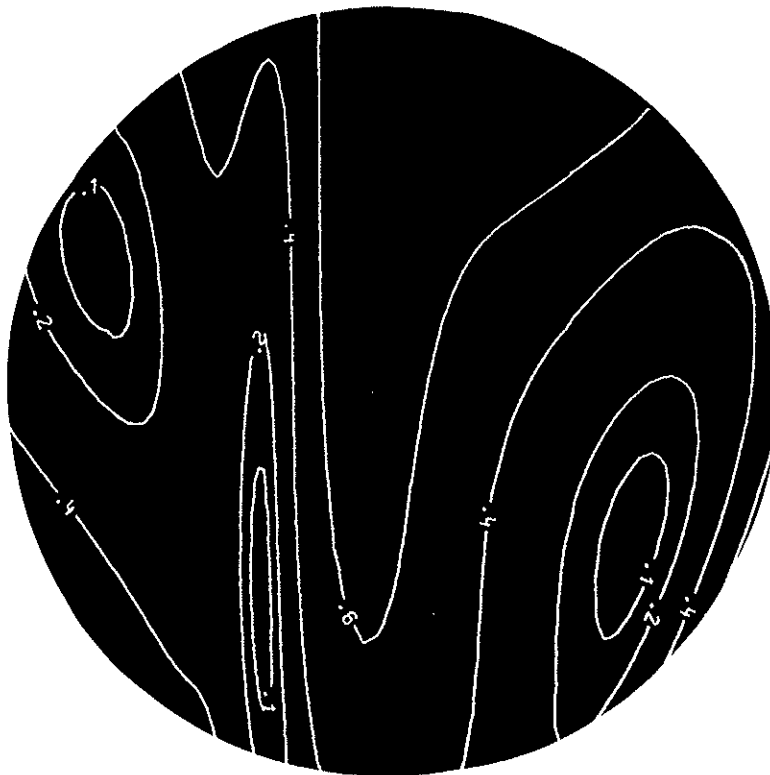

Interface

herausgegeben von der
Abt. Hybridrechenanlage des
EDV-Zentrums der
Technischen Universität Wien

Nummer 23
November 1986



Schichtenliniendarstellung mit HYBSYS

INHALTSVERZEICHNIS

	Seite
HYBSYS PTRAN - Experimentiersystem für den EAI SIMSTAR Multiprozessor	3
Praktische Parameteroptimierung in ACSL	9
Vokalsynthese	15
Simulation der Herzfrequenz unter Belastung	20
Ankergesteuerte Gleichstrommotore zur Bewegung von Maschinenteilen unter Reibungseinfluß	23
Waldsterben	28
Stimulation des Gleichgewichtssinnes mit einem Analogrechner	34
Benutzerversammlung	36
Neuigkeiten vom Betrieb am Hybridrechenzentrum	37
Simulationskurse	39

Redaktion: Irmgard Husinsky

Abt. Hybridrechenanlage des EDV-Zentrums der Technischen Universität Wien, A-1040 Wien, Gußhausstraße 27-29. Herausgeber, Verleger, Hersteller: EDV-Zentrum der Technischen Universität Wien, Abt. Hybridrechenanlage, Leitung: Dipl.Ing.Dr. W. Kleinert, A-1040 Wien, Gußhausstraße 27-29. Telefon: (0222) 588 01/3706 oder 3669. Ttx: (232) 3222467 = TUW, Modem: (0222) 65 98 17. Druck: Hochschülerschaft Technik, A-1040 Wien, Argentinierstr. 8.

HYBSYS PTRAN - EXPERIMENTIERSYSTEM FÜR DEN EAI SIMSTAR MULTIPROZESSOR*)

D. Solar
Hybridrechenanlage, TU Wien

Zusammenfassung

HYBSYS ist ein hybrides Simulationssystem, das für die Simulation kontinuierlicher Systeme mittels eines parallelen Multiprozessors entwickelt wurde. Dieser Artikel beschreibt die Version HYBSYS PTRAN, ein Experimentierwerkzeug, das in einer Kombination der Modellbeschreibungssprache PTRAN und der Experimentiersprache HYBSYS die Verwendung des neuen parallelen EAI SIMSTAR Multiprozessors ermöglicht. In dieser Kombination wird das SIMSTAR Laufzeitsystem (Run-Time Executive) durch das HYBSYS Experimentiersystem (Experimentation Executive) zusammen mit modellabhängigen digitalen Programmteilen und der SIMSTAR Programm-bibliothek für den Parallelprozessor ersetzt. So wird eine modellabhängige Version des im allgemeinen modellunabhängigen HYBSYS Experimentiersystems gebildet. Dieser Artikel gibt einen Überblick über die wichtigsten Eigenschaften dieses Simulationswerkzeugs und zeigt anhand einiger Beispiele, wie einfach die Verwendung des Systems ist, sowohl für einfache als auch für komplexe Experimente mit dem Modell.

Einführung

Seit 1975 verwenden die Benutzer der Hybridrechenanlage der Technischen Universität Wien HYBSYS für Simulationen. Momentan ist es die Version 5 TS, eine beinahe komplette Version der HYBSYS Experimentiersprache, die eine Art SIMSTAR Prototyp für kontinuierliche parallele Datenverarbeitung verwendet. Neben den Vorteilen der Parallelprozessortechnik (rasche Lösungen, keine Probleme mit Integrationsalgorithmen, Unstetigkeiten und Event Finding) sind hohe Interaktivität, einfache Anwendung komplexer Methoden inklusive interaktive Modelländerung, umfangreiche graphische Dokumentation, Bibliothek mathematischer Methoden und benutzerspezifische Methodenoverlays die Vorteile dieses Systems. Weiters ist es das einzige hybride Time-Sharing System, das bis zu 8 Benutzer haben kann, ausgenommen Echtzeitanwendungen. Die Nachteile sind, daß die Modelldeklaration blockstrukturiert anstelle von gleichungsorientiert ist, es keine Modellmakros und keine Experimentiermakros gibt und die Ergebnisse wegen des Prototyp-Charakters des Parallelprozessors (ein modifizierter EAI 680 Analogrechner) relativ ungenau sind. Schließlich führte der zunehmende Umfang der Simulationssoftware zu einer Situation, wo der Digitalteil (PACER 100) keine Erweiterungen mehr zuließ, insbesondere um die Version HYBSYS 6 zu implementieren, deren Konzept in /2/ veröffentlicht wurde.

Aufgrund der Installation des neuen EAI SIMSTAR Parallelprozessors führten folgende Argumente zur Implementierung von HYBSYS PTRAN:

- Das SIMSTAR Laufzeitsystem ist nicht so leistungsfähig wie das HYBSYS Laufzeitsystem (Experimentiermöglichkeiten).
- Kontinuität im Simulationswerkzeug (HYBSYS hat sich als leistungsfähiges und leicht erlernbares Simulationswerkzeug bewährt).
- Um das System so bald wie möglich zugänglich zu machen, wurde beschlossen, zuerst eine reduzierte Version von HYBSYS 6, nämlich HYBSYS PTRAN, zu implementieren. Diese Version verwendet die Simulationssprache PTRAN zur Modelldeklaration und Modellvorbereitung (Hardwarezuordnung).
- Da PTRAN sich von ACSL herleitet, steht nun ein Simulationsstandard für die Modelldeklaration zur Verfügung.

* Bearbeitete Übersetzung von D. Solar: "HYBSYS PTRAN - An Experimentation Tool for the EAI SIMSTAR Parallel Multiprocessor", 2nd European Simulation Congress, Antwerpen, September 1986

Der parallele Simulationsprozessor SIMSTAR

SIMSTAR ist ein neuartiger, vollständig automatisierter Simulationsrechner. Er ermöglicht eine genaue Modellierung des Verhaltens komplexer Prozesse in Echtzeit, schneller oder langsamer als Echtzeit, einschließlich aller nichtlinearen, un stetigen und stochastischen Eigenschaften.

Der Parallel Simulation Processor (PSP) besteht aus einer Parallel Mathematical Unit (PMU), einer Parallel Logic Unit (PLU) und einem Local Control Processor (LCP), wie in Abbildung 1 dargestellt.

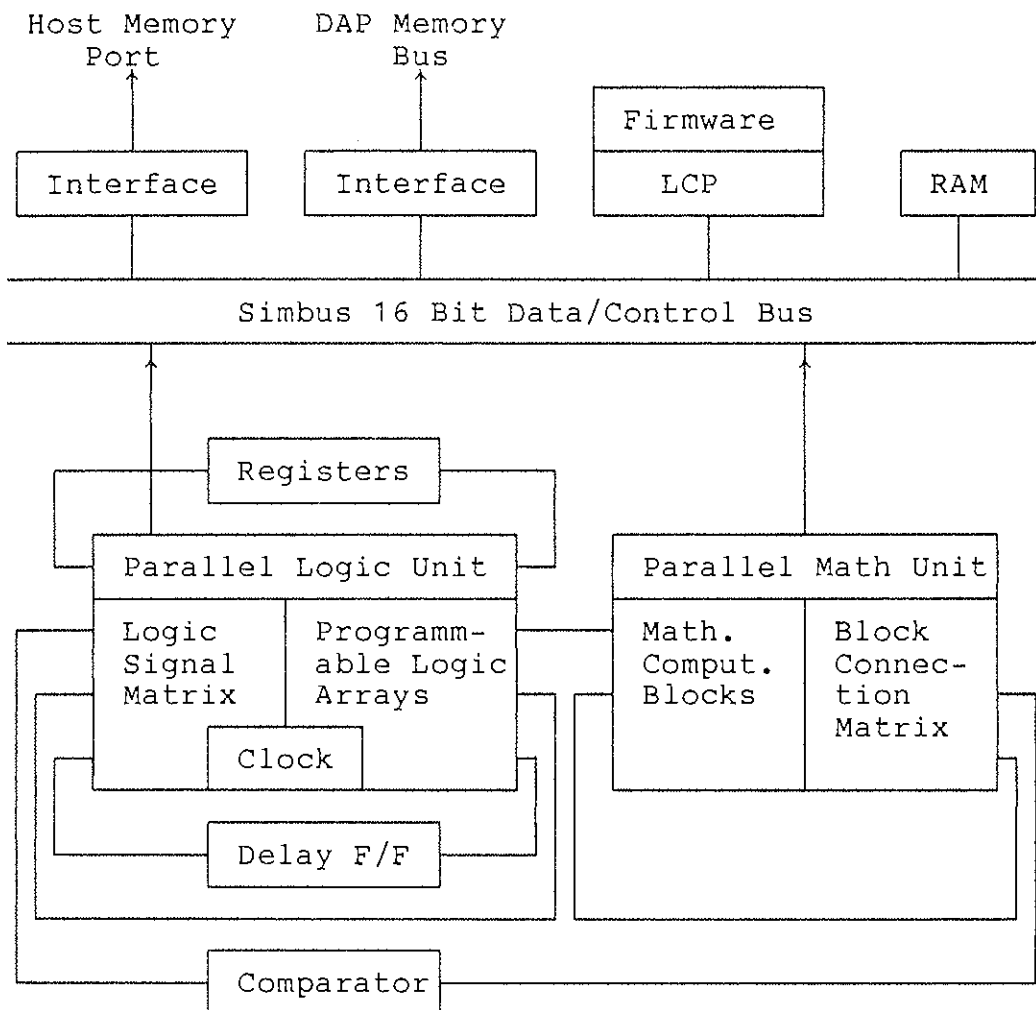


Abbildung 1: Funktionsdiagramm des PSP

Die PMU führt parallel alle arithmetischen Operationen, wie Integration, Summation, Multiplikation, Funktionsgenerierung etc. durch. Jede Variable des Modells wird durch genau einen Rechenblock repräsentiert. Die mathematischen Rechenblöcke können mit Hilfe einer dreistufigen 320x512 Verbindungsmatrix beliebig miteinander verbunden werden. Die Rechenblöcke sind in der Nähe der Verbindungsmatrix angeordnet, um die Länge der Kabel zu minimieren.

Die PLU berechnet jene Teile des Modells, die sequentielle und kombinatorische logische Operationen erfordern, einschließlich Kontrolle der PMU wie Integrierersteuerung, Switch-Steuerung oder Event-Erkennung. Basierend auf einem 1-Bit-Prozessor Design mit Time-Slicing kann die PLU innerhalb einer Mikrosekunde bis zu 320 Ausgänge von 120 Eingängen berechnen. Komponenten dieses Prozesses sind binäre Funktionsgeneratoren, Zähler, Delay Flip Flops etc..

Der LCP ist ein Single Board Computer mit einem 68000 μ P. Er dient für Setup, Kontrolle und Wartung des PSP.

Der PSP ist mit einem Digital Arithmetic Processor (DAP) verbunden, der für die Problemvorbereitung und für Echtzeitberechnungen parallel zum PSP verwendet wird.

Die Modelldeklarationssprache PTRAN

PTRAN ist Teil der Simulationssprache STARTRAN für kontinuierliche Systeme, die sich von ACSL herleitet und die meisten Eigenschaften von ACSL besitzt. Der STARTRAN Compiler trennt das System in seine prozeduralen (DTRAN-Ebene) und seine kontinuierlichen Teile (PTRAN-Ebene). Die PTRAN-Ebene enthält die meisten Eigenschaften von ACSL zur Modellierung kontinuierlicher Systeme, einschließlich Modellmakros, aber ohne alle Prozeduralteile des Modells. Es sind einige zusätzliche Sprachelemente zur Definition spezieller Sektionen (z.B. PARALLEL-Sektion) und zur Angabe von Skalierungsinformationen enthalten, die für das Setup des PSP benötigt werden. ACSL behandelt diese Befehle wie Kommentare. Dadurch kann man mit denselben Programmen digitale wie auch hybride Lösungen zur Verifizierung erhalten.

Der PTRAN Compiler generiert zwei Programme, eine Routine zur Berechnung der PSP-Koeffizienten und eine Gruppe von Anweisungen, die die notwendigen Verbindungen der parallelen mathematischen Rechenblöcke beschreiben. Später wird die Koeffizientenberechnungsroutine zum HYBSYS Laufzeitsystem dazugebunden, um das HYBSYS Experimentierprogramm zu generieren. Die Verbindungsanweisungen werden dazu verwendet, die erforderlichen Rechenblöcke unter Berücksichtigung der aktuellen Hardwarekonfiguration zuzuordnen. Mit Ausnahme des HYBSYS Experimentierprogrammes werden alle diese Aufgaben vom SIMSTAR Program Generation System durchgeführt.

Die Experimentiersprache HYBSYS

HYBSYS ist ein interaktives hybrides Simulationssystem für die Simulation kontinuierlicher Systeme, basierend auf dem Prinzip der Trennung von Modell, Methode und Experiment. Ein Experiment ist definiert als die Anwendung einer bestimmten Methode auf ein bestimmtes Modell.

Das Hauptziel während der Entwicklung von HYBSYS war, ein komfortables Simulationswerkzeug zu schaffen, das Standardexperimente durchführen kann und den Benutzer dabei von methodologischen Problemen und Problemen der Programmierung möglichst fernhält. Es wurden Standardsimulationen definiert, die innerhalb der Standardumgebung des Simulationssystems gelöst werden können. Diese Standardumgebung schließt Modelldeklaration, Änderungen der Modellstruktur, Speichern und Laden des Modellzustands, numerische und graphische Dokumentation der Simulationsergebnisse, Invokation allgemeiner Methoden und Dokumentation der Modellstruktur mit ein.

Weiters wurde eine umfangreiche Unterstützung von Simulationen mit speziellen Anforderungen verlangt, und zwar durch die Möglichkeit der Umgebungserweiterung um spezifische Methoden und Interaktionen.

Durch Definition verschiedener Typen von Methoden und durch ein Overlay-Konzept wurde es möglich, das System um Standardmethoden als auch Benutzermethoden beliebig zu erweitern. Das wird einfach durch Zusammenbinden der neuen Moduln mit dem existierenden System realisiert. Die Befehle für neue Standardmethoden werden während der Systemgenerierung definiert, die Befehle für benutzerprogrammierte Methoden müssen vom Benutzer bei jeder Simulationssitzung definiert werden.

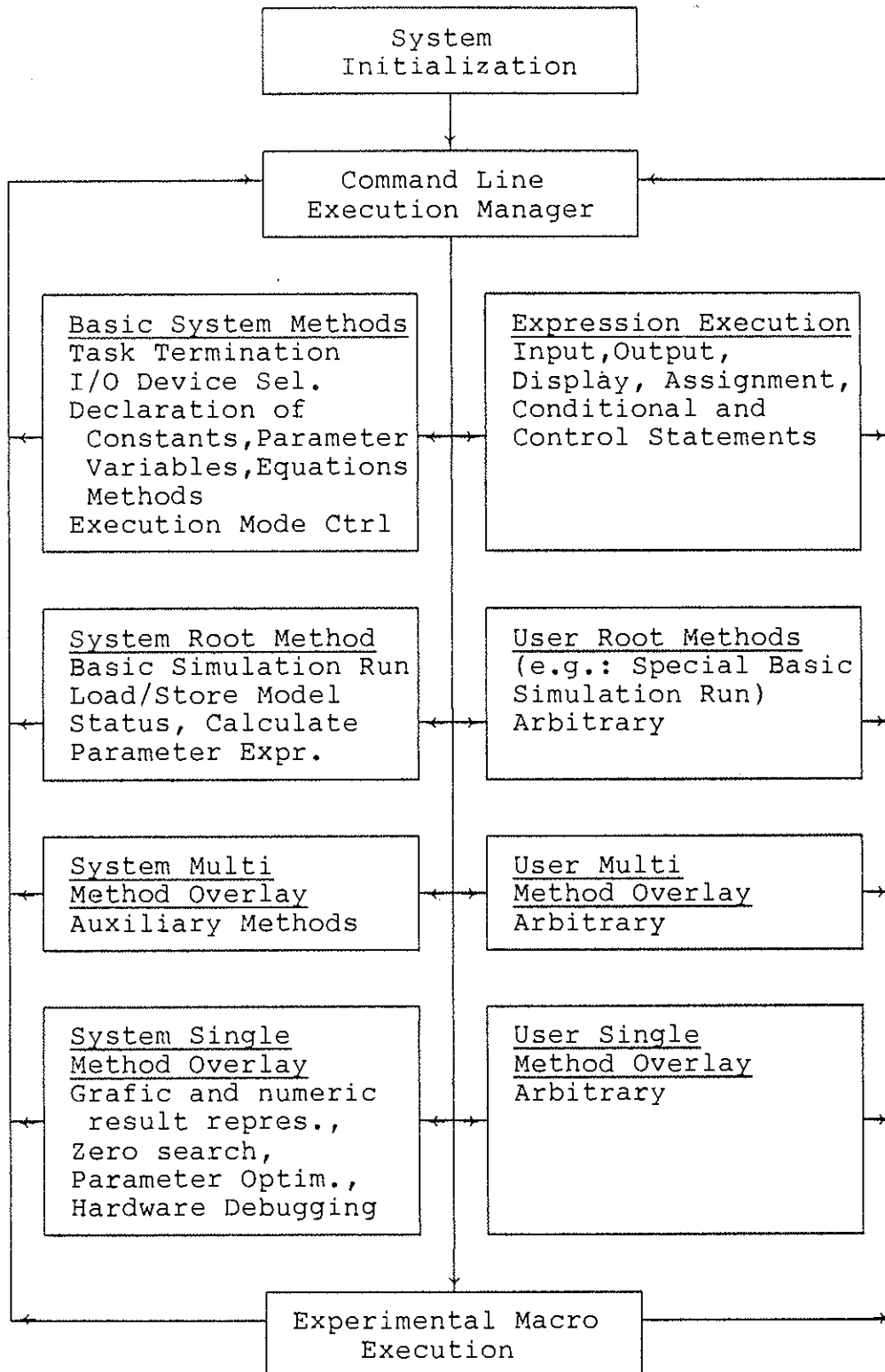


Abbildung 2:
Funktionsdiagramm des HYBSYS Experimentiersystems

Im allgemeinen umfaßt der Umfang der Experimentiersprache HYBSYS die Anwendung von

- Methoden, die die explizite Kenntnis der Modellstruktur voraussetzen. Dies sind Modelldeklaration, Änderung der Modellstruktur, Dokumentation der Modellstruktur, Berechnung von Parameterausdrücken, digitale Integrationsmethoden, Eigenwertberechnung und Fehleranalyse.

- Methoden, die keine explizite Kenntnis der Modellstruktur voraussetzen. Dies sind quantitative Modelländerungen (Änderungen von Parameterwerten und Funktionstabellen), graphische und numerische Lösungsdarstellung, optimale Skalierung, Optimierungsmethoden, Nullstellensuche und die Modelldeklaration selbst.
- Kontrollmechanismen für beliebige komplexe Kombinationen von Methoden. Dies sind Parameterschleifen, Konditionalbefehle, Ein/Ausgabebefehle und experimentelle Makros.

Die Experimentiersprache HYBSYS ist befehlsorientiert und enthält folgende Befehlsgruppen:

- elementare Befehle für die Zuweisung, numerische Ein/Ausgabe von Variablen und graphische Ausgabe von Variablen über der Zeit, über einer anderen Variablen oder einem Parameterintervall.

A,X=.75;	Zuweisung zur Modifikation eines oder mehrerer Parameterwerte oder Skalierungsfaktoren
A,X:	Numerische Darstellung eines oder mehrerer Parameterwerte oder Skalierungsfaktoren
X;	Skalierte graphische Darstellung der Variablen X nach einem impliziten Simulationslauf
DX(X);	Graphische Darstellung des Phasenbildes DX über X
X(A=1,2,.1);	Graphische Darstellung des Endwertes X(t=TEND) über einem Parameterintervall von A
A,B←	Programmierte Eingabe von Parameterwerten während der Experimentierphase.

- allgemeine Befehle zur Aktivierung beliebiger Methoden wie Parameteroptimierung, Nullstellensuche etc., unabhängig vom Typ der Methode.

PLOT	graphische Darstellung von Ergebnissen (unskaliert)
PRINT	numerische Darstellung von Ergebnissen
ZERO	Nullstellensuche
OPTIM	Parameteroptimierung
SCALE	automatische On-line-Skalierung
SIM	Hardware Debugging
GET,STORE	Laden und Speichern des Modellzustands
PAR	Deklaration oder Liste von Parametern
VAR	Deklaration oder Liste von Variablen
MTD	Deklaration von Methoden

- Kontrollbefehle für begrenzte oder unbegrenzte Schleifen und Konditionalbefehle.

A=1,2.5,.1! Parameterschleife mit Wertinkrement und Exekution des darauffolgenden Befehlsblocks

A={0,2,5}! Parameterschleife mit Indexinkrement (Werte aus Wertetabelle)

<command block>! unbegrenzte wiederholte Exekution des Befehlsblocks

A<B? Konditionalbefehl, der den darauffolgenden Befehl exekutiert, falls die Bedingung erfüllt ist

A>B?<command block>! Dieser Befehlsblock entspricht
WHILE A>B DO <command block>

- Befehlsblöcke zur Zusammenfassung von mehreren Befehlen bei komplexer Verwendung von Kontrollbefehlen.

A=0,2,.25!{ZERO EX,EDX BY XO,DXO;PLOT,S DX(X);}

- Befehlsmakros (Experimentiermakros) zur einfachen Exekution komplexer Befehlskombinationen.

MACRO name; Deklaration
command lines
END

name; Exekution

name: Auflisten des Makros

HYBSYS PTRAN

HYBSYS PTRAN ist eine reduzierte Version der Experimentiersprache HYBSYS, ohne diejenigen Methoden, die explizite Kenntnis der Modellstruktur erfordern. Der Grund für diese Reduktion liegt darin, daß die explizite Modellstruktur nicht bekannt ist.

Das Modell wird mit dem PTRAN Compiler übersetzt, es wird ein FORTRAN-Programm generiert, das eine Routine zur Koeffizientenberechnung und ein BLOCK DATA Programm zur Definition der Modelldatenbasis durch COMMON-Blöcke enthält. Statt seine eigene Datenbasis zu verwenden, ist HYBSYS PTRAN nun mit der Datenbasis, die von PTRAN generiert wurde, über ein spezielles Datenbasisinterface verbunden und erzeugt so ein HYBSYS PTRAN-Programm für das aktuelle Modell.

Obwohl Änderungen in der Struktur eines Modells eine zeitaufwendige Neukompilierung und Neukatalogisierung erfordern, hat sich HYBSYS sehr gut für interaktives Experimentieren auf dem parallelen EAI SIMSTAR Multiprozessor bewährt.

Literatur

1. Embley R.W.: "The Technology behind SIMSTAR, an all-new Simulation Multiprocessor". Informatik Fachberichte 85, Springer (1984) 317-327.
2. Solar D.: "Konzepte für die Beschreibung von Modellen und Experimenten im hybriden Simulationssystem HYBSYS 6". Informatik-Fachberichte 85, Springer (1984) 475-479.
3. Solar D.: "HYBSYS PTRAN User Manual". Hybridrechenanlage, EDV-Zentrum, Technische Universität Wien (1986)

PRAKTISCHE PARAMETEROPTIMIERUNG IN ACSL^{*)}

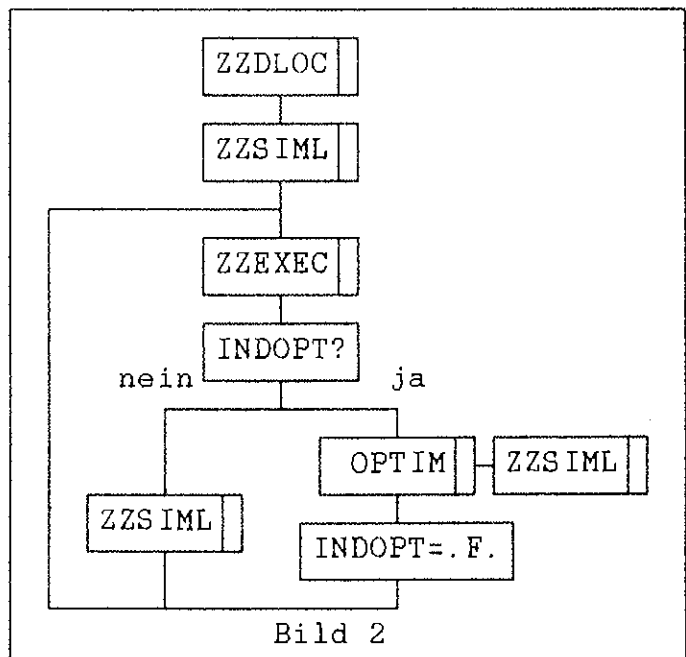
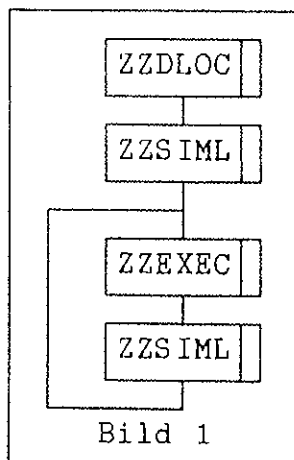
M. Gräff, F. Breitenecker, I. Troch
Institut für Technische Mathematik, TU Wien

Implementierung

Bei der Untersuchung von dynamischen Systemen durch Rechnersimulation stellt sich oft die Frage nach optimalen Werten für Modellparameter, die vorgegebene Gütemaße unter Einhaltung von Beschränkungen (an die Parameter und den Zustand des Systems) minimieren.

Gründe, die für den Einbau von Optimierungsmöglichkeiten in bereits bestehende Simulationssprachen sprechen, und Wege, derartige erweiterte Simulationsprogramme zu schaffen, sind beispielsweise in /1/, /2/ und /3/ zu finden. Dieser Beitrag beschäftigt sich mit Optimierungsumgebungen für die digitale Simulationssprache ACSL, wo das Optimierungsprogramm durch einen Eingriff in das vom ACSL-Precompiler erzeugte Hauptprogramm des FORTRAN-Zwischencodes eingebaut wird.

Zum Verständnis dieser Änderung wird zuerst kurz der Aufbau eines normalen ACSL-Simulationsprogramms skizziert: Das Hauptprogramm, siehe Bild 1, besteht aus Initialisierungen (ZZDLOC, erster Aufruf von ZZSIML), die nur beim Start des Programms ausgeführt werden; dann folgt eine Schleife, die aus dem Runtime-Interpreter ZZEXEC und dem einen Simulationslauf ausführenden Unterprogramm ZZSIML besteht. Der Runtime-Interpreter ist der Programmteil, der mit dem Benutzer kommuniziert, d.h. die Kommandos verarbeitet; er wird nur bei einem START-Befehl verlassen.



*) Unterstützt durch das Forschungsprojekt S3207 "Praktische Berechnung optimaler Steuerungen und Regelungen" im Rahmen des Schwerpunkts "Methodenbank Angewandte Mathematik" des Fonds zur Förderung der wissenschaftlichen Forschung.

Bild 2 zeigt den Aufbau eines erweiterten Hauptprogramms, bei dem abhängig vom Wert des Optimierungsflags INDOPT ein START-Befehl entweder einen Simulationslauf oder einen Optimierungslauf, d.h. eine Iteration mit i.a. mehreren Simulationsläufen, auslöst. Das Löschen des Optimierungsflags nach jedem Optimierungslauf verhindert ein irrtümliches Starten eines Iterationslaufs.

Dieser Weg diente bei der Realisierung nur als Grobgerüst, da das praktische Arbeiten mit derartigen Optimierungsumgebungen einige potentielle Fehlerquellen aufzeigte, die mit geringem Aufwand ausgeschaltet werden können. Die zuletzt realisierten Optimierungsumgebungen mit den Programmen ZXSSQ aus der IMSL für nichtlinearen Ausgleich und Gleichungssysteme und EO4VCF aus der NAGLIB für allgemeine nichtlineare Programme weisen die folgenden Eigenschaften auf:

- * keine Änderungen im ACSL-Modell für die Optimierung notwendig;
- * Trennung des Benutzers von der FORTRAN-Ebene durch automatische Generierung des erweiterten Hauptprogramms (Preprozessor);
- * einfache Anwendung und Bedienung;
- * Nebeneffekte durch Namensgleichheit von Modellvariablen und Optimierungsvariablen weitgehend ausgeschaltet.

Das ACSL-Modell kann unverändert übernommen werden, weil die Optimierungsaufgabe getrennt in der Optimierungsdirektiven-Datei formuliert wird. Für diese Aufteilung ist das später dargestellte Datenkonzept wesentlich, das erlaubt, die Parameter des Optimierungsprogramms als "Nicht-Modellvariable" zu erklären.

Ein Preprozessor erzeugt aus dem ACSL-Modell und der Optimierungsdirektiven-Datei die Eingabe für den ACSL-Precompiler; der Benutzer ist auf diese Weise von Änderungen in den FORTRAN-Programmen befreit und er muß sich nicht mit dem genauen Aufbau dieser Programme beschäftigen. Der ACSL-kundige Benutzer ist bei der Verwendung einer solchen Optimierungsumgebung nicht gezwungen, viel Neues zu lernen. Die Variablen des Optimierungsprogramms sind für ihn mit ACSL-Systemvariablen gleichwertig.

Die größte Schwierigkeit bei der Konzeption der Optimierungsumgebungen war, eine Möglichkeit für einen sauberen Datentransfer zwischen ACSL-Modell und Optimierungsprogramm zu finden. Das Hauptproblem ist, daß ACSL alle Modellvariablen in einem gemeinsamen Commonblock ZZCOM verwaltet. Dieser Commonblock kann zwar ohne Schwierigkeiten durch den ACSL-Precompiler in einen beliebigen FORTRAN-Modul kopiert werden ('\$'-Zeichen in Spalte 1), es gibt aber nur die Entscheidung, entweder alle oder keine Modellvariablen in einem Modul ansprechen zu können. Das bedeutet aber, daß bei Namensgleichheit von FORTRAN-Modul- und Modellvariablen unerwartete Nebeneffekte auftreten können. Für die Lösung dieses Problems ist es wesentlich, daß der Datenfluß bei der Optimierung auf drei Vorgänge beschränkt ist: Lesen der Parameter (aus der 'Modelldatenbasis' ZZCOM), Schreiben der Parameter und Lesen der Ergebniswerte wie z.B. Gütemaß. Außerdem ist ein interaktiver Zugriff auf einige Parameter des Optimierungsprogramms, u.a. Fehlerindikatoren, Toleranzparameter etc., für das Arbeiten mit den Optimierungsprogrammen notwendig.

Bild 3 zeigt den Datenfluß in einer einfachen Ausführung der Optimierungsumgebung. Das Hauptprogramm übergibt beim Unterprogrammaufruf Daten an das Optimierungsprogramm; die Parameter mit Zugriff des Benutzers müssen im Modellteil deklariert werden, weil der Benutzer sie nur über den Commonblock ZZCOM ansprechen kann. Das Optimierungsprogramm ruft wiederholt ein Unterprogramm (FUNC) auf; bei jedem Aufruf soll es zu einem gegebenen Parametersatz eine Funktionsauswertung (Gütefunktion, oder Gütemaß und Gradient, ...) durchführen. FUNC schreibt dazu den Parametersatz auf ZZCOM, führt einen Simulationslauf aus

(oder mehrere Simulationsläufe im Falle von Differenzenquotientenapproximation des Gradienten) und liest die Ergebniswerte vom Commonblock.

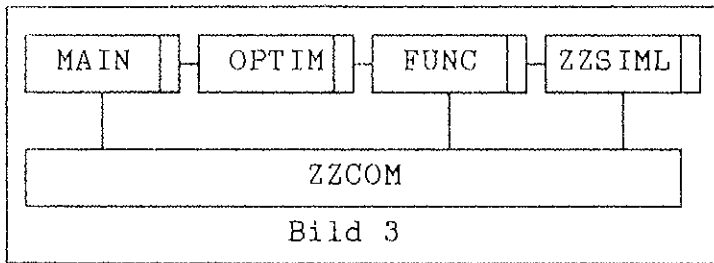


Bild 3

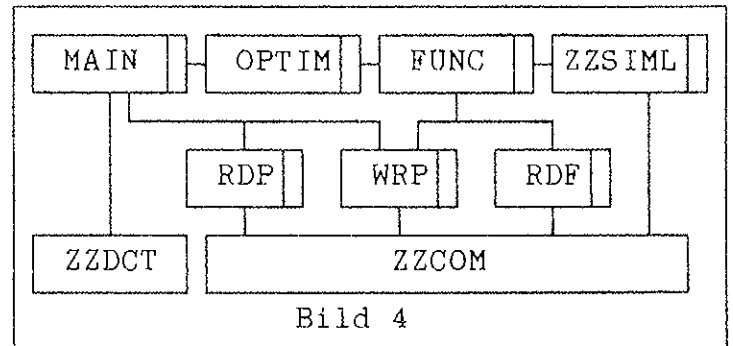


Bild 4

Die modifizierte Version (Bild 4) zeigt den gleichen Aufbau, doch werden die Zugriffe auf die Modelldatenbasis ZZCOM durch eigene Datentransferprozeduren durchgeführt. In den Modulen MAIN und FUNC sind daher die Modellvariablen nicht mehr direkt verfügbar, wodurch die Kollisionsgefahr von Variablenamen und die Datenunsicherheit erheblich vergrößert werden. Die Parameter des Optimierungsprogramms werden jetzt nicht mehr als Modellvariable deklariert, sondern als Variable im Hauptprogramm; der Zugriff erfolgt nur mehr interaktiv über den Runtime-Interpreter, wobei die Informationen über diese ansprechbaren Variablen in einem ACSL-Commonblock ZZDCT (Verzeichnis aller Variablenamen) angelegt werden.

Die Technik, die Optimierungsvariablen bzw. ihre Namen für den Runtime-Interpreter ansprechbar zu machen, ist in /4/, App. B, beschrieben. Ein Aufruf der Prozedur BLDDCT macht eine angegebene Variable unter einem wählbaren Namen so ansprechbar, als wäre der Name eine ACSL-System- oder Modellvariable; gleichzeitig wird überprüft, ob eine gleichbenannte Modellvariable existiert (Fehlermeldung).

Wir unterscheiden bei den Variablen bzw. Namen vier Klassen:

- * Modellvariable,
- * ansprechbare Optimierungsvariable,
- * verborgene Optimierungsvariable,
- * formale Optimierungsvariable.

Modellvariable sind einfach die im Modell vereinbarten Größen. Durch den Preprozessor werden die im Hauptprogramm und den Unterprogrammen verwendeten Variablen auf die drei anderen Klassen aufgeteilt. Die ansprechbaren Optimierungsvariablen sind jene aktuellen Parameter des Optimierungsprogramms, auf die der Benutzer interaktiven Zugriff hat. Verborgene Variable sind jene Parameter, auf die er keinen Zugriff hat, aber auch die Variablen in jenen Unterprogrammen, wo die Modellvariablen nicht ansprechbar sind (z.B. FUNC in Bild 4). Die formalen Variablen haben nur die Aufgabe, Namensgleichheiten in den gekapselten Datentransferprozeduren zu verhindern; sie sperren nur ihre Namen für Modellvariable, obwohl unter den Namen keine sinnvoll ansprechbaren Größen stehen (i.a. nur die Namen der formalen Parameter der Datentransferprozeduren, siehe RDP, WRP, RDF in Bild 4).

Verwendung des Systems

Für den Benutzer einer solchen Optimierungsumgebung steht natürlich die Frage im Vordergrund, welche Form die Benutzerschnittstelle aufweist. An Dokumentation wird ihm eine allgemeine Beschreibung des Aufbaus von ACSL-Umgebungen und des Arbeitens damit, für jedes Optimierungsprogramm ein spezifisches Dokument (Liste der ansprechbaren und der formalen Variablen, Standard-Initialisierungen, Beispiel) sowie die Dokumentation des Optimierungsprogramms unterstützen. Im folgenden werden die Arbeitsschritte kurz beschrieben.

Aufstellen, Validieren und Testen des Modells: Unter Testen des Modells ist dabei eine kleine Untersuchung der numerischen Eigenschaften der Gütefunktion und - falls vorhanden - der nichtlinearen Beschränkungsfunktionen zu verstehen. Falls ein Verfahren mit Differenzenquotientenapproximation der Ableitungen von Gütefunktion und Beschränkungen gewählt wurde, ist darauf zu achten, daß nach der Berechnung der Differenzenquotienten nicht bloß Integrationsfehler stehen bleiben!

Erstellen der Datei mit den Optimierungsanweisungen: Formulierung der Optimierungsaufgabe, Festlegung der Parameter, des Gütemaßes, der Beschränkungen, Möglichkeit zur Initialisierung der Parameter des Optimierungsprogramms. Dann erfolgt der Aufruf des optimierungsprogrammspezifischen Preprozessors, der aus den beiden obigen Dateien die Eingabedatei für den ACSL-Precompiler erzeugt.

Aufruf von ACSL, wobei das Optimierungsprogramm verfügbar gemacht werden muß, d.h. es muß die entsprechende Programmbibliothek geladen werden.

Interaktives Arbeiten: Durch Setzen des Optimierungsflags INDOPT wird durch den START-Befehl anstatt eines Simulationslaufs ein Optimierungslauf durchgeführt. Es ist bei Verwendung der schrittweitengesteuerten Integrationsverfahren (IALG=1 und IALG=2) wichtig, daß die Meldungen der Integratoren durch die Anweisung SET WESITG=.FALSE. ausgeschaltet werden (die Meldungen des State-Event-Finders können durch SET WEDITG=.FALSE. abgeschaltet werden), da ansonsten sehr große Datenmengen auf den Bildschirm oder auf Papier kommen und beispielsweise bei Verwendung des NAG-Programms EO4VCF das Protokoll des Optimierungsprogramms durch diese Meldungen unterbrochen wird.

Beispiel

Das folgende Beispiel soll die aufgezählten Arbeitsschritte veranschaulichen. Die Aufgabenstellung kommt aus der Regelungstechnik, es ist eine zeitoptimale Steuerung für einen Erzentlader (siehe /5/) zu berechnen. Das Problem, die Laufkatze von einer Position A in die Position E zu bringen, wobei das System in A und E jeweils in Ruhe sein soll, führt bei Parametrisierung der Steuerung durch eine stückweise konstante Funktion mit 20 Intervallen auf ein nicht-lineares Optimierungsproblem mit 21 Parametern (Werte der Steuerung auf den Intervallen, Endzeit), festen Grenzen für diese Parameter (Beschränkung der Steuerung, Nichtnegativität der Endzeit) und vier nichtlinearen Beschränkungen (Endbedingungen an den Zustand).

Definition der Optimierungsaufgabe:

Parameter:

```
-1.0 <= UTAB( 1) <= 1.0 $ -1.0 <= UTAB( 2) <= 1.0 $ -1.0 <= UTAB( 3) <= 1.0
-1.0 <= UTAB( 4) <= 1.0 $ -1.0 <= UTAB( 5) <= 1.0 $ -1.0 <= UTAB( 6) <= 1.0
-1.0 <= UTAB( 7) <= 1.0 $ -1.0 <= UTAB( 8) <= 1.0 $ -1.0 <= UTAB( 9) <= 1.0
-1.0 <= UTAB(10) <= 1.0 $ -1.0 <= UTAB(11) <= 1.0 $ -1.0 <= UTAB(12) <= 1.0
-1.0 <= UTAB(13) <= 1.0 $ -1.0 <= UTAB(14) <= 1.0 $ -1.0 <= UTAB(15) <= 1.0
-1.0 <= UTAB(16) <= 1.0 $ -1.0 <= UTAB(17) <= 1.0 $ -1.0 <= UTAB(18) <= 1.0
-1.0 <= UTAB(19) <= 1.0 $ -1.0 <= UTAB(20) <= 1.0 $ 0.0 <= TEND <= 10.0
```

Gütefunktion:

TEND

Gradient Gütefunktion:

0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0
0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 0.0 \$ 1.0

Nichtlineare Gleichheitsbeschränkungen:

X1 - X1E = 0.0 \$ X2 - X2E = 0.0
X3 - X3E = 0.0 \$ X4 - X4E = 0.0

ACSL-Modell:

PROGRAM ERZENTLADER

"=====

INITIAL

"-----KONSTANTEN"

CONSTANT CMK = 20.0, CMG = 10.0, CH = 3.0, DG = 9.806, CUM = 0.5

CBETA = (CMK + CMG)/CMK

COMO2 = CG/CH

CA2 = CBETA*COMO2

CC = COMO2*CMG/CMK

CBETAQ = CBETA*CUM

"-----ANFANGS- UND ENDWERTE DER ZUSTANDSVARIABLEN"

CONSTANT X10 = 0.0, X20 = 0.0, X30 = 0.0, X40 = 0.0

CONSTANT X1E = 0.0, X2E = 0.0, X3E = 1.706, X4E = 0.0

"-----ENDZEIT"

CONSTANT TEND = 5.0

"-----STEUERUNG"

INTEGER IC

ARRAY UTAB(20)

CONSTANT UTAB = 20*0.0

CINT = TEND/20.0

IC = 0

U = UTAB(1)

END \$ "OF INITIAL SECTION"

DYNAMIC

DERIVATIVE

"-----FESTSETZEN DER SCHRITTWEITE UNABHÄNGIG VON CINT"

MAXTERVAL MAXT = 0.025

MINTERVAL MINT = 0.025

X1 = INTEG(X2, X10)

X2 = INTEG(-CA2*X1 + CBETAQ*U, X20)

X3 = INTEG(X4, X30)

X4 = INTEG(-CC*X1 + CBETAQ*U, X40)

END \$ "OF DERIVATIVE SECTION"

"-----BERECHNUNG DER SCHRITTWEITEN KONSTANTEN STEUERUNG, DER AUFRUF"

" VON LOG IST NUR FÜR SCHÖNE ZEICHNUNGEN NOTWENDIG"

CALL LOG

IF (IC .LT. 20) IC = IC + 1

U = UTAB(IC)

"-----ENDBEDINGUNG"

TERMT(T .GE. TEND - 1.E-4)

END \$ "OF DYNAMIC SECTION"

END \$ "OF PROGRAM ERZENTLADER"

Runtime-Kommandos:

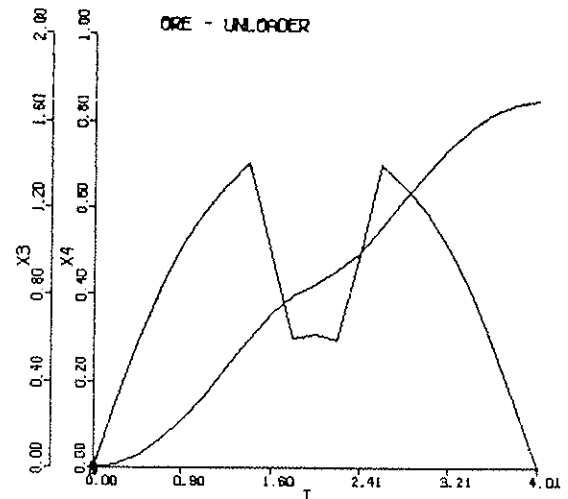
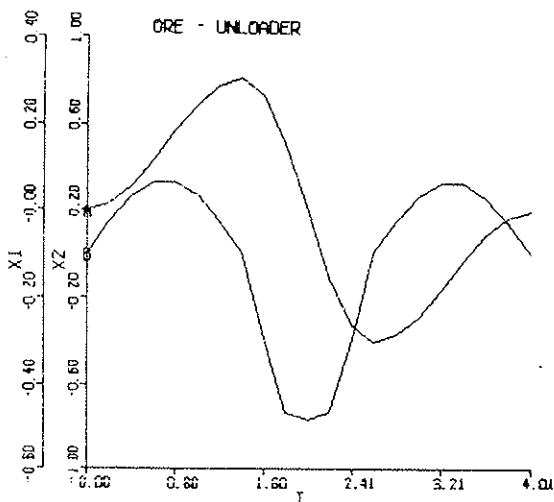
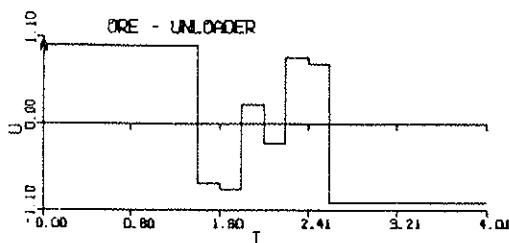
```
SET INDALG=.T.  
START  
PREPAR T,X1,X2,X3,X4,U  
START  
SET STRPLT=.T.,PRNPLT=.F.  
PLOT "XHI"=TEND,U,"LO"=-1.1,"HI"=1.1  
SET CALPLT=.T.,STRPLT=.F.  
PLOT X1,X2  
PLOT X3,X4  
D IFAIL, ITER $ D OBJF $ D C(1), C(2) $ D C(3), C(4)A  
STOP
```

Optimierungslauf

Simulationslauf

Ansprechen von Nicht-
Modellvariablen

Ergebnisse:



```
IFAIL      2          ITER      13  
OBJF 4.00529734  
C(1) -1.2140E-04    C(2) 0.00220887  
C(3) 8.7045E-05    C(4) 8.7604E-04
```

Literatur

1. Bausch-Gall I.: Parameteroptimierung bei technischen Modellen mittels einer kontinuierlichen Simulationssprache. Informatik-Fachberichte 56, Springer.
2. Breiteneker F.: Optimierung in kontinuierlichen Simulationssprachen: Aspekte bei Modellen technischer Systeme. Informatik-Fachberichte 85, Springer.
3. Breiteneker F.: Optimierung in HYBSYS. Interface 22 (1985)
4. ACSL User Guide/Reference Manual. Mitchell & Gauthier Ass., Concordia, Mass.
5. Hippe P.: Zeitoptimale Steuerung eines Erzentraders. Regelungs- und Prozeßdatenverarbeitung, Heft 8 (1970)

VOKALSYNTHESE

F.Rattay

Institut für Technische Mathematik, Technische Universität Wien

Die menschliche Stimme ermöglicht das Bilden verschiedener Phoneme, das sind die Elemente der Sprache, zum Beispiel /a/, /p/, /sch/, durch Verwenden des Sprechapparates (Abb. 1). Die Phoneme sind für verschiedene Sprachen unterschiedlich. Es gibt unter den deutschen Phonemen 8 Vokale: /a/, /e/, /i/, /o/, /u/, /ä/, /ö/, /ü/, während im Englischen 12 Vokale unterschieden werden müssen /l/.

Die beim Sprechen entstehenden Luftschwingungen sind der Informationsträger der Sprache. Verschiedene Personen, aber auch ein und derselbe Sprecher liefern beim Sprechen desselben Wortes unterschiedliche Schwingungsformen, auch wenn die Grundtonhöhe und die Sprechgeschwindigkeit gleich sind. Die Zuordnung der Schwingungsmuster zu den Phonemen ist eine bemerkenswerte Leistung des menschlichen Gehirns, kann aber in der Sprachanalyse durch Computereinsatz leider noch nicht zufriedenstellend gelöst werden.

Von der Lunge wird die Luft über die Stimmbänder gepreßt, die den Luftweg absperren, dann durch entsprechenden Druck wieder freigeben, und auf diese Weise wird die antreibende Grundschwingung von ca. 100 Hz bei männlichen und 200 Hz bei weiblichen Sprechern erzeugt. Durch die nachfolgenden, teils veränderbaren Hohlräume bekommt das Sprachsignal seine besondere Form (Abb. 1 und Abb. 2).

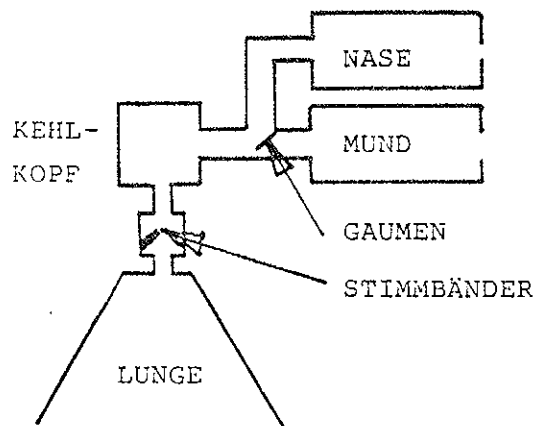


Abbildung 1

Schematische Darstellung des Sprechapparates.

Durch den die Stimmbänder spannenden Muskel ist die Grundtonhöhenverstellung möglich. Veränderungen der Mundöffnung und die Lage der Zunge sind die wichtigsten Hilfsmittel zur Erzeugung der Sprache.

Mittels Spektrogrammen lassen sich Fourierreihendarstellungen von Sprachsignalen ermitteln und die dominanten Frequenzen werden bei den Vokalen als Formanten bezeichnet. Eine grobe Einteilung der deutschen Vokale ist bereits durch Kenntnis der ersten beiden Formanten F_1 und F_2 möglich. Tabelle 1 zeigt die Werte der ersten beiden Formanten und ihrer relativen Amplituden, wie sie sich aus einer Fourierreihendarstellung ergeben.

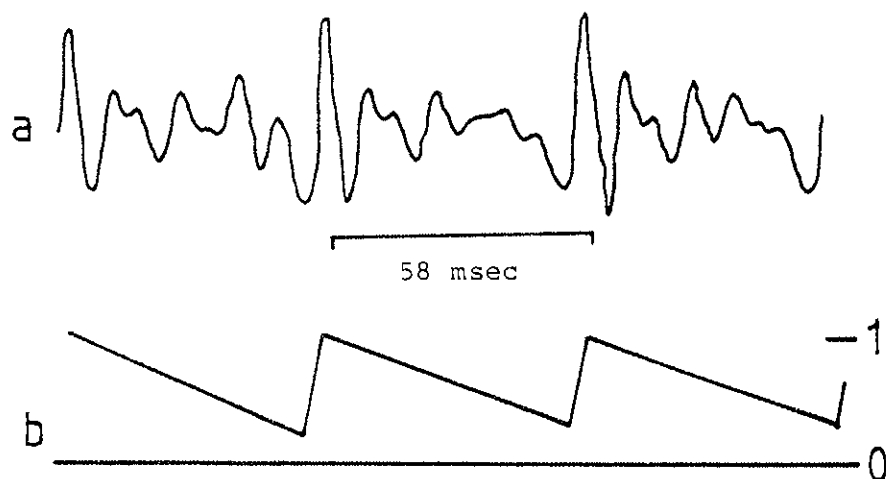


Abbildung 2

a. Drei Perioden eines /a/ von einem weiblichen Sprecher.
Die Stimmbänder schließen sich - periodisch - innerhalb von 58 msec
und bewirken dadurch einen fast linearen Abfall der Intensität des Sprachsignals
b. Darstellung der entsprechenden Dämpfungsfunktion

	F1	F2	A1	A2
/a/	780	1100	4,5	4
/e/	310	2000	2,5	0,4
/i/	270	2000	1	0,4
/o/	320	550	8	5,5
/u/	300	600	5	2,2
/ä/	480	1780	5	0,8
/ö/	320	1300	4	1,5
/ü/	300	1500	2,2	1,8

Tabelle 1

Die ersten beiden Formanten F1 und F2 (in Hz)
und die zugehörigen Amplituden A1 und A2 (relativ)

Die Zuordnung eines Vokals zu (F1,F2,A1,A2) ist, auch abgesehen von der Lautstärke, nicht eindeutig, sondern unterliegt einer Streuung.

Am Hybridrechner EAI PACER 600 wurden unter Verwendung der hybriden Simulationssprache HYBSYS die deutschen Vokale durch Summation von zwei entsprechenden Sinusschwingungen nachgebildet. Um den Effekt der Stimmbänder zu berücksichtigen, wurde diese Summenfunktion noch mit einer Dreiecksschwingung wie in Abb. 2b multipliziert. Die entstehende Schwingung kann über das Revox-Tonbandgerät direkt vom Analogrechner weg abgehört werden. Die synthetischen Vokale klingen - nach kleinen Korrekturen (vgl. Tabelle 2) gegenüber den Werten von Tabelle 1 - zwar ähnlich wie natürliche Vokale, aber doch mehr wie musikalische Töne. Jedenfalls kann man die vom Computer synthetisierten Vokale erlernen. Die Schwingungsformen für die deutschen Vokale sind in Abb. 3 dargestellt.

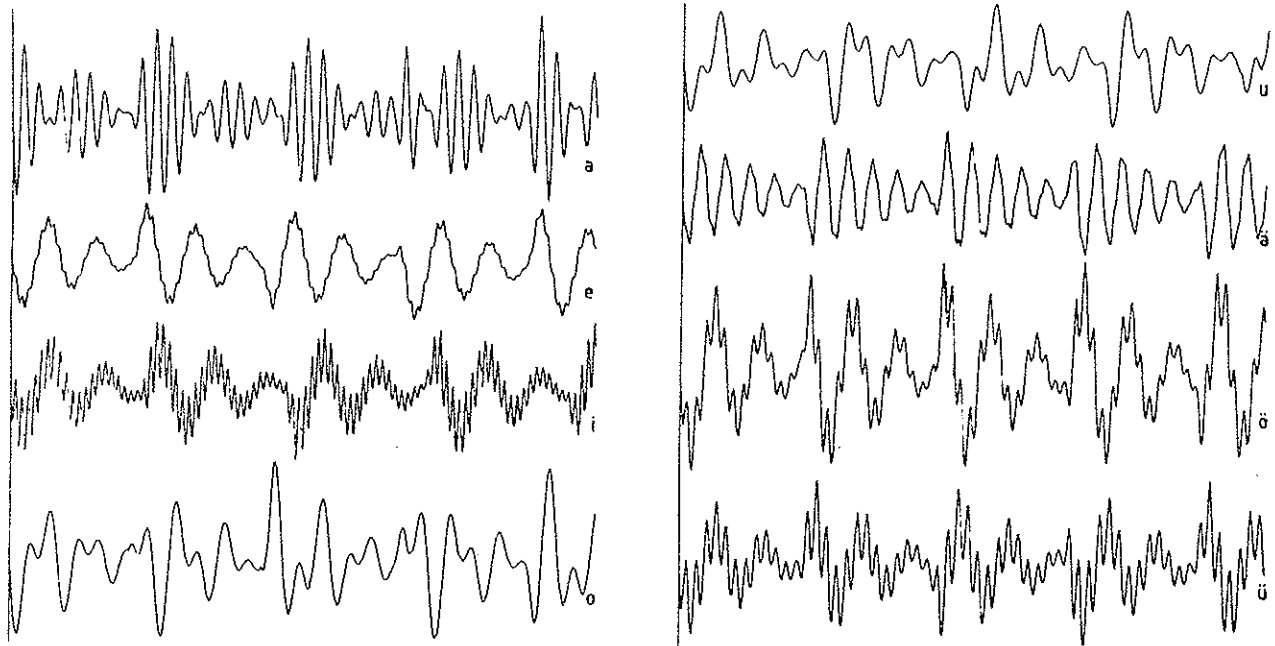


Abbildung 3

Nachbildung der deutschen Vokale durch die Summe zweier Sinusfunktionen.
 Die Summe ist noch mit einer Dämpfungsfunktion, die die Grundfrequenz der Stimmbänder simuliert (Abb. 2b), multipliziert.
 Die Sprachsignale sind über einen Zeitraum von 40 msec dargestellt.
 Frequenzen und Amplituden stimmen mit den Werten von Tabelle 2 überein.

Die relativen Maxima der Koeffizienten der Fourierreihendarstellung eines Vokals bestimmen seine Formanten, doch heißt dies natürlich nicht, daß auch die beste phonetische Approximation durch die Summe zweier harmonischer Schwingungen durch diese Formanten und ihre Amplituden bestimmt ist. Tatsächlich zeigen sich auch bei Vergleich der Tabellen 1 und 2 einige Abweichungen.

	F1	F2	A1	A2
/a/	900	1100	4.5	4
/e/	300	2500	5	0.6
/i/	270	2400	1	0.7
/o/	320	600	5	4
/u/	320	700	8	5
/ä/	600	2500	5	0.6
/ö/	320	1600	4	1.5
/ü/	300	1500	2.2	1.8

Tabelle 2

Frequenzen und Amplituden zweier Sinusschwingungen, deren Summe die Vokale phonetisch möglichst gut nachbildet

Eine bessere Sprachsynthese ist erst durch Verwenden höherer Formanten möglich und man weiß aus der Telefontechnik, daß der Frequenzbereich von Sprachsignalen nicht unter 3.5 kHz abgeschnitten werden soll. Die Verwendung der dreieckigen Grundfrequenz hat keinen besonderen Einfluß auf das Sprachverständnis und kann sogar auch ganz vernachlässigt werden.

Eine andere Möglichkeit der Vokalsynthese berücksichtigt mehr das Verwenden digitaler Technik und produziert das Sprachsignal aus der Überlagerung von rechteckigen Pulsfolgen (Abb. 4). Dieses "digitale" Sprachsignal ist zunächst unverständlich, doch führt bereits ein Tiefpaßfilter dritter Ordnung /2/ das digitale Signal in ein analoges über, das sowohl akustisch als auch optisch dem oben erwähnten ähnelt.

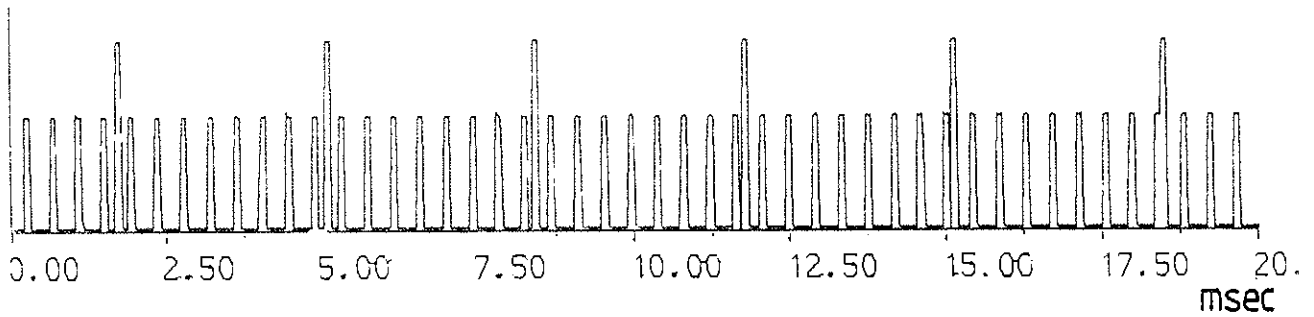


Abbildung 4

Impulsfolge für ein /e/.

Die Rechteckhöhen entsprechen den Amplituden, die Längen der Perioden den Frequenzen gemäß Tab. 2.

Die dargestellte Impulsfunktion wird wieder mit der Dämpfungsfunktion (Abb.2b) multipliziert und vor der akustischen Umsetzung gefiltert.

Die einfache Nachbildung der Vokale durch die Summe von zwei Sinusfunktionen, aber auch die digitale Darstellung, belegen nur einen geringen Teil der Analogkomponenten und es ist daher möglich, die Qualität der Signale durch Hinzunahme von höheren Frequenzen zu verbessern, oder aber parallel dazu noch andere Simulationen durchzuführen.

In einem früheren INTERFACE-Artikel /3/ wurde bereits über die Elektrostimulation des Gehörnervs berichtet: ein von einer implantierten Elektrode einer Hörprothese ausgehendes Stromsignal, das bis auf gewisse Verstärkungen dem Sprachsignal gleicht, kann die Fasern des Gehörnervs stimulieren. Die Reaktion einer Faser kann durch eine erweiterte Van-der-Pol Gleichung beschrieben werden /4/. Die Kapazität des Analogrechners EAI 680 der TU Wien erlaubt es, neben der Vokalsynthese auch noch dieses nichtlineare Nervenmodell dreimal parallel aufzubringen und so die Reizantwort von drei verschieden entfernten Fasern zu simulieren.

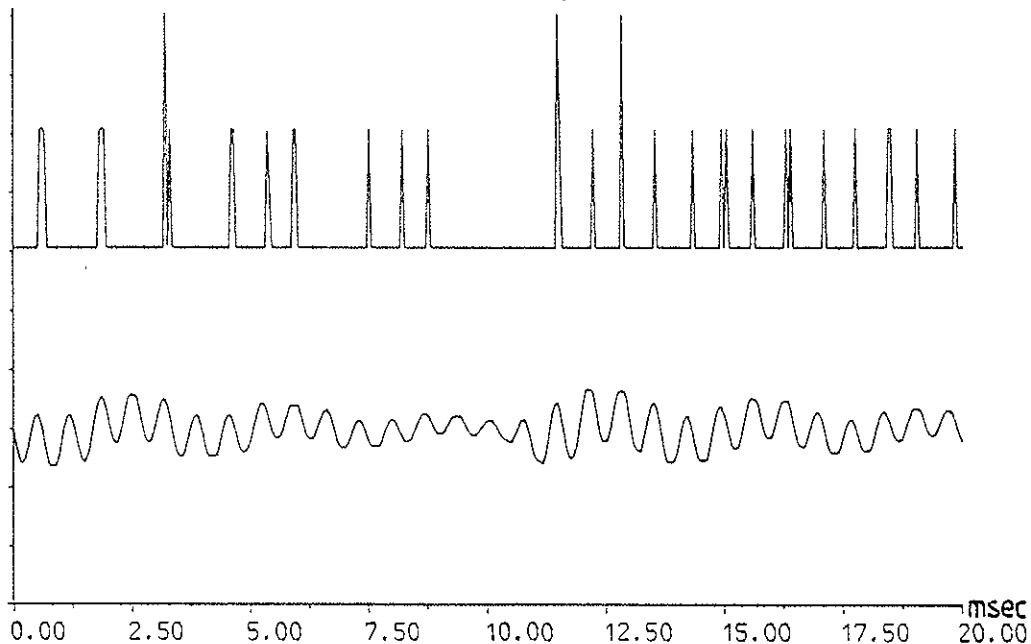


Abbildung 5

Simulation der Summenreaktion von drei Gehörnerven bei Stimulation mit einem synthetischen /ü/.

Die wichtige Information über den 2. Formanten (1500 Hz) kann nicht durch eine einzelne Nervenfasern übertragen werden.

Hingegen erlaubt das Summensignal von verschiedenen Fasern die Übertragung von Maxima des Reizsignals, die weniger als 1 msec auseinander liegen.

oben: Summe der Reizantworten von drei verschieden weit von der stimulierenden Elektrode liegenden Nervenfasern.

unten: stimulierendes Signal (20 msec)

Eine einzelne Faser erbringt bei elektrischer Stimulation eine maximale Übertragungsrate von 1000 Hz und das Unterscheiden von höheren Tönen kann nur auf Informationen verschiedener Fasern beruhen /5/. In Abb. 5 ist die Summe der Antworten von drei verschieden entfernt liegenden Fasern auf ein /ü/ dargestellt. Dieses Nervensignal enthält nun tatsächlich noch die Sprachinformation /ü/, was man durch Filtern und Abhören vom Analogrechner mit den oben besprochenen Techniken nachprüft.

Alle besprochenen Arbeiten konnten am Analogrechner parallel durchgeführt werden:

- Aufbringen von zwei Grundfrequenzen (bei digitaler Darstellung mit anschließender Filterung) inklusive der Multiplikation mit den Dämpfungswerten
- Gleichzeitige Darstellung der Nervenreaktion für drei verschiedene Intensitäten des stimulierenden Sprachsignals
- Filtern des sich ergebenden Nervensummensignals
- Ausgabe des abzuhörenden synthetischen Vokals und der Nervenantwort.

Für sämtliche Untersuchungen brauchte keine einzige Hardware-Modifikation an der Rechenanlage durchgeführt werden. Da auch sonst nur die vorhandenen Peripheriegeräte verwendet wurden, zeigt dies wieder einmal die Vielseitigkeit der Anwendungsmöglichkeiten einer hybriden Rechenanlage.

Literatur

1. J.L. Flanagan: Speech Analysis, Synthesis and Perception. Springer. New York (1972)
2. F. Rattay: Filtersimulation. INTERFACE 22 (1985)
3. F. Rattay: Simulation der Reizentwicklung am Gehörnerv durch elektrische Anregung. INTERFACE 20 (1983)
4. I.J. Hochmair-Desoyer, E.S. Hochmair, H. Motz, F. Rattay: A model for the electrostimulation of the nervus acusticus. Neuroscience 13, 553-562 (1984)
5. F. Rattay: Investigation of the problem of vowel recognition by the profoundly deaf fitted with an electrostimulating prosthesis by means of simulation with nerve models. Hearing Research, in Vorbereitung

SIMULATION DER HERZFREQUENZ UNTER BELASTUNG

F. Breitenecker
Institut für Technische Mathematik, TU Wien
J. Kaliman, G. Reisner
Kardiologische Universitätsklinik

Einleitung

Physische Belastung führt zu einem erhöhten Sauerstoffbedarf, dem das Blutkreislaufsystem durch schnelleren und vermehrten Sauerstofftransport gerecht werden muß. Unter den Kenngrößen des Blutkreislaufsystems ist nun das Herzminutenvolumen (Puls x Schlagvolumen) jene, die den aktuellen Energie(= Sauerstoff-)bedarf am besten charakterisiert, die jedoch nur "blutig" (mit Sonden) gemessen werden kann. Als Ersatzgröße bietet sich der Puls an, dessen belastungsbedingte Erhöhung ebenfalls zur Erhöhung des Herzminutenvolumens führt (bei Belastung in nicht liegender Position erhöhen sich Puls und Schlagvolumen gleichzeitig, sodaß die Reaktion "Erhöhung des Pulses - Erhöhung des Herzminutenvolumens" als nichtlinear anzusehen ist). Für die Modellbildung eignet sich daher als Beschreibungsgrundlage der Zusammenhang "physische Belastung - Puls", da beide Größen sehr leicht gemessen werden können.

Modellbeschreibung

Physiologisch sind zwei Komponenten für eine Pulserhöhung verantwortlich: zu Beginn der Belastung eine rasch eintretende Vagushemmung und als Folge eine später einsetzende wesentlich langsamer verlaufende Sympatikusreizung (/3/). Das kann mathematisch durch die Koppelung einer schnellen nervalen "Feedforward"-Komponente (Vaguskomponente) und einer langsamen neurohumoralen "Feedback"-Komponente (Sympatikuskomponente) beschrieben werden, wenn man den Mechanismus "Belastung - Pulserhöhung" beschreiben möchte (zur Beschreibung des prinzipiellen Blutkreislaufmechanismus muß die gesamte Pulsdruckkurve modelliert werden). Innerhalb dieser Komponenten bieten sich nun regelungsmathematische Methoden zur Beschreibung des Zusammenhanges an. Bild 1 zeigt zwei lineare Modelle in regelungstechnischer Notation.

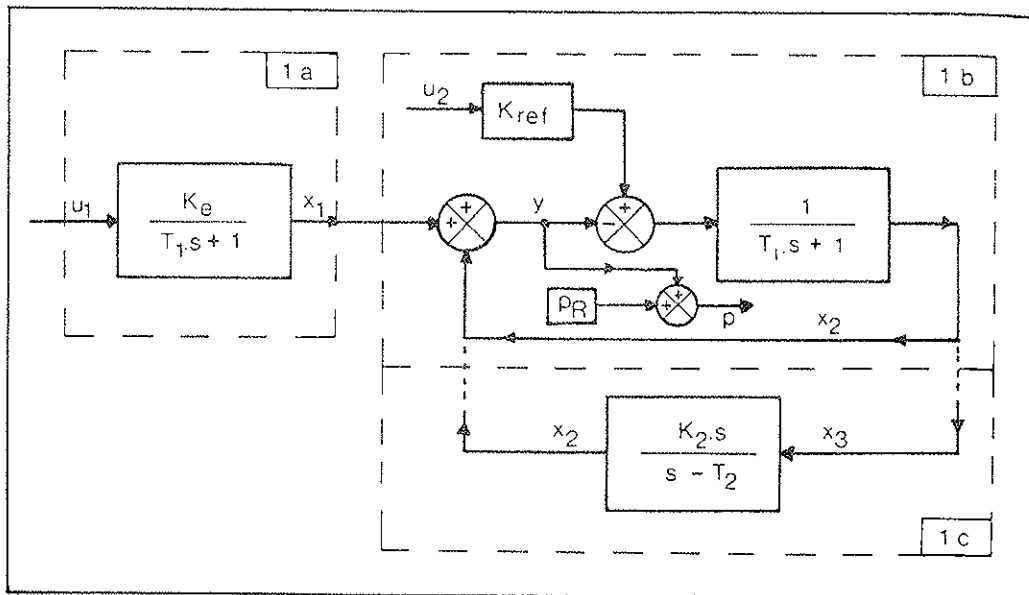


Abbildung 1
Regelungstechnisches Blockdiagramm für mathematisches Modell

Die Eingangsgröße $u_2(t)$ beschreibt quantitativ die Belastung ([W]), $u_1(t)$ ist ein Indikator für die nervale Komponente, der angibt, ob eine Belastung vorliegt oder nicht (dimensionslos). Die Zustandsgrößen $x_1(t)$ und $x_2(t)$ ($[s^{-1}]$) stellen die belastungsbedingten Frequenzerhöhungen in der jeweiligen Komponente dar und ergeben gemeinsam mit dem Ruhepuls P_R die Höhe des Pulses. Die Parameter T_i und T_1 (und T_2) ($[s]$) sind die Zeitkonstanten der Komponenten, K_e repräsentiert den Verstärkungsfaktor des Effekts der Vagushemmung ($[s^{-1}]$), K_{ref} charakterisiert die resultierende Pulserhöhung bei Belastung (von 1W). Die Parameter T_2 und K_2 kennzeichnen ein Korrekturglied $x_3(t)$ für die langsame Komponente.

Die verwendete Modellbeschreibungsmethode - eine Kombination aus "compartmental modelling" (2 Kompartments) und "concept fitting" mit regelungsmathematischen Methoden - ist auch bei der Modellbildung für andere physiologische Prozesse erfolgreich (/1/,/6/).

Simulation

Das mathematische Modell (Abb. 1) wurde mit Hilfe der hybriden Simulationssprache HYBSYS (/5/) simuliert und identifiziert. Besonders bei der Identifizierung (Bestimmung der Modellparameter T_i , T_1 , K_e , K_{ref} zur individuellen Anpassung des simulierten Verhaltens an Meßdaten) kämen die Vorteile der parallelen Simulation in HYBSYS vor allem durch kurze Rechenzeiten zum Tragen. Die Berechnung der Parameter für eine individuelle Meßkurve (die Meßkurven wurden mit Standard-Ergometrie an der kardiologischen Universitätsklinik ermittelt) wurde mit dem HYBSYS-Standard-Overlay ZERO durchgeführt, der die Abweichung Modellkurve - Meßkurve durch Änderung der Parameter T_i , T_1 , K_e , K_{ref} minimiert; für die Identifizierung einer Meßreihe waren durchschnittlich 40 Simulationsläufe nötig.

Abbildung 2 zeigt das Ergebnis für einen Fall (Meßreihe strichliert).

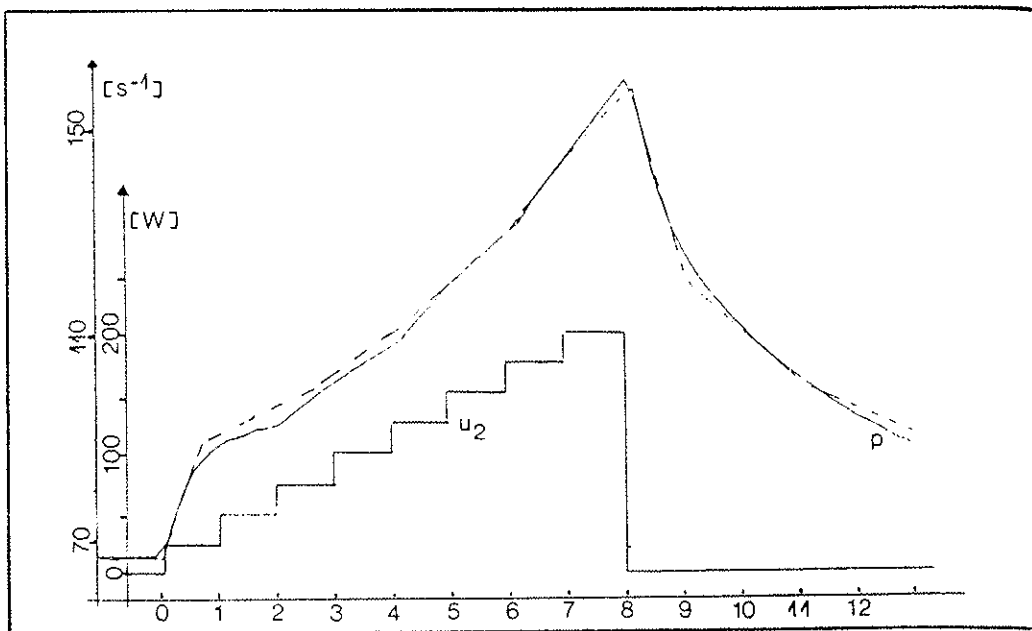


Abbildung 2
Belastungskurve u_2 ,
gemessene Kurve (strichliert) und Modellkurve p für Puls

Es zeigte sich bei allen 98 zur Verfügung stehenden Meßreihen, daß das einfache Modell (ohne Korrekturteil $x_3(t)$) ausreichend ist (vergleiche /2/, /3/).

Statistische Untersuchung der Ergebnisse

In weiterer Folge wurden die individuell identifizierten Modellparameter T_i , T_l , K_e , K_{ref} untersucht und mit den üblichen physiologischen Kenngrößen Alter, Gewicht, Größe, erwartete Maximalleistung (abhängig von Alter, Gewicht, Größe, Geschlecht) und erwarteter Maximalpuls (abhängig vom Alter) der Ergometrie (/4/) in Relation gesetzt. Dabei ergaben sich interessante Beziehungen (siehe Korrelationstabelle, Abb. 3). Von Interesse ist dabei die starke negative Korrelation $K_{ref} - E_{maxL}$, die im wesentlichen die Aussage "je fitter eine Person, desto geringer der Effekt der Belastung auf die neurohumorale Komponente" stützt. Weitere Korrelationen können ebenfalls erfolgreich aus physiologischer Sicht interpretiert werden (siehe /2/).

KORRELATIONSMATRIX								
	Alter	Gewicht	Größe	E_{maxL}	E_{maxP}	T_i	K_{ref}	K_e
Alter	1.000	0.168	-0.181	-0.527	-0.969	0.051	-0.031	-0.144
Gewicht		1.000	0.474	0.538	-0.153	0.144	-0.475	-0.028
Größe			1.000	0.474	0.188	-0.045	-0.494	-0.168
E_{maxL}				1.000	0.540	0.057	-0.449	0.015
E_{maxP}					1.000	-0.076	0.012	0.099
T_i						1.000	0.187	0.229
K_{ref}							1.000	0.312
K_e								1.000

Abbildung 3
Korrelationsmatrix für Modell- und physiologische Parameter

Derzeit wird das Modell um die Beschreibung des Verhaltens des systolischen Blutdruckes unter Belastung erweitert, wofür ebenfalls Meßreihen zur Verfügung stehen.

Literatur

1. Breitenecker F., Kaliman J.: Simulation and analysis of pathological blood pressure behaviour after treadmill test in patients with coarctation of the aorta. Informatik Fachbericht 56, Springer (1983)
2. Breitenecker F., Kaliman J., Reisner G.: Modellbildung und Simulation der Herzfrequenz unter Belastung. Informatik Fachbericht 85, Springer, S. 409-414 (1984)
3. Hajek M., Potucek J., Brodan V.: Mathematical model of heart rate regulation during exercise. Automatica 3 (1980)
4. Niederberger M.: Prinzipien der Ergometrie. Herz 4 (1982)
5. Solar D., Berger F., Blauensteiner A.: HYBSYS - interactive simulation software for a hybrid multiple-user system. Informatik Fachbericht 56, Springer, S. 257 (1983)
6. Troch I., Breitenecker F., Gampe J., et al.: Modelling by simulation of hepatic glucose production in vitro. Proc. Int. IMACS Conf. "European Simulation Meeting on Simulation in Research and Development", Eger, Hungary (1984)

ANKERGESTEUERTE GLEICHSTROMMOTORE ZUR BEWEGUNG VON MASCHINENTEILEN

UNTER REIBUNGSEINFLUSS

F. Rattay
Institut für Technische Mathematik

Bei vielen Bewegungen von Maschinenteilen ist der Reibungseinfluß relativ groß. Wird etwa die Spielfreiheit von Roboterarmen durch Anwendung von Federdruck erreicht, so wird der Hauptteil der aufzubringenden Arbeit durch die Reibung verursacht. Der elektrische Antrieb diverser Stellglieder erfolgt vorzugsweise mit Gleichstrommotoren oder Schrittmotoren.

Ein typischer Anwendungsfall ist der Roboter mit sechs Freiheitsgraden /1/, wo jeder der bewegten Teile durch einen eigenen Motor angetrieben wird. Für jede Einzelbewegung läßt sich ein vereinfachtes Ersatzsystem angeben, wobei alle bewegten Massen zu m vereint werden oder dem Rotor des Motors zugeschlagen werden. Dieses vereinfachte Ersatzsystem (Abb. 1) reduziert die vorgegebene Aufgabe auf das Bewegen der Masse m auf horizontaler Unterlage unter Reibungseinfluß:

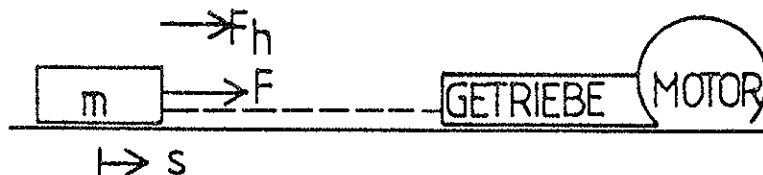


Abbildung 1
Ersatzsystem eines bewegten Maschinenteils. Über das Getriebe wird die Masse m , auf die noch die Störkraft F_h wirkt, bewegt.

Mit Hilfe eines Gleichstrommotors, dessen Leistung über die Ankerspannung gesteuert werden kann, soll die Masse m eine vorgegebene Horizontalbewegung $s(t)$ ausführen, wobei die Drehbewegung des Motors durch ein Getriebe mit konstantem Übersetzungsverhältnis k in eine Linearbewegung übergeführt wird. Alle äußeren Kräfte werden zur horizontal wirkenden Kraft $F_h(t)$ zusammengefaßt, sodaß sich für die Geometrie nach Abb. 1 folgende Kräftebeziehung ergibt

$$F + F_h = m \cdot \ddot{s} + mg\mu \quad (1)$$

In einem vereinfachten Modell läßt sich der Reibungskoeffizient μ im bewegten Zustand durch einen konstanten und einen der Geschwindigkeit proportionalen Anteil darstellen, während erst eine genauere Modellierung mittels geschwindigkeitsabhängiger Reibungskennlinie dem erhöhten Haftreibungskoeffizienten μ_0 gerecht wird. Hier wird also μ angesetzt mit

$$\mu = \alpha \dot{s} + \beta \cdot \text{sign}(\dot{s}) \quad (2)$$

Die für den gewünschten Bewegungsablauf erforderliche Kraft F wird über ein Getriebe mit dem Übersetzungsverhältnis k aus dem vom Motor abgegebenen Moment M bezogen. Es ergeben sich dabei die folgenden Beziehungen für den Verdrehwinkel ϕ des Rotors, die Winkelgeschwindigkeit ω und das Moment M :

$$\phi = k \cdot s \quad (3a)$$

$$\dot{\phi} = \omega = k \cdot \dot{s} \quad (3b)$$

$$\ddot{\phi} = k \cdot \ddot{s} \quad (3c)$$

$$F = k \cdot M \quad (3d)$$

Als Antrieb diene nun ein Gleichstrommotor mit gesteuerter Ankerspannung und konstanten Statormagneten, die oft als Permanentmagnete ausgeführt sind. Hat der Anker die Selbstinduktion L , den Widerstand R , das Trägheitsmoment I_r und die Motorkonstante $c\psi$, so erzeugt die Steuerspannung am Rotor das Moment M , wobei der Ankerstrom I fließt /2/, /3/:

$$L \cdot \dot{I} + R \cdot I = U - c\psi \cdot \omega \quad (4)$$

$$I_r \dot{\omega} = c\psi \cdot I - M \quad (5)$$

Durch (3) wird aus (1) und (5) nach Elimination von M

$$k \cdot c\psi \cdot I + F_h = \ddot{s} \cdot (k^2 I_r + m) + mg\alpha \quad (6)$$

und aus (4) wird

$$\dot{I} = \frac{1}{L} (-RI - c\psi \cdot k \cdot \dot{s} + U) \quad (7)$$

Das lineare Differentialgleichungssystem (6), (7) läßt sich für den bewegten Zustand mittels (2) durch Einführen der Geschwindigkeit $v = \dot{s}$ reduzieren auf

$$\dot{v} = aI - bv \pm c + d \quad (8)$$

und

$$\dot{I} = -eI - fv + \frac{U}{L} \quad (9)$$

mit

$$a = \frac{k \cdot c\psi}{k^2 I_r + m}, \quad b = \frac{mg\alpha}{k^2 I_r + m}, \quad c = \frac{mg\beta}{k^2 I_r + m}, \quad d = \frac{F_h}{k^2 I_r + m},$$

$$e = \frac{R}{L} \quad \text{und} \quad f = \frac{k \cdot c\psi}{L}$$

Wegen der Signumfunktion in (2) ist in (8) $-c$ zu nehmen für $v > 0$ und $+c$ für $v < 0$.

Elimination von I aus (8) und (9) führt zu

$$\ddot{v} + \dot{v}(b+e) + v(eb+af) = \frac{a \cdot U}{L} \pm ec + ed \quad (10)$$

Gleichung (10) erlaubt bei vorgegebener Bewegung $s(t)$ und vorgegebener Störkraft $F_h(t)$ unter Verwendung der Motordaten die Ankersteuerspannung U anzugeben.

In vielen Anwendungsfällen werden in letzter Zeit wegen ihres geringen Ankerträgheitsmomentes oft Scheibenläufermotore eingesetzt. Für einen 500 W Motor sind dabei die folgenden Daten typisch (BBC - Model MA17H):

$$L=0.00015 \text{ H}, \quad R=1.7 \text{ Ohm}, \quad c\psi=0.3 \text{ Vsec}, \quad I_r=0.0007 \text{ kgm}^2. \quad (11)$$

Dieser Motor soll nun eine Masse von $m=10\text{kg}$ mit einer maximalen Geschwindigkeit von 2m/sec bewegen.

Für $\alpha = 0, \quad \beta = 1, \quad F_h = -100 \text{ N}, \quad k = 100 \text{ rad/m}$

ergibt sich für die Koeffizienten (ohne Berücksichtigung der Dimension):

$$a = 1.785, \quad b = 0, \quad c = 5.77, \quad d = -5.88, \quad e = 11300, \quad f = 200000$$

und aus (10) kann die Steuerspannung U berechnet werden zu

$$U = 0.000084 \cdot \ddot{v} + 0.95 \dot{v} + 29.6v + 5.5 + 5.6$$

Diese Gleichung zeigt das bekannte Verhalten, daß die Steuerspannung im wesentlichen proportional zur Drehzahl sein muß, recht deutlich. Die beiden letzten Terme zeigen den Einfluß der Reibung und der Störkraft F_h . Durch die Abb. 2a wird die Spannungsanhebung durch den Reibungseinfluß h für eine harmonische Bewegung mit einer Periodendauer von 2 sec verdeutlicht:

$$v = v_{\max} \cdot \sin(\gamma t); \quad v_{\max} = 2 \text{ m/sec}, \quad \gamma = 3.14 \text{ rad/sec} \quad (12)$$

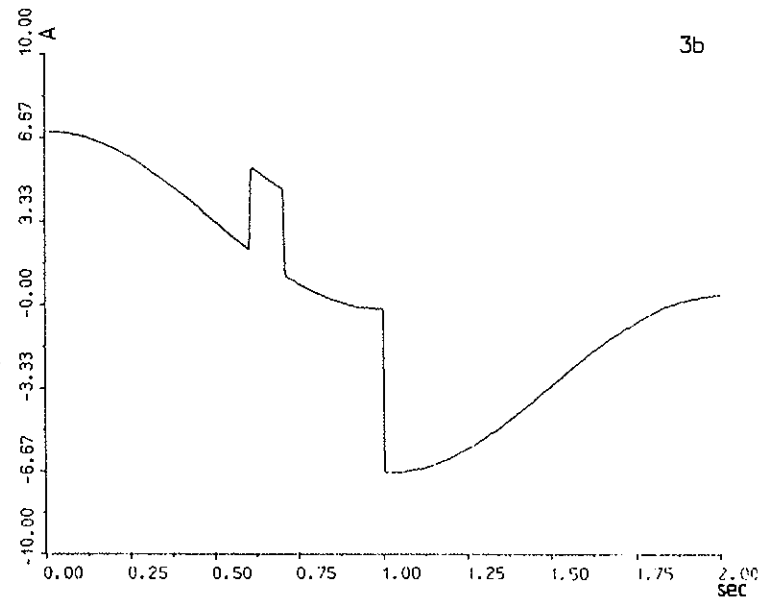
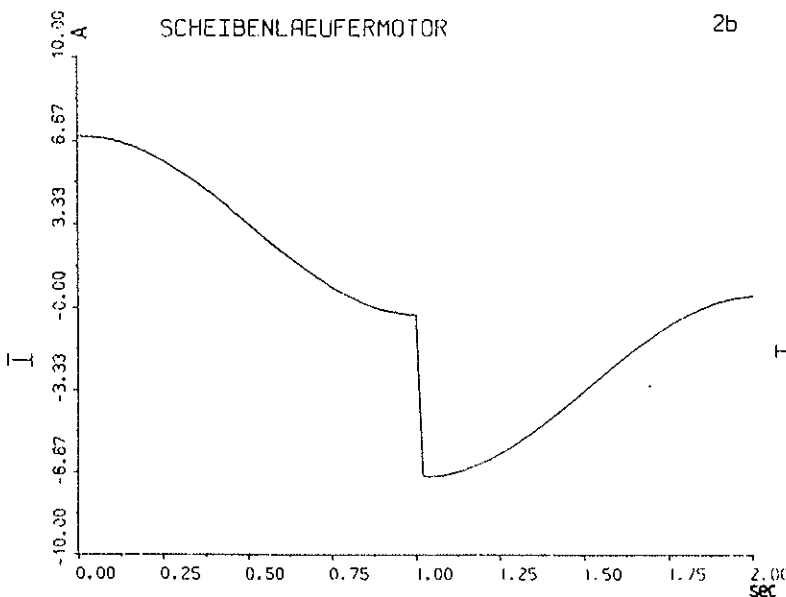
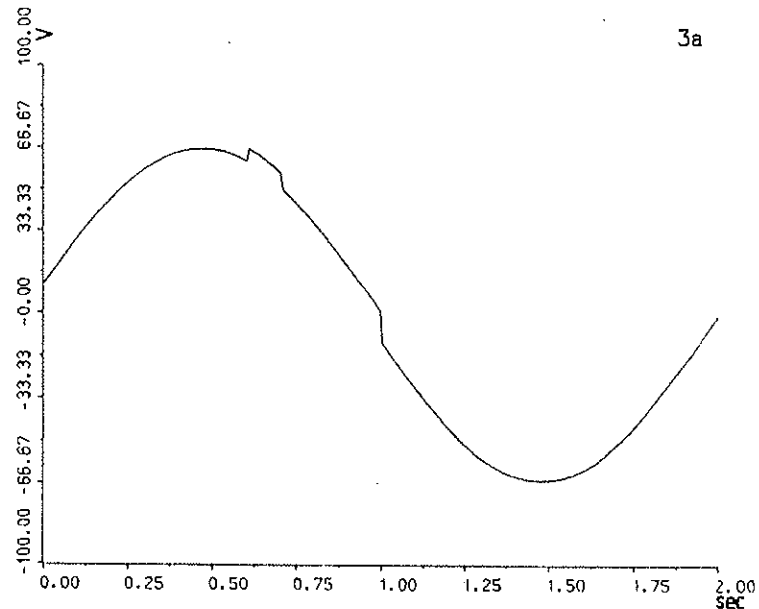
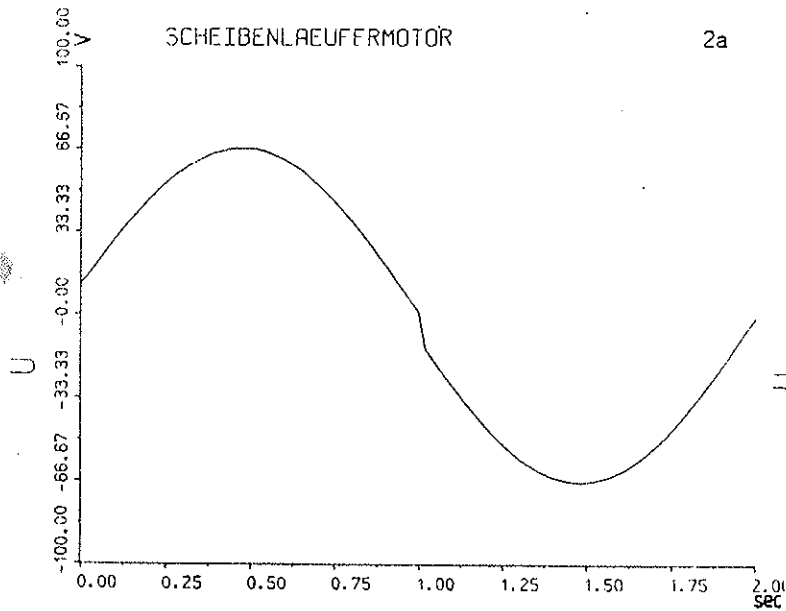


Abbildung 2

Spannung und Stromverlauf eines 500 W Scheibenläufermotors als Antrieb für eine harmonische Bewegung einer Masse von 10kg gemäß (11, 12);

$$\alpha = 0, \beta = 1, k = 100.$$

Die sprunghafte Änderung von Spannung und Strom rührt vom Vorzeichenwechsel der Geschwindigkeit her und wird durch die entgegengesetzt wirkende Reibkraft verursacht.

Abbildung 3

Auswirkung einer kurzzeitigen Störung auf den Spannungs- und Stromverlauf:

$$F_h = -100 \text{ N für } 0.6 \leq t \leq 0.7 \text{ sec;}$$

alle anderen Daten wie bei Abb. 2.

Bei $t = 1 \text{ sec}$ erfolgt mit $v=0$ die Bewegungsumkehr und die Reibkraft wirkt in die andere Richtung. Wesentlich stärker ist das Kräftespiel in der Stromstärke widergespiegelt (Abb. 2b). Abb. 3 zeigt Spannung und Stromverlauf für eine kurz wirkende Störkraft von -100N .

Da dieser Scheibenläufermotor Permanentmagnete im Stator aufweist, läßt sich aus den angegebenen Gleichungen auch die aufgenommene Leistung leicht als $U \cdot I$ ermitteln, wobei der im Generatorbetrieb gewonnene Strom im allgemeinen nicht verwertet wird, sodaß dann für die Leistungsbilanz stets $U \cdot I \geq 0$ gerechnet wird. Abb. 4a stellt die Aufnahmeleistung für $F_h=0$ dar.

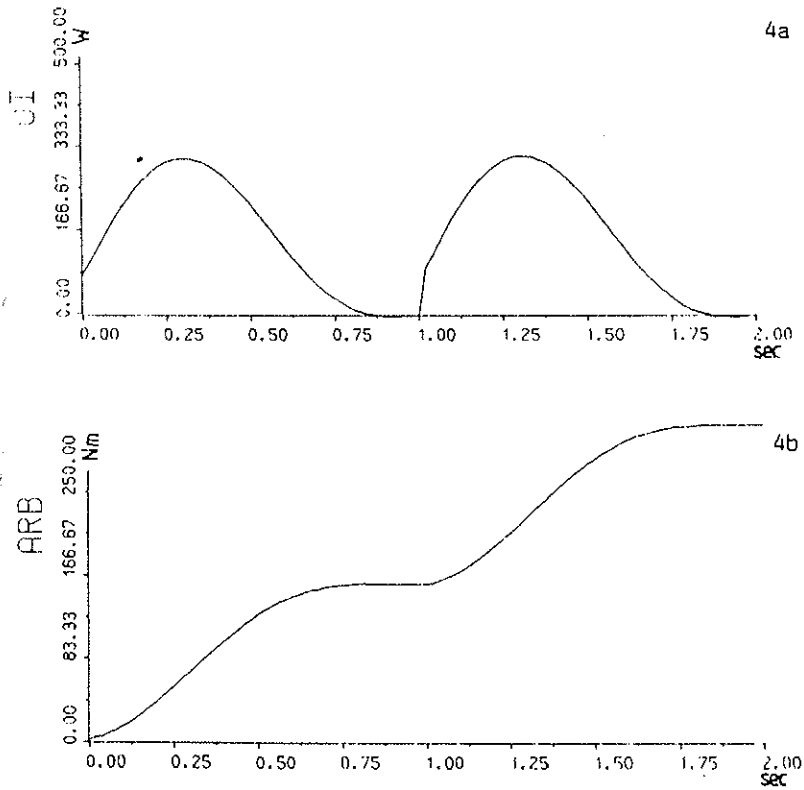
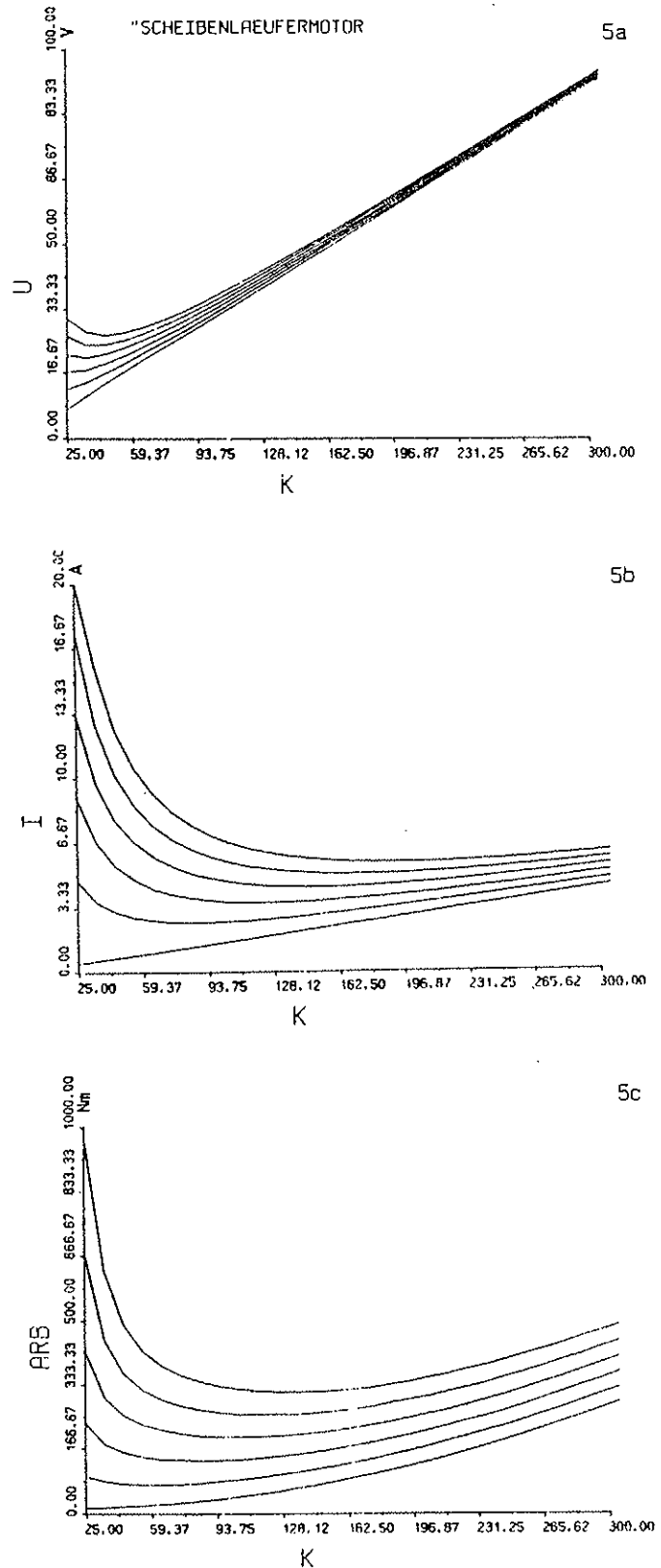


Abbildung 4
Leistung $U \cdot I$ und aufzubringende Arbeit für die Annahmen der Abb. 2.

Abbildung 5
Maximale Spannung und maximale Stromstärke und aufzubringende Arbeit für einen vollen Bewegungsablauf von 2 sec, gemäß den Daten von Abb. 2 in Abhängigkeit vom Übersetzungsverhältnis k des Getriebes, dargestellt für $m = 0, 2, 4, 6, 8$ und 10 kg .



Kennt man den am häufigsten angewandten Bewegungsablauf, so läßt sich dazu das geeignete Übersetzungsverhältnis angeben. In der Abb. 5 wird der Einfluß des Übersetzungsverhältnisses für eine Arbeitsperiode von 2 sec mit obigen Angaben für $m=0,2,4,\dots,10$ kg angegeben. Bei kleinem k ergeben sich große Stromstärken, die wegen starker Motorerwärmung unerwünscht sind, bei zu großem k wird der Bewegungsablauf hohe Winkelbeschleunigungen des Rotors erfordern.

Ist eine Sollbewegung vorgegeben, so kann durch (10) die notwendige Steuerungspannung sofort hingeschrieben werden und (9) liefert auch eine Abschätzung des dabei fließenden Stroms. Durch die Simulation des beschriebenen Modells in HYBSYS lassen sich Steuerungspannung und Stromverbrauch sowie der damit verbundene Energiekonsum schnell veranschaulichen. Außerdem können Parametereinflüsse wie etwa das Getriebeübersetzungsverhältnis (Abb. 5) rasch untersucht werden.

Literatur

1. Desoyer K., Kopacek P. und Troch I.: Industrieroboter und Handhabungsgeräte, Oldenburg, München (1985)
2. Föllinger O.: Regelungstechnik, Elitera Verlag, Berlin (1983)
3. Bödefeld und Sequenz: Elektrische Maschinen, Springer, Wien (1971)

WALDSTERBEN

Informatikpraktikum II
Institut für Analysis, Technische Mathematik
und Versicherungsmathematik, TU Wien
Prof. Troch, Doz. Breitenecker

Gruppe W. Reitgruber, A. Sauberer

Das Modell Baum zur Simulation von Luft- und Bodenbelastung bei Bäumen lag in Form einiger Berichte der Arbeitsgruppe Mathematisierung der Universität Kassel und eines Basicprogrammes vor (H. Bossel, Editor: Dynamik des Waldsterbens, Fachbericht Simulation, Band 4, Springer, Heidelberg 1985).

Implementierung

Dieses Programm wurde für die Simulationssprache ACSL adaptiert und einige Erweiterungen wurden hinzugefügt. So ist es nun möglich, zwei verschiedene Baumarten mit unterschiedlichen Belastungen entsprechend der Annahme, daß Randbäume eines Waldes stärker belastet werden als die inneren Bäume, gleichzeitig zu simulieren. Durch das Setzen der Variablen UMSCH auf TRUE können die Berechnungen dahingehend gekoppelt werden, daß zu den Zeitpunkten, zu denen die Randbäume (Baumart 2) abgestorben sind, die aktuellen Werte der 1. Baumart übernommen werden und mit diesen weitergerechnet wird. Durch den Aufbau mit Makros ist es leicht möglich, die Anzahl der verschiedenen Baumarten noch auszuweiten. Bezeichnet werden die artspezifischen Größen wie in den Berichten, gefolgt von einer Ziffer, die die Baumart angibt.

Die Leistungsfähigkeit des Programmes wird in den anschließend dokumentierten Zeichnungen dargestellt. Zugleich werden auch die Simulationsergebnisse diskutiert, wobei über die Untersuchungen in den zitierten Berichten hinaus noch einige andere Untersuchungen durchgeführt worden sind. So können z.B. durch das Hinzufügen des Commandfiles SETUP (beim Aufruf des Programmes) leicht andere Klimata verwendet werden.

Modellübersicht

Die Entwicklung der folgenden Größen kann durch das Modell beschrieben werden:

BIOM1, BIOM2	holzige Biomasse in 1000 kg _{OTS} /ha
WURZ1, WURZ2	Feinwurzelmasse in 1000 kg _{OTS} /ha
LAUB1, LAUB2	Gesamtlaubmenge in 1000 kg _{OTS} /ha
ASSI1, ASSI2	Assimilatmenge in 1000 kg _{OTS} /ha
L1(1), ..., L1(NJ), L2(1), ..., L2(NJ)	Laubmenge pro Nadeljahrgang in 1000 kg _{OTS} /ha

Wichtigste Parameter

Auf die folgenden Parameter wurde besonderes Augenmerk gelegt:

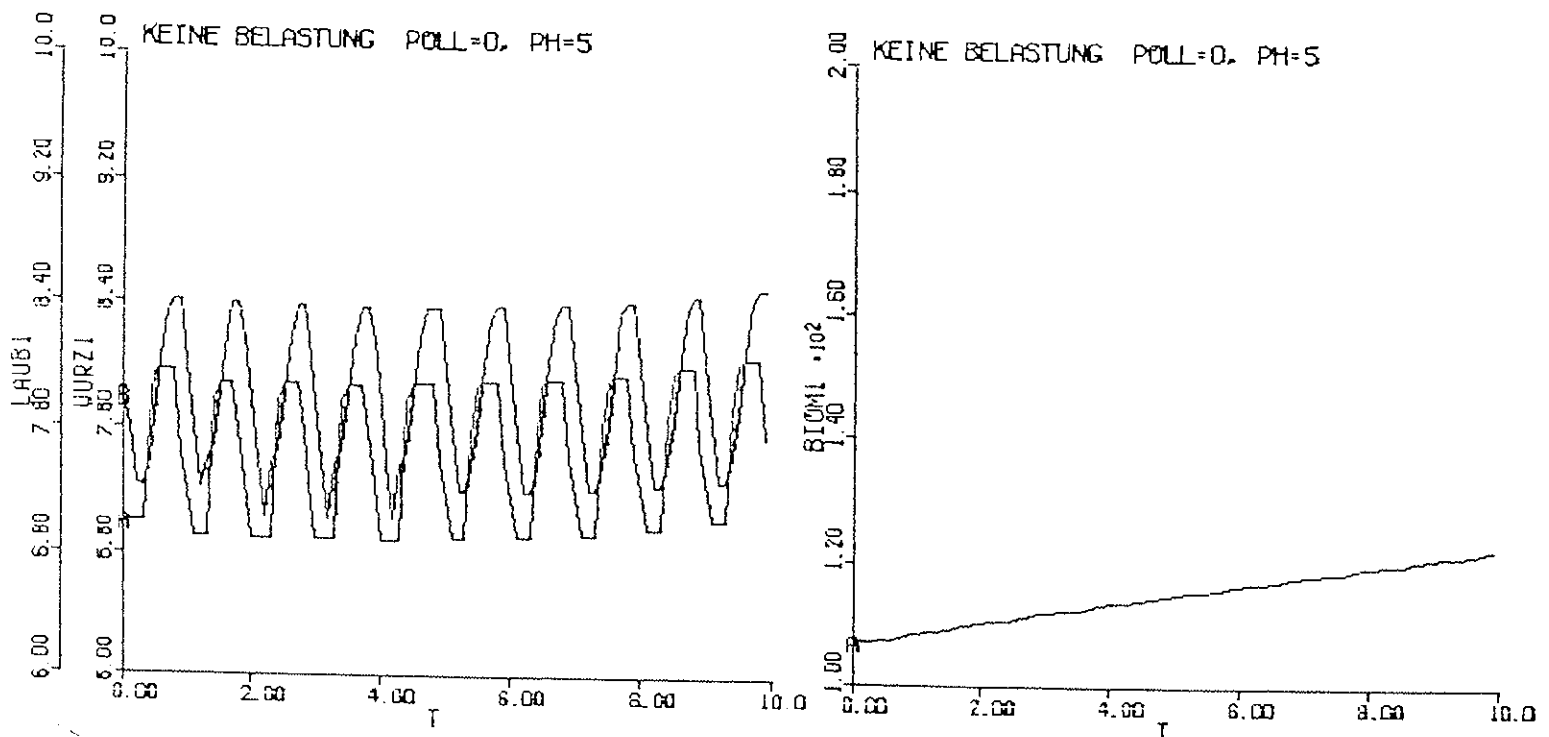
POLL1, POLL2	Schadstoffkonzentration in der Luft $0...50 \mu\text{gSO}_2/\text{m}^3$ Luft
PH1, PH2	pH-Wert des Bodens $2...7$
BWP	Bodenwasserpotential in bar, gibt den Wassermangel im Boden z.B. durch Absinken des Grundwasserspiegels an
LUX	mittlerer Lichteinfall in W/m^2
TTAB	Tabellenfunktion für die Monatsmittelwerte der Lufttemperatur.

Beispiele

Das Verhalten des (der) Baumes (Bäume) wird durch 10 Jahre hindurch dokumentiert.

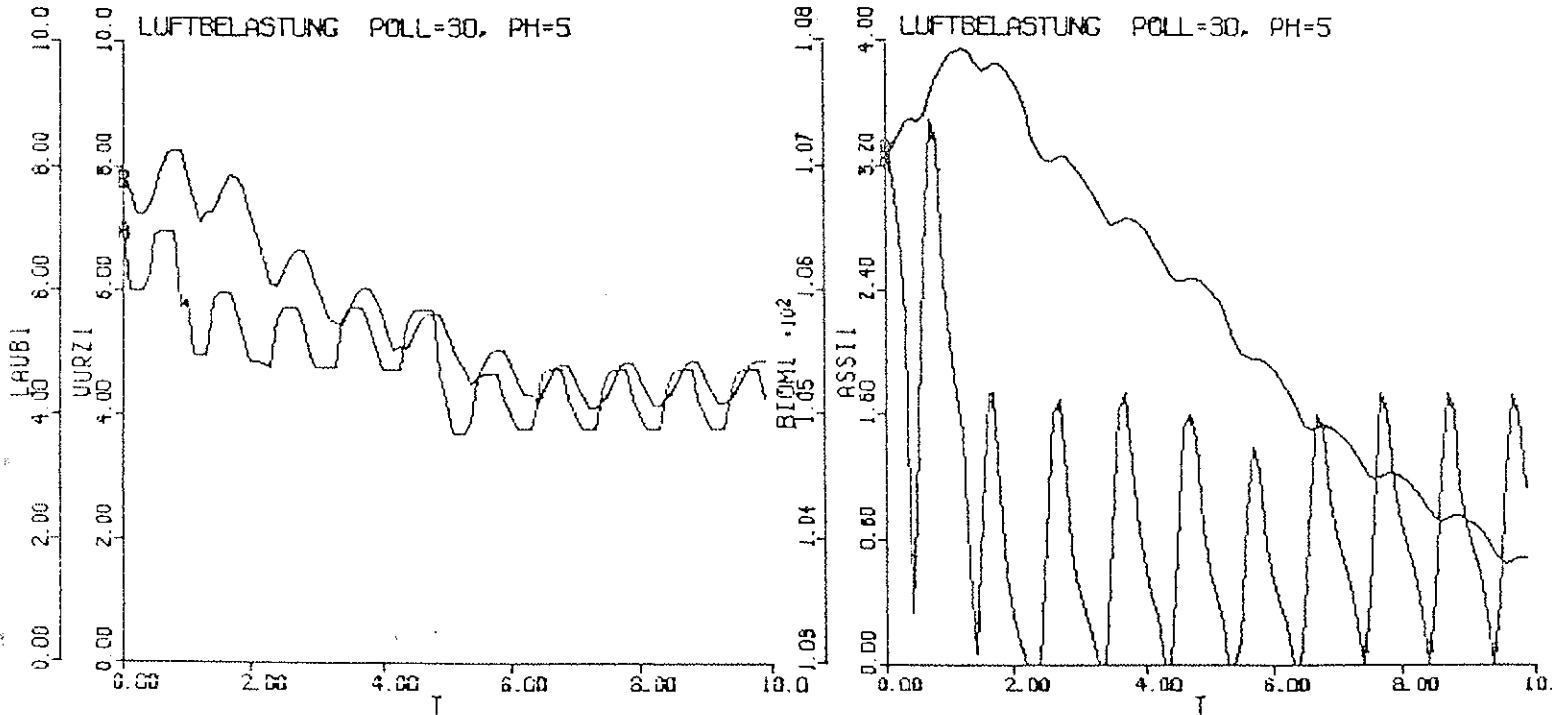
Keine Belastung

Ein gesunder Baum (60 Jahre alt) wird modelliert. Die Gesamtbiomasse steigt langsam aber stetig an und der Laub- und Feinwurzelhaushalt ist stabil; er zeigt jahreszeitlich bedingtes periodisches Verhalten, hervorgerufen durch das Abwerfen alter Nadeljahrgänge und abgestorbener Feinwurzeln, sowie den Aufbau neuer Substanz.



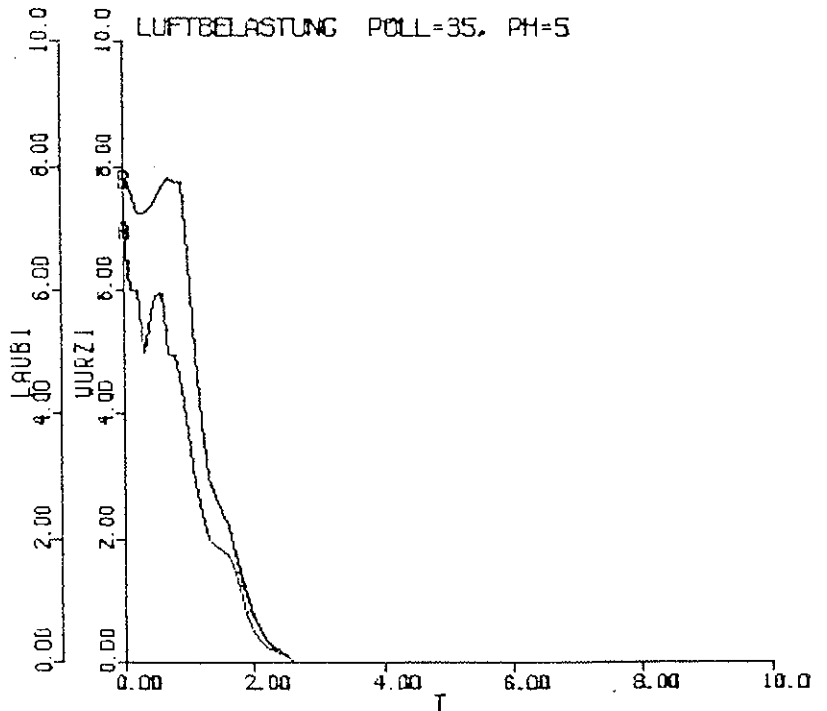
Luftbelastung 1

Hier wird die Luft mit $30 \mu\text{gSO}_2/\text{m}^3$ Luft belastet, während der pH-Wert des Bodens unverändert auf dem Normalwert von 5 bleibt. Es zeigt sich ein Siechen des Baumes, das aber nicht zum Tod führt, sondern zu einem Einpendeln bei niedrigeren Werten. Periodisches Verhalten ist weiterhin erkennbar.



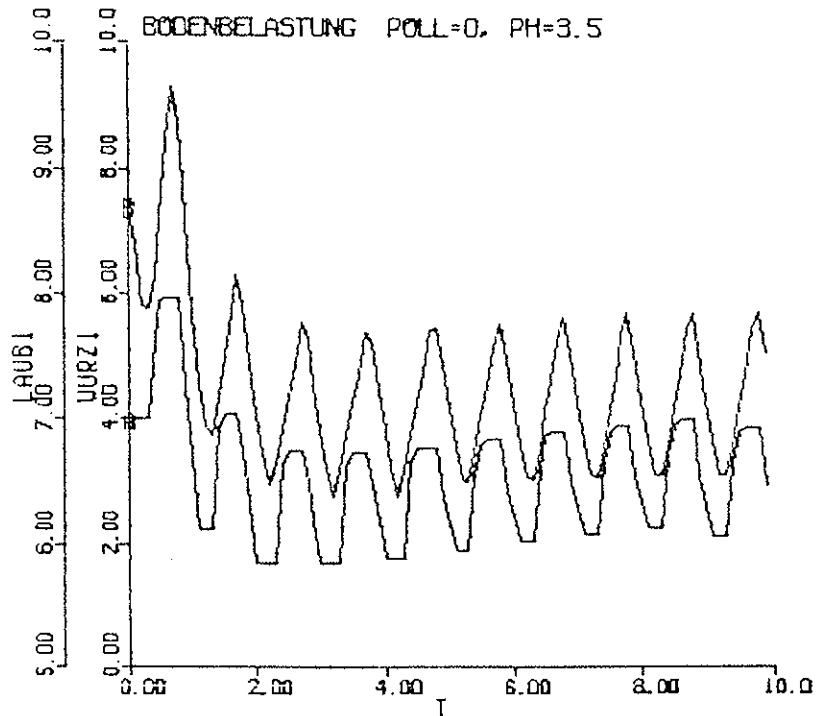
Luftbelastung 2

$35 \mu\text{gSO}_2/\text{m}^3$ Luft führen nach 2.5 Jahren zum Tod des Baumes, wobei zuerst die Laubmenge stark abnimmt und anschließend die Feinwurzelmenge.

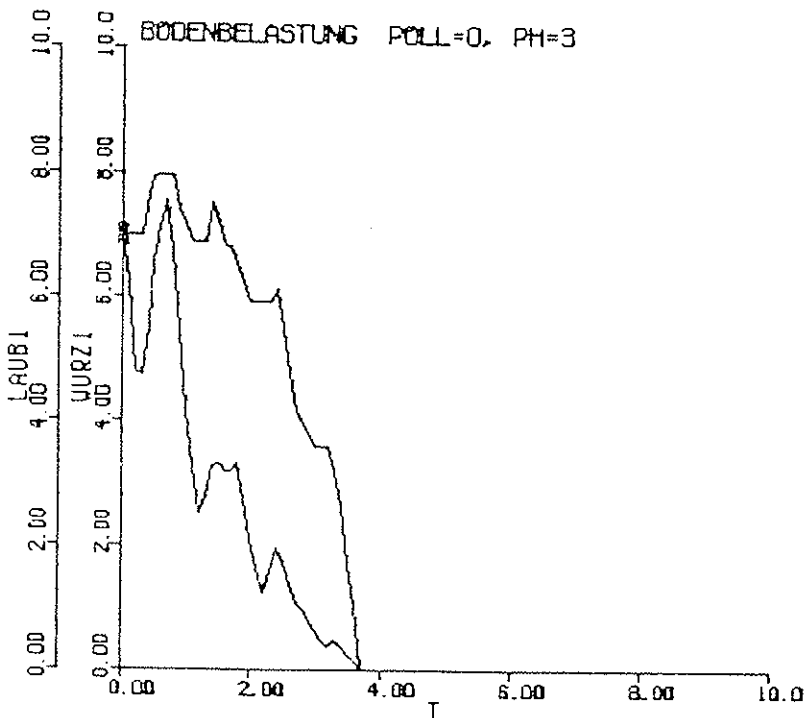


Bodenbelastung 1

Hier wird ausschließlich der pH-Wert des Bodens gesenkt, also übersäuert. Bei pH=3.5 pendeln sich die Laub- und Feinwurzelmenge bei einem niedrigeren Niveau ein. Beachtenswert ist, daß der saisonale Unterschied der Mengen geringer wird (also die Differenz zwischen Höchst- und Tiefststand).



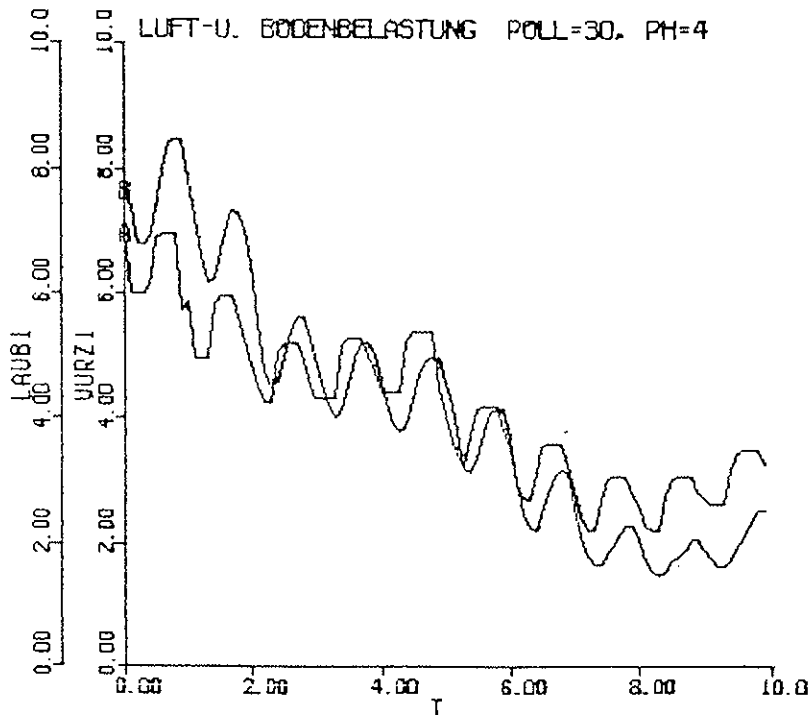
Bodenbelastung 2



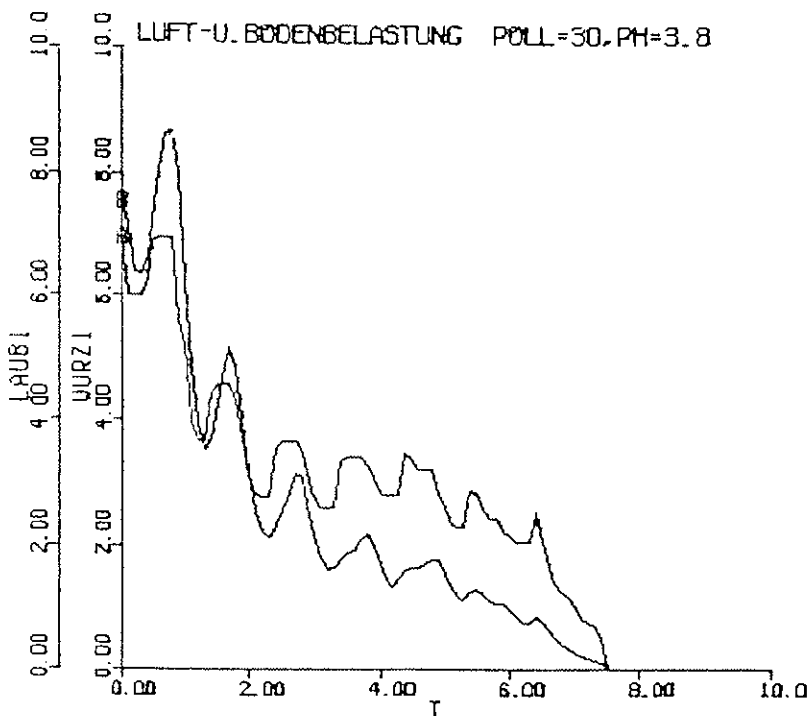
Ein pH-Wert von 3 führt nach fast 4 Jahren zum Tod des Baumes. Hier wird zuerst die Feinwurzelmenge rasch kleiner und erst später die Laubmenge.

Kombinierte Luft- und Bodenbelastung 1

Bei gleichzeitiger Belastung von Luft und Boden ergibt sich ein niedrigeres Niveau als bei den Einzelbelastungen, auch ist das periodische Verhalten nicht mehr so gut erkennbar.



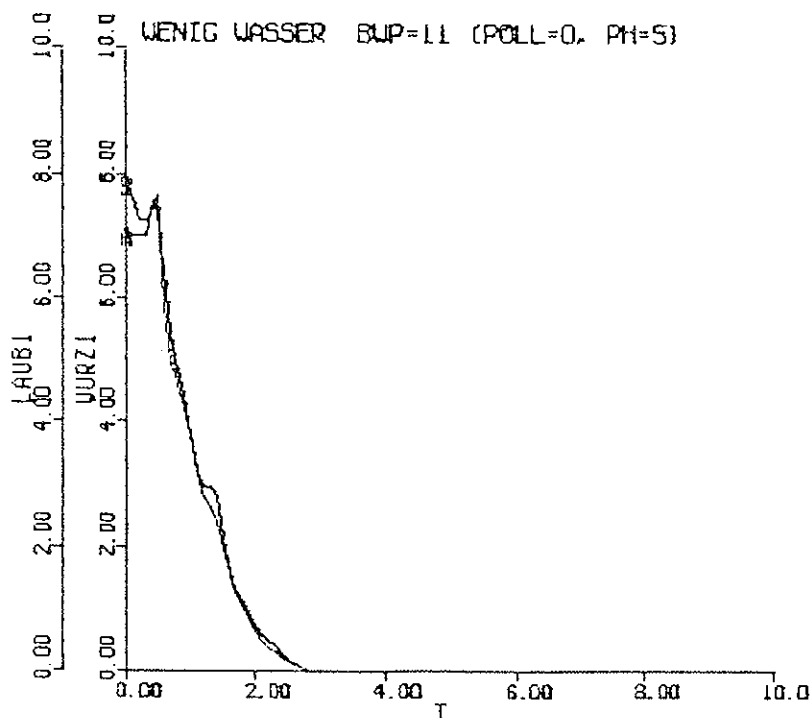
Kombinierte Luft- und Bodenbelastung 2



Weit geringere Einzelbelastung führt in Kombination schon zum Tod des Baumes, wobei im Mittelteil der Rechnung (2. - 6. Jahr) eine scheinbare Erholung des Baumes auftritt. Dieses Verhalten kann man aus der Erfahrung als gerechtfertigt anerkennen.

Wenig Wasser

Das Bodenwasserpotential BWP wurde auf 11 bar gesetzt, also ein extremer Wassermangel simuliert, der auch tatsächlich ohne weitere Belastung zum Absterben des Baumes führt. Laub- und Feinwurzelmenge gehen im Gleichklang zurück.



Modellkritik

Außer dem Umstand, daß das Modell qualitative plausible Ergebnisse liefert, wurde es keiner Gültigkeitsprüfung unterzogen. Es ist also noch die Frage offen, ob das Waldsterben (besser eigentlich: Baumsterben, da man nicht von einer epidemischen Krankheit, sondern nur von einem krankhaften Verhalten sprechen kann, das immer mehr Bäume zeigen) wirklich in erster Linie durch den pH-Wert des Bodens und die SO₂-Belastung der Luft gesteuert werden. Außerdem fehlen wichtige Kreisläufe, wie z.B. der Wasserhaushalt. Wie die Erfahrungen des Winters 1984/85 gezeigt haben, beschleunigt ein sehr strenger Winter die Waldschäden durch die Umweltbelastung sehr stark; dieses Verhalten läßt sich aus diesem Modell allerdings nicht erkennen.

Verwendbar wäre das Modell im Rahmen einer umfangreichen Strategie, in der man mittels Landkarten, auf denen die Schädigungsgrade des Waldsterbens verzeichnet sind (unter Berücksichtigung der Bodenbeschaffenheit (Erosion)), die über einen größeren Zeitbereich vorhanden sind, verschiedene Modelle testet. Die Vorgangsweise wäre dabei folgende: Man nimmt die Landkarte mit dem Zustand von vor 10 Jahren, wendet darauf die jeweilige Hypothese (Modell) an und vergleicht den Zustand heute mit dem Simulationsergebnis. Im Rahmen eines solchen Projekts könnte das vorliegende Modell sicher als Hypothese verwendet werden.

STIMULATION DES GLEICHGEWICHTSSINNES MIT EINEM ANALOGRECHNER

F. Rattay

Institut für Technische Mathematik, Technische Universität Wien

H. Mark

Facharzt für Hals-, Nasen- und Ohrenkrankheiten, Wien

Das Gleichgewichtsorgan (Vestibularapparat) liegt im inneren Ohr und besteht unter anderem aus je drei rechtwinkelig aufeinander stehenden Bogengängen, die von einer Flüssigkeit, der Endolymphe, erfüllt sind. Bei einer Drehbewegung des Kopfes kommt es als Folge der Trägheit der Endolymphe zumindest in einem der Bogengänge zu einer Flüssigkeitsströmung und die dadurch verursachten Bewegungen feiner Härchen der Sinneszellen bewirken eine Erregung des Gleichgewichtsnerfs (nervus vestibularis).

Bei längerem gleichmäßigem Drehen des Körpers wird infolge der Reibung die Endolymphe mitgenommen und die Sinneshärchen stellen sich wieder in ihre Ruhestellung ein. Ein plötzliches Stillhalten führt zum "Drehnachschwindel". Ein momentaner lokaler Temperaturunterschied in der Endolymphe ruft ebenfalls eine Flüssigkeitsbewegung hervor und dies kann z.B. durch Spülung eines Ohres mit kaltem Wasser bewirkt werden.

Bei Drehung des Kopfes fixieren wir einzelne Punkte der Umgebung immer nur kurz mit den Augen und rücken dann rasch zum nächsten Punkt weiter, sodaß sich die Augen ruckartig mit einer langsamen und einer raschen Phase hin und her bewegen. Dieselben Augenbewegungen werden auch vom Vestibulariskerngebiet im Hirnstamm über das Kerngebiet der Augenmuskeln reflektorisch ausgelöst (=Nystagmus).

Dieser Nystagmus dient unter anderem dem Mediziner zum Nachweis des Drehschwindels: dreht man eine Versuchsperson eine Zeit lang möglichst gleichmäßig auf einem Drehsessel nach links und blockiert dann plötzlich die Drehung, so kann für einige Sekunden der von dem Drehschwindel ausgelöste sogenannte postrotatorische oder Nachnystagmus nach rechts beobachtet werden, das heißt die Augen zucken schnell nach rechts, gehen langsamer wieder zurück, zucken wieder usw. Eine andere Methode zum Nachweis des Drehschwindels ist der Unterbergersche Tretversuch: mit geschlossenen Augen und vorgestreckten Händen versucht die "schwindelige" Versuchsperson auf der Stelle zu treten. Der oben beschriebene Drehversuch führt zu einer Abweichung nach links.

Auch die Kaltspülung führt zu einem Schwindel und Kaltspülung im linken Ohr hat die gleiche Wirkung wie der obige Drehversuch und verursacht einen "Unterberger" nach links und einen Nystagmus nach rechts. Eine genauere Darstellung des Gleichgewichtsorgans befindet sich in /1/.

Elektrische Stimulation

Das Gleichgewichtsorgan kann auch elektrisch stimuliert werden. Bei unseren Versuchen geschah dies mittels dreier auf die Haut aufgesetzter Elektroden. Zwei Elektroden werden hinter den beiden Ohren befestigt, während die Masselektrode im Bereich des oberen Teiles des Brustbeins aufgebracht wird.

Zunächst sollte nun versucht werden, mit welchen Stromsignalen der Vestibularapparat am leichtesten zu reizen sei und wie sich eine Phasenverschiebung der an den beiden Ohrelektroden einfließenden Ströme auswirkt. Für diese Aufgabe war der Analogrechner wegen seiner universellen Verwendbarkeit ein geeignetes Hilfsmittel, wenn auch die Ausgangsspannung des verwendeten EAI 1000 Tischrechners mit $\pm 5V$ zu gering für die elektrische Reizung ist. Es mußte eine Verstärkung in dem Bereich $\pm 15V$ für die beiden Stimulationselektroden erfolgen.

Da eine "rechteckige" Pulsanregung bereits eine unangenehme Hautreizung erzeugt, bevor noch der Schwellwert für die Stimulation des Vestibularapparates erreicht wird, wurde diesem ein einfaches Tiefpaßfilter 1. Ordnung /2/ nachgeschaltet, sodaß die Elektroden mit einer Spannung gemäß Abb. 1 versorgt wurden.

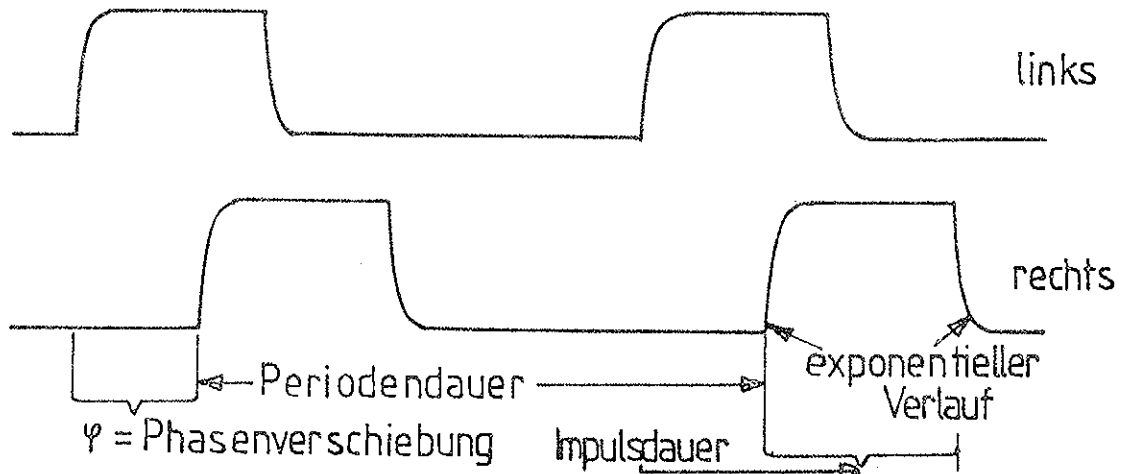


Abbildung 1
Spannungsverläufe an den stimulierenden Elektroden

Als günstige Periodendauer ergaben sich 250 msec. Obwohl Stimulation des Gleichgewichtssinnes mit einseitiger Reizung, also nur an einem Ohr, möglich ist, läßt sich bei beidseitiger Anregung mit $\phi=0^\circ$, d.h. ohne Phasenverschiebung kein Schwindel erzeugen. Die niedrigste Schwelle zur Stimulation des Vestibularapparates ergibt sich für $\phi=180^\circ$.

Verändern der Impulsdauer zeigt für kurze Impulse ein notwendiges Anheben der Spannung, um die Reizschwelle zu erreichen. Am einfachsten ist daher die Reizung mit der Impulsdauer gleich der halben Periodendauer und $\phi=180^\circ$. Es ist nun naheliegend, das rechte Ohr mit dem inversen Signal des linken Ohres zu versorgen, und diese Anordnung reduziert die notwendige Spannung für die vestibuläre Reizschwelle und kann als optimal angesehen werden.

Die Reizung wurde nun stets mit dieser "optimalen" Spannungsversorgung durchgeführt. Positive Impulse an der linken Elektrode und negative an der rechten bewirken bei überschwelliger Anregung eine Abweichung im Unterbergerschen Tretversuch nach rechts.

Nystagmusregistrierung

Da die Netzhaut gegenüber der Hornhaut negativ geladen ist, bildet das menschliche Auge einen elektrischen Dipol. Diese Eigenschaft läßt sich ausnützen, um Augenbewegungen am Oszillographen sichtbar zu machen oder sie in der Elektronystagmographie aufzuzeichnen. Dazu werden zwei Elektroden im einfachsten Fall links und rechts eines Auges angebracht und die schwache Potentialdifferenz von ca. 4-12 μV pro Grad Augenbewegung wird zunächst vorverstärkt, um dann am Oszillographen sichtbar zu werden /1/. Auch diese Vorverstärkung konnte mit den Operationsverstärkern des Analogrechners durchgeführt werden und damit wurden sowohl normale Augenbewegung als auch Nystagmus registrierbar.

Leider ist bei vestibularer Elektrostimulation der Einfluß des Stimulations-
signals an den an den Augen liegenden Meßelektroden etwa 50mal größer als bei
starkem Nystagmus. Da sich aber das Stimulationssignal der Abb. 1 dabei auch
noch verzerrt abbildet und sich außerdem größere Genauigkeitsanforderungen
ergeben, reicht dazu der Rechner EAI 1000 nicht aus und man müßte die
Differenzbildung mit einem Funktionsgeber auf dem EAI 680 oder am SIMSTAR
ausführen.

In der zur Verfügung stehenden Zeit konnte nicht festgestellt werden, ob der bei
Elektrostimulation entstehende Nystagmus synchron zur Reizung erfolgt.

Literatur

1. H. Scherer: Das Gleichgewicht. Springer (1984)
2. F. Rattay: Filtersimulation. INTERFACE 22 (1985)

Hybridrechenanlage
EDV-Zentrum
Technische Universität Wien

BENUTZERVERSAMMLUNG

Mittwoch, 26. November 1986, 14 Uhr

Die nächste Benutzerversammlung findet am 26. November 1986 um
14 Uhr im Seminarraum Physikalische Elektronik, 1040 Wien,
Gußhausstraße 27-29, 4. Stock, Raum CD 04 04, statt.

In Kurzvorträgen und Diskussionen sollen u.a. die folgenden
Themen angesprochen werden:

- Die Verwendung der interaktiven Simulationssprache HYBSYS am
EAI PACER 600A und am EAI SIMSTAR
- Umstellungshilfen für eine Übertragung von alten Modellen auf
den EAI SIMSTAR
- Unterstützung für die Modellerstellung
- ACSL an der CYBER 860 und auf IBM kompatiblen PCs
- Übersiedlung der Hybridrechenanlage auf die Freihausgründe
- Geplante neue Simulationswerkzeuge an der Hybridrechenanlage

Alle bisherigen Benutzer der Hybridrechenanlage sowie alle, die
an der Verwendung von spezieller Hard- und Software bei der
Lösung von Simulationsaufgaben interessiert sind, werden herz-
lich eingeladen, an dieser Benutzerversammlung teilzunehmen.

NEUIGKEITEN VOM BETRIEB AM HYBRIDRECHENZENTRUM

F. Blöser
Hybridrechenanlage, TU Wien

NEUE SOFTWARE UND HARDWARE

Am Simulationsmultiprozessor EAI SIMSTAR steht nun die Version HYBSYS PTRAN des hybriden Simulationssystems HYBSYS für die Simulation kontinuierlicher Systeme zur Verfügung. Es handelt sich dabei um ein Experimentierwerkzeug, das durch Kombination der Modellbeschreibungssprache PTRAN und der Experimentiersprache HYBSYS die Verwendung des SIMSTAR Simulationssystems ermöglicht (s. Seite 3).

Am EAI PACER 600 steht weiterhin die Version 5 TS der interaktiven Experimentiersprache HYBSYS für hybride Simulationen zur Verfügung. Einerseits als Ersatz für die Pertec Disk Drives, andererseits um den vermehrten Massenspeicherbedarf abzudecken, ist seit längerer Zeit ein CDC Lark Drive 9457 mit einer Speicherkapazität von 50 Megabytes installiert. Das entsprechende Interface zum PACER wurde an der Hybridrechenanlage selbst entwickelt.

ÖFFNUNGS- UND BETRIEBSZEITEN

Die Hybridrechenanlage ist

Montag bis Donnerstag von 8 - 18 Uhr,
Freitag von 8 - 16 Uhr

geöffnet. Während dieser Zeiten kann am PACER 600 uneingeschränkt gerechnet werden. Das SIMSTAR Simulationssystem steht von Montag bis Freitag täglich von 8 bis 14 Uhr für die Benutzer zur Verfügung, wobei auch die Möglichkeit der Reservierung beim Operator besteht (Tel. 588 01 - 3706 od. 3669 DW).

TERMINAL- UND TUNET-ANSCHLÜSSE

Der Zugang zu den Rechenanlagen des Hybridrechenzentrums ist über direkte Terminalanschlüsse, TUNET- und Modem-Anschlüsse möglich.

Ein am TUNET angeschlossenes graphisches Terminal HP 2623A befindet sich im Terminalraum der Hybridrechenanlage (Zimmer CA 0434), der während der oben angeführten Öffnungszeiten für Benutzer zugänglich ist. Damit kann man sowohl an den Rechenanlagen des Hybridrechenzentrums rechnen als auch die Simulationssprache ACSL an der CYBER 180-860 verwenden. Das Terminal hängt am Server DIG3, die Anwahl der Rechner erfolgt mittels der Prozeduraufrufe

DO HYB.EOFF für den PACER 600,
DO SIM.EOFF für den SIMSTAR,
DO NOS2.EON für die CYBER 860.

Das Terminal im Terminalraum steht den Benutzern der Hybridrechenanlage während der Öffnungszeiten zur Verfügung. In Zeiten verstärkten Andrangs kann eine Reservierung beim Operator (Tel. 588 01 - 3706 od. 3669 DW) vorgenommen werden.

Über das TUNET sind die Rechenanlagen PACER und SIMSTAR des Hybridrechenzentrums auch von jedem anderen ans TUNET angeschlossenen Terminal erreichbar.

Außerdem besteht die Möglichkeit der Verwendung der Rechner über Modem-Anschlüsse (300 Baud), sowohl über die TU-internen Hausklappen 194 und 195 als auch über die Wiener Telefonnummer 65 98 17. Dabei ist eine telefonische Kontaktnahme mit dem Operator zweckmäßig, da die Modems wahlweise an beide Rechner angeschlossen werden können.

Zusätzlich besteht ein DATEX-P Anschluß am SIMSTAR Simulationssystem.

UMSTELLUNG DES DIGITALTEILS

Im Zuge der schrittweisen Ersetzung des Hybridsystems PACER 600 durch das neue System SIMSTAR/TUNIX wird es mit Beginn des Sommersemesters 1987 zu Einschränkungen des Rechenbetriebs am PACER kommen. Während das hybride Rechnen mit HYBSYS weiter möglich sein wird, kommt es zu Beschränkungen von digitalen Programmen durch Wegfall bestimmter Peripheriegeräte:

Ab 1. März 1987 wird am Hybridrechenzentrum kein Card Reader und auch keine Magnetbandstation mehr verfügbar sein. Durch Wegfall der Bandstation wird es am PACER keine Backup Tapes und keine Filesicherung von seiten des Hybridrechenzentrums mehr geben. Die Plotterausgabe in ACSL wird bis dahin so umgestellt, daß die Ausgabe von Zeichnungen am Plotter des Digitalrechenzentrums erfolgt.

Außerdem ist es ab diesem Zeitpunkt durch Einschränkung der Plattenkapazität nicht mehr möglich, den Benutzern eigene Massenspeicher am PACER (Platten oder Districts) zur Verfügung zu stellen.

Es wird dann allerdings das Rechnersystem TUNIX als Frontend-System für den Simulationsmultiprozessor SIMSTAR verfügbar sein. Bei TUNIX handelt es sich um ein unter UNIX V.2 laufendes Mikrosystem auf der Basis des NS 32000 von National Semiconductor.

Bis zum 1. März 1987 besteht für Benutzer die Möglichkeit, Programme und Daten auf das neue System übertragen zu lassen. Dabei sind die Mitarbeiter des Hybridrechenzentrums gerne behilflich und stehen auch für weitere Informationen und die Beantwortung von Fragen in Zusammenhang mit den ab 1. März 1987 wirksamen Änderungen des Rechenbetriebs zur Verfügung.

In der Benutzerversammlung des Hybridrechenzentrums, die am 26. November 1986 um 14 Uhr im Seminarraum Physikalische Elektronik, 1040 Wien, Gußhausstraße 27-29, 4. Stock, Raum CD 04 04, abgehalten wird, werden für die Benutzer genauere Informationen über die Umstellung gegeben werden.

SIMULATIONSKURSE

Das Hybridrechenzentrum der TU Wien bietet dem Benutzer Soft- und Hardware zur Lösung verschiedenster Simulationsaufgaben an (Simulation technischer, ökologischer, biologischer etc. Prozesse). Angeboten und gewartet wird auch Simulationssoftware, die an anderen Rechnern der TU Wien installiert ist, z.B. ACSL (auf CYBER und PC) und SYSMOD (VAX); Installation weiterer Simulationssoftware (auch für diskrete Prozesse) ist geplant.

Um den Benutzern einen Einstieg in die verfügbare Simulationssoft- und Hardware zu bieten, veranstaltet die Abt. "Simulationstechnik" des Institutes für Technische Mathematik in Zusammenarbeit mit dem Hybridrechenzentrum von 1. bis 12. 12. 1986 "Simulationskurse". Drei abgeschlossene Kursteile, bestehend aus Vorträgen und praktischen Übungen, stellen verschiedene Simulationsmöglichkeiten vor.

Der erste Kursteil Simulation mit HYBSYS stellt die Möglichkeiten zur Simulation mit Hilfe der (hybriden) Simulationssprache HYBSYS vor. HYBSYS (HYBRID Simulation SYStem; entwickelt und implementiert am Hybridrechenzentrum der TU Wien) ist eine CSSL-Sprache und erlaubt Beschreibung und Simulation dynamischer Prozesse in komfortabler und hardware-unabhängiger Weise. Die Simulation erfolgt entweder auf dem Hybridrechner, wobei alle Vorteile der echt parallelen Arbeitsweise (auch bei Logik!) ausgenutzt werden, oder man verwendet eine digitale Version, die im Prinzip rechnerunabhängig ist. HYBSYS wird in neuer Form auch am SIMSTAR zur Verfügung stehen (siehe Kursteil III).

Der zweite Kursteil Digitale Simulation mit ACSL stellt die Simulationssprache ACSL (Advanced Continuous Simulation Language) vor. Implementationen stehen auf der CYBER des Digitalrechenzentrums und für kompatible PCs zur Verfügung, gewartet vom Hybridrechenzentrum. ACSL ist eine der meist-implementierten digitalen Simulationssprachen und bietet neben der Simulation und Dokumentation im Zeitbereich auch Ereignisbehandlung und Möglichkeiten für kombinierte Simulation sowie für Frequenzbereichsanalyse an.

Der dritte Kursteil Simulation am SIMSTAR stellt die Simulationsmöglichkeiten am neuen Simulationsrechner SIMSTAR vor. Dieser Hybridrechner neuester Technologie unterstützt wegen seiner echt parallelen Arbeitsweise insbesondere die Simulation von komplexen nichtlinearen Systemen mit stark unterschiedlichen Zeitkonstanten und Unstetigkeiten. Beschreibung und Experimentieren mit einem Modell erfolgen mit PTRAN (hybrides ACSL) und mit HYBSYS, weswegen für diesen Kursteil Kenntnisse von ACSL und HYBSYS günstig sind.

ANTWORTKARTE

Ich (wir) melde(n) mich (uns) an zu den Simulationskursen (Kursteilen):

Name(n): _____

Institut _____
/Firma _____

Adresse: _____

Dr.F.Breitenecker
Inst.f.Technische Mathematik
TU Wien
Wiedner Hauptstrasse 8-10
1040 Wien

Telefon: _____

Teil I	Teil II	Teil III
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Kursteil "Simulation mit HYBSYS" (Teil I)

Inhalt: Einführung - Aufbau von HYBSYS - Simulation mit HYBSYS - Tabellen - spezielle Makros und Fähigkeiten (Fallstudien)

Kurstermin: Montag, 1.12.1986 bis Donnerstag, 4.12.1986

Vortragender: DI.Dr.F.Rattay

Kursteil "Digitale Simulation in ACSL" (Teil II)

Inhalt: Einführung - Aufbau von ACSL - Simulation mit ACSL - Tabellen - Ereignisbehandlung in ACSL - spezielle Eigenschaften von ACSL - diskrete Simulation in ACSL - Optimierung in ACSL

Kurstermin: Dienstag, 9.12.1986 bis Donnerstag, 11.12.1986

Vortragender: Doz.DI.Dr.F.Breitenecker

Kursteil "Simulation am SIMSTAR" (Teil III)

Inhalt: Einführung - Modellbeschreibung in PTRAN (hybrides ACSL) - Simulation am SIMSTAR - HYBSYS und ACSL als Runtime-Interpreter - Simulationsfallstudien

Kurstermin: Freitag, 12.12.1986

Vortragender: Doz.DI.Dr.F.Breitenecker

Generelle Kurszeiten: Beginn täglich 14.00 Uhr, offenes Ende

Kursort: TU Wien, Besprechungszimmer des Institutes für Technische Mathematik (Raum DA 03 C22), 3.Stock, Wiedner Hauptstrasse 8-10, A-1040 Wien

Kursunterlagen: Unterlagen werden zur Verfügung gestellt

Kurskosten: keine

Zielgruppe: Studenten höherer Semester, Diplomanden, Dissertanten, Assistenten, beliebige Interessenten (auch außerhalb der TU)

Rückfragen: Dr.F.Breitenecker, Dr.F.Rattay, M.Schandl, Tel. 0222-58801
DW 5374, 5379, 3706

Bemerkungen: Die Kursteile können unabhängig voneinander belegt werden.
Die Kurse können auch als Vorlesung/Übung (101.162, 101.205, 101.216, 101.382, 101.393, 101.404) inskribiert und kolloquiert werden und sind anrechen- bzw. austauschbare Lehrveranstaltungen verschiedener Studienrichtungen
