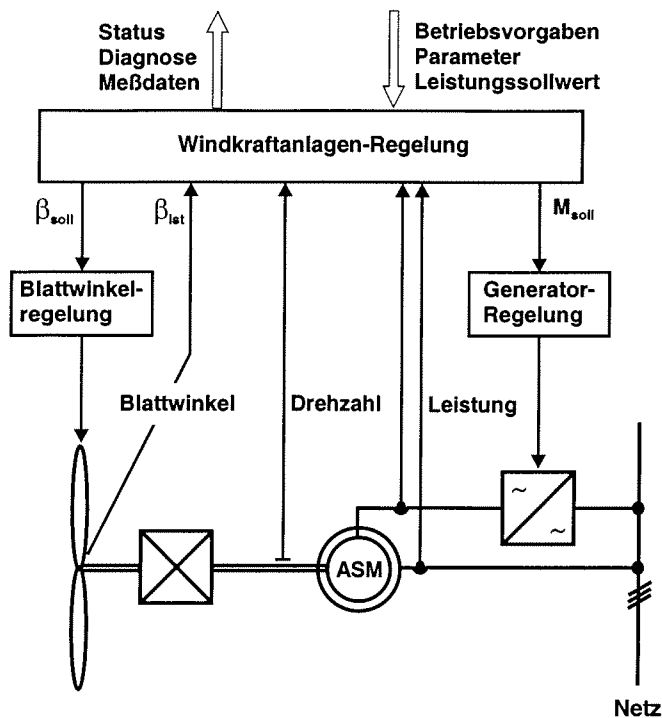


Bild 2: Regelungsstruktur einer Windkraftanlage

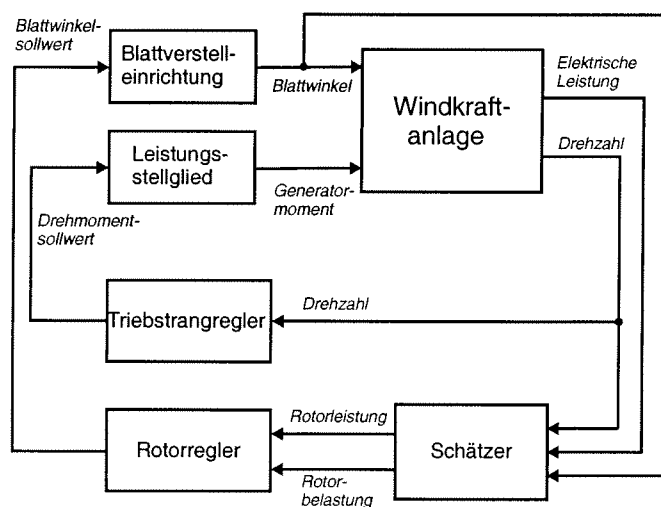


Bild 3: Prinzip einer WKA-Regelung mit Belastungsschätzer

2.2 Regelungskonzept für die TW 1.5

Die Großwindkraftanlage TW 1.5 der Tacke Windenergie GmbH (Bild 1) ist eine moderne Anlage der Megawattklasse mit einer Nabenhöhe von 67 – 80 m, 65 m Rotordurchmesser und einem Generatorsystem mit einer doppeltgespeisten Asynchronmaschine von 1500 kW Nennleistung.

Für diese Anlage wurde eine neuartige Regelung entworfen, die eine gezielte Reduktion der mechanischen Beanspruchungen ermöglicht [1, 2]. Kern des neu entwickelten Regelalgorithmus ist ein Schätzfilter zur Bestimmung der nicht direkt meßbaren Kräfte und Momente am Rotor (Bild 3). Die wesentlichen Belastungsgrößen der Anlage werden dadurch direkt als Regelgrößen verfügbar und können entsprechend den jeweiligen Entwurfsanforderungen beeinflusst und begrenzt werden. Analytische Verfahren sind bei diesem stark nichtlinearen System nur sehr beschränkt einsetzbar. Somit fußt die Entwicklung der Reglerstruktur und die Auslegung der Reglerparameter wesentlich auf einer detaillierten Modellbildung und der Simulation repräsentativer Lastfälle (siehe Abschnitt 4).

Als Hardwareplattform der Regelung dient ein VMEbus-System bestehend aus einem 68k-Mikrorechner, diversen I/O-Modulen und einem VMEbus-IndustriePC für die Meßdatenerfassung und Visualisierung. Das System ist so

konzipiert, daß es für FuE-Zwecke parallel zu einer vorhandenen Anlagensteuerung eingesetzt werden kann und durch Umschaltung zwischen beiden Controllern vergleichende Untersuchungen bezüglich des Erreichens der Regelziele ermöglicht.

3 Simulationsbasierte Entwurfs- und Testumgebung

Die zum Test von VME-Mikrocontrollern aufgebaute Simulations- und Testumgebung basiert auf einem VME-Bus-System des Typs HP 9000/748i. Das Prozessorboard stellt eine Workstation der HP 9000/700-Serie dar, auf der unter HP-UX jede für diese Familie verfügbare Standardsimulationssoftware betrieben werden kann. Ein zweites Board auf dem VME-Bus enthält die I/O-Module für die Ein- und Ausgabe analoger und digitaler Signale.

Als Simulationssystem ist Matlab 5.2/Simulink 2.2 installiert, so daß neben der grafischen Oberfläche für die Modellbildung und Simulation eine breite Palette von mathematisch-regelungstechnischen Funktionen und Optimierungsroutinen zur Verfügung steht. Die Simulationsschnittstellen zu den I/O-Modulen wurden in C geschrieben und in Form einer Blockbibliothek angelegt (Bild 4). Die Signalausgabe erfolgt also durch einfaches Einfügen entsprechender Blöcke in ein Simulink-Modell. Es wird kein zusätzliches echtzeitfähiges Prozessorsystem für die "Hardware-in-the-Loop" – Simulation benötigt.

Das Abarbeiten der Simulation in Echtzeit wird in den Schnittstellenroutinen durch Vergleich der Simulationszeit mit der Systemzeit (Auflösung: 1 μ sec) erzwungen. Einschränkungen ergeben sich allerdings aus den Eigenschaften des UNIX-Betriebssystems, bei dem sich sporadische, kurze Prozeßunterbrechungen nicht vermeiden lassen. Diese bewegen sich im Millisekundenbereich. Dem steht der Vorteil gegenüber, daß der gesamte Entwurf – Modellentwicklung und Validierung, Softwareeinbindung und Hardwaretest – hier unter ein- und derselben Simulations- und Analysesoftware durchgeführt wird.

Trotz der eingeschränkten Echtzeiteigenschaften läßt sich mit dem System ein weiterer Anwendungsbereich abdecken. Beispielsweise lassen sich bei allen extern getakteten Geräten

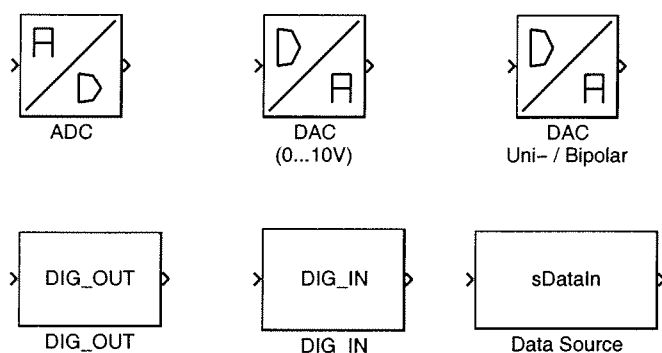


Bild 4: Blockbibliothek der I/O-Schnittstellen

die Softwarefunktionen ohne Probleme testen, da die Abtastfolge durch den Takt vorgegeben wird und keine zeitlich äquidistante Verarbeitung vorausgesetzt wird. Das ist auch bei der oben beschriebenen WKA-Regelung der Fall, die von der vorhandenen Anlagensteuerung getriggert wird. Den Testaufbau für die Prototypentwicklung dieser Regelung zeigt Bild 5.

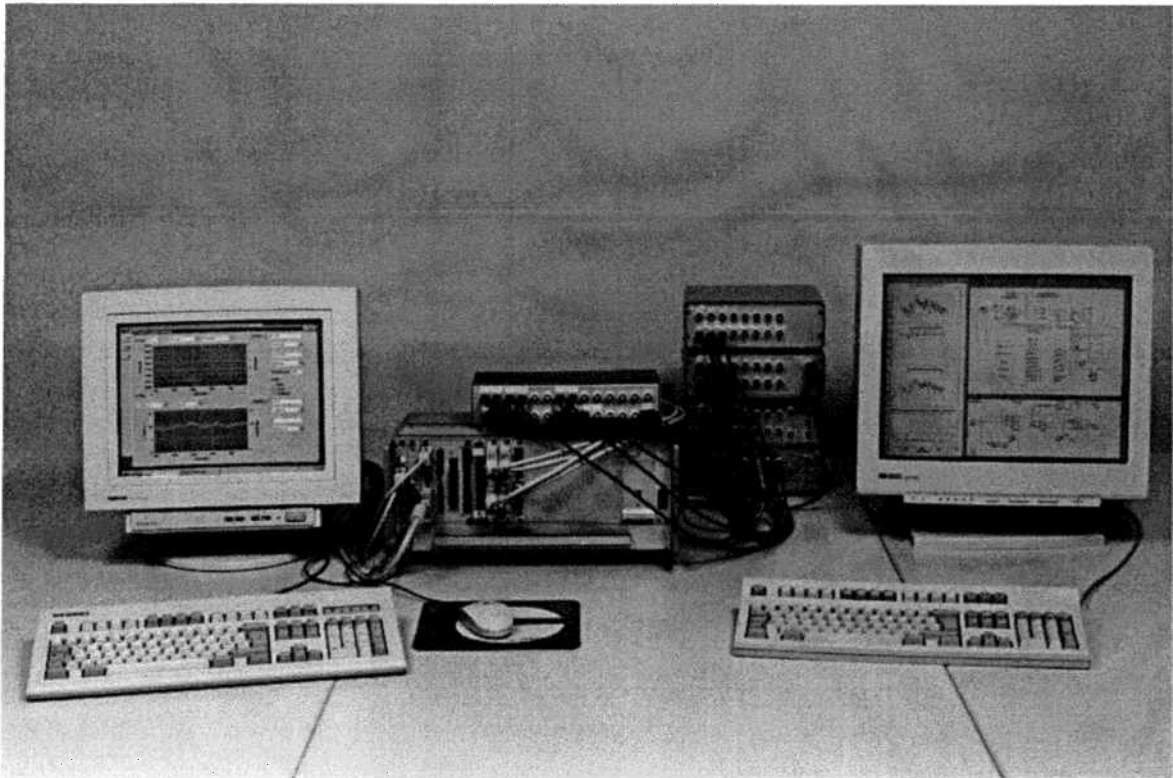


Bild 5: HIL-Simulationsumgebung zum Test der Regelung einer Windkraftanlage

4 Entwicklung und Test der Regelungssoft- und -hardware

Erster Schritt des simulationsbasierten Entwurfs ist die Modellbildung und -verifikation. Die Umgebung stellt zu diesem Zweck alle notwendigen Analysewerkzeuge und Visualisierungsmöglichkeiten zur Verfügung. Für den hier dargestellten Reglerentwurf wurde ein detailliertes aeroelastisches Modell der Windkraftanlage entwickelt, dessen Struktur in Bild 6 zu sehen ist.

Für die Phase des theoretischen Reglerentwurfs (Reglerstruktur, Regelungsverfahren, Auslegung) ist die nahtlose Verbindung von Simulation und regelungstechnischer Analyse

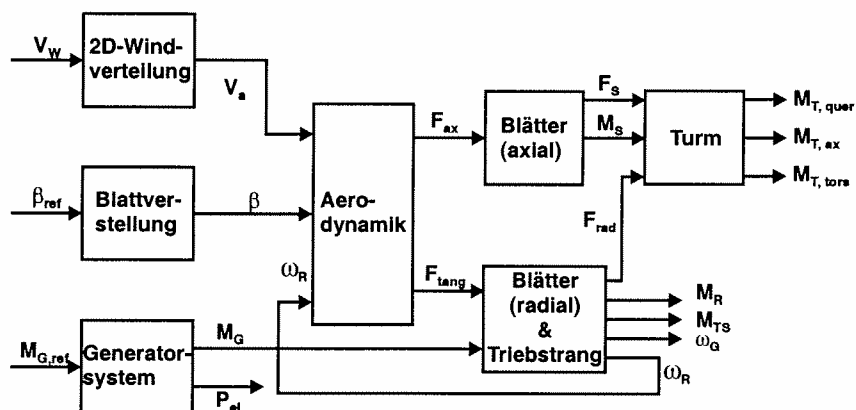


Bild 6: Struktur des Windkraftanlagenmodells für den Reglerentwurf

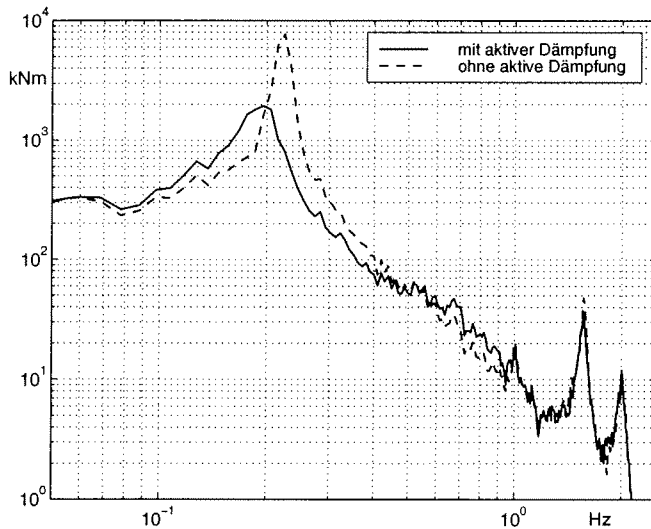


Bild 7: Leistungsdichtespektrum der Turmbeschleunigung, geregelt/ungeregelt

wesentlich. Ein Beispiel ist die Auswertung von Simulationsergebnissen im Frequenzbereich. Sie ermöglicht hier z. B. eine Beurteilung unterschiedlicher Regelungsverfahren zur Dämpfung der Turmschwingungen (Bild 7).

Auch die Umsetzung der entwickelten Regelverfahren in Software für das Zielsystem konnte unter der Simulationsoberfläche getestet und verifiziert werden ("Software-in-the-Loop" – Simulation). Voraussetzung ist dabei eine konsequente Modularisierung, die eine einfache Einbindung in das Simulationssystem ("MEX-Function") einer-

seits und in die Betriebsführungssoftware des Zielsystems andererseits ermöglicht. Unter dem Aspekt der Portierbarkeit oder späterer Modifikationen und Anpassungen ist diese Art der Softwareentwicklung in vielen Fällen einer automatischen Code-Generierung vorzuziehen.

Letzter Schritt der Entwicklung vor dem Einsatz des Controllers in der Anlage ist die Hardware-in-the-Loop – Simulation. Der Übergang von der Software-in-the-Loop – Simulation ist dabei denkbar einfach. Lediglich das Reglermodell ist durch einen Schnittstellenblock mit den in Bild 4 dargestellten I/O-Funktionen zu ersetzen.

Bei dem hier entwickelten Regelsystem wird parallel zum Mikrocontroller ein PC-basiertes Meßsystem eingesetzt, mit dem neben den Meß- und Stellsignalen auch die internen Größen der implementierten Regelung visualisiert werden können. Bild 8 zeigt als Beispiel den Verlauf der wesentlichen Stell- und Regelgrößen eines simulierten Lastfalles, wobei die Werte für das Biegemoment an den Rotorblättern aus dem implementierten Schätzfilter ausgelesen werden. Die durch die Regelung erreichte Reduktion der Spitzenbelastungen an den Rotorblättern läßt sich anhand solcher Testläufe aussagekräftig darstellen.

Wesentlicher Vorteil dieser Labor-Testumgebung ist die Möglichkeit, die Funktion des Controllers für alle relevanten Betriebssituationen in kurzer Zeit testen zu können. Dazu gehört auch das Verhalten bei Störungen der Anlage. Für bestimmte Teile der Betriebsführung ist dies sogar die einzige Testmöglichkeit, da einige Betriebszustände einer Windkraftanlage, wie etwa die Einstellung auf extreme Windgeschwindigkeiten, während der Inbetriebnahme normalerweise nicht auftreten.

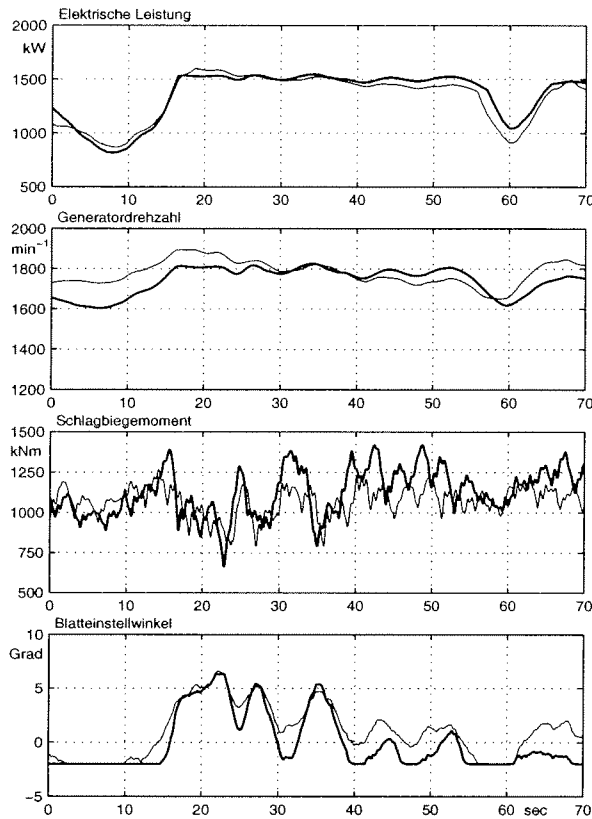


Bild 8: Ergebnis eines HIL-Tests des Prototypen der Windkraftanlagenregelung

5 Ausblick

Die verringerte Komplexität der dargestellten Entwicklungsumgebung wird in erster Linie dadurch erreicht, daß der Zugriff auf die I/O-Hardware aus demselben Betriebssystem heraus erfolgt, unter dem auch die Simulationsoberfläche läuft. Gegenüber bekannten, PC-basierten Realisierungen eines solchen Konzepts hat das System den Vorteil, daß die Entwicklung von vornherein auf industrietauglicher Hardware stattfindet. Sinnvoll erscheint allerdings auch eine Umsetzung auf PC – Plattformen für Industrie-Bussysteme (z. B. Compact-PCI) unter gängigen Windows-Betriebssystemen mit Echtzeiterweiterungen. Die hier entwickelten Schnittstellenroutinen (C MEX S-Functions) sind soweit portabel gestaltet, daß eine solche Umsetzung problemlos möglich ist.

Die Vorteile derart kompakter HIL-Simulationsumgebungen bestehen in der einfacheren Handhabung, in kurzen Einarbeitungszeiten und in der Minimierung von Fehlerquellen beim Übergang von der Offline-Simulation zum Hardwaretest. Dabei ist die Anwendung nicht auf die Controllerentwicklung beschränkt. Die beschriebene Umgebung wird derzeit z. B. auch zur Generierung von Testsignalen für die Auslegung und Konfiguration eines Fehlerfrüherkennungssystems für Windkraftanlagen eingesetzt.

Literatur

- [1] P. Caselitz, J. Giebhardt, T. Krüger, M. Mevenkamp, J. Petschenka, M. Reichardt, "Neue Verfahren zur Regelung von Windkraftanlagen", "Forschungsverbund Sonnenenergie – Themen 96/97: Energie aus Sonne und Wind, Köln, 1997
- [2] T. Krüger, *Regelungsverfahren für Windkraftanlagen zur Reduktion der mechanischen Belastung*, Dissertation, Universität Gh Kassel, 1998
- [3] A. Wohnhaas, "Einsatz rechnergestützter Methoden bei der Entwicklung neuer Funktionen im Kraftfahrzeug", ASIM-Fachgruppe Simulation Technischer Systeme, Heidelberg, 1998

Simulation menschlicher Handlungen in dynamischen Mensch-Maschine-Systemen

Thomas Jürgensohn

juergensohn@zmms.tu-berlin.de, 030/31472996

ISS-Fahrzeugtechnik, Technische Universität Berlin, Gustav-Meyer-Allee 25, 13355 Berlin

Zusammenfassung

Die Modellierung und Simulation menschlichen Verhaltens in dynamischen Umgebungen erfordert Herangehensweisen, die sich in vielen Punkten wesentlich von denen in technischen Systemen unterscheiden. Hauptschwierigkeiten sind die große Flexibilität im menschlichen Verhalten und das Fehlen der physikalischen Meßbarkeit von mentalen Prozessen. Eine Folge davon ist, daß Theorien über menschliche mentale Tätigkeiten nur sehr vage, oft nur sprachlich formuliert werden können. Eine Übersetzung dieser Theorien in simulierbare Modelle kann deshalb nicht nach einem festen Kalkül erfolgen, sondern erfordert eine hohes Maß an Heuristik und den Einsatz sehr unterschiedlicher Modellierungswerkzeuge. Die Simulation komplexer Handlungsgeflechte vereinigt diese Werkzeuge zu sogenannten Hybriden Modellen.

1. Einleitung

Der Mensch ist das wohl am schwersten zu modellierende System. Menschliche Handlungen sind dadurch gekennzeichnet, daß sie hochgradig nichtlinear, nichtstationär, adaptiv und zeitvariabel sind. Dies ist die Hauptursache dafür, daß trotz 50jähriger Tradition die Simulation des menschlichen dynamischen Verhaltens bei weitem noch nicht den Stand erreicht hat wie die Simulation technischer Systeme. Es gibt bislang kaum Modelle, die den Einfluß des Menschen bei komplexen dynamischen Aufgaben ausreichend genau präzisieren können. Insbesondere für Notfallsituationen ist deshalb bisher eine Vorhersage des Systemverhaltens nicht möglich. Ebenso kann der Einfluß künftiger Automatisierung beispielsweise durch halbautomatische Lenksysteme, rechnergestützte Konfliktbewältigung in der Flugüberwachung oder im Wartebereich nicht wirklich präzisiert werden.

Dabei ist es gerade heute – da dank des technischen Fortschritts es immer leichter wird, menschliche Tätigkeiten von Maschinen erledigen zu lassen – notwendig, die technische Entwicklung „menschgerecht“ und „menschoptimiert“ voranzutreiben [12]. Dies erfordert neben der Unterstützung von seiten der Ergonomie, Psychologie und Arbeitswissenschaft

insbesondere in Situationen mit dynamischen Wechselwirkungen zwischen Mensch und technischem System die Entwicklung mathematischer oder anderer formaler Modelle des menschlichen Bedieners.

Die Ursache für das heutige Fehlen geeigneter Bedienermodelle ist hauptsächlich darin zu suchen, daß in der Vergangenheit auf Entwicklerseite fast ausschließlich die technische Optimierung und Automation im Vordergrund standen. Eine Modellierung des Bedienerverhaltens erschien nicht notwendig, da der Mensch in der Regel normgerecht „funktionierte“ und Sondersituationen nicht in die Entwicklungskalküle einbezogen wurden. Die wenigen, hauptsächlich von Ingenieuren entwickelten Ansätze reichten als Instrument für eine Systementwicklung nicht aus.

Eine der wesentlichen Forderungen an eine „menschadäquate“ Menschmodellierung ist die Einbeziehung von psychologischen Theorien und physiologischen Erkenntnissen. Allerdings müssen die Theorien oft erst mathematisch quantisiert oder in eine formale Sprache übersetzt werden. Dabei zeigt sich, daß sehr ausgearbeiteten Theorienkomplexe oft nur einen sehr kleinen quantisierbaren Kern haben. Ein Beispiel sind die Versuche, psychologische Konstrukte wie ‘Motivation’ [19,20,8] oder ‘mentales Modell’ [10,29] in formale Modelle zu überführen.

Mehr noch als bei komplexen nichtlinearen Prozessen in der Technik ist eine Simulation menschlichen Verhaltens nur mit einer breiten Palette unterschiedlicher Methoden und Werkzeuge zu leisten. Dabei werden die klassischen Modelle, die menschliches Verhalten in Form von Differentialgleichungen beschreiben, mehr und mehr abgelöst von flexibleren Modellierungsmethoden, die dem adaptiv-flexiblen Charakter menschlichen Verhaltens besser gerecht werden.

2. Modellierungswerkzeuge zur Verhaltensmodellierung

Trotz einer durch die modernen Simulationstechniken eröffneten Methodenvielfalt dominieren zur Beschreibung menschlichen Verhaltens in dynamischen Umgebungen auch heute noch die seit den 40er Jahren [35] bekannten regelungstechnischen Modelle [27]. Diese „Tracking“-Modelle sind mathematische Beschreibungen des dynamischen Verhaltens des Menschen mit dem aus der Technik bekannten systemtheoretischen Instrumentarium [14]. Die meisten Bedienermodelle bilden den Menschen in seiner Eigenschaft als Regler nach, der eine Maschine nach Maßgabe einer von außen vorgegebenen dynamischen Sollgröße manipuliert. Die wichtigsten Modelle sind das Quasilineare Modell [25], das Crossover-Modell [24] und das optimaltheoretische BBN-Modell [3]. Erst in den letzten Jahren werden zunehmend Modelle entwickelt, die zwar weiterhin das Regelverhalten des Bedieners mit linearen Modellen beschreiben, darüberhinaus aber die kognitiven Eigenschaften des Bedieners in den

Vordergrund stellen [6,4,11] und die Bedeutung einer Situationsmodellierung und einer Abbildung des Situationsbewußtseins (Situation Awareness) betonen [13,33].

Das Methodeninventar, das zur Modellierung menschlichen Verhaltens in dynamischen Umgebungen eingesetzt werden kann, ist breit gefächert. Tatsächlich findet man in veröffentlichten Modellen einen Großteil der aus dem Ingenieur- oder Informatikbereich bekannten Methoden. Angefangen über Beschreibungen mit Hilfe von Differential- oder Differenzengleichungen über spezielle Methoden der Regelungs- und Systemtheorie bis zu den „modernen“ Methoden wie Intervallnetze [22], Künstliche Neuronale Netze [21,23], Petri-Netze [32], Netzwerkmodelle [7], Fuzzy und Fuzzy-Control [38] oder Mehrkörpersysteme [15] (Bild 1).

Methoden in der Menschmodellierung	
Ingenieurmethoden	Methoden der KI
• lineare Differentialgl. • nichtlineare Differentialgl. • Differenzengleichungen • Zustandsraummodelle • stochastische Modelle • Petri-Netze • Intervallnetze • Fuzzy-Entscheidungsfindung • Fuzzy-Control • Mehrkörpersysteme • Qualitative Modellierung	• semantische Netze • Produktionensysteme • Expertensysteme • Begriffsgraphen • Handlungsgraphen • Maschinelle Lernverfahren • behavioral cloning • Künstliche Neuronale Netze

Bild 1: Modellierungsmethoden für die Verhaltenssimulation (modifiziert nach [18]).

Hinzu kommen Methoden, die sich in der psychologischen Forschung und KI entwickelt haben, wie semantische Netze [34], Produktionensysteme [1], Begriffs- und Handlungsgraphen [30] oder Maschinelle Lernverfahren [39,17]. Die Menschmodelle der Künstlichen Intelligenz und Kognitiven Modellierung bilden in der Regel statische Problemlöseaufgaben und Zielsetzungen ab [31]. In den beiden letzten Jahrzehnten wurden eigene Programmiersprachen und Implementierungswerkzeuge für diesen Zweck geschaffen, unter anderem die Systeme ACT-R [2], SOAR [28], und EPIC [26]. Erst in jüngster Zeit kommen relativ dynamische Aufgaben hinzu, beispielsweise die Optimierung von Systemeingriffen in der Überwachung eines simulierten Kohlekraftwerks [36] oder die Flugüberwachung durch Fluglotsen [10,29].

3. Die Problematik der Validierung

Quantitative Verhaltensmodellierung in dynamischen Mensch-Maschine-Systemen mit dem

Ziel der Simulation bedarf einer Methodik, die sowohl Elemente der rein technischen Modellbildung als auch der allgemeinen psychologischen Theorienentwicklung und der Modellbildung in der Kognitionswissenschaft enthält. Aus dieser Synthese ergeben sich Implikationen, die außerhalb der Paradigmen der Einzelwissenschaften liegen. Dies betrifft sowohl den Modellbildungsprozeß selber als auch die dafür eingesetzten Werkzeuge und schließlich die Beurteilung und Überprüfung der Simulationsergebnisse. Der dominierende Unterschied zu den Modellen der Psychologie und KI liegt in der Beschäftigung mit kontinuierlich ablaufenden Vorgängen in der Zeit. Unterschiede zu den Modellen der Technik ergeben sich vor allem daraus, daß wegen der Zielgerichtetheit menschlicher Handlungen nicht die explizite Nachbildung der Dynamik, sondern die Modellierung übergeordneter Strategien im Vordergrund steht.

Daß die quantitativen dynamischen Benutzermodelle Methoden benötigen, die weder in der technischen noch in der psychologischen Modellbildung notwendig sind, läßt sich besonders deutlich an Hand der Validitätsproblematik zeigen. Wegen des Anspruchs, die Wirklichkeit möglichst gut abzubilden, ist in psychologischen Modellen des Menschen der Vergleich von Modellverhalten und experimentellen Daten, die Validierung, ein wesentlicher Teil der Modellbildungsprozesses. Es gibt deshalb in der Psychologie ein ausgereiftes Instrumentarium zur Validitätsmessung [9]. Diese sind aber bei der Verhaltensmodellierung in der Regel nicht anwendbar. Ein Beispiel soll dies demonstrieren.

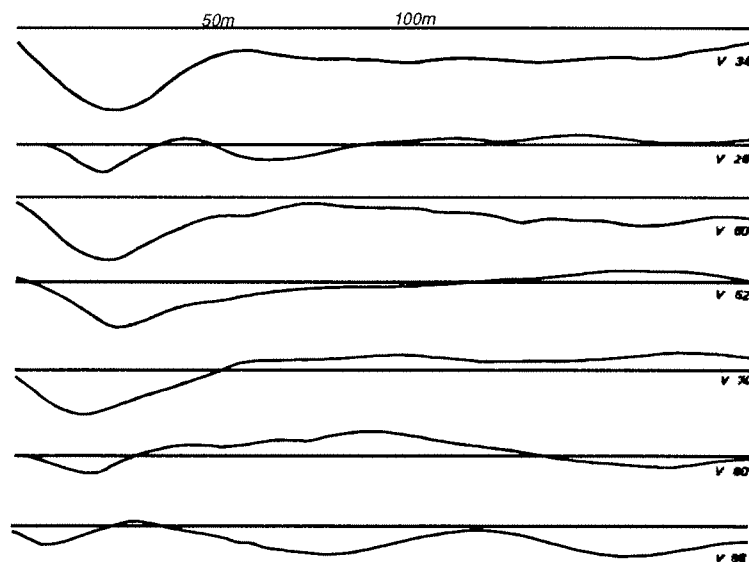


Bild 2: Fahrspur von Versuchsfahrern nach Seitenwindstoß (aus [16]).

Bild 2 zeigt das Verhalten von Autofahrern beim Befahren von Teststrecken mit einem künstlich erzeugten Seitenwindstoß. Dargestellt ist die Abweichung des Kurses hervorgerufen durch die Störung. Man sieht, daß ein Modell eines „mittleren“ Fahrers keines der Fahrer-

halten exakt nachbilden kann. Eine Überprüfung des Modells auf Validität müßte aus dem Vergleich von gemessenen Verläufen und Modellverhalten ein Maß ableiten, das eventuell mit dem Maß für andere Modelle verglichen werden könnte. Da es wegen der unvermeidlichen großen Unterschiede von Modelldaten und Realität kein „natürliches“ Validierungsmaß gibt, können unterschiedliche Modelle auch nicht bezüglich einer *absoluten* Validität, sondern allenfalls bezüglich eines bestimmten Validitätsmaßes verglichen werden. Da dieses wiederum willkürlich ist, macht es keinen Sinn, eine absolute Richtigkeit des Modells zu fordern. Da sich für jedes Modell immer ein Validitätsmaß finden läßt, das für dieses Modell maximal wird, könnte man folgern, daß alle Modelle gleich gut sind. In der Tat ist dies der Fall, wenn man *Güte* mit Abbildungsgüte gleich setzt. Anders ist es, wenn 'Güte' mit den Zielen der Modellbildung in Verbindung gebracht werden. *Gut* ist dann ein Modell, wenn es besonders zweckmäßig ist. Nur bezüglich des Zielsystems der Modellierungsproblematik ist dann ein Vergleich zwischen unterschiedlichen Modellen und im nachhinein bezüglich des erreichten Nutzens möglich.

4. Hybride Modellierung

Das Beispiel im letzten Abschnitt hat gezeigt, daß die quantitative Verhaltenssimulation stark von den Maximen des Nutzens bestimmt sein muß. Es ist sinnvoll, für die Modellierung nicht die Nachahmung der „Wirklichkeit“ zu fordern, sondern immer nur bestimmte Aspekte eines hochkomplexen Systems abzubilden. Diese Aspekte sind in der Regel so prinzipiell unterschiedlich, daß sie mit einem einzigen Modellierungswerkzeug, wie z.B. Differentialgleichungen oder Produktionensystemen nur unzureichend abgebildet werden. Neuere Ansätze setzen deshalb mehr und mehr das Mittel der Hybriden Modellierung [16,37] ein. Hier werden unterschiedliche Modellierungswerkzeuge unter dem Dach der Simulation vereinigt. Problematisch an dem Vorgehen ist, daß es zwar sehr einfach ist, ein Hybrides Modell praktisch zu implementieren, es aber nur wenige Richtlinien gibt, dies auch systematisch zu tun. Angesprochen wurde schon die Validierungsproblematik als eines der wichtigsten ungelösten Probleme.

Wiechert [37] schlägt für die ebenfalls stark heterogenen Systeme Ökonomie, Ökologie, Verfahrenstechnik oder biologische Prozesse mehrere Vorgehensweisen vor, wie man das Problem der „Methodenanarchie“ in den Griff kriegen könnte. Er fordert u.a. möglichst kleine Überschneidungen der durch räumliche und zeitliche Parameter unterscheidbaren Gültigkeitsbereiche von Teilmodellen, um Widersprüchlichkeiten möglichst zu vermeiden. Übertragen auf die Menschmodellierung kann das z.B. eine Trennung von biomechanisch modellierten Eigenschaften körperlicher Bewegungen auf der einen Seite und von Wahrnehmungsvorgängen und kognitiver Wissensverarbeitung auf der anderen.

Ein eng mit der Validierung verknüpft Problem ist die Festlegung freier Parameter des Systems. Da mentale Vorgänge nicht physikalisch meßbar sind, können Modelle dieser Vorgänge immer nur aus Beobachtung (auch Introspektion) von Eingangs-Ausgangs-Relationen entwickelt werden. Diese „eigengesetzlichen“ Modelle unterscheiden sich von „physikalischen“ Modellen dadurch, daß sie nicht aus anerkanntem Wissen deduktiv abgeleitet, sondern an Hand standardisierter Messungen induktiv mit Hilfe semantikarmer mathematischer Methoden konstruiert werden. Die freien Parameter dieser Modelle können außerhalb des Modellverbundes mit keiner Bedeutung belegt werden, wobei mit Parameter nicht nur mit Zahlen assoziierbare Variable, sondern auch Strukturen gemeint sind.

Dieses Problem der semantikarmen Parameter menschlicher Handlungsmodelle ist eines der Gründe, weshalb die regelungstechnischen Modelle des Menschen auf die industrielle Praxis keinen nennenswerten Einfluß hatten. Wegen ihrer stark mit dem Experiment verkoppelten Bedeutung sind die Parameter der eigengesetzlichen Modelle des Menschen nur schwer auf unbekannte Situationen übertragbar – die Modelle haben nur eine kleine Prädiktionskraft. In [16] wird als Lösung des Problems eine möglichst große Fremdstrukturierung gefordert. Dadurch werden die erfahrbaren Daten über ein System in einen strukturellen Zusammenhang gestellt, dessen Begründungen außerhalb des modellierten Systems selber liegen. Mögliche Fremdstrukturierung kann z.B. über physiologisches oder psychologisches Wissen erfolgen. Je weiter dieses Wissen von den jeweils modellierten Eigenschaften entfernt ist, desto höher wird die Wahrscheinlichkeit, daß diese aufgabeninvariant sind und damit ein bestimmtes Maß an Allgemeingültigkeit haben. Das Mittel der Fremdstrukturierung ersetzt also bei der Verhaltensmodellierung das in der Technik angewandte Mittel der Physikalischen Modellbildung.

5. Zusammenfassung

Die quantitative Verhaltensmodellierung steht zwischen der exakten Modellierung der Technik und der bewußt ungenauen der Psychologie und muß Exaktheit und Mehrdeutigkeit in sich vereinigen. Es wurden in dem Artikel einige Aspekte der Modellierung menschlichen Verhaltens in dynamischen Umgebungen dargestellt. Die Simulation des Menschen steht trotz einer langen Tradition noch immer relativ am Anfang. Es bleibt zu hoffen, daß die steten Fortschritte in der Simulationstechnik uns der Lösung dieser spannenden Aufgabe näher bringen.

6. Literatur

- [1] Aasman, J.. Modelling Driver Behaviour in Soar. *Proefschrift, Rijksuniversiteit Groningen*. Leidschendam: KPN Research, 1995.
- [2] Anderson, J.R.. *The architecture of cognition*. Cambridge, MA: Harvard University Press, 1983.

- [3] Baron, S.; Kleinman, D.L.. The Human as an Optimal Controller and Information Processor. *IEEE Transactions on Man-Machine Systems*, Vol. MMS-10, No. 1, 1969, S. 9-17.
- [4] Cacciabue, P.C.. *Modelling and Simulation of Human Behaviour in System Control* (Advances in Industrial Control). London: Springer, 1998.
- [5] Cacciabue, P.C.; Cojazzi, G.. A Human Factors Methodology for Safety Assessment Based on the DYLAM Approach. *Reliability Engineering and System Safety*, 45, 1994, S. 127-138.
- [6] Corker, K.; Pisanich, G.; Bunzo, M.. A Cognitive System Model for Human/Automation Dynamics in Airspace Management. *1st USA/Europe Air Traffic Management R&D Seminar, Sacly, France, June 17-20, 1997*, 1997.
- [7] Döring, B.. Application of SLAM for modeling and analysing pilot's rule-based behaviour on an automated landing approach. In McMillan, G.R. et al. (Eds.), *Applications of Human Performance Models to System Design*, 1989, S. 243-258. Plenum Press.
- [8] Dörner, D.; Hamm, A.; Hille, K.. EmoRegul, Beschreibung eines Programmes zur Simulation der Interaktion von Motivation, Emotion und Kognition bei der Handlungsregulation. *Otto-Friedrich-Universität, Bamberg, Lehrstuhl Psychologie II, Memorandum 2*, 1996.
- [9] Erdfelder, E.; Bredenkamp, J.. Hypothesenprüfung. In Herrmann, T.; Tack, W.H. (Hrsg.), *Methodologische Grundlagen der Psychologie* (Enzyklopädie der Psychologie, Bereich B, Serie I, Band 1). Göttingen: Hogrefe, 1994, S. 604-648.
- [10] Eyferth, K.; Bierwagen, T.; Niessen, C.. Ein implementiertes Modell mentaler Leistungen in der Streckenflugkontrolle. *Proceedings der 3. Fachtagung der Gesellschaft für Kognitionswissenschaft*. Jena, Friedrich Schiller-Universität, 1997.
- [11] Feraric, J.P.; Onken, R.. DAISY - a driver assisting system which adapts to the driver. *Proceedings of the Man-Machine Systems (MMS'95), Cambridge, Mass, 27-29 June*, 1995.
- [12] Gill, K.S. (Ed.). *Human Machine Symbiosis*. Berlin: Springer, 1996.
- [13] Grashey, S.; Onken, R.. Adaptive Verhaltensmodellierung für den Autofahrer auf der Basis von statistischen Klassifikatoren. In Willumeit, H.-P.; Kolrep, H. (Hrsg.), *Wohin führen Unterstützungssysteme? - Entscheidungshilfe und Assistenz in Mensch-Maschine-Systemen*, 2. *Berliner Werkstatt Mensch-Maschine-Systeme*, 7.-9. Okt. 1997, Berlin (ZMMS-Spektrum, Band 5), 1998, S. 56-69. Sinzheim: Pro Universitate Verlag.
- [14] Johannsen, G.; Boller, H.E.; Donges, E.; Stein, W.H. (Hrsg.). *Der Mensch im Regelkreis, Lineare Modelle*. München, Wien: Oldenbourg Verlag, 1977.
- [15] Jürgensohn, T. et al.. Rückwirkungen des Fahrerkörpers auf die Stabilität des Systems Fahrer-Fahrzeug. *9. internationaler Kongreß, Berechnung und Simulation im Fahrzeugbau*, 24.-25. Sep., Würzburg (im Druck). Düsseldorf: VDI-Verlag, 1998.
- [16] Jürgensohn, T.. *Hybride Fahrermodelle* (ZMMS-Spektrum, Band 4). Sinzheim: Pro Universitate Verlag, 1997.
- [17] Jürgensohn, T.. Maschinelle Lernverfahren zur Nachbildung vom Verhalten und von Urteilen von Autofahrern. *Tagungsband der FGML-98*, 17.-19. August, Berlin, 1998.
- [18] Jürgensohn, T.. Modellierung und Simulation. In Timpe, K.-P.; Kolrep, H. (Hrsg.), *Mensch-Maschine-Systemtechnik* (im Druck). Berlin: Springer-Verlag, 1998.
- [19] Jürgensohn, T.; Irmscher, M.; Willumeit, H.-P.. A Model of Individually Acting Drivers Influenced by Motivational Factors. *Proceedings of 16th European Annual Conference on Human Decision Making and Manual Control* Dec. 9.-11. 1997, Kassel, Germany, 1997.
- [20] Jürgensohn, T.; Irmscher, M.; Willumeit, H.-P.. The Motivation Concept for Driver Models in Driver Assistant Systems. *Proceedings of the AVEC 98*, Sep. 14.-18., Nagoya, Japan (in press), 1998.
- [21] Jürgensohn, T.; Willumeit, H.-P.. Modellierung von Mikro-Steuerstrategien des Fahrers mit künstlichen neuronalen Netzen. *Automobiltechnische Zeitschrift*, Heft 6, 1997, S. 348-354.
- [22] Kraiss, K.-F.. Modellierung von Mensch-Maschine-Systemen. In Willumeit, H.-P.; Kolrep, H. (Hrsg.), *Verlässlichkeit von Mensch-Maschine-Systemen*, 1. *Berliner Werkstatt Mensch-Maschine-Systeme*, 11.-13. Okt. 1995, Berlin (ZMMS-Spektrum, Band 1). Berlin: Technische Universität Berlin, 1995, S. 15-35.

- [23] Kraiss, K.F.; Küttelwesch, H.. Identification and Application of Neural Operator Models in a Car Driving Situation. *5th IFAC Symposium on Analysis, Design and Evaluation of Man Machine Systems, The Hague, The Netherlands, June 6-8*. The Hague, The Netherlands, 1992, S. 121-126.
- [24] McRuer, D.T.; Graham, D.; Krendel, E.; Reisener, W.. Human Pilot Dynamics in Compensatory Systems. *Wright-Patterson Air Force Base, Ohio, AFFDL-TR-65-15*, 1965.
- [25] McRuer, D.T.; Krendel, E.S.. The Human Operator as a Servo System Element Part I+II. *Journal of The Franklin Institute*, Vol. 267, No. 5+6, 1959, S. 381-403, 511-536.
- [26] Meyer, D.E.; Kieras, D.E.. A Computational Theory of Executive Cognitive Processes and Multi-Task Performance: Part 1+2. *Psychological Review*, Vol. 104, No. 1+4, 1997, S. 3-65+749-791.
- [27] Mitschke, M.; Ahring, E.. Control Loop for Driver-Vehicle with Four Wheel Steering. *Proceedings of the International Symposium on Advanced Vehicle Control 1994, SAE - Paper, No. 9437980*, 1994, S. 28-35.
- [28] Newell, A.. *Unified Theories of Cognition*. Cambridge: Harvard University Press, 1990.
- [29] Niessen, C.; Leuchter, S.; Eyferth, K.. A Psychological Model of Air Traffic Control and Its Implementation. *Proceedings of European Conference on Cognitive Modelling '98*, 1998.
- [30] Nirschl, G.. *Verfahren zur integrierten Gestaltung und Bewertung von Mensch-Maschine-Dialogen im Kraftfahrzeug, basierend auf einem Entwicklermodell des Fahrerwissens* (VDI-Fortschrittberichte, Reihe 10, Nr. 142). Düsseldorf: VDI-Verlag, 1990.
- [31] Opwis, K.. *Kognitive Modellierung*. Bern: Hans Huber, 1992.
- [32] Ruckdeschel, W.. *Modellierung regelbasierten Pilotenverhaltens mit Petrinetzen* (Dissertation, UniBw-München, LRT), 1997.
- [33] Sarter, N.B.; Woods, D.D.. "How in the world did we ever get into that mode": Mode error and awareness in supervisory control. *Human Factors*, 37(1), 1995.
- [34] Schmid, U.; Kindsmüller, M.C.. *Kognitive Modellierung: Eine Einführung in die logischen und algorithmischen Grundlagen*. Spektrum Akademischer Verlag, 1996.
- [35] Tustin, A.. The Nature of the Operator's Response in Manual Control and Its Implications for Controller Design. *Journal, IEE, Vol. 94, Part II A2*, 1947, S. 190-202.
- [36] Wallach, D.. Learning to control a coal-fired power plant: Empirical results and a model. In Harris, D. (Ed.), *Engineering Psychology and Cognitive Ergonomics*, 1997. Hampshire, Mass: Ashgate Publishers.
- [37] Wiechert, W.. Modellkonsistenz und Konfliktlösung bei Hybriden Modellansätzen. In Kuhn, A.; Wenzel, S. (Hrsg.), *II. Symposium Simulationstechnik, Dortmund* (Fortschritte in der Simulationstechnik, Band 11), 1997, S. 643-648. Braunschweig: Vieweg.
- [38] Wolter, T.-M.; Jürgensohn, T.; Willumeit, H.-P.. Ein auf Fuzzy-Methoden basierendes Situations-Handlungs Modell des Fahrerverhaltens. *Automobiltechnische Zeitschrift, Heft 3*, 1997, S. 142-147.
- [39] Wysotzki, F.. Maschinelles Lernen. *Automatisierungstechnik (at)*, Nov., 1997, S. 526-536.

Geschäftseinheiten planen und simulieren mit dem BB-Tool®

Dr. U. La Roche, W. Heinzmann, Dr. M. Simon

Heilighüsli 18, CH-8053 Zürich

ularoch@ibm.net

Zusammenfassung

Das BB-Tool® (Basic Business Tool) ist ein für Geschäftsführer, Manager und Prozesseigner entwickeltes, einfach zu bedienendes Stand-alone-Fließmodell mit integrierter Kostenrechnung. Ein Laptop-/Desktop-PC genügt, um für bestehende Prozesse die dynamische Führung des Geschäftsverlaufs mit einer Zeitreihe des Bestellungseingangs als Vorgabe durchzuplanen, der Kennzahlenverlauf wird durch Simulation mit einer Genauigkeit besser als ± 5 Prozent erfasst. Dazu werden mit Hilfe einer überschaubaren Anzahl Parameter die bestehenden Geschäftsprozesse auf vorbereitete Strukturen im BB-Tool® abgebildet. Die Teilkostenrechnung mit geringem Datenimport- und Datenexportbedarf ermöglicht den direkten Zugriff auf die drei Kostenelemente Volumen-, Zeit- und Bestandeskosten und baut daraus die Erfolgsrechnung auf. In Planungssitzungen können interaktiv am Bildschirm die Kostenfolgen von Geschäftspolitik, Marktdynamik und Eingriffen in die Prozesssteuerung mittels Simulation durchgespielt und protokolliert werden. In der gleichen Planungssitzung wird daraus eine optimale Prozessführung für einen Geschäftsverlauf erarbeitet.

1. Übersicht

Das hier vorgestellte BB-Tool® gehört zur Klasse der Fließmodelle, jedoch mit einem wesentlichen Zusatznutzen, der betriebswirtschaftlich vollständigen, nahtlos integrierten und simulationsfähigen Kostenrechnung. Dieser Beitrag beschreibt am Beispiel des BB-Tool® einige der Möglichkeiten und Anwendungsbereiche für die Geschäftsprozesssimulation. Eine kurze Gegenüberstellung wichtiger Charakteristika von flussorientierten und von ereignisorientierten Modellen in Abschnitt 2 dient einleitend einer begrifflichen Klärung. Darauf folgt eine Skizze der Entwicklungsmotive für das BB-Tool® aus der Erfahrung mit Business-Process-Engineering-Projekten. Abschnitt 4 und 5 schildern Eigenschaften, Einsatzvorbereitung und Anwendungsbeispiele des BB-Tool®. Die Abschnitte 3 bis 5 bieten einen Diskussionsbeitrag zu unserer These, dass Leistungsspektrum und Anwendernutzen von konsequent entwickelten Fließmodellen für die Geschäftsprozesssimulation noch unterbewertet werden.

2. Fließmodelle und Geschäftsprozess-Simulation

Neben den ereignisorientierten Modellen zur Ablaufsimulation gibt es eine lange Tradition der flussorientierten Modelle. Beide Ansätze bilden dieselben Prozesse ab, die Modelle sind prinzipiell wechselseitig übersetzbar, doch die Ansätze betrachten einen Prozess aus unterschiedlichen Perspektiven, mit verschiedenen Werkzeugen, und mit je eigenen Vor- und Nachteilen. Die Frage nach der Perspektive – „Wo befindet sich ein Beobachter, der eine Prozesskette steuern möchte?“ – beantwortet ein ereignisorientiertes Modell grundsätzlich mit einem Beobachter, der zusammen mit dem Einzelobjekt, einem Werkstück etwa, und mit der

Bearbeitungsliste in der Hand den Prozess Stufe für Stufe durchlebt. Der Fliessprozess-Beobachter dagegen sucht die Vogelperspektive, um die Prozesskette zu überblicken, und lässt – zunächst ungeachtet der Objekteinheiten – einen Material-Strom kontinuierlich vorüberfließen. Die Detail-Auflösung ist in dieser Perspektive absichtlich begrenzt, denn anstelle der Detaildatensätze aus der Auftragsabwicklung werden beim Fliessmodell bewusst Durchschnittswerte verwendet. Die Konsequenzen daraus werden im folgenden deutlicher.

Die explizite Darstellung von Prozessraten, Beständen und Informationsverknüpfungen, wie wir sie im BB-Tool® verwenden, macht Kapazitäten, Lieferfähigkeit und Verfügbarkeit einer abgebildeten Prozesskette transparent. Dies trägt der Tatsache Rechnung, dass Kapazitätsengpässe, Lieferverzögerungen und Verfügbarkeitseinbrüche wichtige Ursachen für unerwünschte Kostenentwicklungen sind. Wie die praktische Erfahrung zeigt, werden aber sowohl Kostenverlauf als auch Kapitalbindung durch die Steuerung des Ablaufs und die daraus resultierende Verfügbarkeit von Beständen entlang der Prozesskette entscheidend beeinflusst. Daher gehört in die Geschäftsprozesssimulation unbedingt der Zugang zur Prozessdynamik. Wird die mittel- und langfristige Ablaufdynamik folgerichtig erfasst und bildet die prozesskettengestützte Kostenrechnung zugleich einen integralen Bestandteil des Simulationstools, steht den Entscheidungsträgern mit einem solchen Werkzeug der Weg offen, günstige Steuerungsvarianten für die Geschäftsprozesse zu ermitteln und allenfalls ungenutzte Potentiale in den bestehenden Prozessen wirksam und nachhaltig zu erschliessen.

Die Fliessmodell-Abbildung einer Prozesskette im BB-Tool® gestattet stichpunktartig gefasst eigenständige Darstellung der Steuerung, stand-alone Erfassung der Ablaufdynamik, zeitrichtige Integration der Prozesskosten und raschen Zugang zu den Führungskennzahlen für die vorausschauende Planung und Führung des abgebildeten Prozesses, allerdings um einen Preis. Wir müssen zugunsten der übersichtlichen Prozessbearbeitung bewusst auf die Detailtreue der ereignisorientierten Modellierung verzichten.

	Unikat	Mehrfachprodukt	Produktgruppen	Serienprodukt
Jobshop	✖			
Batchprocess		✖		
Montagelinie			✖	
Prozesslinie	⊙	⊙	⊙	⊙ / ✖

Table 1: Klassische Zuordnung (✖) und aktueller Trend der Zuordnung (⊙) in der Produkt-Prozess-Matrix nach Moody [2]

3. Entwicklung des BB-Tools® aus BPE-Projekten

Der in allen Branchen beobachtbare Trend zur kundenorientierten, flexiblen Erstellung der Marktleistung eines Unternehmens erfordert Verbesserungen bei der Beherrschung von Durchlaufzeit und Kapitalbindung. Moody [2] hat darauf hingewiesen, dass sich damit die

herkömmliche Zuordnung in der Produkt-Prozess-Matrix nicht nur für Produktgruppen, sondern sogar für Unikate zu den Prozesslinien hin verschiebt (vgl. Table 1).

Auf der Suche nach einem geeigneten Werkzeug zur Unterstützung von Business-Process-Engineering-Projekten (BPE) liessen sich vor diesem Hintergrund folgende sieben kritische Erfolgsfaktoren für ein solches Tool identifizieren:

- top-down Modellierung nach dem Grundsatz „so grob wie möglich, so fein wie nötig“
- systemunabhängige, simulationsfähige dynamische Verifikation mit grossem Zeithorizont
- Nachweis von Belastungs-/Kapazitätsengpässen und Verfügbarkeitsschwierigkeiten
- finanzielle Beurteilung über genügend lange Zeithorizonte integriert in die Simulation
- prozesslogische Verknüpfung von Prozessteilnehmern, Steuerungsarchitektur und Daten
- lokales und globales Benchmarking (Prozesskosten der Prozess-Stufen, Verfügbarkeit, Lieferbereitschaft, Leistungskosten) für Prozessvarianten und -alternativen
- kompatibel mit ereignisorientierten Prozessabbildungen

Mit der ständig zunehmenden Kundenorientierung einhergehend wurde auch bei BPE-Projekten ein wachsender Druck auf Zeit- und Mittelbudgets spürbar. Unsere BPE-Projekte haben vor Augen geführt, dass Fliessmodelle mit kontinuierlicher Simulation im Sinne der kritischen Erfolgsfaktoren eine hocheffiziente Lösung bieten. Wir sehen vier massgebliche Gründe dafür:

Zunächst erlaubt die *mittels flussorientierter Abbildung der Prozessketten gewonnene Vereinfachung* nicht nur einen kurzen Entwicklungszyklus für das Simulationsmodell, sondern anschliessend auch schnelle und zielgerichtete Problemanalysen, da die nötigen Simulationsläufe interaktiv und mit bescheidenem Ressourcenbedarf durchgeführt werden können.

Die *vollständige betriebswirtschaftliche Erfassung* der volumenabhängigen, zeitabhängigen und bestandesabhängigen Kosten jeder einzelnen Prozessstufe im Modell holt die Betriebserfahrung der Buchhaltungsverantwortlichen prozesslogisch in die Simulation, potenziert deren Nutzen für die vorausschauende Planung und unterstützt die Kostenbewertung von Lösungsvorschlägen quantitativ verlässlich.

Da Fliessmodelle *massstabsrobust* gebaut werden, können Teilmodelle problemgerecht mit unterschiedlichem Detaillierungsgrad ausgeführt werden, und trotzdem lassen sich die Teilmodelle ohne Schwierigkeiten koppeln. Hierdurch wird eine Grundvoraussetzung erfüllt, um vor- und nachgelagerte Prozesse mit geringstmöglichem Aufwand in die Analyse einbeziehen zu können. Weder der Zeitpunkt des Einbeziehens, noch Detailkenntnisse über die umliegenden Abläufe zwingen zu Kompromissen bei der lokalen Analyse, lediglich die Reichweite der Schlussfolgerungen wird beeinflusst. Anders betrachtet sind wir nicht darauf angewiesen, alle Modellteile auf der gleichen Detaillierungsebene anzugehen, um zu kompatiblen Modulen zu gelangen. Ausserdem wird hierdurch die logische Darstellung und Einbindung der Datenbanken eines Geschäfts mit ihren Entity-Relations-Diagrams (ERD) in

den Prozessablauf wesentlich erleichtert, denn die Schnittstellen können bedarfsgerecht und gleichzeitig budgetschonend so schlank wie möglich ausgeführt werden.

Schliesslich beschleunigt die eingangs erwähnte Begrenzung der Detailauflösung das Erarbeiten der Prozessoptimierung. Die *kontinuierliche Simulation auf der Basis der Prozessmittelwerte* lässt die Regelqualität der gewählten Steuerungsvariante unmittelbar hervortreten. Gegenüber ereignisorientierten Modellen mit einer Ereignisstatistik als Primärinformation schafft die kontinuierliche Simulation im Fließmodell mit ihrem günstigen Signal/Rausch-Verhältnis und den explizit ausgewiesenen Beständen entlang der Prozesskette sehr gute Voraussetzungen dafür, regeltechnische Konzepte für die Prozessoptimierung direkt nutzbar zu machen.

Ganz im Sinne einer 20/80-Regel haben unsere flussorientierten Simulationsmodelle gezeigt, dass der überwiegende Anteil der Prozessverbesserungspotentiale schon in der Strukturierungs- und Optimierungsarbeit mit bescheidenem Aufwand erreichbar ist gegenüber dem Aufwand, den ein stückgutorientiertes Werkzeug in dieser Phase abverlangen würde. Bei allen durchgeführten BPE-Projekten haben sich zudem die Prozessgruppen Zulieferung, Leistungserstellung, Vertrieb und Steuerung als strukturelle Invariante in der Modellierung herausgestellt. Wir schlagen daher vor, sie als kanonische Prozessgruppen zu bezeichnen, in Anlehnung an den mathematischen Begriff des kanonischen Koordinatensystems, in welchem sich ein Problem besonders einfach und übersichtlich darstellen lässt. In sorgfältiger Entwicklungsarbeit ist aus der breiten Erfahrungsbasis mit BPE-Projekten schliesslich eine verallgemeinerte und standardisierte Prozesskettenstruktur mit eben jenen kanonischen Prozessgruppen als Kernelementen hervorgegangen, das BB-Tool® oder Basic Business Planning Tool.

4. Das BB-Tool®

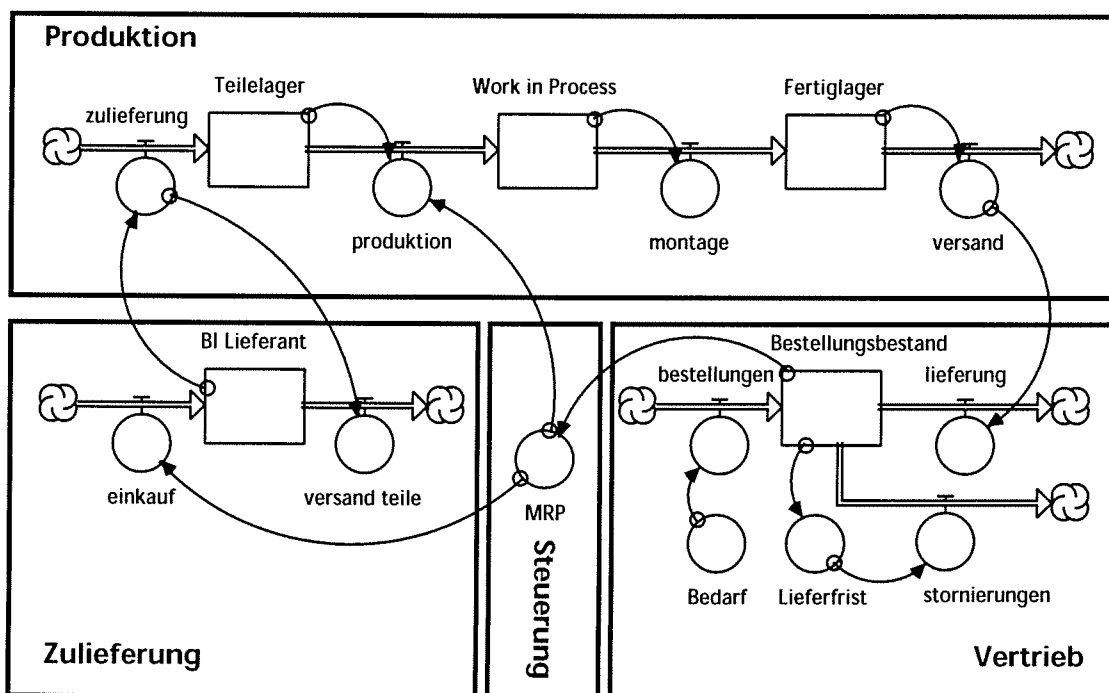
Die Prozesskettenstruktur im BB-Tool® widerspiegelt einerseits unsere Erfahrung aus BPE-Projekten und kommt andererseits dem Kundenwunsch nach einer systemunabhängigen, rasch einsetzbaren und einfach zu bedienenden Geschäftsprozesssimulation einen grossen Schritt entgegen. Das BB-Tool® enthält eine produktdefinierende Basis-Prozesskette mit den beiden vorbereiteten Schnittstellen zwischen Leistungserstellung und Umwelt, Zulieferung und Vertrieb, sowie die Steuerung des Systems (vgl. Bild 1).

Die kanonische Grundstruktur des BB-Tool® vereinfacht die Arbeit beim Kunden zur Feinabstimmung auf die individuellen Geschäftsprozesse. Hierzu wird das BB-Tool® durch Eingabe von vierzig Parametern, die alle aus den Kennzahlen der Betriebsführung des jeweiligen Geschäfts entnommen oder abgeleitet werden, den tatsächlichen Abläufen realitätsnah angepasst.

Vor dem Einsatz in Budgetplanung und Controlling Audit wird die Realitätsnähe des Tools mit Hilfe von betriebsbuchhalterisch dokumentierten Geschäftsverläufen der Vergangenheit überprüft und validiert. Die Validierungsarbeit erfüllt vor allem kundenseitig wichtige Funktionen. Neben dem Aufbau der nötigen Fertigkeiten im Umgang mit dem Tool bildet sich in dieser Phase die Kopplung zwischen Betriebserfahrung und Simulationstool heraus, und die AnwenderInnen verschaffen sich Sicherheit in der Interpretation und Umsetzung der

Simulationsergebnisse. Gleichzeitig wächst aus der Parallele von bekanntem Geschäftsverlauf und Simulation das Vertrauen in die Verlässlichkeit der Resultate. Soweit bedeutet Tool-Validierung nichts Besonderes, jedes vertrauenswürdige Simulationswerkzeug muss diese Hürde nehmen. Neu hingegen ist die kurze Zeitspanne, innerhalb der diese Phase bei Verwendung des BB-Tool® erfolgreich abgeschlossen werden kann. In der Regel genügen zwei Beratertage beim Kunden vom Kennenlernen des BB-Tool® und der Erarbeitung der Parameterwerte bis zum ersten tragfähigen Geschäftsplanungseinsatz vollauf.

Zur Ausrüstung für eine funktionstüchtige Simulation verlangt das BB-Tool® zuerst eine Prozessbeschreibung, welche die vier angesprochenen Elemente Leistungserstellung, Zulieferung, Vertrieb und Steuerung identifizieren hilft, sowie Angaben zur Definition der Prozesssteuerung. Zweitens werden Kostendaten benötigt, welche die Leistungskosten den drei Prozessabschnitten der Leistungserstellung zuzuordnen erlauben und die Aufteilung in Prozesskosten und Umlagekosten sichtbar machen. Drittens muss eine Betriebsrechnung für eine vergangene Geschäftsperiode vorliegen, welche einen Zeitraum von mindestens dreimal der gesamten Durchlaufzeit einschliesslich Zulieferung umfassen sollte.



© 1992 Dr. Ulrich La Roche 1998

Bild 1: Die Kernelemente im BB-Tool® in einer vereinfachten Darstellung

Anschliessend können mittels Simulation tatsächliche oder künftig erwartete Geschäftsverläufe mit der gewählten Steuerungsvariante durchgespielt und auf mögliche Potentiale, Schwachstellen oder vorzubereitende Massnahmen hin untersucht werden. Die besondere Realitätsnähe der BB-Tool®-Simulation ist dabei mit der Qualität vergleichbar, die in einem andern Zusammenhang von professionellen Flugsimulatoren erwartet wird: Nur mit einem Flugsimulator können in der Aus- und Weiterbildung von Airline-Piloten Erfahrungen im

Umgang mit komplexen oder gefährlichen Situationen der Flugpraxis zu vertretbaren Risiken und Kosten erworben und geschult werden. Übertragen auf die Geschäftsprozesssimulation mit dem BB-Tool® bedeutet dies, Prozesspotentiale, Planungs- und Steuerungsvarianten können ausgelotet werden, obwohl Änderungen im simulierten Ausmass den realen Prozess einem unzulässigen Risiko aussetzen würden. So wird dank Simulation auch in komplexen Geschäftssituationen sicheres Handeln ermöglicht. *Vor* der Umsetzung in den aktuellen Geschäftsablauf werden mit dem BB-Tool® die Massnahmenpakete erarbeitet, *während* der Umsetzung dient das BB-Tool® der Projektbegleitung zur Überprüfung der Massnahmen-Wirksamkeit, zur Aktualisierung der Kennzahlen, sowie zur Überbrückung von Kontroll-Lücken, die durch die Umstellungen bedingt sind. Zusammengefasst konzentriert sich der Einsatz des BB-Tool® hauptsächlich auf die folgenden Anwendungsbereiche:

- Replay von Geschäftsverläufen der Vergangenheit: Benchmarking verschiedener Geschäftseinheiten, Best Practices, Ursachenanalysen, Mitarbeiterschulung.
- Simulation verschiedener Massnahmen wie Ändern der Steuerung, der Durchlaufzeiten, der Kapazitäten und der Kostenparameter. Risikoanalysen für Prozessabschnitte und für den ganzen Prozess, Massnahmenvorbereitung: Welche Massnahmenpakete greifen, und wann ist der richtige Einsatzmoment für eine nachhaltige Wirkung?
- Simulation von Zukunfts-Szenarien: Wie reagieren die Prozesse auf Änderungen im Bestellungseingang? Welche Massnahmen sind vorzubereiten? Wie verhalten sich die Führungskennzahlen im Zeitraum zwischen Bestellungseingang und Abschluss der Betriebsbuchhaltung?

Das BB-Tool® ist wie schon erwähnt aus einer Reihe von BPE-Projekten hervorgegangen. Es ist besonders gut geeignet, um für bereits bestehende Geschäftsprozesse vorausschauende Planungs- und Führungsaufgaben zu unterstützen. Darüber hinaus ist das BB-Tool® aber auch ein empfehlenswertes Instrument, um kostengünstig die Vorgaben der Kennzahlen für ein Business-Process-Engineering Projekt zu ermitteln. In einem solchen Fall kann die Erfahrung mit dem BB-Tool® vollumfänglich in ein jeweils kundenspezifisch entwickeltes, im Unternehmensdetail strukturisomorphes Prozessmodell, das BPE-Tool®, übernommen werden.

Modul	Kundenziel	Beratungsziel	Aufwand
BB-Tool® mit Ausbildung	Tool für Geschäftsleitung und lokale Prozesseigner zur operativen Unterstützung	Einführen und Festigen des dynamischen Prozessden- kens inkl. Verständnis der Teilkostenrechnung	Tool plus 2 Tage Ausbildung
Pilot-Projekt	Messbarer Ersterfolg an begrenztem Teilablauf	Handwerk einführen und Vertrauen schaffen	vgl. Pilotprojekt- beschreibung
BPE-Tool®	Potentialanalyse und Master-Plan zur Prozessverbesserung	Teilprojektdefinition im BPE-Gesamtrahmen	einzelner Baustein ca. 1- bis 4-mal Pilotprojekt

Table 2: Das BB-Tool® im Rahmen des Beratungskonzepts

Beim BPE-Tool® fallen die durch Vorstrukturierung des BB-Tools® auferlegten Beschränkungen weg, und die volle Leistungsfähigkeit der individuellen Fließprozess-Modellierung kann für die Prozessoptimierung ausgeschöpft werden. Tabelle 2 zeigt die Einbettung des BB-Tool® in das simulationsgestützte Beratungskonzept und gibt eine grobe Übersicht zu den Einsatzgebieten beider Tools mit dem jeweils zugehörigen Aufwand.

5. Zwei Beispiele für den Anwendernutzen

5.1 Benchmarking mit dem BB-Tool®

Zwischen der summarischen Beschreibungsweise in den Vorgaben eines Verwaltungsrats und der konkreten, operativen Ausführung gilt es häufig, eine Verständnis- oder Umsetzungslücke zur Entwicklung ablaufwirksamer Massnahmen zu schliessen. Die Kapitalbindung aus der klassischen Betriebsbuchhaltung wird dementsprechend umgesetzt in Bestandes- und Verfügbarkeitskosten und wird verknüpft mit der Steuerung als wirksamem Hebel zur Beeinflussung der Verfügbarkeiten und Bestände. Weitere Begriffe wie Fixe und Variable Kosten, Margen, Kapitalkosten, Synergien, Umsatzwachstum und Umsatz erfahren sinngemäss eine Übersetzung in prozessrelevante Begriffe. Hervorzuheben ist hier die Übersetzung der fixen und variablen Kosten in Zeitkosten und Volumenkosten. Diese Übersetzung führt dazu, dass in den Zeitkosten auch die kapitalisierten Personalkosten einbezogen werden, wodurch eine Gleichbehandlung aller Kostenursachen erreicht wird. Damit werden bei der Bewertung von Geschäftseinheiten auch traditionelle Umlagebereiche denselben Beurteilungskriterien zugänglich, die üblicherweise bei den Prozessketten der Leistungserstellung zum Einsatz kommen.

5.2 Geschäftsführung und Beurteilung der Kreditwürdigkeit in KMU

Das BB-Tool® verknüpft die Prozessstufen über Durchsatzraten und Bestände einschliesslich der Steuerungseinflüsse. Die als Attributrechnung ausgeführte Teilkosten-/Stückkostenrechnung gibt für jeden einzelnen Prozessabschnitt und in jedem Zeitmoment über den dort zugeordneten, aktuellen Stückkostenwert Aufschluss. So werden nicht nur die beiden Fragen nach der Höhe der Kosten und nach dem Entstehungsort der Kosten beantwortet. Vielmehr kann nun auch die Frage nach dem Zeitpunkt ihrer Entstehung geklärt werden. Auch wenn die Belastung der Prozesskette von Zulieferung bis Endkunde bei unregelmässigem Geschäftsgang dynamischen Schwankungen ausgesetzt ist, liefert diese integrierte Kostenrechnung alle betriebswirtschaftlichen Kenngrössen prozessrichtig, einschliesslich der üblichen Bilanzmittelwerte. Damit können für die Zeitspanne vom letzten Abschlussdatum der Betriebsbuchhaltung bis zum nächsten Datum bereits aus dem bekannten Bestellungseingang betriebswirtschaftlich tragfähige Kennzahlen durch Simulation mit dem BB-Tool® gewonnen werden. Wie die Praxis gezeigt hat, bietet schon der universelle Prozessknoten BB-Tool® im Vergleich zur später nachvollzogenen Betriebsbuchhaltung eine Genauigkeit besser als ± 5 Prozent. Die Beurteilung der Kreditwürdigkeit des Unternehmens – eine der wichtigen Planungsaufgaben der Geschäftsleitung – kann deshalb mit dem BB-Tool® wesentlich belastbarer begründet werden als dies gestützt alleine auf den Bestellungseingang und die groben Erfahrungszahlen der Vergangenheit möglich wäre.

6. Marken; Hardware- und Softwarevoraussetzungen

BB-Tool® und BPE-Tool® sind geschützte Marken der Fast Focus Consulting Group. Für den Einsatz des BB-Tool® ist die Standardsoftware POWERSIM Voraussetzung (Powersim AS, Isdalstø, Norway; e-mail: powersim@modeld.no). Wir empfehlen als Hardware-Plattform IBM-kompatible PC mit einem Prozessor ab Intel 486 unter Windows 3.1x/95/98/NT oder Macintosh PowerPC mit VirtualPC & Windows95/98/NT. Datenschnittstellen: Windows DDE.

7. Literaturverzeichnis

- [1] Ulrich La Roche; Optimales Ablaufkonzept – Maximale Rationalisierung; *Technische Rundschau*, Vol. 45 (1987), pp. 76-79
- [2] Patricia E. Moody; Strategic Manufacturing; Dow Jones-Irwin/APICS, 1990; ISBN 1-55623-193-8
- [3] Robin Cooper, Robert S. Kaplan; Activity Based Costing: Resource Management at its Best; *Harvard Manager* 4/1991, pp. 87-94
- [4] Ulrich La Roche; System Dynamics Analysis of a Business as Prerequisite for Efficient Factory Planning; *Proc. of EMS*, Copenhagen, 1991, pp. 17-19
- [5] Ulrich La Roche; System Dynamics Analysis, a Prototyping Tool for Production System Design; *Proc. of the Int. System Dynamics Conf. 1992*, Utrecht, 1992, pp. 355-364; ISBN 92-9129-001-7
- [6] Ulrich La Roche; Fließprozessmodelle als Werkzeug zur Entwicklung von Geschäftsabläufen; *Technische Rundschau*, 1994, p. 32f
- [7] Ulrich La Roche; A Basic Business Loop as a Starting Template for Customized Business Process Engineering Models; *Proc. of the Int. System Dynamics Conf. 1994*, Stirling/UK, 1994, Band *Change Management*, p. 39f

Diskrete Simulation auf Stufe Auftragsabwicklung

Dipl. Ing. HTL Viktor Metzler

viktor.metzler@bluewin.ch, Fax./Tel. +41 56/441 56 27

Informatik & Beratung, Kestenbergrasse 29, CH-5210 Windisch

Zusammenfassung

Das Ziel der Entwicklung von MetSim® war, den Aufwand für eine Modellerstellung erheblich zu senken. Als Folge treten einfach editierbare Tabellen, nach dem Prinzip WYSIWYG, an die Stelle von sonst üblichen Programmiersprachen. Ihre Umsetzung in das eigentliche Simulationsprogramm geschieht voll automatisiert mittels eines Modellgenerators.

Mit diesem Konzept ist es möglich, komplexe Netzwerke mit mehreren Tausend Prozessen, in einer einfachen und übersichtlichen Art zu modellieren. Hilfen, die zur Visualisierung der Modelle dienen, stehen zur Verfügung. Sie leisten wertvolle Dienste für die Kunde zu Kunde Kommunikation, aber auch für das Process Reengineering.

1 Aufbau und Ablauf

1.1 Der Prozess Simulieren

Über die Auftrags-tabelle wird die Simulation initiiert. Der objektorientierte Aufbau gewährleistet eine hohe Flexibilität, bei hoher Geschwindigkeit, wenn neue Auftrags-situationen abzubilden sind.

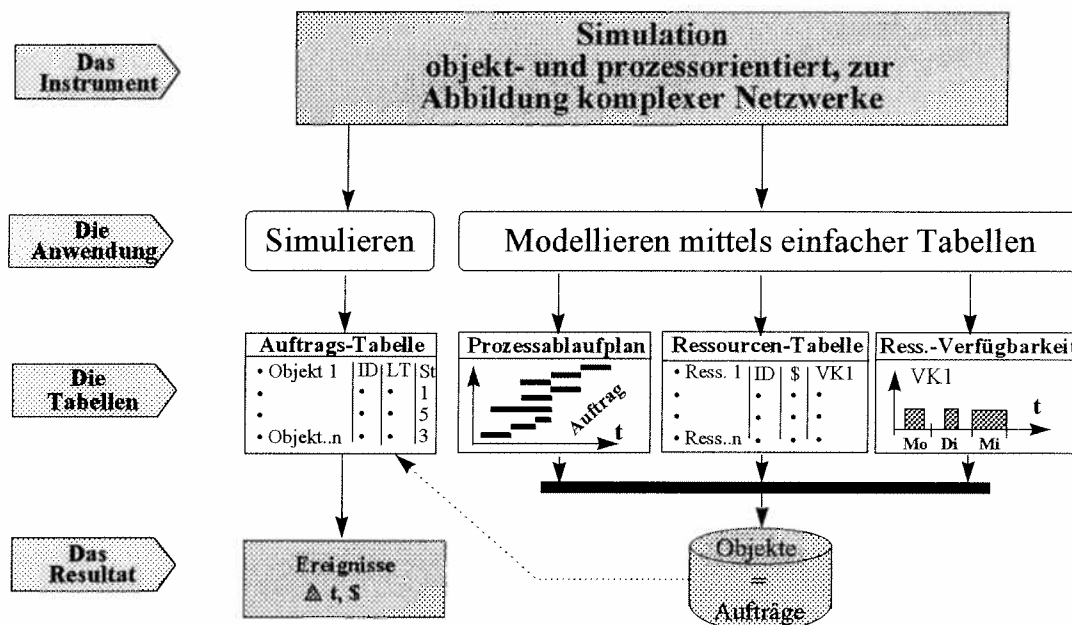


Bild 1: Simulieren und Modellieren mittels einfacher Tabellen

1.2 Der Prozess Modellieren

Über den Prozessablaufplan, im folgenden PAP genannt, wird das Modell kreiert. Die beiden Tabellen Ressourcen und Ressourcenverfügbarkeit sind notwendig, um den Prozessen Attribute zuzuordnen, wie z.B der Verrechnungssatz oder die Verfügbarkeit für eine Prozessstelle.

Der Modellerstellungsprozess gliedert sich in drei Phasen. In der ersten Phase wird die Produktstruktur gebildet (linke Seite im PAP, Bild 3). In Phase 2 definiert man die Prozesse für diesen Strukturbaum mit den drei Grössen, Prozesszeit, Einrichtzeit und Prozessstellen-ID (rechte Seite im PAP, Bild 3). In Phase 3 ist die Frage zu beantworten, welche weiteren Prozesse in das Modell miteinbezogen werden sollen. Gemeint sind Prozesse, die für einen reibungslosen Ablauf eingeplant werden müssen, wie z.B die Herstellung spezieller Betriebsmittel, oder die Erstellung von Dokumenten, Offerten, externe Zulieferanten etc. Diese Supportprozesse sind im PAP an den entsprechenden Stellen einzufügen, insbesondere dann, wenn sie auf dem kritischen Weg eines Auftrages liegen.

1.3 Die einstellbaren Parameter einer diskreten Simulation

Die Puffergrösse und die Anzahl Transporter stehen in quasi unendlicher Kapazität zur Verfügung. Zwei Gründe die dafür sprechen:

- ☐ Rasche Ergebnisse erzielen, mit einem Minimum an Einstellmechanismen
- ☐ Die frühzeitige Erkennung eines Problems gibt Spielraum für das Einleiten von Massnahmen

Zur Problemerkennung werden Hilfen bereitgestellt, die Ereignisse bzgl. Pufferbelegung und Transporte auf der Zeitachse ausweisen. Sie reichen aus, um an signifikanten Stellen mit entsprechenden Massnahmen rechtzeitig gegenzusteuern.

Mögliche Massnahmen zur Problembeseitigung:

- Anpassen der Ressourcenverfügbarkeit an die jeweilige Situation.
- Auswärtsvergabe
- Signifikante Transporte als Prozess im PAP definieren

2 Die Tabellen

2.1 Die Auftrags-tabelle als Konfigurator für Simulationsläufe

Wie schon erwähnt, setzt sie die Simulation in Gang. Mit drei Parametern wird ein Objekt modifiziert oder hinzugefügt und die Simulation kann gestartet werden. Die drei Parameter sind: Auftrags-ID mit Losgrösse und Abliefertermin. Die Auftrags-ID identifiziert immer den PAP und kann mehrmals in der Tabelle auftauchen, solange die Prozesse dieselben bleiben. Ergänzende Daten in ihr erleichtern die Kommunikation und sind frei wählbar, wie z.B. Typenbezeichnung, Dokumenten-ID, Projektleiter usw.

Auftragstabelle Stand: 30.06.1998							Aussteller: H. Muster
							Datum: 30.06.98
							File: Sim_001.csv
	Auftrags ID	Benennung	Typ	Losgrösse	vereinbarter Endtermin	Kunde	Verantwortlicher PL
Kol.-Nr	1	2	3	4	5	6	7
Zeilen-Nr							
1	MU0001.CSV	Motor	GM-150-A	5	12.12.98	A	Müller
2	MO0003.CSV	Motor	GM-150	2	13.12.98	B	Müller
3	MO0010.CSV	Motor	GM-350-A	1	14.12.98	C	Müller
4	MU0001.CSV	Motor	GM-150-A	10	15.12.98	D	Müller

Bild 2: Instrument zum Definieren von Auftragssituationen (objektorientiert)

2.2 Die Tabelle Prozessablaufplan (PAP), Produktstruktur und Prozesse

Der Prozessablaufplan verbindet die Produktstruktur mit ihren Abwicklungsprozessen in idealer Weise.

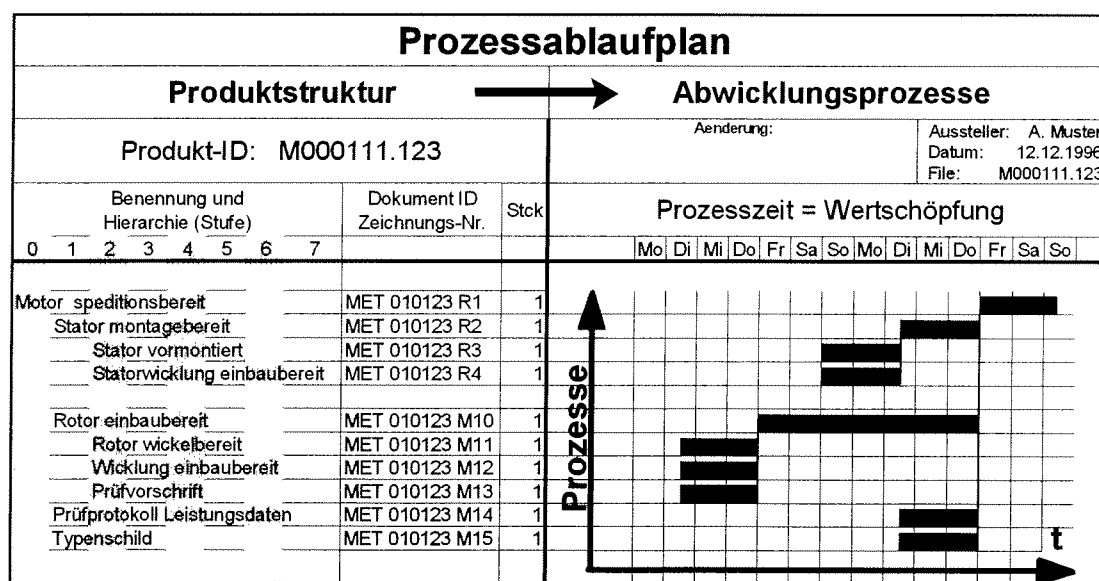


Bild 3: Instrument zum Definieren der Abwicklungsprozesse eines Produktes

Die Produktstruktur bildet das Rückgrat, auf dem die Prozesse aufgesetzt werden. Supportprozesse wie z.B. das Erstellen eines Betriebsmittels, einer Prüfvorschrift etc., also kein Lieferbestandteil des Auftrages, aber zur Abwicklung notwendig, können frei an entsprechender Stelle eingefügt werden und beeinflussen damit den Gesamtablauf mit Zeit und Kosten.

Die Produktstruktur wird abgeleitet aus dem physisch gegebenen Montageablauf des Produktes. Er bestimmt, welche Teile in den jeweiligen Montagestufen zusammenfließen. Die Frage nach Vorgänger- und Nachfolgerprozess bringt eine klare Struktur der Abwicklungsprozesse. Diese Vorgehensweise führt zu eindeutigen und korrekten Prozessabläufen und zeigt auf, wo parallele und serielle Prozesse liegen, und welche Auswirkungen auf den Gesamtablauf zu erwarten sind.

Als erstes Resultat, bevor simuliert wird, liefert der Modellgenerator ein Zwischenergebnis, das die theoretische Durchlaufzeit mit Angaben des spätest möglichen Starttermins ausgibt. Im Kapitel 4 "Nutzen der Simulation" wird darauf näher eingegangen (Bild 7).

2.3 Die Ressourcen-Tabelle

Die Ressourcenbeschreibung ist weiträumig angelegt. Mehrere Hundert Positionen wie Personen, Maschinen und Anlagen aber auch z.B. externe Zulieferanten und Dienstleister finden darin Platz.

Ressourcentabelle							
Benennung	Kapazität				Kosten/Erträge		
Kostenstellen ID	Verfügbarkeits ID	Tage pro Woche	intern=1 extern=2	Bediener	Verr.-Satz	Budgetierte Leistung	Budgetierte Erträge
ID	ID	Anzahl		Anzahl	SFr./Std	Std/Jahr	Fr./Jahr
Müller	IDENT_01.CSV	5.00	1	1.00	100	1600	160'000
Huber	IDENT_02.CSV	6.00	2	1.00	85	3000	255'000
4711	IDENT_05.CSV	3.00	2	1.00	65	1000	65'000
110111	IDENT_02.CSV	6.00	1	1.00	120	4000	480'000
110111	IDENT_05.CSV	5.00	1	1.00	70	1800	126'000
110111	IDENT_02.CSV	5.00	1	1.00	75	1800	135'000
110111	IDENT_10.CSV	5.00	1	1.00	100	1500	150'000
Total				7		14'700	1'371'000

Bild 4: Instrument zur Ressourcen-Definition

2.4 Die Tabelle Ressourcenverfügbarkeit

Sie ist im Sinne eines Kalenders aufgebaut, und ist das Instrument, mit dem jeder Ressource das Verfügbarkeits-Attribut zugeteilt wird. Es handelt sich hier um Schichteinstellungen, Pausen, Ferien und deterministische Ausfälle an den Ressourcen.

3 Standard Outputs

3.1 Business Daten (Fokus auf Unternehmung)

Der Verlauf des Geschäftsganges wird aufgezeigt (Bild 6). Er basiert auf dem in der Auftragsstabelle (Bild 2) definierten realen Auftragsvolumen. Eine Abbildung auf Stufe Auftrag ermöglicht detaillierte Untersuchungen und Vergleiche.

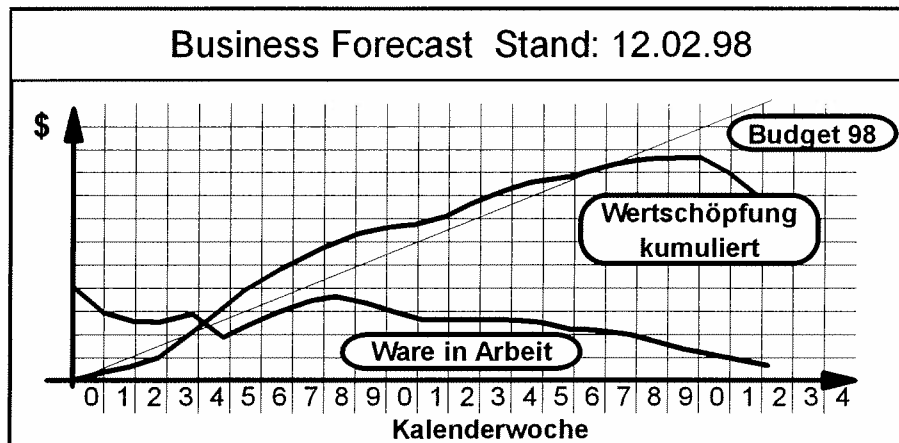


Bild 5: Kennzahlen der Unternehmung

3.2 Lieferprognose

Die Summe der Liegezeiten und die Summe der Prozesszeiten entlang des kritischen Pfades eines Auftrags sind die wesentlichen Grössen, die eine diskrete Simulation zu erbringen hat. Als Nebenprodukt der Simulation wird zusätzlich das Kostentotal pro Auftrag ausgewiesen.

Termin-Forecast Stand: 30.06.1998						Prognose Simulation		Liegezeit		Prozesszeit auf kritischem Pfad		H. Muster 30.30.98 Sim_001.csv											
	Auftrags ID	Los- grösse	Kosten [kFr]	Kunde	verein- barter Endtermin	spätest möglicher Start	Lieferung am:	Kalenderwoche															
Kol.-Nr	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	
Zeile-Nr																							
1	MU0001.CSV	5	10.5	A	18.02.98	01.01.98	15.02.98																
2	MO0003.CSV	2	10.5	B	05.03.98	02.01.98	08.03.98																
3	MO0010.CSV	1	10.5	C	09.03.98	03.01.98	14.03.98																
4	MU0001.CSV	10	21.1	D	10.03.98	04.01.98	17.03.98																
5	MO0012.CSV	4	10.5	E	13.03.98	05.01.98	11.03.98																
6	MU0001.CSV	1	2.1	A	17.03.98	06.01.98	19.03.98																
7	MO0050.CSV	5	15.9	A	20.03.98	07.01.98	20.03.98																
8	MU0001.CSV	5	10.5	A	27.04.98	08.01.98	27.04.98																
Total																							

Bild 6: Die wesentlichen Aussagen einer Simulation

3.2 Weitere mögliche Aussagen aus den OUTPUT-Dateien

MetSim® legt die Ergebnisse eines Simulationslaufes in ASCII-Dateien ab. Daraus können beliebige eigene Darstellungen, über Excel-Spreadsheets erzeugt werden.

4 Nutzen der Simulation

MetSim® berechnet die Startzeiten jedes einzelnen Prozesses. Ausgehend vom vorgegebenen Liefertermin, werden die Vorgängerprozesse durchgehend bis zum Ende der Prozesskette berechnet. Als Resultat erhält man die 3 Durchlaufzeit-Faktoren. Bild 7 zeigt die Entstehung und deren Nutzen.

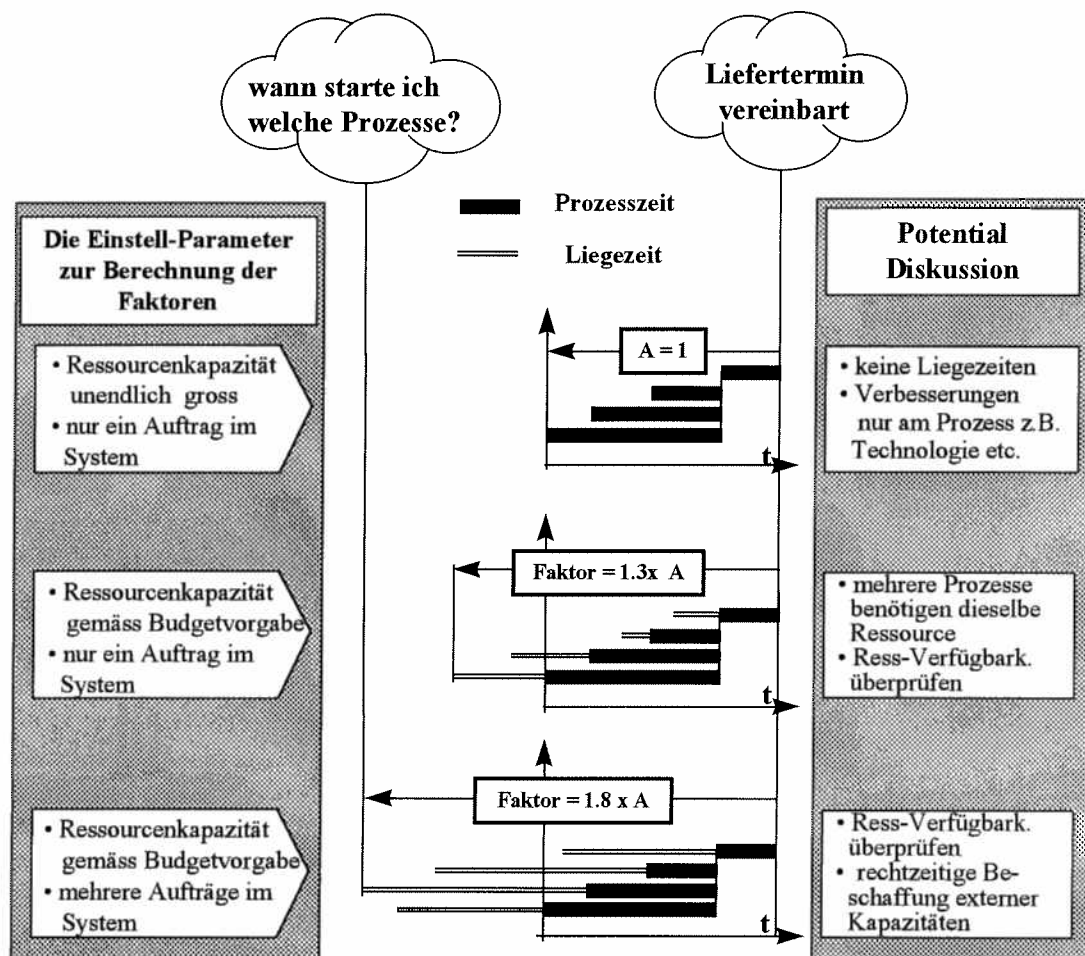


Bild 7: Die Berechnung der Durchlaufzeit-Faktoren

Weitere Anwendungsgebiete liegen in der Prozessoptimierung und im Reengineering. Die Vorteile liegen in der umfassenden Beschreibungsmöglichkeit von Prozessen und Ressourcen, und den daraus resultierenden konkreten Messgrößen bzgl. Durchlaufzeit und Kosten.

5 Einsatzgebiete in der Unternehmung, zentral/lokal

MetSim® ist flexibel in der Frage der Segmentierung. In der heutigen Produktionswelt mit prozessorientierten Strukturen, wird die Simulation sinnvoll dort installiert, wo die Auftragsverantwortung liegt. Das heisst aber auch, dass diese verantwortliche Stelle die Prozessbeschreibung (Modellieren) mitsamt Datenpflege zu erbringen hat.

6 Schlusswort

Die Beschreibungssprache erfüllt das Postulat einer einfachen Handhabung für das Simulieren und Modellieren und wird mittels der Standard-Tabellen auch gewährleistet. Sie dienen der innerbetrieblichen Kommunikation und leisten einen Beitrag zur Vereinheitlichung in der Darstellung von Prozessabläufen und deren Dokumentation.

7 SW-Limits

Anzahl Aufträge in einem Modell:	$\leq$	100
Anzahl Prozesse in einem Modell:	$\leq$	10'000
Anzahl Ressourcen	$\leq$	500
Max. Simulationszeitraum in Jahren	$\leq$	10

Bestandsentwicklung in einer Halbleiterfabrik nach dem Ausfall der Engpaßmaschinengruppe

Oliver Rose

rose@informatik.uni-wuerzburg.de

Universität Würzburg, Am Hubland, D-97074 Würzburg

Zusammenfassung

In Halbleiterfabriken kann nach einem Ausfall der Engpaßmaschinengruppe selbst noch nach Wochen ein erhöhter Bestand beobachtet werden. In dieser Studie wird ein einfaches Fabrikmodell entwickelt, um das Studium dieses Phänomens zu ermöglichen. Die vereinfachte Fabrik besteht aus einem detaillierten Modell der Engpaßmaschinengruppe und einem Verzögerungsglied, das die restliche Fabrik repräsentiert. Nach dem Durchlaufen des Verzögerungsglieds werden die Lose zur Engpaßmaschinengruppe zurückgeleitet, um den zyklischen Losfluß realer Halbleiterfertigungen zu modellieren. Durch das Experimentieren mit zahlreichen Szenarien wird gezeigt, daß das Modell das erwähnte Bestandsanstiegsphänomen erfolgreich reproduzieren kann.

1 Einführung

Um bei der Halbleiterherstellung den Einfluß verschiedener Abfertigungsregeln zu ermitteln, werden in der Regel Simulationsstudien eingesetzt. In zahlreichen Studien (z.B. [1], [6]) wird das Langzeitverhalten der Fabrikation in bezug auf mittlere Durchlaufzeiten, mittlere Bestandsniveaus, etc. bestimmt. Diese Art von Studien unterstützt die Auswahl von Abfertigungsregeln unter der Berücksichtigung von Zielkriterien wie z.B. einer gegebenen Wahrscheinlichkeit, den Liefertermin einzuhalten.

Es gibt jedoch Fabrikphänomene, die sich mit solch klassischen Simulationsansätzen nicht erklären lassen, da hier nur langfristige bzw. Gleichgewichtsleistungskriterien in Betracht gezogen werden. Eines dieser Phänomene ist die Beobachtung von großen Bestandsmengen noch nach Wochen nach einem katastrophalen Ausfall der Engpaßmaschinengruppe, d.h. alle Maschinen der Maschinengruppe, die den Fabrikdurchsatz bestimmen, fallen zugleich für mehrere Tage aus. Dieses auffällige und unerwünschte Verhalten wurde uns von Fertigungsleitern einer Speicherfabrik der Siemens AG berichtet.

Im Gegensatz zu klassischen Simulationsstudien muß die Entwicklung der Fabrik untersucht werden, z.B. der Bestand über der Zeit, nachdem der katastrophale Ausfall stattfand und eben nicht das Langzeitverhalten der Fabrik. Um dies zu erreichen, werden Simulationstechniken angewendet, die entwickelt wurden, um das Verhalten von Routing-Algorithmen nach dem Ausfall eines Knoten in einem Kommunikationsnetz zu vergleichen [5]. Der größte Nachteil dieser Techniken besteht darin, daß – anstatt einiger Dutzend Simulationsläufe bei klassischen Studien – Hunderte von Läufen benötigt werden, um das zeitliche Verhalten des Systems zu ermitteln.

Aus diesem Grund muß zuerst ein einfaches Fabrikmodell entwickeln werden, das das Verhalten der kompletten Fertigung mit Bezug auf obiges Problem widerspiegelt. Es ist wegen der extremen Laufzeit von hunderten von Simulationsläufen nicht möglich, die Experimente

mit einem kompletten Fabrikmodell durchzuführen. Es werden anhand des vereinfachten Modells einige Teilstudien durchgeführt, die zeigen, wie sich die Simulationsergebnisse ändern, wenn das Modell bzw. die Abfertigungsregeln geändert werden. Es zeigt sich, daß das Modell geeignet ist, das Ansteigen des Bestandes nach dem katastrophalen Ausfall zu reproduzieren. Darüberhinaus wird gezeigt, daß während der Zeit nach dem Ausfall lieferzeitbasierte Abfertigungsdisziplinen wie Critical Ratio in bezug auf Bestandsniveau und Durchlaufzeitvariabilität zu einer schlechteren Fertigungsperformance führen als FIFO. Das steht im krassen Gegensatz zu den Langzeitsresultaten, bei denen lieferzeitbasierte Regeln klar besser sind als FIFO [2].

Die Studie gliedert sich wie folgt. Das vereinfachte Fabrikmodell und die Simulationsdetails werden in Abschnitt 2 vorgestellt. Der Einfluß verschiedener Verzögerungsgliedmodelle wird in Abschnitt 3 untersucht.

2 Fabrikmodell

Typische Halbleiterfertigungen bestehen aus mehreren hundert Maschinen, die zugleich an mehreren Dutzend verschiedenen Produkten arbeiten. Die Halbleiter werden nach sog. Rezepten gefertigt, die meist mehrere hundert Prozeßschritte beinhalten. Wegen der schichtweisen Herstellung von Halbleitern durchlaufen die Scheiben bzw. Wafer bestimmte Maschinensequenzen mehrmals, d.h. sie bewegen sich in Zyklen durch die Fabrik. Speicherchips können bis zu 30 Schichten besitzen. Diese zyklische Besuchsreihenfolge der Maschinen ist für einen großen Teil der logistischen Probleme in der Halbleiterfertigung verantwortlich, da stets Lose mit unterschiedlichen Lieferzeitanforderungen um die Maschinen konkurrieren. Falls lieferterminabhängige Abfertigung angewendet wird, werden Lose, deren Liefertermin näher liegt, auf Kosten der Wartezeit anderer Lose bevorzugt. Die Konsequenzen dieses permanenten Neuordnens der Losprioritäten werden in Abschnitt 3 offenbar.

Um eine Simulationsstudie in bezug auf die Laufzeit durchführbar zu machen, wird ein Fabrikmodell benötigt, das die oben erwähnten Eigenschaften, aber erheblich geringere Komplexität bzw. Maschinen besitzt. Abbildung 1 zeigt das vorgeschlagene Fabrikmodell. Es besteht aus der Engpaßmaschinengruppe, einem Verzögerungsglied und einem Kontrollglied. Die Engpaßmaschinengruppe bestimmt größtenteils die Fabrikperformance [1] und wird daher detailliert unter Berücksichtigung der Maschinenanzahl, Prozeßzeiten und Abfertigungsdisziplinen modelliert. Die verbleibenden Maschinen werden als Verzögerungsglied modelliert. Jedes Mal, wenn ein Los den Engpaß verläßt wird es um eine zufällige Zeitspanne verzögert bevor es entweder die Fabrik verläßt oder erneut eine Engpaßmaschine anfordert. Das Kontrollglied entscheidet, ob bereits die geforderte Zahl von Zyklen vom Los durchlaufen wurde und sendet das Los entweder zum Fertigungsausgang oder zurück zur Engpaßmaschinengruppe.

Simulationsdetails

Für die Simulationsexperimente wurden folgende Parameter berücksichtigt. Die Engpaßmaschinengruppe besteht aus vier Maschinen. In der Fabrik werden vier Produkte gefertigt. Jedes Produkt hat eine Startrate von 0.0925 Losen/Stunde. Dabei ist die Zeit zwischen zwei Losstarts konstant. Die Prozeßzeiten an der Engpaßmaschinengruppe sind 0.7, 0.9, 1.1 bzw. 1.3 Stunden. Es wird angenommen, daß die Prozeßzeiten für jede Schicht identisch sind. Alle Produkte ha-

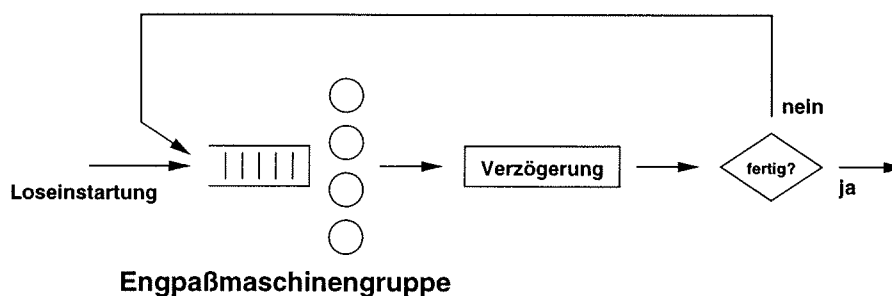


Abbildung 1: Fabrikmodell

ben 10 Schichten. Aus diesen Parametern resultiert eine Engpaßmaschinengruppenauslastung von 92.5 % ($= 4[\text{Produkte}] \cdot 0.0925[\text{Lose/Stunde}] \cdot (0.25 \cdot (0.7 + 0.9 + 1.1 + 1.3))[\text{Stunden}] \cdot 10[\text{Schichten}] / 4[\text{Maschinen}]$). Das ist ein sinnvoller Auslastungswert, wenn angenommen wird, daß der mittlere Prozentsatz von Ausfallzeiten weniger als 7.5 % beträgt. Die mittlere Zeitdauer, die ein Los im Verzögerungsglied verbringt beträgt 125 Stunden für jedes Produkt und für jede Schicht. Um die Anwendung von lieferzeitabhängigen Abfertigungsregeln zu ermöglichen, werden die Prozeßzeiten aller Maschinen benötigt. Aus diesem Grund wird die Verzögerungszeit in einen konstanten Betrag von 50 Stunden Prozeßzeit und in einen zufallsabhängigen Betrag von im Mittel 75 Stunden für Wartezeiten und sonstige Nicht-Prozeßzeiten aufgeteilt. Detaillierte Angaben über die Verteilungen der Wartezeiten werden im nächsten Abschnitt gemacht. Der Zielflußfaktor wurde auf 2.5 gesetzt, d.h. der Quotient aus der Durchlaufzeit und der Rohprozeßzeit soll 2.5 betragen. An der Engpaßmaschinengruppe soll dies durch entsprechende Abfertigungsregeln erreicht werden, während sich die restlichen Maschinen repräsentiert durch das Verzögerungsglied immer konform verhalten, da die mittlere Verzögerungszeit 125 Stunden bei 50 Stunden Prozeßzeit beträgt. Bei Losstart wird jedem Los ein Liefertermin von $\text{aktuelle Zeit} + 10 \cdot (125 \text{ Stunden} + 2.5 \cdot \text{Engpaßprozeßzeit})$ zugewiesen.

Wir betrachten die folgenden vier Abfertigungsregeln an der Engpaßmaschinengruppe.

FIFO (First In First Out) Die wartenden Lose werden in der Reihenfolge ihrer Ankunft abgefertigt. Dies ist die einzige Regel, die nicht zu einer Umordnung der Losreihenfolge führt.

SPTF (Shortest Processing Time First) Die wartenden Lose werden in der Reihenfolge ihrer Prozeßzeiten abgefertigt. Lose mit der kürzeren Prozeßzeit werden bevorzugt.

CR (Critical Ratio) Immer dann, wenn ein Los aus der Warteschlange genommen werden soll, wird der folgende Index für jedes wartende Los berechnet:

$$\text{CR} = \frac{\text{Liefertermin} - \text{aktuelle Zeit}}{\text{verbleibende Prozeßzeit}}.$$

Das Los mit dem kleinsten Index wird ausgewählt. Folglich werden Lose, deren Liefertermin näherliegt, bevorzugt.

ST (Slack Time) Im Vergleich zu CR wird der Index, der zur Auswahl des nächsten Loses

benötigt wird, durch eine Differenz und nicht durch einen Quotienten bestimmt.

$$CR = \text{Liefertermin} - \text{aktuelle Zeit} - \text{verbleibende Prozeßzeit}.$$

Auch hier wird das Los mit dem kleinsten Index aus der Warteschlange genommen. Die ST-Regel steigert die Prioritäten der Lose langsamer als CR, falls die Lose kurz davor sind, ihren Liefertermin zu überschreiten.

In den letzten drei Fällen wird bei gleicher Priorität nach FIFO abgefertigt. Im Gegensatz zu den ersten beiden Abfertigungsregeln berücksichtigen die letzten beiden die Liefertermine der Lose.

Das obige Fabrikmodell dient der Ermittlung der Fabrikperformance nach einem vollständigen Ausfall der Engpaßmaschinengruppe. Das Simulationsmodell wurde in ARENA [3] implementiert. Die statistische Nachbearbeitung der Meßwerte wurde mit eigenen Algorithmen durchgeführt.

Zuerst wurde die Länge der Warmlaufphase des Modells empirisch bestimmt [4]. Nach dem Start mit einem leeren System ist das so geschätzte Ende der transienten Phase bei etwa 1500 Stunden. Aus diesem Grund wurde der Ausfall nach 2000 Stunden simulierter Systemzeit gestartet. Alle Maschinen der Engpaßmaschinengruppe stehen augenblicklich für die Bearbeitung von Losen nicht mehr zur Verfügung. Nach einer Reperaturzeit von 50 Stunden fangen alle Maschinen zugleich an zu arbeiten. Die Simulation endet nach 5000 Stunden Fabrikzeit.

Während der 5000-Stunden-Läufe werden die Bestandsänderungen und die Durchlaufzeit der fertigen Lose erfaßt. Um die Datenmenge zu reduzieren und um die Messungen verschiedener Wiederholungsläufe synchronisieren zu können, wird folgendes Verfahren angewendet. Die simulierte Zeit wird in 10-Stunden-Intervalle eingeteilt. Für jedes Intervall wird der Mittelwert der Durchlaufzeiten für die Lose gebildet, die während dieser Zeit die Fertigung verlassen. In bezug auf den Bestand wird der zeitgewichtete Mittelwert der während des Intervalls beobachteten Bestandsniveaus berechnet. Auf diese Weise erhält man für jeden Lauf eine kondensierte Bestands- sowie Durchlaufzeitsequenz mit jeweils 500 Werten (5000 Stunden/10 Stunden). Um statistisch aussagekräftige Resultate zu erhalten, wurde jedes Experiment 500 mal wiederholt. Die Kurven, die im weiteren zu sehen sind, basieren alle auf der Mittelung der kondensierten Sequenzen von jeweils 500 Läufen. Die 95%-Konfidenzintervalle der Bestandsequenzen waren dann in allen Fällen hinreichend schmal.

3 Verzögerungsgliedmodellierung

Der Hauptteil der Studie besteht in einer Reihe von Experimenten zur Bestimmung des Einflusses des Verzögerungszeitmodells auf das Verhalten des Fabrikmodells. Zuerst wird ein Verzögerungsglied betrachtet, bei dem die Verzögerungszeit konstant ist, d.h. 125 Stunden für alle Produkte und alle Schichten.

Abbildung 2 zeigt die Bestandsentwicklung bei konstanter Verzögerung der vier betrachteten Abfertigungsregeln für die Engpaßmaschinengruppe. Um einige interessante Effekte zeigen zu können, wurde in diesem Fall die Simulationszeit auf 8000 Stunden ausgedehnt.

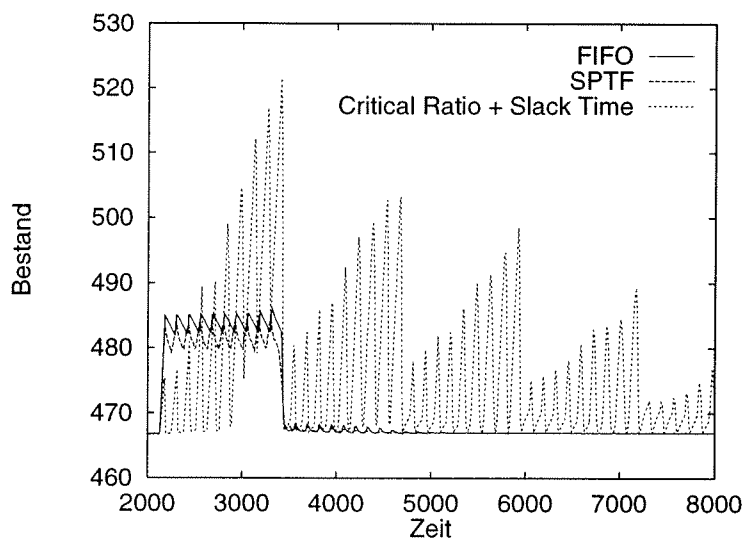


Abbildung 2: Bestandsentwicklung bei konstanter Verzögerung

Die Abfertigungsregeln FIFO und SPTF, die beide den Liefertermin nicht berücksichtigen, zeigen ein grundsätzlich anderes Verhalten als CR und ST. Während des Ausfalls steigt der Bestand bei FIFO und SPTF sehr schnell an, bleibt für etwa 1250 Stunden auf konstantem Niveau und fällt dann augenblicklich auf den Wert vor dem Ausfall. Bei SPTF sind die Bestände niedriger als bei FIFO. Nach etwa 3500 Stunden können in der Fabrik fast keine Auswirkungen des Ausfalls mehr beobachtet werden. Bei CR und ST sind die Verhältnisse jedoch anders. Die Bestände bilden sich langsamer, aber nach etwa 500 Stunden nach dem Ausfall ist das Bestandsniveau signifikant höher als bei FIFO. Sogar nachdem alle Lose, die den katastrophalen Ausfall selbst erlebt hatten, die Fabrik verlassen haben, ist dieses Systemverhalten zu beobachten, wenn auch in mit ständig sinkenden Spitzenwerten des Bestandsvolumens für jedes 1250-Stunden-Intervall. Zudem ist festzustellen, daß bei einem Zielflußfaktor von 2.5 kaum ein Unterschied im Verhalten von CR und ST beobachtet werden kann. In allen Fällen oszilliert das Bestandsniveau mit einer Frequenz von etwa $1/125[\text{Stunde}^{-1}]$. Die Ursache dieser Schwingungen ist die konstante Verzögerungszeit der zur Engpaßmaschinengruppe rückgekoppelten Lose. Die an den Engpaßmaschinen wartenden Lose werden im Verhältnis zur Rückkopplungsverzögerung sehr schnell bearbeitet.

Im folgenden wird das auffällige Systemverhalten bei der Verwendung von lieferterminabhängigen Abfertigungsregeln untersucht. Direkt nach Beendigung der Reparatur der Engpaßmaschinen werden die Lose durch CR priorisiert, die am nächsten zu ihrem Liefertermin liegen. Da alle Lose, die den Ausfall erlebten, einen näherliegenden Liefertermin besitzen als solche, die erst nach dem Reperaturende ankommen, werden diese Lose immer vor den neuen Losen bearbeitet. Neue Lose haben immer erst dann Zugriff auf die Maschinen, wenn alle blockierten Lose die Warteschlange verlassen haben. Durch die fortwährende Ankunft von neuen Losen steigt der Bestand an. Sobald alle Lose, die den Ausfall erlebt hatten, die Fabrik verlassen haben, übernehmen die neuen Lose, die durch diese Lose ständig blockiert wurden, ihre Rolle, und das Phänomen des Bestandsanstiegs wiederholt sich. Da die Engpaßmaschinengruppe eine Restkapazität von 7.5% besitzt wird die Höhe der Bestandsspitzen bei jedem Zyklus kleiner.

Wenn man nur die Bestandsgraphen betrachtet, liegt der Schluß nahe, daß SPTF eine vernünftige Abfertigungsregel darstellt, da sie beim katastrophalen Ausfall aller Engpaßmaschinen zum geringsten Bestandsniveau und zu den geringsten Bestandsschwankungen führt. In bezug auf die Durchlaufzeiten ist dies jedoch nicht mehr der Fall. Während die FIFO-Durchlaufzeiten das gleiche Verhalten zeigen wie die FIFO-Bestände, sind bei SPTF die Durchlaufzeiten meist geringer, in einigen Fällen jedoch erheblich höher als die FIFO-Zeiten. Der Grund liegt in der Priorisierung von Losen mit kurzen Prozeßzeiten an der Engpaßmaschinengruppe. Deshalb durchteilen Lose mit einer Prozeßzeit von 0.7 Stunden den Engpaß, während Lose mit 1.3 Stunden Prozeßzeit warten müssen, bis alle anderen Produkte abgefertigt worden sind. Sobald sowohl Bestand als auch Durchlaufzeit in Betracht gezogen wird ist erkennbar, daß durch FIFO-Abfertigung die Bestandsbildung verhindert und die Durchlaufzeitvariationen deutlich vermindert werden.

Bis jetzt wurde eine Fabrik betrachtet, die vollständig deterministisch arbeitet. Das trifft natürlich bei realen Fertigungen nicht zu. Insbesondere sind die Wartezeiten der Lose nicht im entferntesten deterministisch. Deshalb wird nun ein Verzögerungsmodell mit höhervarianzigen Verzögerungszeiten betrachtet. Die folgenden Experimente werden mit um 50 Stunden verschobenen, exponentiell verteilten Verzögerungszeiten mit dem Mittelwert von 75 Stunden durchgeführt, d.h. es wird eine konstante Summe von Prozeßzeiten und eine exponentiell verteilte Summe von Wartezeiten und anderen Nicht-Prozeßzeiten angenommen. Abbildung 3 zeigt die Bestandsentwicklung für verschobene exponentiell verteilte Verzögerungszeiten.

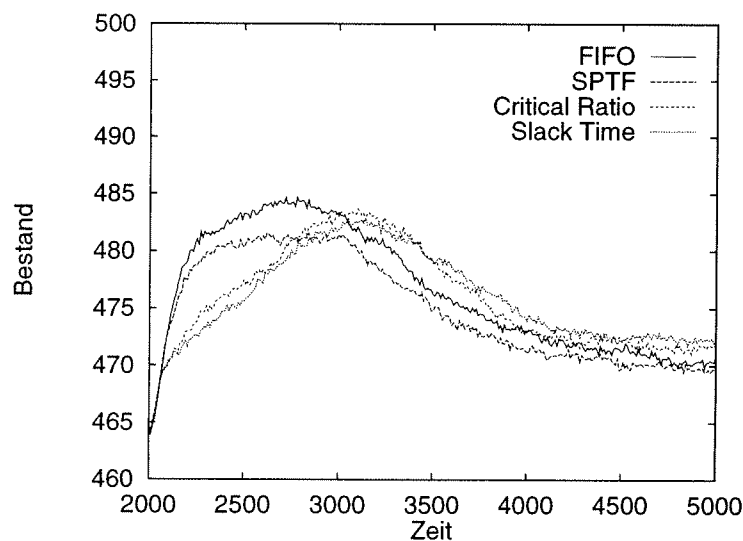


Abbildung 3: Bestandsentwicklung bei verschobener exp. verteilter Verzögerung

Bis auf die Schwingungen ist die Bestandsentwicklung ähnlich wie bei konstanten Verzögerungen. Für FIFO und SPTF bleibt der Bestand über etwa 1250 Stunden auf gleichem Niveau und fällt dann auf Gleichgewichtsniveau, wohingegen für CR und ST der Bestand während dieser 1250 Stunden ständig steigt. Der Bestandsabfall ist langsamer als bei der Fabrik mit konstanter Verzögerungszeit. Die Bestandsoszillationen verschwinden, da die Lose wegen der zufälligen Verzögerungszeiten nicht länger synchronisiert bleiben.

Die beiden bisher geschilderten Modellierungsansätze für das Verzögerungsglied stellen hin-

sichtlich der Verzögerungszeitvariabilität Extremfälle in bezug auf reale Fabriken dar. Im folgenden wird ein Verzögerungsmodell betrachtet, daß das reale Fabrikverhalten besser annähert: die Verzögerungszeit ist Erlang-5-verteilt mit einem Mittelwert von 75 Stunden und verschoben um 50 Stunden. Abbildung 4 zeigt die Bestandsentwicklung bei verschobener Erlang-verteilter Verzögerung.

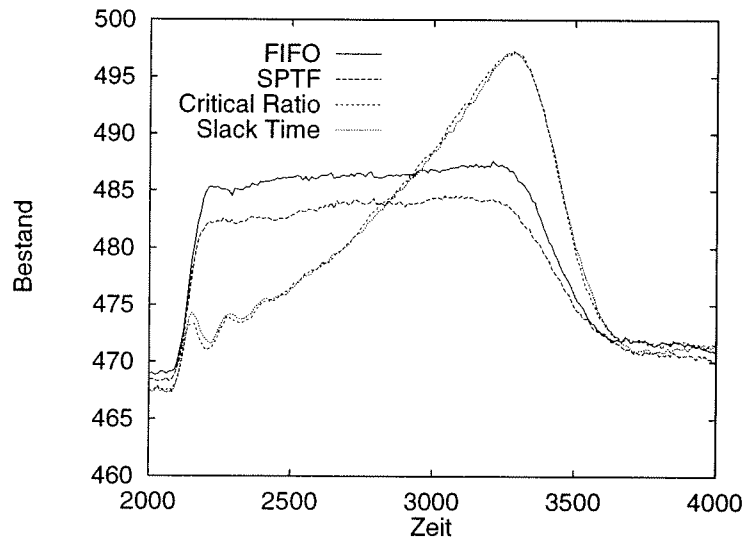


Abbildung 4: Bestandsentwicklung bei verschobener Erlang-verteilter Verzögerung

Das beobachtete Verhalten ist eine Mischung der Szenarien mit konstanter Verzögerung und verschobener exponentiell verteilter Verzögerung. Abgesehen von wenigen Schwingungen direkt nach der Beendigung der Reperatur sind wegen der zufälligen Verzögerungszeiten keine weiteren Oszillationen zu beobachten. Wie bei den anderen Ansätzen führt FIFO- bzw. SPTF-Abfertigung zu nahezu konstanten Beständen und CR- bzw. ST-Abfertigung zu beständig steigenden Bestandsniveaus.

Aus den obigen Resultaten kann man folgende Schlußfolgerungen ziehen. Nach dem Ende des Ausfalls der Engpaßmaschinengruppe zeigt das vorgestellte vereinfachte Fabrikmodell das gleiche Verhalten wie die reale Halbleiterfertigung. Bei CR-Abfertigung steigen Bestandsniveau und Durchlaufzeit an und erreichen ihr Maximum erst Wochen nach dem Ausfall. Dieses Verhalten wurde bei allen betrachteten Verzögerungsmodellen festgestellt. Dabei reichte die Variabilität der Verzögerungszeiten von Null bei konstanten Verzögerungen bis zu der einer verschobenen Exponentialverteilung. Bei geringer Variabilität tendiert das Bestandsniveau zu Schwingungen. In bezug auf das mittlere Bestandsniveau und die mittlere Durchlaufzeit für das Intervall ab dem Beginn des Ausfalls bis zum Wiedererreichen des Bestandsniveaus vor dem Ausfall führt FIFO-Abfertigung zu etwa denselben Werten wie CR-Abfertigung. Für die FIFO-Regel sind jedoch geringere Bestands- und Durchlaufzeitvariationen sowie Maximalwerte dieser beiden Meßgrößen zu beobachten. Falls die Verzögerungszeitvariabilität nicht zu nahe bei Null liegt, bietet die SPTF-Abfertigungsregel sogar noch bessere Resultate. In bezug auf den Bestandsanstieg in der Fabrik, der Wochen nach der erfolgreichen Reperatur augenscheinlich wird, kann man schließen, daß dieses Verhalten durch das Zusammenwirken von lieferterminabhängigen Abfertigungsdisziplinen und dem zyklischen Losfluß durch die Fertigung

zurückzuführen ist. Nur diese beiden, für die Halbleiterfertigung typischen Charakteristika zusammen führen zu einer Blockierung von neu ankommenden Losen und induzieren somit den permanenten Bestandsanstieg.

4 Zusammenfassung

In der vorliegenden Studie wurde ein vereinfachtes Fabrikmodell für eine Halbleiterfertigung untersucht. Es besteht aus einem detaillierten Modell der Engpaßmaschinengruppe und einem Verzögerungsglied zur Modellierung der restlichen Maschinen der Fabrik. Die eingestarteten Lose durchlaufen wiederholt Engpaßmaschinengruppe und Verzögerungsglied, um den schichtweisen Fertigungsprozeß von Halbleitern nachzubilden. Für die Experimente werden vier Abfertigungsregeln an der Engpaßmaschinengruppe betrachtet. Zwei von ihnen ohne Berücksichtigung von Lieferterminen (FIFO und Shortest Processing Time First) und zwei mit Liefertermineinfluß (Critical Ratio und Slack Time).

Das Fabrikmodell wird verwendet, um die Bestands- und Durchlaufzeitentwicklung nach einem katastrophalen Ausfall der Engpaßmaschinengruppe, d.h. alle Maschinen fallen gleichzeitig für einen längeren Zeitraum aus, zu untersuchen. Besondere Aufmerksamkeit wird der Modellierung des Verzögerungsgliedes gewidmet. Es zeigt sich jedoch, daß die Abfertigungsregeln einen größeren Einfluß auf das Fabrikverhalten haben als das jeweilige Verzögerungszeitmodell. Mit dem vorgeschlagenen vereinfachten Fabrikmodell war es möglich, das Verhalten von realen Halbfertigungsanlagen zu reproduzieren. Es zeigt sich, daß das Phänomen des permanent steigenden Bestandes durch eine Kombination von lieferterminorientierter Abfertigung und zyklischem Losfluß zustande kommt.

Literatur

- [1] L. F. Atherton und R. W. Atherton. *Wafer Fabrication: Factory Performance and Analysis*. Kluwer, Boston, 1995.
- [2] S. Brown, J. Fowler, H. Gold und A. Schömig. Measurable improvements in cycle-time-constrained capacity. In *Proceedings of the 6th International Symposium on Semiconductor Manufacturing*, 1997.
- [3] W. D. Kelton, R. P. Sadowski und D. A. Sadowski. *Simulation with Arena*. McGraw-Hill, New York, 1997.
- [4] A. M. Law und W. D. Kelton. *Simulation Modeling & Analysis*. McGraw-Hill, New York, 2nd edition, 1991.
- [5] W. Lovegrove, J. Hammond und D. Tipper. Simulation methods for studying nonstationary behavior of computer networks. *IEEE Journal on Selected Areas in Communications*, 8(9):1696–1708, 1990.
- [6] L. M. Wein. Scheduling semiconductor wafer fabrication. *IEEE Transactions on Semiconductor Manufacturing*, 1(3):115–130, 1988.

Eine modulare Prozessumgebung zur Visualisierung, Simulation und Auswertung von Verkehrsmessdaten

Th. Riedel

Thomas.Riedel@usa.net

Adaptive Traffic Control AG, Langgasse 27, 8400 Winterthur

Zusammenfassung

Simulation von Verkehr kann zum Entwurf, zur Bewertung und zur Verbesserung der Regelung einer Kreuzung beigezogen werden. Vor allem zur Bewertung einer Regelung muss die Simulation mit gemessenen Verkehrsdaten als Eingangsgrössen durchgeführt werden. Verkehr nach statistischen Kriterien im Simulator zu erzeugen, genügt nicht immer.

Nichts spricht dagegen, die Eingangsgrössen direkt zu messen und zeitverzugslos vom Simulator verarbeiten zu lassen. Der Simulator führt in einem solchen Fall gleichzeitig eine Visualisierung des gegenwärtigen Verkehrs auf der Kreuzung durch. Dafür versucht er, sein Bild mit der Wirklichkeit in Übereinklang zu bringen und wird dadurch zu einem Beobachter.

Um aus der Simulation Rückschlüsse ziehen zu können, müssen die Simulationsergebnisse nach verkehrstechnischen Kriterien ausgewertet werden können. Geschieht eine solche Auswertung nach Daten, welche in Echtzeit im Simulator verarbeitet werden, so muss sie gleichzeitig mit der Simulation durchgeführt werden. Nach einer Interpretation der Resultate ermöglicht dies sogar eine Einflussnahme auf den Regelungsprozess oder gar in einem Versuchsaufbau die Regelung selbst.

Die Auswertungen verschiedener Prozesse können einem Leitsystem über ein grösseres Gebiet zugeführt werden, welches damit in Echtzeit über die gewünschten Daten verfügt und sie zur Anzeige von Informationen oder zur Intervention benutzen kann.

Der Artikel beleuchtet das technische Konzept des Zusammenspiels der einzelnen Komponenten. Zu den beschriebenen Punkten ist in den letzten Jahren ein Paket von Programmen entstanden. Sie tragen den Namen p-Coq, was für "Protection for Control Quality" steht. Unterstützte Betriebssysteme sind Windows 95 und Windows NT. Gewisse Komponenten sind auch auf UNIX-Systemen lauffähig oder binden ältere Komponenten von Windows 3.x mit ein.

Das Protokoll zwischen den Prozessen ist öffentlich und dokumentiert, so dass auch Programme anderer Hersteller als Module mit einbezogen werden können.

Module aus der Familie p-Coq werden zur Zeit in der Stadt Zürich eingesetzt und auch in Verbindung mit Standard-Entwicklungsumgebungen für Verkehrsingenieure. Vor allem aber wird p-Coq für die interne Forschung verwendet.

1 Einführung

Die Welt der Verkehrsregelungsanlagen für den Strassenverkehr ist heute sehr heterogen. Neben wenigen modernen Umgebungen sind viele Systeme im Einsatz, welche in den Siebzigerjahren eingeführt worden sind.

Es ist wichtig, die Vorgänge auf den geregelten Strassenkreuzungen beobachtbar, messbar und vor allem **bewertbar** zu machen, sei es für die Einführung einer neuen Regelanlage, sei es zum Erkennen von Schwachstellen in geregelten Gebieten, oder sei es gar um zu überprüfen, ob die Regelung die Erwartungen erfüllt, die man in sie gesetzt hat. Der letzte Punkt kann in naher Zukunft stark an Bedeutung gewinnen, falls die Regelung von Kreuzungen an private Anbieter vergeben werden soll.

Es gibt bis heute eigentlich keine Normen, wie mathematisch und technisch eine solche Bewertung geschehen soll. Der vorliegende Artikel beleuchtet das technische Konzept, wie mit Messdaten aus dem Strassenverkehr umgegangen werden kann. Sie sind auch in [1] bereits am Rande skizziert worden. Für mathematische und verkehrstechnische Ansätze sei auf die Untersuchungen [3] verwiesen.

Die drei Aufgaben

- Bereitstellung der Eingangsgrößen (Quelle),
- Simulation und Visualisierung und
- Interpretation und Auswertung der Resultate

müssen für die Echtzeitverarbeitung **parallel** und, vor allem im Fall der Auswertung, **iterativ** durchgeführt werden: die Resultate der Auswertung müssen nach jedem Messwert mit wenig Aufwand neu berechnet werden können.

Die drei Aufgaben lassen sich als voneinander unabhängige Prozesse implementieren. Die Prozesse können **modular** kombiniert werden.

1.1 Die Hauptkomponenten

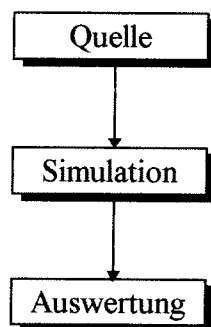


Bild 1: Einbettung der Simulation zwischen Quelle und Auswertung

Bild 1 zeigt die drei Hauptkomponenten Quelle, Simulation und Auswertung, aus welchen ein

System zur Visualisierung, Simulation und Auswertung von Verkehrsmessdaten mindestens bestehen muss. Die Quelle beliefert die Simulation mit Daten, die Auswertung interpretiert die Resultate der Simulation.

Die Komponenten sind *unabhängige Programme*, welche als Prozesse parallel nebeneinander her laufen können. Sie organisieren sich selbständig zu einer Client/Server-Architektur. Die Kommunikation zwischen den Prozessen geschieht entweder als Kommunikation zwischen parallelen Prozessen oder über TCP/IP, was die Verteilung der Programme auf verschiedene Rechner ermöglicht.

Die Komponenten sind *austauschbar*. So kann eine Quelle in Echtzeit Daten einer Kreuzung liefern oder Aufzeichnungen von Messungen einspielen.

Es ist auch denkbar, die Komponente "Simulation" wegzulassen und die Messdaten direkt auszuwerten, wenn zum Beispiel nur die Belegungszeiten von Detektoren ausgewertet werden sollen. Dazu müssen die Datenströme *lenkbar* und *genormt* sein.

1.2 Test eines Reglers

Die Hauptkomponenten können durch einen Regler ergänzt werden, den Prüfling, der getestet werden soll. Auch hier kann der Regler irgend ein Programm auf irgend einem Rechner sein, welcher über das standardisierte Protokoll mit dem übrigen System kommunizieren kann. Interne Größen des Reglers (beispielsweise der Inhalt von Variablen, welche zu Regelentscheiden führen) können auch direkt ausgewertet werden, ohne den Umweg über den Simulator nehmen zu müssen.

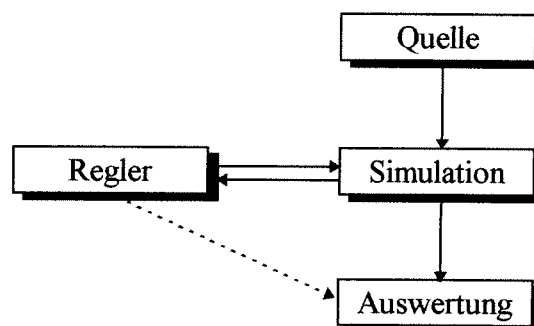


Bild 2: Test einer neuen Regelung in der Simulation

1.3 Regelung im Versuchsaufbau

Falls die Quelle die Daten in Echtzeit von einer Kreuzung bezieht und Simulation und Auswertung iterativ arbeiten, können die Entscheide des Reglers nicht an den Simulator, sondern direkt an die Kreuzung zurückgegeben werden. Dabei bildet die Verbindung von Simulation und Auswertung einen *Beobachter* im regelungstechnischen Sinn. Ein Beobachter berechnet

Eingangsgrößen für einen Regler, welche nicht direkt dem zu regelnden Prozess entnommen werden können.

Dies ist der letzte Schritt für den Test eines neuen Regelungsverfahrens, wie er in [4] beschrieben ist.

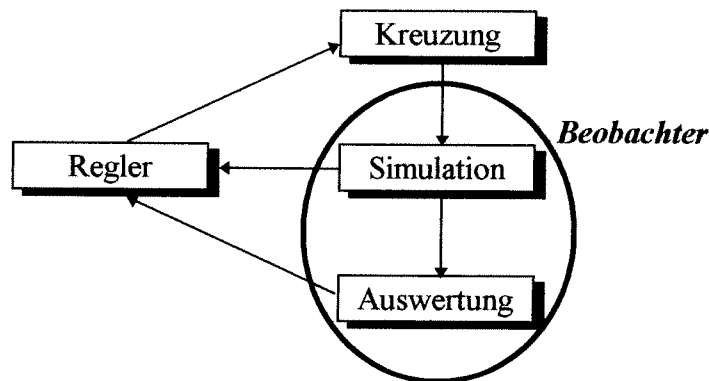


Bild 3: Regelung einer Kreuzung mit den beschriebenen Komponenten

2 Quelle

Es sind verschiedene Prozesse der Datenlieferung möglich, beispielsweise aus Aufzeichnungen, in Echtzeit über eine serielle Schnittstelle oder mit TCP/IP über das Netz. Mehrere Quellen können gleichzeitig eine Simulation mit Daten beliefern.

Vor allem der Weg über Standard-Entwicklungsumgebungen, hier am Beispiel von VS-PLUS gezeigt, erschliesst die Daten verschiedenster Hersteller von Steuergeräten und Kreuzungsregelungen.

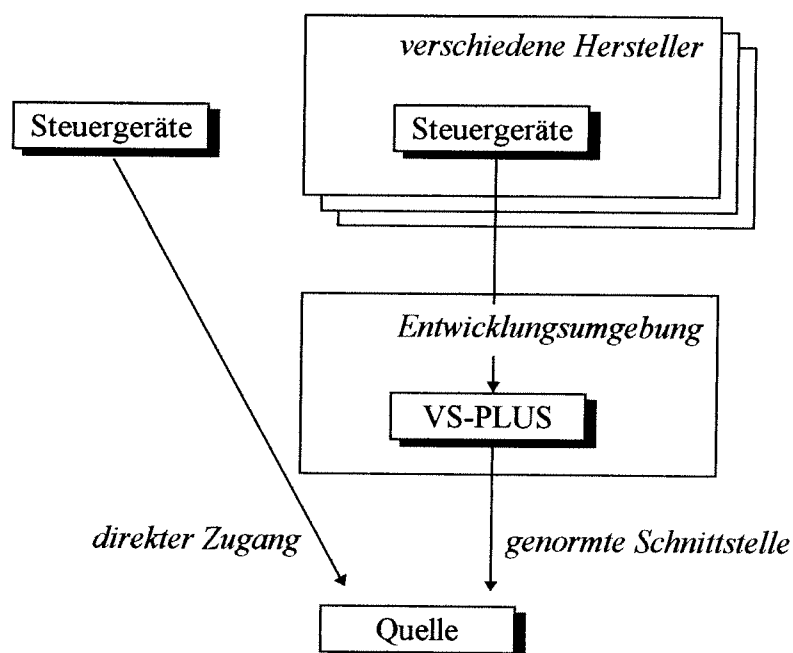


Bild 4: Erschliessung von Messdaten über die Steuergeräte verschiedener Hersteller

Das Archiv verwaltet Messdaten in komprimierter Form. Im folgenden Bild ist das Programm abgebildet, welches als Quelle die Daten aus dem Archiv holt.

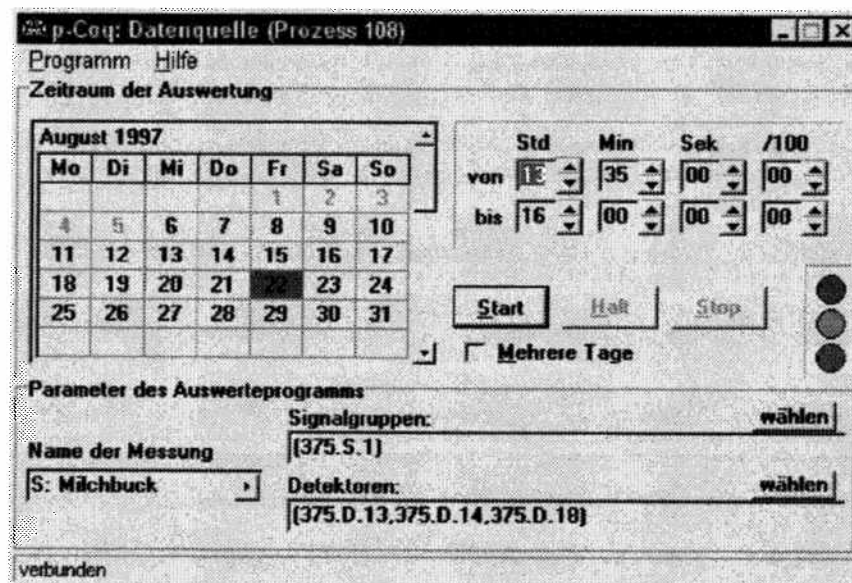


Bild 5: Zugriff auf das Archiv der Verkehrsmessdaten mit "p-Coq Archive"

3 Simulation

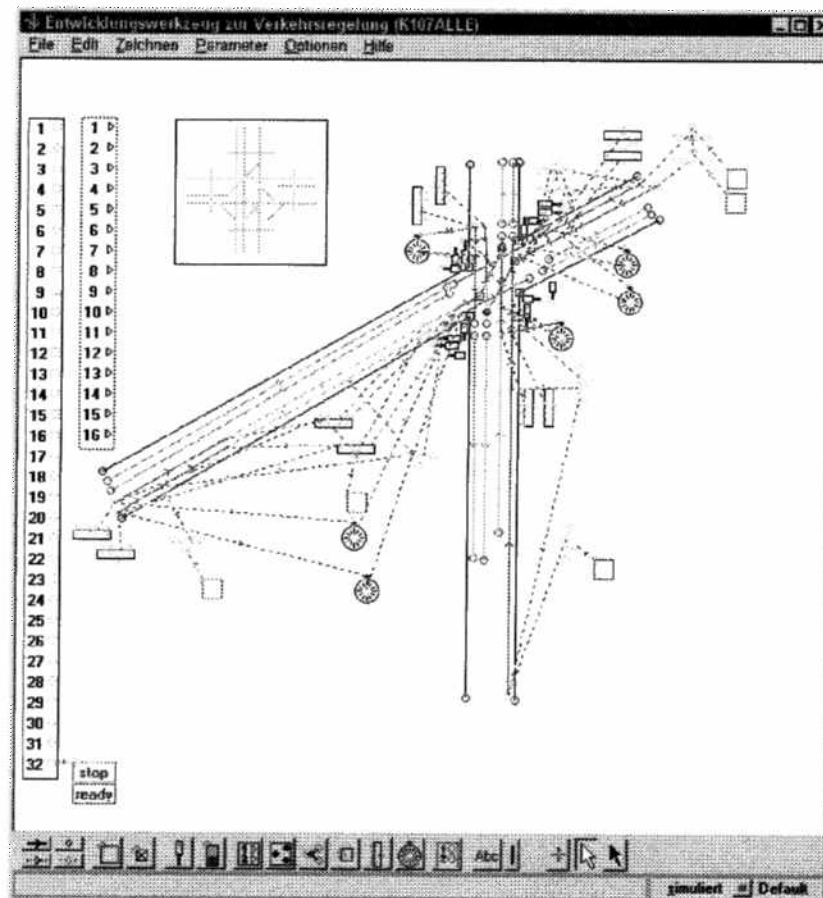


Bild 6: Beispiel einer simulierten Kreuzung mit "p-Coq Spy"

Zur Simulation wird ein Mikrosimulator verwendet, welcher über Synchronisationsmechanismen verfügt, um den inneren Zustand mit der Wirklichkeit abgleichen zu können [6, 7]. Er lässt in einfacher Art und Weise die graphische Definition von Kreuzungen zu [2, 5].

Der Simulator führt in einem solchen Fall gleichzeitig eine **Visualisierung** des gegenwärtigen (oder aufgezeichneten) Verkehrs auf der Kreuzung durch. Dafür versucht er, sein Bild mit der Wirklichkeit in Übereinklang zu bringen, und wird dadurch im regelungstechnischen Sinn zu einem **Beobachter**.

Pro Simulationsprozess können zur Zeit bis zu rund einem Dutzend Kreuzungen simuliert werden. Die Simulation kann aufgezeichnete Abläufe auch gerafft (begrenzt durch die Leistung des Rechners) verarbeiten oder im korrekten Zeitmass.

4 Auswertung

Verschiedene Prozesse der Auswertung können gleichzeitig die ermittelten Resultate in Echtzeit darstellen, interpretieren oder archivieren. Vor allem die Darstellung in Echtzeit lässt Entwicklungen augenfällig werden.

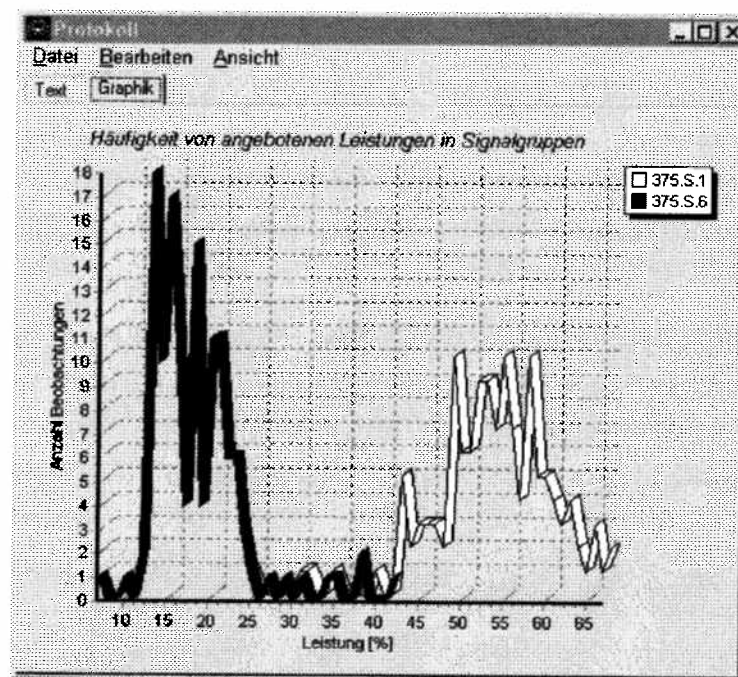


Bild 7: Beispiel einer Auswertung

Auswertung und Darstellung geschehen nach geläufigen Kriterien wie Phasendiagramm, Umlaufzeit, genormte Wartezeit, aber auch Verteilungsfunktionen und anderen statistischen Eigenschaften.

Weiter ist eine Bewertung der Regelung mit Qualitätskriterien, wie beispielsweise Kontrollgüte oder Stärke der Busbevorzugung, möglich.

5 Schlussfolgerungen und Ausblick

Wir verfolgen zwei Ziele:

- Die Vorgänge im Strassenverkehr sollen *veranschaulicht* und bewertet werden können.
- Zur Verbesserung des Verkehrsflusses sollen neue *Regler getestet* werden können.

Voraussetzung dafür ist, dass

- die Kreuzungsregler über die Steuergeräte die nötigen *Daten* liefern können,
- die Daten für jegliche Simulation oder Auswertung *einfach zugänglich* sind und
- die beteiligten Programme untereinander einfach *Informationen austauschen* können.

In einem ersten Schritt, der hier beschrieben worden ist, sind dafür die softwaretechnischen Grundlagen geschaffen worden. Weitere Schritte sind

- die hardwaretechnische Erschliessung der Kreuzungen über die Steuergeräte der verschiedenen Hersteller und
- die Entwicklung und Anwendung mathematischer Modelle zur Bewertung und anschließenden Regelung komplexerer Verkehrssysteme.

Es bleibt die Hoffnung, dass sich in Zukunft möglichst viele Programme an den beschriebenen Standards ausrichten, um dem Verkehrsingenieur eine interessante Programmumgebung bieten zu können, damit er so gut wie möglich die Probleme des Strassenverkehrs lösen kann.

Literatur

- [1] Th. Riedel. *Regelung in Verkehrssystemen*, Dissertation Nr. 10349, ETH Zürich, 1994
- [2] Th. Riedel. *A Traffic Control Development Tool - User's Guide*, Matsushita Communication Industrial Co., Ltd., 1995
- [3] Th. Riedel. *Auswertung von Verkehrsmessdaten am Beispiel der Kreuzung 107 in Zürich*, Bericht Nr. 95-01 der Adaptive Traffic Control AG, 1995
- [4] Th. Riedel. *Regelung der Kreuzung 107 (Hagenholz- / Binzmühle- / Thurgauerstrasse)*, Bericht Nr. 96-02 der Adaptive Traffic Control AG, 1996
- [5] Th. Riedel. Adaptive Regelung komplexer Strassenkreuzungssysteme, *Jahrestagung der Fachgruppe 4.5.8 Simulation und Verkehr: Simulation in der strategischen Verkehrsplanung*, ASIM Mitteilungen aus den Fachgruppen, 1996, Seiten 37 bis 40
- [6] Th. Riedel. *Verfahren und Vorrichtung zur Verkehrsregelung*, Europäischer Patentantrag 96111984.9 (noch in Bearbeitung), 1996
- [7] Th. Riedel. An Observer for Adaptive Intersection Control, *8th IFAC Symposium on Transportation Systems*, 16.-18. Juni 1997 in Chania, Griechenland, 1997, Seiten 899 bis 904

PUBLIKATIONEN VON UND MIT ASIM

ASIM - Buchreihen

Seit 1989 gestaltet ASIM die Buchreihe *Fortschritte in der Simulationstechnik* im Vieweg - Verlag. In dieser Reihe erscheinen Monographien zum Thema Simulationstechnik und die Proceedings der ASIM - Tagungen. Nach Anregung von Mitgliedern, auch eine Möglichkeit zur Veröffentlichung von speziellen Monographien, Dissertationen, Habilitationen und Software Guides zu schaffen, wurde 1996 die zweite Buchreihe *Fortschrittsberichte Simulation*, im ASIM - Eigenverlag in Zusammenarbeit mit dem ARGESIM - Verlag, eingerichtet.

Ab 1999 werden beide Reihen gemeinsam mit den Reihen *Advances in Simulation* und *Frontiers in Simulation* von SCS Europe (SCS Europe Publishing House) herausgegeben.

ASIM-Mitteilungen

Die ASIM - Fachgruppen berichten über ihre Treffen in den *ASIM Mitteilungen*. Diese werden von ASIM im Eigenverlag herausgegeben; die Hefte des eigenen Arbeitskreises („Primär-Fachgruppe“) sind teilweise kostenlos für seine Mitglieder, ansonsten gegen geringe Schutzgebühr erhältlich. Bisher sind 60 Hefte erschienen bzw. im Erscheinen. In dieser Reihe erscheinen auch Tagungsunterlagen von Tagungen und Seminaren mit ASIM - Beteiligung.

ASIM Nachrichten, ASIM Jahresbericht

Seit Sommer 1996 erscheinen *ASIM Nachrichten*, die einerseits aktuellen Themen zur Entwicklung der Simulationstechnik diskutieren und andererseits auch kurzfristige Mitgliederinformationen enthalten sollen (Fachgruppentreffen, Organisatorisches etc.). Diese Nachrichten werden jeweils mit der März-, Juli- und Novemberausgabe von *SNE* mitverschickt, sie können auch als deutschsprachige und ASIM - spezifische Ergänzung zu *SNE* gesehen werden.

Der *ASIM Jahresbericht* gibt einen Rechenschaftsbericht über das vergangene Jahr, zeigt Entwicklungen für das kommende Jahr auf, und gibt als *ASIM Vorstellungsbroschüre* einen allgemeinen Leistungsbericht von ASIM.

EUROSIM Simulation News Europe - SNE

ASIM ist Gründungsherausgeber von *EUROSIM Simulation News Europe (SNE)*, der Informationszeitschrift der europäischen Simulationsgesellschaften. *SNE* erscheint dreimal jährlich und wird den Mitgliedern sofort nach Erscheinen zugestellt. Der Bezug von *SNE* ist im ASIM - Mitgliedsbeitrag enthalten.

Neben Informationen und Neuigkeiten von ASIM und den anderen europäischen Simulationsvereinigungen bietet *SNE* unter anderem Softwarevergleiche, Essays und Publikationen zu aktuellen Themen der Simulationstechnik und zu Entwicklungen, Industry News und einen Veranstaltungskalender über Simulationstagungen und Simulationenkurse. *SNE* entwickelt sich derzeit in Richtung einer allgemeinen europäischen Zeitschrift für „Simulation News“, sowohl mit Neuigkeiten, als auch mit kurzen technischen und wissenschaftlichen Beiträgen

ARGESIM Reports

Die *ARGESIM Reports* werden von ASIM mitvertrieben. Eine Tabelle der lieferbaren Titel ist im Bericht über die ARGE Simulation News (ARGESIM) zu finden. ARGESIM (ARGE Simulation News) organisiert die Verwaltung von ASIM, den betreibt den ASIM WWW - Server und gibt mit ASIM die Reihe *Fortschrittsberichte Simulation* heraus.

Unterstützung und Mitherausgabe von Publikationen

ASIM unterstützt die Herausgabe des wissenschaftlichen *EUROSIM - Journals Simulation Practice and Theory (SIMPRA)*, die *EUROSIM* herausgibt und in deren Editorial Board auch Mitglieder von ASIM mitwirken. *SIMPRA* kann von ASIM - Mitgliedern vergünstigt abonniert werden.

Die Förderung weiterer Publikationen geschieht z. B. durch Kooperationsverträge mit den Zeitschriftenreihen der internationalen Gesellschaften SCSi und IMACS (d.h. mit *Simulation*, *Transactions in Simulation* und *Mathematics and Computers in Simulation*), bzw. durch Gestaltung von Sonderheften in bestimmten Reihen wie *it & ti*, etc.

ASIM - BUCHREIHEN

Seit 1989 gestaltet ASIM die Buchreihe *Fortschritte in der Simulationstechnik* im Vieweg - Verlag. In dieser Reihe erscheinen Monographien zum Thema Simulationstechnik und die Proceedings der ASIM - Tagungen. Nach Anregung von Mitgliedern, auch eine Möglichkeit zur Veröffentlichung von speziellen Monographien, Dissertationen, Habilitationen und Software Guides zu schaffen, wurde 1996 die zweite *Fortschrittsberichte Simulation*, im ASIM - Eigenverlag in Zusammenarbeit mit dem ARGESIM - Verlag, eingerichtet.

Generell geben die in diesen Reihen erscheinenden Bücher einen Querschnitt durch die Simulationstechnik, deren Entwicklung und Anwendung. Auf wissenschaftlichem Niveau werden Ergebnisse der Forschung zusammengetragen, Tests und Entwicklungen bewertet, Methoden zur Lösung von Problemen vorgestellt, bzw. neue Entwicklungen rasch präsentiert. Damit soll die Reihe Forum für die Beteiligten des Arbeitsfeldes Simulationstechnik sein.

Sie wenden sich an Ingenieure in: Physik, Biomedizin, Maschinenbau und Elektronik, Forschungsinstitute im Hochschulbereich sowie an Studierende mit themenbezogenen Aufgaben.

Die Reihen haben sich zum Ziel gesetzt, die Theorie aufzuarbeiten, ohne den Blick auf die Anwendungen zu verlieren. Sie verbinden so die naturwissenschaftlichen Grundlagen mit den ingenieurgerechten Anwendungen.

Ab 1999 werden beide Reihen gemeinsam mit den Reihen *Advances in Simulation* und *Frontiers in Simulation* von SCS Europe (SCS Europe Publishing House) herausgegeben.

Darüber hinaus publizieren die Fachgruppen bei einer Zusammenarbeit mit anderen Gesellschaften, Institutionen, etc. auch in anderen Buchreihen wie z. B. in den SCS - Book Series.

Buchreihe Fortschritte in der Simulationstechnik

Die Bände der Reihe erscheinen im Vieweg - Verlag und sind im Buchhandel erhältlich. Diese Reihe konzentriert sich auf folgende Inhalte:

- Statusbände der einzelnen ASIM-Fachgruppen (früher: Arbeitskreise) in Form von Handbüchern, deren Inhalt alle zwei bis drei Jahre aktualisiert wird.
- Übersichtswerke zu Modellbildung und Simulation
- Tagungsbände zum jährlichen ASIM - Symposium Simulationstechnik
- Monographien mit allgemeinem oder einführendem Charakter

Erschienen sind bisher:

- F. Breitenecker, I. Troch, P. Kopacek (Hrsg.); Simulationstechnik, 6. Symposium in Wien, September 1990; ISBN 3-528-06401-3
- F. Breitenecker, H. Ecker, I. Bausch-Gall; Simulation mit ACSL; ISBN 3-528-06381-5
- Dj. Tavangarian; Simulation und Entwurf elektronischer Schaltungen
- Dj. Tavangarian (Hrsg.); Simulationstechnik, 7. Symposium in Hagen, September 1991, ISBN 3-528-06440-4
- O. Rathjen; Digitale Echtzeitsimulation; ISBN 3-528-06517-6
- A. Sydow (Hrsg.); Simulationstechnik, 8. Symposium in Berlin, September 1993, ISBN 3-528-06555-9
- A. Kuhn, A. Reinhardt, H.-P. Wiendahl (Hrsg.); Simulationsanwendungen in Produktion und Logistik
- J. Biethan, J. Hummelt, B. Schmidt, Th. Witte (Hrsg.); Simulation als betriebliche Entscheidungshilfe; ISBN 3-528-06641-5
- G. Kampe, M. Zeitz (Hrsg.); Simulationstechnik, 9. Symposium in Stuttgart, Oktober 1994, ISBN 3-528-06622-9
- W. Krug (Hrsg.); Simulationstechnik, 10. Symposium in Dresden, September 1996, ISBN 3-528-06889-2
- R. Grützner (Hrsg.); Modellbildung und Simulation im Umweltbereich, Oktober 1997, ISBN 3-528-06940-6
- A. Kuhn, S. Wenzel; Simulationstechnik, 11. Symposium in Dortmund, November 1997, ISBN 3-528-06956-2

Buchreihe Fortschrittsberichte Simulation

Die Bände der neuen ASIM - Buchreihe *Fortschrittsberichte Simulation* zeigen neueste Lösungsansätze, Methoden und Anwendungen der Simulation in Theorie und Praxis. Die Reihe umspannt neue Entwicklungen in Grundlagen und Anwendung der Simulation in einem immer breiter werdenden Spektrum, z. B. Ingenieurwissenschaften, Naturwissenschaften, Medizin, Ökonomie, Ökologie und Soziologie. Auf wissenschaftlichem Niveau werden Ergebnisse der Forschung zusammengetragen und neue Methoden zur Lösung von Problemen vorgestellt, bzw. neue Entwicklungen rasch präsentiert. Neben der Dokumentation des jeweiligen aktuellen Standes und der Entwicklungen der Simulationstechnik werden aber auch deren Grenzen aufgezeigt.

ASIM hat diese Reihe 1996 im Eigenverlag als Ergänzung zur ASIM - Buchreihe *Fortschritte in der Simulationstechnik* ins Leben gerufen, um ein rasches und kostengünstiges Publikationsmedium für neue Entwicklungen in der Simulationstechnik anbieten zu können.

Die Publikation der Reihe erfolgt innerhalb der Reihe *ARGESIM Reports* der ARGE Simulation News. Diese Kooperation vermittelt dabei zum europäischen Umfeld und zur internationalen Publikation.

Die *Fortschrittsberichte Simulation* konzentrieren sich entsprechend ihrer Zielrichtung auf

- Monographien mit speziellem Charakter, wie z. B. Dissertationen und Habilitationen
- Berichte zu größeren Arbeitskreistreffen (mit referierten Beiträgen)
- Berichte von Forschungsprojekten
- Handbücher zu Simulationswerkzeugen (User Guides, Vergleiche, Benchmarks), und ähnliches.

Die Bände erscheinen im ASIM / ARGESIM - Verlag und sind direkt über ASIM (München) und ARGESIM (TU Wien), aber auch über Bestellung im Buchhandel erhältlich. Bisher sind fünf Bände erschienen (siehe folgende Tabelle), weitere in Vorbereitung. Für Autoren sind sehr günstige Preisstaffelungen bei Mengenabnahmen (z.B.: Dissertations-Pflichtexemplare) vorgesehen.

Interessierte Autoren für beide Reihen mögen sich an die Betreuer der Reihen wenden:

- Dr. I. Bausch-Gall, München
- Prof. Dr. F. Breitenecker, TU München
- Prof. Dr. G. Kampe, Fachhochschule Esslingen
- Prof. Dr. D.P.F. Möller, TU Clausthal

Band Nr.	Autor(Hrsg.), Titel, ISBN	ASIM-Mitglieder	Allg.
1	C. Westerkamp: Anwendung der Mehrgrößen-Parameterschätzung zur Simulation von linearen passiven Netzwerken; 1996; ISBN 3-901608-51-6	DM 30.-	DM 40.-
2	M. Salzmann: Genetische Algorithmen in diskreter Simulation; 1996; ISBN 3-901 608-52-4	DM 30.-	DM 40.-
3	J. Plank: State Events in Continuous Modelling and Simulation - Concepts, Implementation and New Methodology; 1997; ISBN 3-901608-53-2	DM 30.-	DM 40.-
4	P. Acel: Methode zur Durchführung betrieblicher Simulationen - Effiziente Optimierung der diskreten Simulation; 1997; ISBN 3-901608-54-0	DM 30.-	DM 40.-
5	M. Kinder: Stochastische Simulation biotechnischer Prozesse - Entwurf von Filtern und Reglern; 1997; ISBN 3-901608-55-9	DM 30.-	DM 40.-
6	M. Lingl: Hybrid Modelling Approaches in Continuous, Discrete and Combined Simulation; 1998. ISBN 2-901608-56-7; erscheint Ende 1998	DM 30.-	DM 40.-

ASIM - MITTEILUNGEN

ASIM-Fachgruppen berichten über ihre Treffen in den *ASIM Mitteilungen*. Diese werden von ASIM im Eigenverlag herausgegeben; die Hefte der eigenen Fachgruppe („Primär-Fachgruppe“) sind teilweise kostenlos für die Mitglieder der Fachgruppe, ansonsten gegen geringe Schutzgebühr erhältlich. In dieser Reihe erscheinen auch Tagungsunterlagen von Tagungen und Seminaren mit ASIM - Beteiligung, sowie die *ASIM Jahresberichte*.

Inhalte der *ASIM-Mitteilungen* sind:

- Berichte / Beiträge von Fachgruppentreffen (Arbeitskreistreffen)
- *ASIM Jahresberichte*
- Leitfäden und andere Publikationen der Fachgruppen (Arbeitskreise)
- Workshop - Berichte / Beiträge von mit ASIM zusammenarbeitenden Gruppen

Die *ASIM-Mitteilungen* sind allgemein gegen eine geringe Schutzgebühr erhältlich, für ASIM-Mitglieder zum verbilligten Preis. Bisher sind 60 *ASIM Mitteilungen* erschienen bzw. im Erscheinen. Im folgenden eine Übersicht über die letzten Ausgaben.

Bestellungen richten Sie bitte an Fr. Dr. Ingrid Bausch-Gall. Sie erhalten die gewünschten Publikationen kurzfristig zugesandt (mit Rechnung inklusive Portokostenanteil).

Ein Bestellformular finden Sie am Ende dieses Jahresberichtes, die Bestellung kann auch über den ASIM WWW - Server erfolgen:

<http://www.asim-gi.org/publikationen>

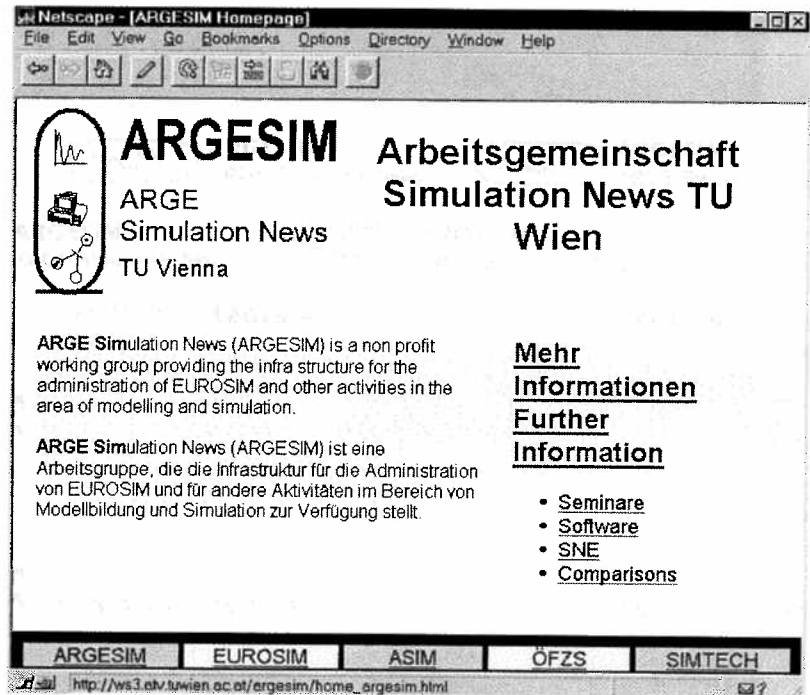
Heft Nr.	Titel	ASIM-Mitglieder	Andere
JB96	Jahresbericht 1996	Kostenlos	
55	Treffen der FG 4.5.8 "Simulation in der strategischen Verkehrsplanung", 14.6.1996, Zürich	DM 30.-	DM 40.-
56	Treffen der FG 4.5.2 „Simulationssoftware und -Hardware“, FG 4.5.3 „Simulation und Künstliche Intelligenz“ und FG 4.5.4 „Simulation in Biologie, Medizin und Ökologie“, 17.4 - 19.4.1997, Ebernburg	in Vorbereitung	
57	Treffen der FG 4.5.5 "Simulation Technischer Systeme" und "Simulationsmethoden und -Sprachen für verteilte Systeme und parallele Prozesse", 3-4. 3. 1997, Rostock	DM 30.-	DM 40.-
58	Leitfaden für Simulationsbenutzer in Produktion und Logistik (ehemals ASIM-Mitteilungen Heft Nr. 7a)		
JB97	Jahresbericht 1997 - ASIM Vorstellungsbroschüre	Kostenlos	
59	Treffen der FG 4.5.5 "Simulation Technischer Systeme", 2.3-3. 3. 1998, Heidelberg	in Planung	
60	Treffen der FG 4.5.2 „Simulationssoftware und -Hardware“, 4. 3. 1998, Magdeburg	in Planung	

ARGESIM

ARGE Simulation News (ARGESIM) ist eine gemeinnützige („non-profit“) Arbeitsgruppe, die Infrastruktur für die Verwaltung und Durchführung von EUROSIM- und ASIM - Aktivitäten und anderer Aktivitäten im Bereich Modellbildung und Simulation bereitstellt.

ARGESIM organisiert und stellt Infrastruktur zur Verfügung für:

- Herausgabe, Produktion, Druck, und Versand von *EUROSIM Simulation News Europe (SNE)*
- Mitgliederverwaltung von ASIM, Herausgabe des *ASIM Jahresberichtes* und Unterstützung bei der Herausgabe von *ASIM Mitteilungen* und der *ASIM Nachrichten*
- Herausgabe der *ARGESIM Reports*
- Definition, Durchführung, Verwaltung und Auswertung der EUROSIM Software Comparisons
- Veranstaltung der Seminarreihe „Seminare über Modellbildung und Simulation“ an der Technischen Univ. Wien (bisher 49 Seminare)
- Entwicklung von Simulationssoftware, z. B. mosis - kontinuierliche (parallele) Simulation
- Betreuung von MATLAB, ACSL und Computer Algebra Systemen an der Technischen Universität Wien
- Unterstützung der deutschsprachigen ACSL User Group
- Betrieb eines WWW-Servers für EUROSIM, ASIM, SNE, ARGESIM etc.:
 - <http://www.argesim.org> (ARGESIM)
 - <http://www.asim-gi.org> (ASIM)
 - <http://www.eurosim.org> (EUROSIM)
 - <http://simtech.tuwien.ac.at> (Abt. Simulationstechnik, Technische Univ. Wien)



ARGESIM Reports

ARGESIM organisierte den EUROSIM - Kongreß **EUROSIM'95** (September 1995, Wien) mit. Im Rahmen dieser Arbeit wurde ARGESIM auch Herausgeber und Verleger und begann die Reihe **ARGESIM Reports**.

Diese Berichte sollen kurze Monographien über neuen Entwicklungen in Modellbildung und Simulation, Lehrunterlagen für Kurse, Proceedings kleinerer Tagungen, Berichte über State-of-the-Art in der Simulationstechnik (z.B. Zusammenfassungen der EUROSIM Comparisons), etc., veröffentlichen.

Kurz nach Gründung dieser Reihe wurde der neue Verlag ARGESIM von ASIM als Verleger für die ASIM - Buchreihe *Fortschrittsberichte Simulation* ausgewählt. Alle Bände erhalten eine ISBN - Nummer; die Bände der Reihe *Fortschrittsberichte Simulation* wird im A5-Format gedruckt, die *ARGESIM Reports* weiterhin im Format A4.

ASIM hat mit dieser Zusammenarbeit eine kostengünstige Lösung mit ISBN Nummern für die Buchreihe *Fortschrittsberichte Simulation* gefunden, ARGESIM kann aufgrund des Übereinkommens die Reports über ASIM vertreiben.

Vertrieben werden beide Publikationen sowohl von ASIM (Büro Dr. I. Bausch-Gall, München, gemeinsam mit den *ASIM Mitteilungen*) und von der ARGESIM (TU Wien). Im Anhang ist ein Verzeichnis der lieferbaren ARGESIM Reports zu finden.

Rep. Nr.	<i>ARGESIM Reports</i>	ASIM-Mitglieder	Andere
1	Congress EUROSIM'95 - Late Paper Volume; F. Breiteneker, I. Husinsky (Hrsg.), ISBN 3-901608-01-X	DM 30.-	DM 40.-
2	Congress EUROSIM'95 - Session Software Products and Tools; F. Breiteneker, I. Husinsky (Hrsg.), ISBN 3-901608-02-8	DM 30.-	DM 40.-
3	EUROSIM'95 - Poster Book; F. Breiteneker, I. Husinsky (Hrsg.) ISBN-Nr.: 3-901608-03-6	DM 30.-	DM 40.-
4	Seminar Modellbildung und Simulation -Simulation in der Didaktik; F. Breiteneker, I. Husinsky, M. Salzmann (Hrsg.), ISBN 3-901608-04-4	DM 30.-	DM 40.-
5	Seminar Modellbildung und Simulation -COMETT - Course 'Fuzzy Logic'; D. Murray-Smith, D.P.F. Moeller, F. Breiteneker, ISBN 3-901608-05-2	DM 30.-	DM 40.-
6	Seminar Modellbildung und Simulation -COMETT - Course 'Object-Oriented Discrete Simulation; N. Kraus, F. Breiteneker, ISBN 3-901608-06-0	DM 30.-	DM 40.-
7	EUROSIM Comparison 1 - Solutions and Results; F. Breiteneker, I. Husinsky (Hrsg.), ISBN 3-901608-07-9	DM 30.-	DM 40.-
8	EUROSIM Comparison 2 - Solutions and Results; F. Breiteneker, I. Husinsky (Hrsg.), ISBN 3-901608-08-7	DM 30.-	DM 40.-
9	Bericht zum Treffen der ASIM -FGs „Simulationssoftware und -Hardware“, „Simulation und Künstliche Intelligenz“ und „Simulation in Biologie, Medizin und Ökologie“, Ebernburg, ISBN 3-901608-09-5	in Vorbereitung	
10	Proceedings TU-BioMed Symposium 1998 „Brain Modelling“, F. Rattay (Hrsg.) ISBN 3-901608-10-9	DM 15.-	DM 30.-
11	Proceedings 2 nd MATHMOD Conference, I. Troch, F. Breiteneker (Hrsg.), ISBN 3-901608-11-7	DM 90.-	DM 120.-
13	Posterband ASIM98 – 13. Symposium Simulationstechnik, Zürich, September 1998; M. Engeli, V. Hrdliczka (Hrsg.), ISBN 3-901608-13-3	DM 30.-	DM 40.-

Band Nr.	Autor(Hrsg.), Titel, ISBN	ASIM Mitglieder	Allg.
1	C. Westerkamp: Anwendung der Mehrgrößen-Parameterschätzung zur Simulation von linearen passiven Netzwerken; 1996; ISBN 3-901608-51-6	DM 30.-	DM 40.-
2	M. Salzmann: Genetische Algorithmen in diskreter Simulation; 1996; ISBN 3-901 608-52-4	DM 30.-	DM 40.-
3	J. Plank: State Events in Continuous Modelling and Simulation - Concepts, Implementation and New Methodology; 1997; ISBN 3-901608-53-2	DM 30.-	DM 40.-
4	P. Acel: Methode zur Durchführung betrieblicher Simulationen - Effiziente Optimierung der diskreten Simulation; 1997; ISBN 3-901608-54-0	DM 30.-	DM 40.-
5	M. Kinder: Stochastische Simulation biotechnischer Prozesse - Entwurf von Filtern und Reglern; 1997; ISBN 3-901608-55-9	DM 30.-	DM 40.-
6	M. Lingl: Hybrid Modelling Approaches in Continous, Discrete and Combined Simulation; 1998. ISBN 2-901608-56-7; erscheint Ende 1998	DM 30.-	DM 40.-

BESTELLUNG ASIM-PUBLIKATIONEN

ASIM Mitteilungen / ASIM Fortschrittsberichte Simulation / ARGESIM Reports

Hiermit bestelle ich:	Anzahl	Betrag
<i>ASIM Mitteilung</i> Nr. _____		_____
<i>ASIM Mitteilung</i> Nr. _____		_____
<i>ASIM Mitteilung</i> Nr. _____		_____
<i>FB Simulation</i> Nr. _____		_____
<i>ARGESIM Report</i> Nr. _____		_____
Zwischensumme:		_____
Verpackung und Porto:		_____
Rechnungsbetrag:		_____

Kosten für Versand und Porto:

	Deutschland	andere Länder
1-2 <i>Exemplare</i>	DM 5,00	DM 8,00
3-7 <i>Exemplare</i>	DM 10,00	DM 16,00
mehr <i>Exemplare</i>	DM 20,00	DM 25,00

Versand- und Rechnungsadresse (Die Rechnung wird mit der Lieferung versandt):

Name:	_____		
Institution/Firma:	_____		
Abteilung:	_____		
Straße/Postfach:	_____		
Land-Postleitzahl-Ort:	_____		
Telefon:	_____	Fax:	_____
Ort, Datum:	_____	Unterschrift	_____

Bitte senden Sie das Bestellformular an:

Dr. Ingrid Bausch-Gall
 Wohlfahrtstraße 21b
 D-80939 München
 Fax: +49-89 3231063

ASIM - AUFNAHMEANTRAG / DATENKORREKTUR

Datenkorrektur
Aufnahmeantrag

☐
☐

persönliche Mitgliedschaft
Firmenmitgliedschaft

☐
☐

Nachname

Vorname

Titel/Anrede

Adresse (Zustelladresse)

Adresse 1

Adresse 2

Straße

Ländercode

Postleitzahl

Ort

Telefon

Fax

Email

WWW

Zweitadresse

Adresse 1

Adresse 2

Straße

Ländercode

Postleitzahl

Ort

Telefon

Fax

Email

WWW

ASIM-spezifische Daten

Primär Fachgruppe Nr.:

FG 1 Verteilte Systeme u. parallele Prozesse

FG 2 Simulationssoftware und -Hardware

FG 3 Simulation und Künstliche Intelligenz

FG.4 Simulation in Medizin, Biologie und Ökologie

FG 5 Simulation technischer Systeme

FG 6 Simulation in Produktion und Logistik

FG 7 Simulation in der Betriebswirtschaft

FG 8 Simulation von Verkehrssystemen

FG 9 Simulation in Umweltsanwendungen

(max. 4 Fachgruppen)

Ich bin GI - Mitglied:

☐ ja

☐ nein

Meine Adresse darf weitergegeben werden für:

wissenschaftliche Zecke

Fachverlage / Veranstalter

Firmen

☐
☐
☐

Datum: _____

Unterschrift: _____

Bitte senden an:

ASIM - Mitgliederverwaltung

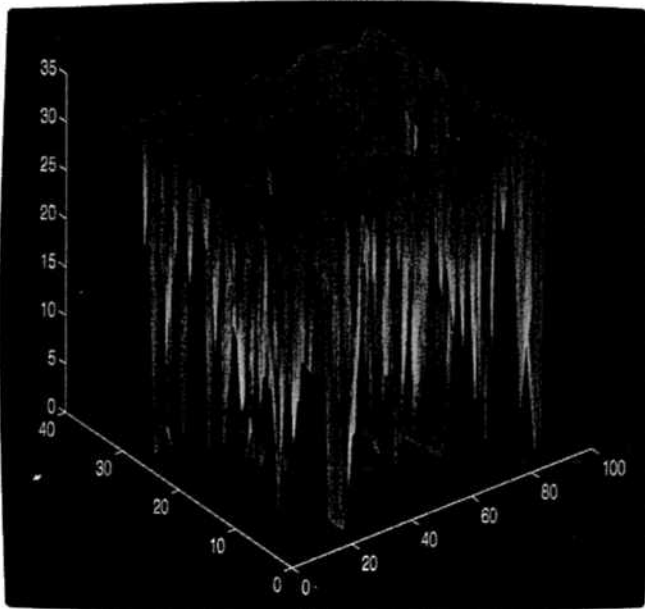
Prof. Dr. Felix Breitenacker

Abt. Simulationstechnik, Techn. Univ. Wien

Wiedner Hauptstraße 8-10, A-1040 Wien

Tel.: +43-1-58801-5374 oder -5386, Fax: +43-1-5056849

Email: admin@asim-gi.org



Dieser Oberflächenplot zeigt Stoßbeschädigungen einer Hubschrauber-Verbundwerkstoffstruktur. Zur Automatisierung zerstörungsfreien Prüfens klassifizierte die MATLAB Neural Net Toolbox Echos von Ultraschallsignalen. Die Daten wurden freundlicherweise von McDonnell Douglas unter einem AATD contract zur Verfügung gestellt.

DIE SPRACHE DER INGENIEURE

MATLAB—eine leistungsfähige, schnelle, interaktive Software—ist das beste Bindeglied zwischen Forschung und technischer Ausführung.

MATLAB ist eine Programmierungsumgebung für die Entwicklung von Algorithmen, Simulation und Analyse mit Visualisierung, numerischen Berechnungen und einer technischen Sprache.

MÄCHTIG FÜR UNTERSUCHUNGEN UND PROTOTYPING

In MATLAB werden Aufgabenstellungen und Lösungen so formuliert, wie es in der Mathematik üblich ist—ohne eine Zeile C oder FORTRAN Code zu schreiben.

Hunderte mächtiger Funktionen, die auf Effizienz und Zuverlässigkeit optimiert sind, sind mit einer leistungsfähigen und intuitiven Programmiersprache gekoppelt.

FACHWISSEN IN MATLAB VERFÜGBAR

Toolboxen bieten eine große Auswahl an optimierten Funktionen für Datenreduktion, Analyse, Modellierung und Systementwurf.

Mit den MATLAB-Toolboxen, die von anerkannten Fachleuten entwickelt werden, können erprobte, dem neuesten Wissensstand entsprechende mathematische Vorgehensweisen erlernt und auf eigene Aufgabenstellungen angewandt werden.

VISUALISIERUNG KOMBINIERT MIT MÄCHTIGEN ANALYSE-FUNKTIONEN

Leistungsfähige objektorientierte Grafik erlaubt interaktive Analyse und dynamische Modellbildung. Die umfangreichen Visualisierungsfunktionen umfassen 2-D, 3-D und 4-D Darstellung sowie Beleuchten von Oberflächen und Schattieren.

Hinter vielen
technologisch
fortschrittlichen
Entwicklungen steht die
Sprache der Ingenieure
und Wissenschaftler:

MATLAB



McDonnell Douglas verwendet MATLAB zur Entwicklung von automatischen zerstörungsfreien Prüfprozessen für Hubschrauber, wie z.B. den Longbow Apache

MATLAB

für ingenieur-
technische Aufgaben

MATLAB

MATLAB Compiler

MATLAB C Math Library

MATLAB C++ Math Library

Anwendungs-Toolboxen für:

Signalverarbeitung

Reglerentwurf

Financial Engineering

Bildverarbeitung

Datenanalyse & Modellbildung

ENTWICKLUNG VON MATLAB-PROGRAMMEN UND STANDALONE-ANWENDUNGEN

Umfangreiche GUI-Entwicklungswerkzeuge erlauben das individuelle Gestalten interaktiver MATLAB-Anwendungen.

Man kann MATLAB mit C und FORTRAN Programmen linken, Toolboxen einbeziehen, Daten mit anderer Software austauschen und MATLAB als ein Analyse- und Visualisierungs-Werkzeug einbauen.

Mit dem neuen MATLAB Compiler und der C Math Library lassen sich automatisch MATLAB-Algorithmen in standalone Programme umwandeln.



WEITERE INFORMATIONEN...

Nehmen Sie mit uns Kontakt auf, und fragen Sie nach kostenlosen, technischen Unterlagen zur MATLAB-Produktfamilie:

Tel.: 089/995 901 0

Fax: 089/995 901 11

<http://www.scientific.de>

The
**MATH
WORKS**
Inc.



scientific COMPUTERS



ASIM

From 1999 on together with SCS series
"Advances/Frontiers in Simulation"



ARGESIM REPORT



ARGESIM REPORT

ASIM - Buchreihen / ASIM Book Series

Reihe/Series Fortschritte in der Simulationstechnik

VIEWEG Verlag, Wiesbaden, Deutschland

erhältlich im Fachbuchhandel / available at book stores

kürzlich erschienen / appeared recently:

- R. Grützner (Hrsg.); Modellierung und Simulation im Umweltbereich
- A. Kuhn, S. Wenzel (Hrsg.); Simulationstechnik 11. Symp. Dortmund, 1997

Schwerpunkte / Topics:

- Statusberichte über Simulation in den ASIM Fachgruppen / Status reports
- Allgemeine Monographien / General monographies
- Proceedings der ASIM Tagungen / Proceedings of the ASIM conferences

Reihe/Series Fortschrittsberichte Simulation

ARGESIM, Wien, Österreich; erhältlich bei ASIM / available at ASIM

kürzlich erschienen / appeared recently:

- J. Plank; State Events in Continuous Modelling and Simulation - Concepts, Implementation and New Methodology
- P. Acel; Methode zur Durchführung betrieblicher Simulationen - Effiziente Optimierung der diskreten Simulation
- M. Kinder; Stochastische Simulation biotechnischer Prozesse - Entwurf von Filtern und Reglern
- M. Lingl; Hybrid Modelling Approach in Discrete, Continuous, and Combined Simulation (geplant für Ende 1998 / to appear at the end of the year)

Schwerpunkte / Topics:

- Spezielle Monographien (Dissertationen, ...) / Special monographies (PhD-thesis, ...)
- Erweiterte Berichte der ASIM Fachgruppentreffen / ASIM Workshop proceedings
- Handbücher für Simulationssprachen / User Guides for simulation languages

Preis / Price: DM 40.- (ASIM-Mitglieder DM 30.-) + Versandkosten

Andere Reihen / other Series

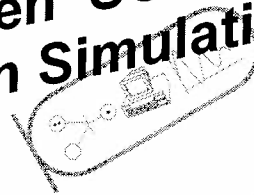
kürzlich erschienen / appeared recently:

- A. Kuhn, M. Rabe (Hrsg.); Simulation in Produktion und Logistik
- K. Mertens, M. Rabe (Hrsg.); Erfahrungen aus der Zukunft

Bestellungen, Informationen für Autoren / Orders:

Dr. Ingrid Bausch-Gall, Wohlfahrtstraße 21b, D-80939 München
Tel: +49-89-3232625; Fax: +49-89-3231063;
Email: 100564.302@compuserve.com

Ab 1999 zusammen mit den SCS-Reihen
"Advances/Frontiers in Simulation"



ARGESIM REPORT