
Interface

herausgegeben von
der Abt. Hybridrechenanlage
des Rechenzentrums
der Technischen Universität Wien

Nummer 10
November 1977



Konventionelle Anologschaltung -
überholt durch AutoPATCH

INHALTSVERZEICHNIS

	Seite
Zum Jubiläum	3
PACER 600A - AutoPATCH	4
Aktuelle Mitteilungen	8
Kurse	12
Source Processor	14
Graphische Ausgabe im HOI	15
Data Plotter als Hardcopy-Einheit für das Graphic Display	16
CYBER 74 Jobs	18
Job Accounting in JCS/TS 7	19
Magnetfeldberechnung für einen adiabatischen Neutronenspindrehler	22
Lösung der Wärmeleitungsgleichung mit Ortsdiskretisierung und Time-Sharing der analogen Komponenten	25
Einsatzmöglichkeiten eines Hybridsystems in Biologie und Medizin	36
EAI Users Group Meeting 1977	42
IMACS Symposium über Simulation von Regelungssystemen	43
RECON Informationssystem	44
SICOS - eine Programmiersprache zur Simulation dynamischer Systeme für die Rechenanlage NOVA 2/10	47
Pseudozufällige Binärsignale und ihre Eignung zur Identifikation linearer Eingrößensysteme	51
Automatische Auswertung von Verkehrszählungen am Hybridrechner	55
Ein Vergleich einiger Verfahren zur Merkmalsbestimmung für die Spracherkennung	57

Redaktion: Irmgard Husinsky

Eigentümer, Herausgeber, Verleger: Rechenzentrum der Technischen Universität Wien, Abteilung Hybridrechenanlage, Vervielfältigung: Österreichische Hochschülerschaft Technik, für den Inhalt verantwortlich: Dipl.Ing.W. Kleinert, alle: Gußhausstraße 27-29, 1040 Wien
 Telex: 76875 rzhw a

Das kleine Jubiläum zur Herausgabe der Nummer 10 der Zeitschrift INTERFACE wollen wir zum Anlaß für einen Rückblick über die bisherigen Nummern nehmen.

Als im Oktober 1974 die erste Nummer erschien, stand auf der ersten Seite über die Ziele der Herausgabe einer solchen Zeitschrift:

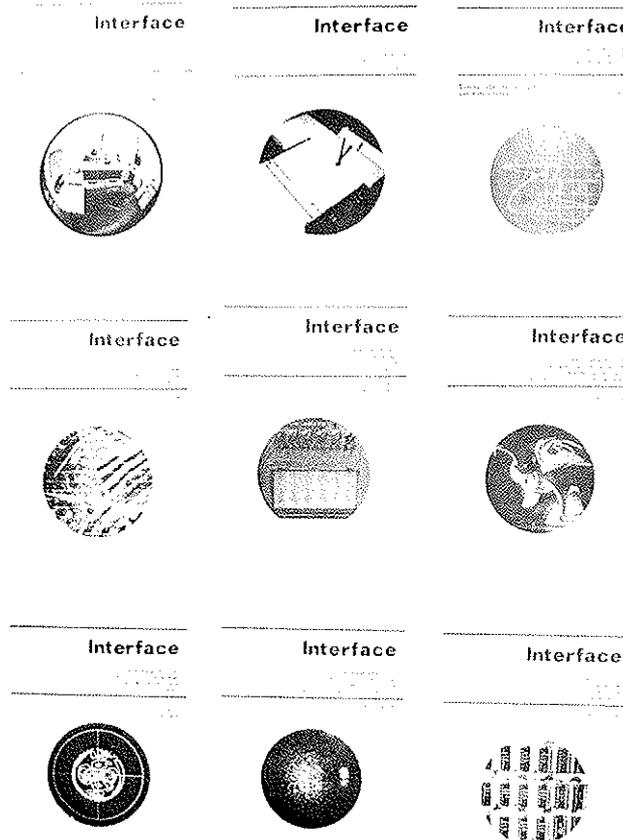
INTERFACE soll

- eine neue Verbindung (interface) zwischen der Hybridrechenanlage der TU Wien und dem Kreis der Benutzer herstellen,
- dem gegenseitigen Erfahrungsaustausch der Benutzer dienen,
- Möglichkeiten und Einsatzgebiete des zur Zeit einzigen in Österreich installierten Hybridsystems aufzeigen.

In den letzten drei Jahren fand INTERFACE großes Interesse im In- und Ausland. Es ist die einzige Publikation dieser Art an europäischen Hybridrechenanlagen.

Deshalb sind wir auch gerne bereit, Berichte über interessante Projekte, die an anderen Hybridrechnern behandelt wurden, zu veröffentlichen.

INTERFACE hat jetzt eine Auflage von etwa 2000 Stück. Es wird an alle Benutzer des Rechenzentrums und an alle Institute der TU Wien, an Institute anderer österreichischer Universitäten und an Interessenten in Industrie und Wirtschaft versandt.



Ein großer Interessentenkreis befindet sich in Deutschland (Universitäten und Hybridrechenanlagen). Ferner erhalten Rechenzentren in Europa, an denen EAI Hybridrechner stehen, unsere Zeitschrift. Unsere internationalen Kontakte (z.B. EAI Users Group Meeting) haben gezeigt, daß auch in nicht-deutschsprachigen Ländern Europas unsere Zeitschrift mit großem Interesse verfolgt wird. Aus diesem Grund werden wir ab dieser Nummer bei Berichten über interessante Anwendungen des Hybridrechners auch eine kurze englische Zusammenfassung bringen.

Im folgenden soll ein kurzer Überblick über den Inhalt der Nummern 1 bis 9 gegeben werden:

- aktuelle Benutzerinformationen, Kursankündigungen, Vorstellung neuer Geräte,
- Programmieranleitungen für spezielle Problembereiche, Beschreibung von Programmpaketten,
- Berichte über Softwareentwicklungen (Betriebssystem JCS/MP 6 und JCS/TS 7, Neuzugänge der Programmbibliothek, etc.),
- Berichte über Hardwareeigenentwicklungen,
- allgemeine Artikel über Einsatzmöglichkeiten von Hybridrechnern und Simulationen, Zukunft des Hybridrechners (Auto-Patch),
- Berichte über Projekte, die am Hybridrechner der TU Wien durchgeführt wurden (21 Artikel aus Physik, Regelungstechnik, Elektrotechnik, Maschinenbau, etc.).

Wir werden uns auch in Zukunft bemühen, INTERFACE interessant zu gestalten. Der Firma EAI danken wir für die finanzielle Unterstützung der Jubiläumnummer.

Irmgard Husinsky
Redaktion

PACER 600A - AutoPATCH

In der letzten Nummer von INTERFACE wurde allgemein auf die AutoPATCH Erweiterung des PACER 600 Hybridrechners eingegangen, welche seit kurzer Zeit von EAI angeboten wird. Diesmal soll die Bauweise eines solchen AutoPATCH Hybridsystems genauer beschrieben werden. Der vorliegende Artikel wurde nach dem Aufsatz "PACER 600 AutoPATCH Computing System" von J.E. Mawson mit freundlicher Genehmigung der Firma EAI gestaltet.

1. Das AutoPATCH Hardware System

Die EAI AutoPATCH Hardware besteht im wesentlichen aus drei Systemen:

- o Das AutoPATCH Panel, mit dem eine von der Schaltmatrix benötigte grundlegende Konfiguration analoger Komponenten erstellt wird (firmwired),
- o Die analoge Schaltmatrix, mit der die problemspezifischen benötigten Verbindungen analoger Komponenten softwareweise gesteuert werden,
- o Der Parallel Logic Processor zur softwareweisen Definition paralleler logischer Funktionen.

Weiters wird das herkömmliche Analogsystem erweitert durch:

- o programmierbare DCAs,
- o einzelprogrammierbare Zeitkonstante bei allen Integrierern,
- o programmierbare Rückführwiderstände (=Verstärkung) bei Summierern,
- o digital setzbare Funktionsgeber bzw. Multi-Variable-Funktionsgeber.

2. Das AutoPATCH Panel - AutoPATCH Computing Elements

Bisher war das Steckbrett eine flexible Verbindung zu allen Rechenkomponenten des 600 Analogrechners. Da eine Schaltmatrix, mit der alle möglichen Schaltverbindungen erstellt werden können, zu umfangreich wird, muß die Anzahl der möglichen Verbindungen auf sinnvolle Weise reduziert werden. Dies wird durch folgende Vorgangsweise erreicht:

- Von den am Steckbrett mehrfach vorhandenen Ausgängen eines Verstärkers geht nur ein Ausgang in die Schaltmatrix ein, die Verzweigung zu mehreren Elementen wird in der Schaltmatrix durchgeführt.
- Die bisher fix verdrahteten Verbindungen Verstärkerausgang - Potentiometereingang sowie die variablen Verbindungsmöglichkeiten Potentiometer-Verstärker wurden eliminiert, indem jeder Eingang, bei dem ein Potentiometer sinnvoll ist, mit einem Pot fest verbunden wird.
- Die Anzahl der möglichen Eingänge wird bei Summierern mit 2 bzw. 3 beschränkt, bei Integrierern wird nur mehr ein Eingang zugelassen (Elimination der Summation).
- Die Verwendung verschiedener Eingangswiderstände bei Integrierern und Summierern wird durch eine variabel programmierbare Rückführung des Verstärkers überflüssig.

Auf diese Weise werden Standard Rechenelemente definiert, die nun mit dem AutoPATCH Panel firmwired realisiert werden, d.h. daß die Definition der Standard Elemente auf bestimmte Problemkreise zugeschnitten werden kann. Die Information über die Analogkomponenten wird an die Software weitergegeben.

Abbildung 1 zeigt einige der für das AutoPATCH neu definierten Rechenelemente. A, B und C sind Problemparameter, die beliebige reelle Zahlen sein können, während a, b und c maschinenabhängige Parameter sind, die sich aus den Problemparametern A, B und C, den problembedingten Skalierungsfaktoren und der Verstärkung der analogen Komponenten ergeben und die im Bereich (-1,+1) bzw. (0,+1) liegen müssen.

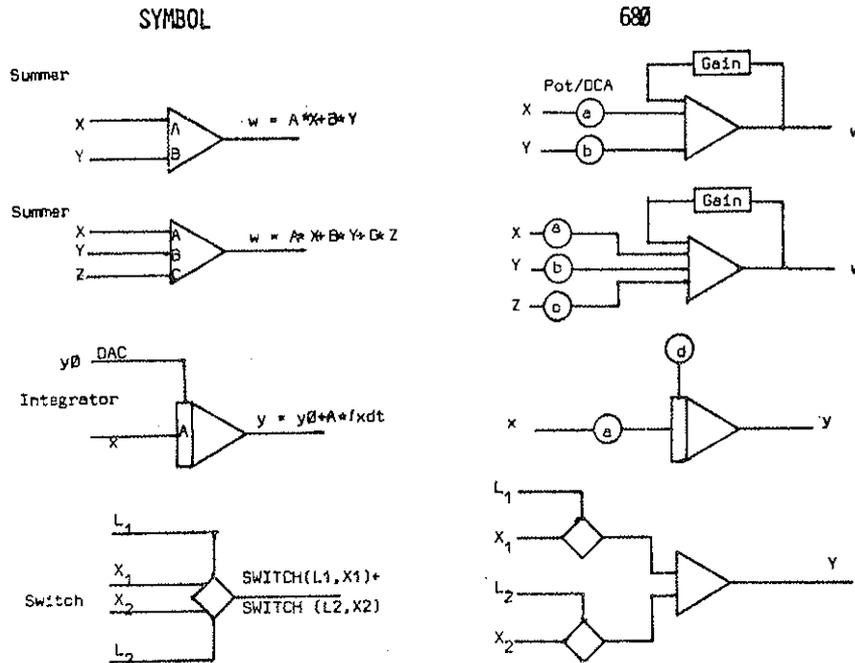


Abbildung 1
AutoPATCH Computing Elements

Der wesentliche Vorteil dieses Konzeptes liegt darin, daß

- die benötigten Schaltverbindungen reduziert werden,
- die Standard Rechenelemente auf bestimmte Problemkreise abgestimmt, bzw. neu definiert werden können,
- durch Austauschen des AutoPATCH Panel gegen ein herkömmliches Steckbrett der Analogrechner wie bisher benutzt werden kann.

3. Die Schaltmatrix

Als Schaltmatrix wird ein 3-stufiges Verbindungsnetzwerk verwendet, das eine hohe Effektivität bei geringem Aufwand gewährleistet. Abbildung 2 zeigt das Schema einer 3-stufigen Schaltmatrix mit N Eingängen und M Ausgängen, die durch das Tupel (n, X, Y, m, Z) definiert ist, wobei

- n Anzahl der Eingänge je Eingangsblock,
- X Anzahl der Eingangsblöcke der Größe $n \cdot X$,
- $N = n \cdot X$ Eingänge insgesamt,
- Y Anzahl der Mittelblöcke der Größe $X \cdot Z$,
- m Anzahl der Ausgänge je Ausgangsblock,
- Z Anzahl der Ausgangsblöcke der Größe $Y \cdot m$,
- $M = Z \cdot m$ Ausgänge insgesamt.

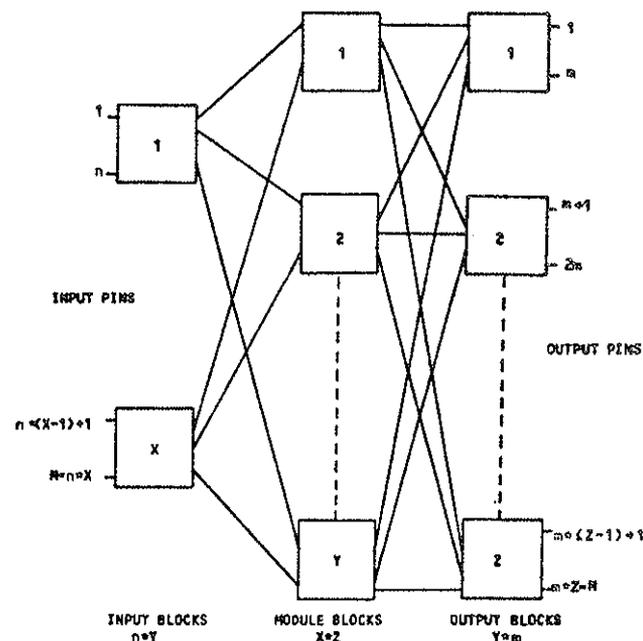


Abbildung 2
3-stufige Schaltmatrix

Zwischen jedem Eingangs- und Mittelblock gibt es nur eine Verbindung, ebenso zwischen jedem Mittel- und Ausgangsblock. Die Verbindung von einem Eingang mit einem Ausgang wird daher im wesentlichen durch einen Mittelblock definiert. Da dieser jedoch mit jedem Ein- und Ausgangsblock nur einmal verbunden ist, kann es vorkommen, daß eine benötigte Verbindung bereits blockiert ist. Um die Wahrscheinlichkeit des Auftretens einer solchen Situation möglichst gering zu halten, werden die Ein- bzw. Ausgänge der Ein- und Ausgangsblöcke nach folgenden Gesichtspunkten definiert:

- Externe Ein- und Ausgänge sowie selten benutzte Elemente werden in den Ein- und Ausgangsblöcken gleich verteilt.
- Die Eingänge von Komponenten mit zwei oder mehr Eingängen werden auf verschiedene Ausgangsblöcke verteilt.
- Komponenten desselben Types werden auf verschiedene Eingangsblöcke verteilt.
- Die vorhandenen Rechenelemente werden so verteilt, daß sie bei einer Reduzierung der Schaltmatrix entsprechend ihrem Typ proportional reduziert werden.

Abbildung 3 zeigt die auf diese Überlegungen aufgebaute Struktur der Ein- und Ausgangsblöcke.

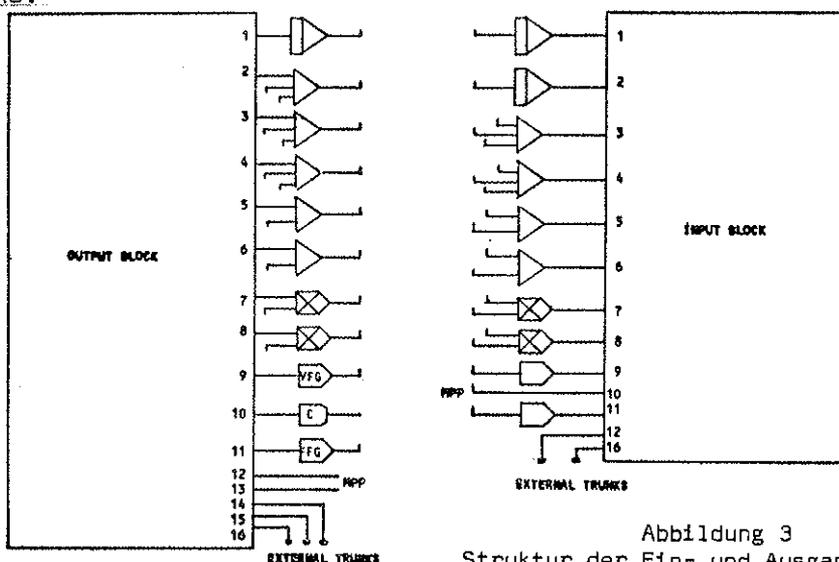


Abbildung 3
Struktur der Ein- und Ausgangsblöcke

Abbildung 4 zeigt die insgesamt möglichen Schaltverbindungen des PACER 600 AutoPATCH Systems.

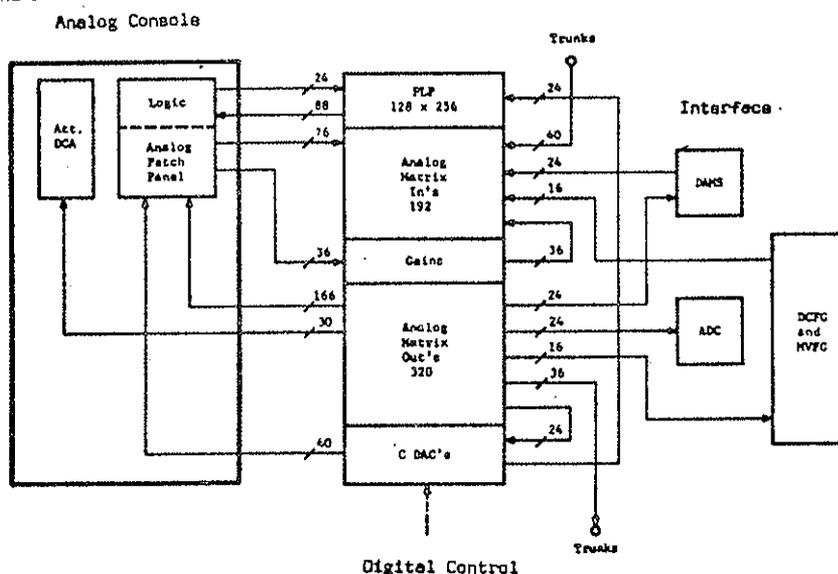


Abbildung 4
Mögliche Schaltverbindungen des AutoPATCH Systems

4. Der Parallel Logic Processor

Der Parallel Logic Processor (PLP) ermöglicht die Realisierung paralleler logischer Funktionen mit kombinatorischen und sequentiellen Eigenschaften, die mit logischen Operatoren wie AND, OR, NOT etc. beschrieben werden können. Ein Logic Statement Processor (LSP) übersetzt diese Funktionen und erstellt eine Wahrheitstafel, die in den Parallel Logic Processor geladen wird, der weiters über eine 3-stufige Schaltmatrix die benötigten Verbindungen zu den Interface-Komponenten des Analogrechners herstellt. Abbildung 5 zeigt die Struktur eines voll ausgebauten PLP.

Zusammengefaßt bestehen die wesentlichen Unterschiede und Vorteile des PLP gegenüber der Verwendung der parallelen Logik des Analogrechners darin, daß

- die logischen Funktionen mit logischen Operatoren auf Problemebene beschrieben werden können,
- Boolesche Operationen nicht mehr mit den Komponenten der parallelen Logik, sondern mit Booleschen Funktionsgebern durchgeführt werden, die aus Random Access Memory (RAM) Blöcken bestehen,
- keine Komponenten der parallelen Logik verwendet werden, Monostabile und logische Differentiatoren werden mit Zählern simuliert.

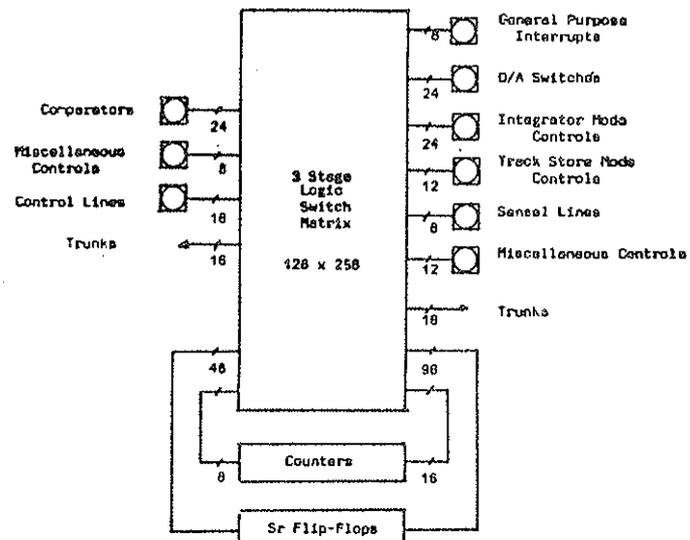


Abbildung 5
Parallel Logic Processor

5. Ausblick

Die Entwicklung des AutoPATCH und des Parallel Logic Processors bedeuten nicht nur eine enorme Verkürzung des Programmieraufwandes, sondern gleichzeitig den Beginn einer neuen Ära der hybriden Rechentechnik, die eine Reihe bisher verschlossener Möglichkeiten eröffnet:

- o Bedienung des Analogrechners auf Problemebene
- o Möglichkeit der Speicherung des momentanen Zustandes des Simulationsmodells, und einfaches und schnelles Wiederherstellen der gespeicherten Schaltung, dadurch leicht auf andere Rechenanlagen transportierbar
- o Ständig abrufbare Dokumentation des momentanen Modellzustandes
- o Automatisches Auffinden und Verwalten fehlerhafter Komponenten
- o Entwicklung problemorientierter hybrider Bibliotheks-Software

Insbesondere durch Entwicklung geeigneter Software für bestimmte Problemklassen kann die vom Benutzer geforderte Kenntnis über die Rechentechnik soweit reduziert werden, daß ein hybrider Terminalbetrieb auf Problemebene ermöglicht wird, wobei hybrides Time Sharing nicht auszuschließen ist.

D. Solar

aktuelle mitteilungen

ÖFFNUNGSZEITEN:

Die Hybridrechenanlage ist von

Montag bis Freitag von 8 Uhr bis 20 Uhr

geöffnet. Die Closed-Shop-Eingabe ist zu jeder vollen Stunde.

NEUE SUBROUTINE N O T E:

Mit der Subroutine NOTE können vom Programm aus Mitteilungen an den Operator erfolgen. Die Wirkung ist die gleiche wie bei der NOTE-Option auf einer Steuerkarte: NOTE bewirkt das Ausschreiben eines Textes auf der Operatorkonsole. Das Programm wird gestoppt und erst dann, wenn der Operator die Meldung zur Kenntnis genommen hat, von diesem fortgesetzt.

Verwendung: CALL NOTE (TEXT)

wobei TEXT ein Feld beliebigen Typs mit gepacktem Text (INTEGER: A2-Format) oder eine Hollerithkonstante ist. Das Ende des Textes wird durch das Zeichen \ angezeigt.

Beispiel: CALL NOTE (36HBITTE PAPER TAPE READER EINSCHALTEN\)

NOTE ist in der Systembibliothek, es ist keine LOAD-Karte nötig.

RECORDERSTEUERUNG:

Der 6-Kanalschreiber kann wieder vom Analogrechner gesteuert werden. Beschreibungen sind in der Programmberatung erhältlich. Insbesondere kann die Start/Stop-Steuerung über die Logik erfolgen.

NEUE RTL :

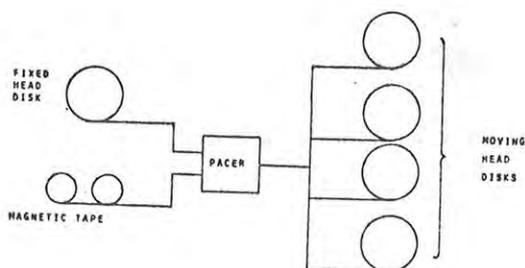
Die auf den Files JCSRTL, NONFPP, DC/FPP und DC/NON zusammengefaßten Runtime-Library-Routinen wurden überarbeitet und Fehler in einigen Routinen behoben (z.B. SIN-Routine, Ein/Ausgabe auf Lochstreifen). Vor allen Dingen aber wurden die Runtime-Fehlermeldungen verbessert und auf Platte abgespeichert. Die Ausgabe dieser Meldungen erfolgt nun direkt über das Betriebssystem. Dadurch konnte der benötigte Kernspeicherplatz für die RTL Routinen wesentlich verringert werden.

MAGNETPLATTENSPEICHER-ERWEITERUNG:

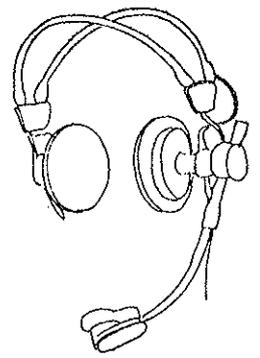
Ende Oktober wurde der Hybridrechner PACER 600 durch die Installation einer zweiten Dual-Magnetplattenspeichereinheit von PERTEC in seiner Massenspeicherkapazität wesentlich erweitert. Die neue Einheit besteht wie die bisher vorhandene aus einer fest eingebauten und einer auswechselbaren Magnetplatte. Die Massenspeicherkapazität wurde um 2286 K-Worte auf insgesamt 4933 K-Worte Plattenspeicher bzw. 7183 K-Worte insgesamt (mit Magnetband) erhöht.

Für das Betriebssystem und die Benutzer ergaben sich keine Veränderungen. Es werden aber 27 statt bisher 21 Districts angesprochen. Es können daher ab sofort für permanente Files von allen Benutzern die Districts 19-24 programmiert werden. Benutzer, die durch Rechenzeitreservierung im OPEN-SHOP-Betrieb an der Anlage arbeiten, können zusätzlich die Districts 7,8,10,15,17 und 27 auf ihrer eigenen Magnetplatte ansprechen. Die Einteilung von eigenen Platten in Districts und Vergabe derer ist direkt an der Hybridrechenanlage zu erfragen.

MASSENSPEICHER DES PACER 600



NEU: KOPFHÖRER-MIKROFON-KOMBINATION AKG K 158



Technische Daten:

Kopfhörer (dynamischer Wandler):

Übertragungsbereich: 25 - 14000 Hz

Normaler Leistungsbedarf: 2 mW für einen Sprechpegel
von 112 db bei 1000 Hz, entsprechend 775 mV

Elektrische Impedanz: 300 Ohm

Mikrofon (dynamischer Druckgradientenempfänger):

Übertragungsbereich: 100 - 12000 Hz

Impedanz bei 1000 Hz: 230 Ohm

Grenzschalldruck für einen Klirrfaktor von 0.5% (1000 Hz):
500 μ bar

Diese Kopfhörer-Mikrofon-Kombination wurde bei den Arbeiten zum Vergleich einiger Verfahren der Merkmalbestimmung zur Spracherkennung (siehe auch Benutzerforum, Seite 57) verwendet.

NEUE DRUCKSORTEN:

Die "Benützung des Betriebssystems JCS/TS 7" wurde überarbeitet und enthält nun eine ausführliche Beschreibung der Steuerkarten und Steueroptions, Fileprogrammierung und Fileorganisation, Besonderheiten des EAI-FORTRAN, (eine Tabelle der erlaubten Mixed Modes) und eine Liste aller Systemunterprogramme und -funktionen. Diese Unterlagen kann der Benutzer in der Programmberatung kostenlos bekommen.

Für das Programmpaket IPP1 ist jetzt auch "Teil I. Allgemeine Beschreibung" um 50 S erhältlich, der die Darstellung des Konzepts und die mathematischen Grundlagen enthält.

NEUE BEFEHLE LABEL UND REWIND

Im FORTRAN können zwei neue Befehle kodiert werden, und zwar die Befehle LABEL und REWIND.

Mit LABEL i (i = INTEGER-Variablen oder -Konstante) kann die momentane Position im Systemfile i markiert werden.

Mit REWIND i wird der Systemfile i an die jeweils zuletzt mit LABEL i markierte Stelle zurückpositioniert.

Mögliche Werte für i sind:

i = 1	Card Reader Input
i = 2	Line Printer Output
i = 3	Data Plotter Output

Da Datenkarten immer über den Systemfile 1 eingelesen werden, können mit LABEL und REWIND Datenkarten in einem Programm mehrfach eingelesen werden.

```
Beispiel:      :
                LABEL 1
                READ (5,4) I,J
4 FORMAT (2I5)
                REWIND 1
                READ (5,4) M,N
                :
```

Auf M,N werden die gleichen Werte wie auf I,J eingelesen, wobei nur eine Datenkarte mit den beiden Werten bereitgestellt werden muß und verarbeitet wird.

Line Printer und Data Plotter Output in einem Programm erfolgen über die Systemfiles 2 und 3 zunächst auf Scratchfiles, die erst nach dem Ende des Jobs gespooled werden. Dadurch können mit LABEL und REWIND Teile des Outputs gelöscht werden, sofern sich dieser im nachhinein bei der Exekution des Programms als nicht notwendig erweist.

```
Beispiel:      :
                LABEL 2
                WRITE (6,1) Kontrollwerte
                :
                :
                IF (IER .EQ. 0) REWIND 2
                :
```

Durch den Befehl REWIND 2 wird der Line Printer Output File bis vor die Kontrollwerte zurückpositioniert, die Ausgabe dieser Werte unterbleibt.

Bei Verwendung der Plottergrundsoftware mit COPY-Option (File CPYPLT, siehe Seite 16) sind die Befehle LABEL 3 und REWIND 3 im Benutzerprogramm verboten.

Zu beachten ist, daß LABEL und REWIND keine Ersparnis der Input bzw. Output-Zeichen selbst bewirken. Bei den obigen Beispielen werden also gleich viele Eingabezeichen verrechnet wie für zwei Datenkarten, ebenso wird auch die Ausgabe der Kontrollwerte verrechnet, da sie ja zuerst tatsächlich auf den Scratchfile geschrieben werden. Wenn also die tatsächliche Kartenanzahl gering bzw. die Line Printer-Liste oder Data Plotter-Zeichnung kurz ist, so kann die IO-Option auf der /RUN-Karte trotzdem erforderlich sein.

F. Blöser

kurse

Folgende Kurse werden an der Hybridrechenanlage abgehalten werden:

RH1 GERÄTETECHNIK EAI PACER 600

Zentraleinheit, Interruptsystem, I/O System (inklusive Interface und Analogrechner)

Dauer: 3 Tage

Termin: 1978/01/16 bis 1978/01/18

Vortragender: Dipl.Ing. W.Kleinert

RH3 HINWEISE FÜR FORTRAN-PROGRAMMIERER AN DER HYBRIDRECHENANLAGE

1 Vortrag

Termin: 1978/03/14

Vortragender: Dipl.Ing.F. Blöser

RH4 HYBRIDE FORTRAN-PROGRAMMIERUNG MIT ÜBUNGEN

Einführung in das Hybrid Linkage Softwarepaket

Dauer: 1 Woche

Termin: 1977/11/21 13-15 Uhr, 1977/11/24 11-13 Uhr,
1977/11/25 9-11 Uhr

Ort: Seminarraum des Instituts für Geophysik,
1040 Wien, Gußhausstraße 27-29, 2. Stock

Vortragender: H.Hummer

RH9 EINFÜHRUNGSKURS IN DIE HYBRIDE PROGRAMMIERUNG

Dieser Kurs wird nach Bedarf, mindestens aber alle zwei Monate für wissenschaftliche Benützer, die keinerlei Vorkenntnisse auf dem Gebiet der hybriden Programmierung besitzen und an einer möglichst raschen Problemlösung interessiert sind, abgehalten. Der Kurs ist als Anregung zum Selbststudium gedacht.

Dauer: 2 Tage

Vortragender: D. Solar

Nähere Auskünfte bei Herrn D. Solar (1040 Wien,
Gußhausstraße 27-29, 4. Stock, Zimmer 1401,

Tel.: 65-37-85/747 DW).

RH10 DATENVERBINDUNG MIT CYBER 74

1 Vortrag

Termin: 1978/01/19

Vortragender: Dipl.Ing. A.Blauensteiner

RH11 ASSEMBLER PROGRAMMIERUNG FÜR FORTGESCHRITTENE

Dauer: 1 Woche

Termin: 1978/03/13 bis 1978/03/17

Vortragender: Dipl.Ing. A.Blauensteiner

Der genaue Kursort sowie die Kurszeiten werden noch rechtzeitig im Schaukasten der Hybridrechenanlage sowie durch eine Aussendung von Einladungen bekanntgegeben.

Nähere Auskünfte und Anmeldungen zu den Kursen telefonisch oder persönlich bei Herrn M. Schandl (1040 Wien, Gußhausstraße 27-29, 4. Stock, Zimmer 1404/05, Tel.: 65-37-85/803 DW).

Vom Institut für Technische Mathematik werden folgende Kurse abgehalten:

AH1 PROGRAMMIEREN VON ANALOGRECHENANLAGEN

AH2 Grundlagen und Anwendungsmöglichkeiten des Analogrechners mit praktischen Übungen an den Rechnern EAI-18Ø und EAI-68Ø

Termin: 1978/01/30 - 1978/02/03

AH3 HYBRIDE SPRACHEN UND VERFAHREN

AH4 Grundprinzipien und Einsatzgebiete des Hybridrechnens, aufbauend auf Vorkenntnisse über Analogrechnen, wie sie in den Kursen AH1 und AH2 vermittelt werden. Simulationssprache CSMP, hybride Programmiersprache HOI, FORTRAN-Hybrid-Ergänzungen.

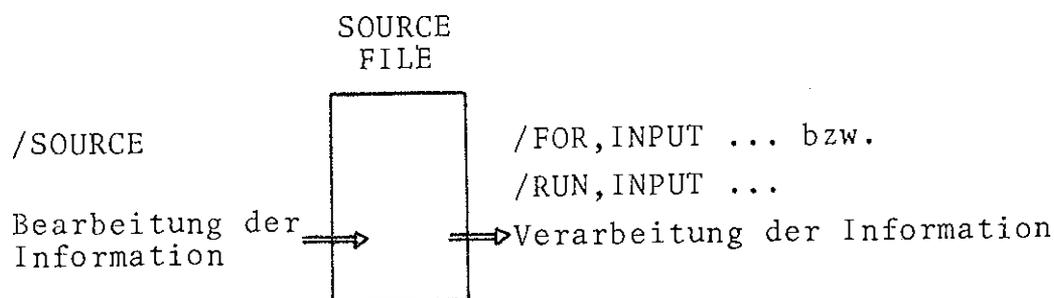
Termin: 1978/02/20 - 1978/02/24

Auskünfte und Anmeldungen zu den Kursen bei Dipl.Ing. Frank Rattay (Klappe 899), Institut für Technische Mathematik, Gußhausstraße 27-29, 1040 Wien).

SOURCE PROCESSOR

Am Hybridrechner PACER 600 ist nun ein Source Processor JCSSOP ansprechbar, welcher das Erstellen, Verbessern, Reduzieren und Erweitern von Files ermöglicht, deren Inhalt wiederholt als Quellen (Daten, Programme) für andere Prozessoren bzw. Programme verarbeitet werden soll. So ist es möglich, Programme zu erstellen und zu modifizieren, und diese dem FORTRAN-Compiler zuzuleiten ohne mit Lochkarten zu arbeiten.

Genauso können aber auch Datenbestände z.B. blockweise für die Exekution bestimmter Programme zugeordnet werden. Der Source Processor wird durch die /SOURCE-Steuerkarte geladen und durch eigene Kontrollbefehle gesteuert, die sich auf die zu treffenden Aktionen eines SOURCE-Files beziehen. Dieser SOURCE-File wiederum kann indirekt als "Lochkarten-Ersatz" durch die INPUT-Option bei einem beliebigen Task angesprochen und verarbeitet werden.



Die Syntax des Source Processors ist so definiert, daß sowohl eine absolute Orientierung innerhalb eines SOURCE-Files (Line-Nummer) möglich ist, genau so aber eine Orientierung nach Zeichen, bzw. Zeichengruppen, die gewisse Positionen innerhalb eines SOURCE-Files spezifizieren. So kann man z.B. in einem SOURCE-File, der mit einem FORTRAN-Programmsystem belegt ist, einfach alle Zeilen ermitteln, in denen die Zeichenkette >FORMAT< vorkommt.

Der Source Processor kann sowohl im BATCH Betrieb über Karten als auch über alle Terminals im Dialogbetrieb angesprochen werden. Nähere Information und genaue Unterlagen sind an der Hybridrechenanlage erhältlich.

A. Blauensteiner

GRAPHISCHE AUSGABE IM HOI

In der neuen Version des Hytran Operations Interpreter JCShOI sind die Möglichkeiten der graphischen Darstellung von Kurven wesentlich erweitert. Zum ersten Mal kann nun auch das Tektronix Graphic Display für die graphische Ausgabe verwendet werden, und zwar zusätzlich zum Data Plotter. Die gewünschte Ausgabeeinheit (Graphic Display oder Data Plotter) wird bei der Initialisierung der graphischen Ausgabe festgelegt. Gegenüber früheren Versionen von HOI können nun unabhängig voneinander Skalierungsfaktoren für die x- und y-Koordinaten gesetzt werden.

Im Zuge dieser Erweiterungen wurde auch die Ausgabe an den Data Plotter neu programmiert. Die Verwendung des Wiederholungsfaktors, der auf den nächstfolgenden Zeichenbefehl wirkt (vgl. "Modifikationen des Data Plotter Controllers" im Interface Nr. 5), bringt eine Verkleinerung der JCShOI Data Plotter Spool Files auf etwa ein Viertel der früheren Größe.

Die Befehle für die graphische Ausgabe in JCShOI sind:

Initialisierung der graphischen Ausgabe:

xf, yf, id, P;

xf Skalierungsfaktor für x-Koordinaten
yf Skalierungsfaktor für y-Koordinaten
id = -1 Graphic Display, Bildschirm wird gelöscht
= 1 Graphic Display, Bildschirm wird nicht gelöscht
= 0 Data Plotter

Bei Verwendung des Graphic Display ist zu beachten, daß nach der Initialisierung und vor dem Abschluß der graphischen Ausgabe auf diesem Gerät kein Text-Input/Output zu und von JCShOI erfolgen darf, da während dieser Zeit der Graphic Mode gesetzt ist.

Bewegung des Lichtstrahls bzw. Zeichenstifts:

x, y, ip, P;

x x-Koordinate, $-1.25 \leq x*xf \leq 15.75$ Graphic Display
 153.00 Data Plotter
y y-Koordinate, $-1.50 \leq y*yf \leq 9.50$
ip = 2 sichtbar (Stift gesenkt)
= -2 sichtbar, neuer Ursprung in (x,y)
= 3 nicht sichtbar (Stift gehoben)
= -3 nicht sichtbar, neuer Ursprung in (x,y)

Änderung der Skalierungsfaktoren (ohne neue Initialisierung):

xf, yf, 5, P;

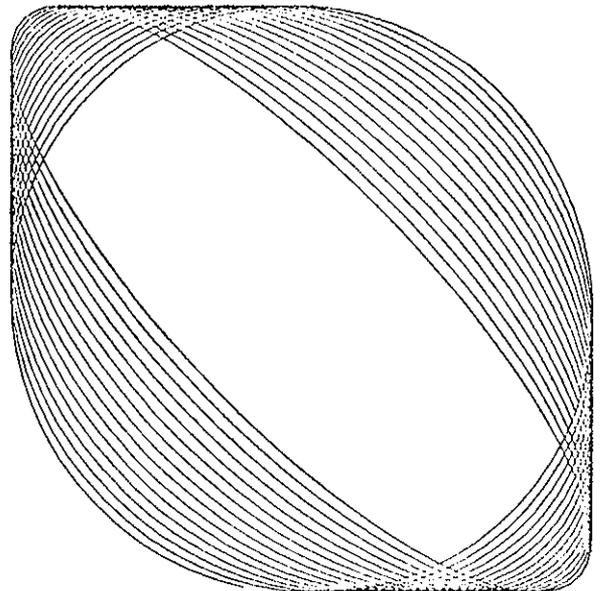
Abschluß der graphischen Ausgabe:

x, y, 999, P;

Es wird auf den Punkt (x,y) positioniert und die graphische Ausgabe abgeschlossen. Bei Ausgabe am Graphic Display wird in den Alpha Mode umgeschaltet. Der Cursor befindet sich am Beginn der Zeile, die dem Wert von y entspricht.

Beispiel mit Ausgabe am Data Plotter:

```
1.010  INIT = .002
1.020  STEP = .001
1.030  FINAL = 1.
1.040  1., 1., 0, P;
1.050  3., 4., -3, P;
1.060  4*SIN(99*INIT), 4*COS(100*INIT), 3, P;
1.070  I = INIT, STEP, FINAL 2;
1.080  5.5, -4., 999, P;
2.010  4*SIN(99*I), 4*COS(100*I), 2, P;
```



F. Blöser

DATA PLOTTER ALS HARDCOPY-EINHEIT FÜR DAS GRAPHIC DISPLAY

Die für die graphische Ausgabe am Data Plotter erforderliche Grundsoftware war bisher (und bleibt auch weiterhin) auf dem Bibliothekssammelfile PLOTSR zusammengefaßt. Sollte die graphische Ausgabe auf dem Graphic Display Terminal im Rechenraum erfolgen, so mußten die für das Graphic Display spezifischen Unterprogramme auf dem Sammelfile TEKPLT beim Binden des Programms vor dem File PLOTSR geladen werden.

Im Zuge von Verbesserungen an den bestehenden Programmen (mit dem Zweck, Kernspeicherplatz einzusparen) wurde der File TEKPLT so erweitert, daß nur mehr entweder der File PLOTSR oder der File TEKPLT geladen werden muß, je nachdem, auf welchem Gerät die graphische Darstellung erfolgen soll. In diesem Zusammenhang sei daran erinnert, daß die Bildschirmfläche des Graphic Display einem Doppelblatt auf dem Data Plotter entspricht (17 x 11 Inch). Für jedes "Doppelblatt" (jede neue Zeichnung) sollte beim Graphic Display die Software mit der Subroutine PLOTS neu initialisiert werden, damit die alte Zeichnung am Bildschirm gelöscht wird. Im übrigen werden x-Werte größer als 17 Inch ohnedies auf diesen Wert abgeschnitten.

Wird nun das Graphic Display interaktiv (z.B. zur Bestimmung günstiger Glättungskoeffizienten für die graphische Darstellung von Glättungskurven) verwendet, so will man unter Umständen von einzelnen Zeichnungen eine Hardcopy haben. Da aber an das Graphic Display keine Hardcopy-Einheit angeschlossen ist, mußte man bisher stets das Programm neu binden (mit der Software für den Plotter) und bei der neuerlichen Exekution des Programms die gewünschten Zeichnungen nochmals anfertigen.

Aus diesem Grund wurde nun die Möglichkeit geschaffen, den Data Plotter softwaremäßig als Hardcopy-Einheit an das Graphic Display anzuschließen. Verwendet man statt der Files PLOTSR oder TEKPLT den Sammlerfile CPYPLT, so erfolgt die Ausgabe auf dem Graphic Display, wobei aber einzelne Zeichnungen zusätzlich auch auf dem Plotter gezeichnet werden können. Grundlagen dafür sind das Data Plotter Output Scratching im Rahmen des Betriebssystems JCS/TS 7 und die Möglichkeit, den Data Plotter Scratch File an einer bestimmten Stelle zu markieren und zu dieser Marke zurückzupositionieren. Bei Verwendung von CPYPLT sind daher die Befehle "LABEL 3" und "REWIND 3" im Benutzerprogramm verboten (siehe Aktuelle Mitteilungen, Seite 11).

Ist ein Programm mit dem File CPYPLT gebunden worden, so erfolgt die graphische Ausgabe immer auf das Graphic Display und den Data Plotter Scratch File, wobei der Beginn der Zeichnung markiert wird. Bei Abschluß der Zeichnung durch den Aufruf CALL PLOT (x,y,999) hat der Benutzer die Wahl, die Zeichnung auch auf dem Plotter gezeichnet zu bekommen oder nicht. Durch Tippen der Taste COPY auf dem Graphic Display bleibt die Zeichnung auf dem File stehen, bei Eingabe eines BREAK wird der File auf die Marke zurückpositioniert, d.h. die (nicht gewünschte) Zeichnung wird durch eine neue Zeichnung überschrieben. Auf diese Weise stehen am Ende eines Jobs genau jene Zeichnungen auf dem Data Plotter Scratch File, von denen durch Tippen der COPY-Taste eine Hardcopy gewünscht wurde. Beim Spoolen des Files werden dann genau diese Zeichnungen am Data Plotter ausgezeichnet.

Durch die Copy-Möglichkeit wurden einige Plottergrundsoftware-Unterprogramme länger und benötigen auch mehr Rechenzeit. Daher sollten Programme nur dann mit dem File CPYPLT gebunden werden, wenn sinnvoll von der Copy-Option Gebrauch gemacht wird. In allen anderen Fällen - im Closed Shop und wenn keine Hardcopy gewünscht wird - sollten die Files PLOTSR bzw. TEKPLT verwendet werden.

Bei der Verwendung von CPYPLT ist zu beachten, daß jede graphische Ausgabe auf Display und Plotter durchgeführt wird und für beide Geräte verrechnet wird. Es kann also die IO-Option für den Data Plotter notwendig werden, auch wenn nur wenige Zeichnungen tatsächlich am Plotter gezeichnet werden. Die PLOTTER-Option muß bei CPYPLT immer verwendet werden.

Abschließend sei noch erwähnt, daß es für die Bedienung des Graphic Display spezielle Routinen (z.B. für das Löschen des Bildschirms, Umschalten in den Graphic oder Alpha Mode) gibt, die sowohl im File TEKPLT als auch im File CPYPLT enthalten sind.

F. Blöser

JOB ACCOUNTING IN JCS/TS 7

Die Einrichtungen der Hybridrechenanlage des Rechenzentrums der Technischen Universität Wien können von Universitätsangehörigen für Lehrtätigkeit, wissenschaftliche Arbeiten und Verwaltung kostenlos benützt werden. Personen von anderen Stellen können gemäß einem vereinbarten Vertrag Rechenzeit und Betriebsmittel gegen Bezahlung erhalten.

Für die Benützung der Hybridrechenanlage ist eine Benützungsbewilligung erforderlich, die projektbezogen vergeben wird. Zu diesem Zweck muß der Benützer ein Formular ausfüllen, das an der Hybridrechenanlage in Zimmer 1403 erhältlich ist und auf dem eine Projektbeschreibung und die benötigten Betriebsmittel angegeben werden müssen. Das Formular muß vom Antragsteller und vom Institutsvorstand unterschrieben werden. Bei Abgabe des ausgefüllten Formulars (Zimmer 1403 oder beim Operator) erhält der Benützer eine Jobnummer (oder Accountnummer), die für eine begrenzte Zeit Gültigkeit hat. Für ein neues Projekt muß um eine neue Jobnummer angesucht werden.

Eine Jobnummer an der Hybridrechenanlage besteht aus einem Buchstabencode, der eine Benützerkategorie angibt, und einer Integer-Zahl (z.B. SY.H1.945). Die ersten zwei Zeichen geben an, um welche Art Projekt es sich handelt:

DP	Diplomarbeit	DS	Dissertation
HP	Hybridpraktikum	MP	Mathematisches Praktikum
WI	wissenschaftliche Arbeit	CM,CE	kommerzielle Arbeit (gegen Bezahlung der Betriebsmittel)
SY	systemunterstützende Arbeiten	VD	Arbeit von Angestellten der Hybridrechenanlage für Benutzerunterstützung, Softwareentwicklungen und Tests

Die mittleren zwei Zeichen der Jobnummer setzen sich aus einem Buchstaben (H = hybride, D = digitale Verwendung des Hybridrechners) und einer Ziffer zusammen, die angibt, von welcher Stelle der Benutzer kommt (1 = TU Wien, 2 = andere österreichische Universität, 3 = andere Stellen). Die anschließende Nummer wird fortlaufend vergeben.

Durchschnittliche Aufteilung der Benutzer der Hybridrechenanlage		
	Jobnummern	verbrauchte CPU-Zeit
Wissenschaftliche Arbeiten, Diplomarbeiten, Dissertationen	48 %	25 %
Praktika	14 %	23 %
kommerzielle Kunden	11 %	8 %
Softwareentwicklungen, Service, Tests, administrative Arbeiten an der Hybridrechenanlage	27 %	44 %

Eine Jobnummer ist für die vom Benutzer angegebene Zeit gültig (maximal 2 Jahre). Um eine rationelle Verwendung der Betriebsmittel zu gewährleisten, sind Betriebsmittelbeschränkungen festgelegt:

- das CPU-Time Limit definiert die Maximalanzahl von CPU-Minuten, die der Benutzer rechnen kann,
- das I/O Limit ist die Maximalanzahl der Zeichen von Ein/Ausgabeoperationen.

Diese Limits werden pro Monat Gültigkeit festgelegt. Die Standard-Limits, mit denen die meisten Benutzer das Auslangen finden, sind

480 CPU-Minuten für Hybridbenutzer	}	pro Monat
120 CPU-Minuten für Digitalbenutzer		
3 Megaworte Input/Output Operationen		

Für spezielle Benutzeranforderungen (z.B. sehr viel Plotterausgabe, längere hybride Simulationen) können größere Limits vereinbart werden.

Die Jobnummernkontrolle und das Job Accounting werden vom Betriebssystem JCS/TS 7 automatisch durchgeführt. Der Benutzer muß auf der

/JOB - Steuerkarte,

mit der jeder Job beginnt, seine Jobnummer angeben. Das Betriebssystem überprüft die im Batch-Betrieb eingelesene Steuerkarte auf eine gültige Jobnummer. Beim Auftreten einer ungültigen Jobnummer wird dem Operator eine entsprechende Meldung (INVALID JOBNUMBER) gegeben und der Job wird übergangen. Der Line Printer Output für einen Job beginnt mit einem Deckblatt, auf dem zur Identifikation des Outputs der Codename, der vom Benutzer gewählt wird, in Balkenschrift gedruckt ist.

Alle Jobnummern befinden sich auf dem Jobfile, der außerhalb der Districteinteilung auf Massenspeicher liegt. Die Erledigung eines Jobnummernansuchens erfolgt bei Abgabe des vollständig ausgefüllten und unterschriebenen Jobnummernformulars sofort. Der Operator aktiviert einen System Overlay, der jederzeit parallel zu den zwei Benutzertasks auf den zwei Partitions des Betriebssystems JCS/TS 7 laufen kann, und errichtet die neue Jobnummer auf dem Jobfile.

Zur regelmäßigen Überprüfung der Limits und des Ablaufdatums dient das Programm DAYJOB, das täglich in der Früh unmittelbar nach dem Systemanstieg exekutiert wird. Es wird eine Kopie des Jobfiles und eine Liste erstellt. Etwa 14 Tage vor dem Ablauf einer Jobnummer wird automatisch ein Brief gedruckt, der den Jobnummernbesitzer davon verständigt. Unter Verwendung des beigelegten Formulars kann um eine Verlängerung der Jobnummer angesucht werden. Bei Erreichen des Ablaufdatums ohne vorherige Verlängerung wird eine Jobnummer automatisch gesperrt. Nach Ablauf eines weiteren Monats wird sie gelöscht. Innerhalb dieses Monats kann noch eine Verlängerung erfolgen. Wenn 90 % der durch die Limits beschränkten Betriebsmittel verbraucht wurden, wird der Benutzer ebenfalls verständigt. Wenn mehr Betriebsmittel benötigt werden, muß wie bei einer Verlängerung ein neues Ansuchen erfolgen.

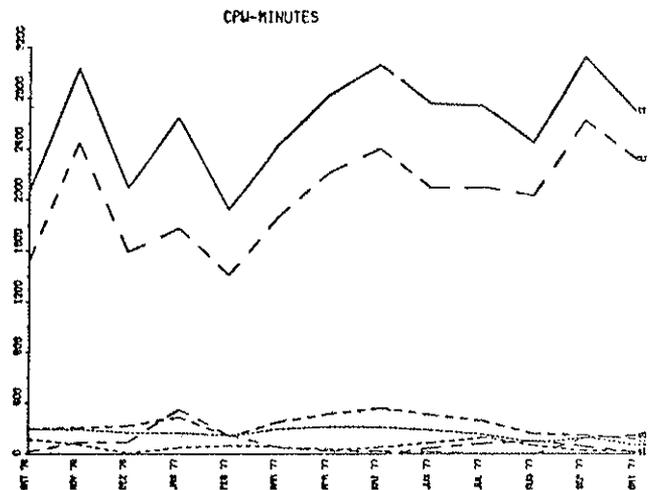
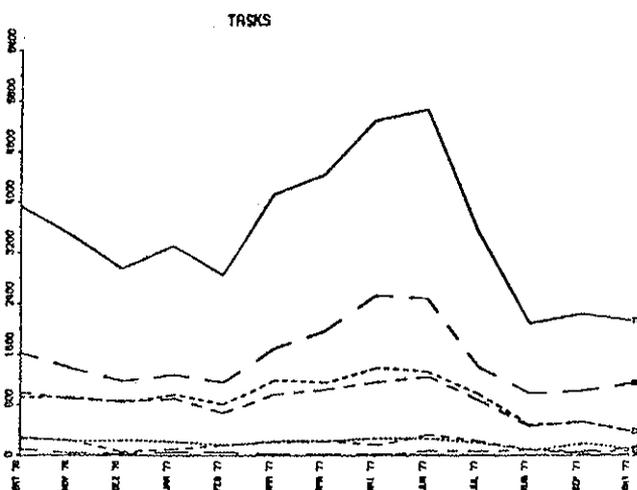
Die Accounting-Werte für jede Jobnummer sind auf dem Jobfile gespeichert. Die Jobnummer auf der /JOB-Steuerkarte gibt die Nummer an, unter welcher die Werte für den folgenden Job verrechnet werden sollen. Die Verrechnung erfolgt beim Start und Exit jedes User Tasks (ein Job besteht aus mehreren Tasks).

Es werden folgende Werte mitgeführt:

- o Taskanzahl (getrennt nach Prozessoren: Source Processor, FORTRAN-Compiler, Assembler, Core Image Generator, HDI oder User Program Run)
- o Anzahl der fehlerhaften Tasks (Gesamtanzahl): ein fehlerhafter Task ist jeder Task, der eine Systemfehlermeldung erzeugt, die zu einem Programmabbruch führt.
- o Exekutionszeit (reine CPU-Zeit, getrennt nach Prozessoren): als CPU-Zeit wird jene Zeit verstanden, in der die Central Processing Unit für diesen Task im Multiprogramming aktiv ist. Wartezeiten (z.B. auf das Bereitwerden von Ein/Ausgabegeräten) werden nicht verrechnet.
- o Kernspeicherbedarf (gesamt)
- o File Attach Count, der angibt, wie oft auf Daten- oder Programmfiles zugegriffen wurde.
- o Anzahl der 16-Bit Worte, die ein/ausgegeben wurden (getrennt nach Ein/Ausgabeeinheiten).

Wenn eine Jobnummer abgelaufen ist, erhält der Benützer ein Abschlußformular, auf dem der Betriebsmittelverbrauch für diese Jobnummer angeführt ist. Bei Bedarf kann der Benützer einen Ausdruck des aktuellen Standes der Accounting-Werte seiner Jobnummer beim Operator anfordern.

Jeden Monat wird eine Betriebsmittelabrechnung für jede Jobnummer durchgeführt. Rechnungen mit den fiktiven Kosten für den Betriebsmittelverbrauch werden an die Institutsvorstände gesandt, um einen Überblick über die Aktivitäten der Mitarbeiter zu geben. Es werden die Rechnungen für die kommerziellen Kunden erstellt. Um einen Überblick über den Rechenbetrieb an der Hybridrechenanlage zu erhalten, werden die monatlich ausgewerteten Accounting-Daten auch graphisch am Data Plotter dargestellt. Die folgenden Diagramme zeigen die Anzahl der gerechneten Tasks und die verbrauchten CPU-Minuten seit der Installation des Betriebssystems JCS/TS 7.



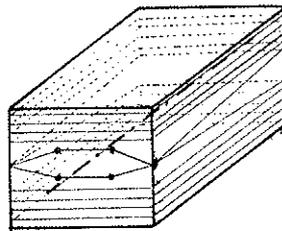
MAGNETFELDBERECHNUNG FÜR EINEN ADIABATISCHEN NEUTRONENSPINDREHER

Beispiel aus dem "Praktikum für Hybridrechner II"
Gruppe Hammer - Markum

In diesem Praktikum werden die Kenntnisse für wissenschaftliches Arbeiten an der Hybridrechenanlage vermittelt. Dabei werden öfters auch Aufgaben gerechnet, die zur Vorbereitung auf eine Diplomarbeit oder Dissertation der Praktikumsteilnehmer dienen. Die Erkenntnisse aus der Lösung des hier beschriebenen Problems aus der Atomphysik sollen auch in einer Dissertation weiterverwendet werden.

Aufgabenstellung:

Auf den Seitenflächen eines Quaders wird eine Wicklung aus lackisolierten Cu-Drähten aufgebracht. Die Form der Wicklung kann durch die Seitenlängen des Quaders und durch das Setzen von Stützen (Kabelstifte) beeinflusst werden.



Interessant ist das aus einer solchen Stromverteilung resultierende Magnetfeld an verschiedenen, innerhalb und außerhalb des Quaders gelegenen Aufpunkten in allen drei kartesischen Komponenten.

An den Feldverlauf einer solchen Anordnung sind manchmal besondere Anforderungen gestellt, die im Regelfall erst nach der realen Ausführung durch Magnetfeldmessung kontrolliert werden. Damit nicht erst nach dem Bau des Solenoids der Feldverlauf vermessen wird, scheint es sinnvoll, eine Stromverteilung vorzugeben, das Magnetfeld zu berechnen und gegebenenfalls noch Änderungen an der Drahtführung vorzunehmen. Auch die Anpassung an eine gewünschte Feldform durch eine Optimierungsrechnung rückt dadurch in greifbare Nähe, wenngleich die Schwierigkeiten dabei hier nicht aufgezeigt werden sollen.

Grundlagen:

Die physikalische Behandlung des Systems gründet sich auf das Biot-Savartsche Gesetz, das aus der Maxwellgleichung

$$\oint \vec{H} d\vec{s} = \int_Q \vec{j} d\vec{q}$$

bei der Annahme von Drahtstärken, deren Dimension klein gegen die geometrischen Abmessungen der Anordnung ist, folgt

$$\vec{H} = I \int \frac{d\vec{l} * (\vec{r} - \vec{r}')}{|\vec{r} - \vec{r}'|^3}$$

H	magnetische Erregung
J	Stromdichte
I	Stromstärke
Q	Querschnitt des Leiters
P	Aufpunkt

Sogleich erkennt man die Schwierigkeit der Berechnung, wenn man sich vorzustellen vermag, daß über eine Anzahl Drähte zu integrieren ist. Eine Berechnung am Digitalrechner wäre sicherlich möglich, die Anwendung des Diodenfunktionsgebers erscheint jedoch deshalb sinnvoll, weil dadurch die Drähte nicht diskret sondern, wie in der Realität auch, kontinuierlich in der Rechnung berücksichtigt werden können.

Die zur Verfügung stehenden Ausgabegeräte - insbesondere der Plotter - lassen überdies eine anschauliche Darstellung des Magnetfeldverlaufes zu.

Anwendungen:

Neutronenphysik: elektromagnetische Führungsfelder
 Spinfliepeinrichtungen (Mezei, Badurek)
 Spindrehher (Okorokov)

Sonstige Gebiete: Ablenkeinheiten

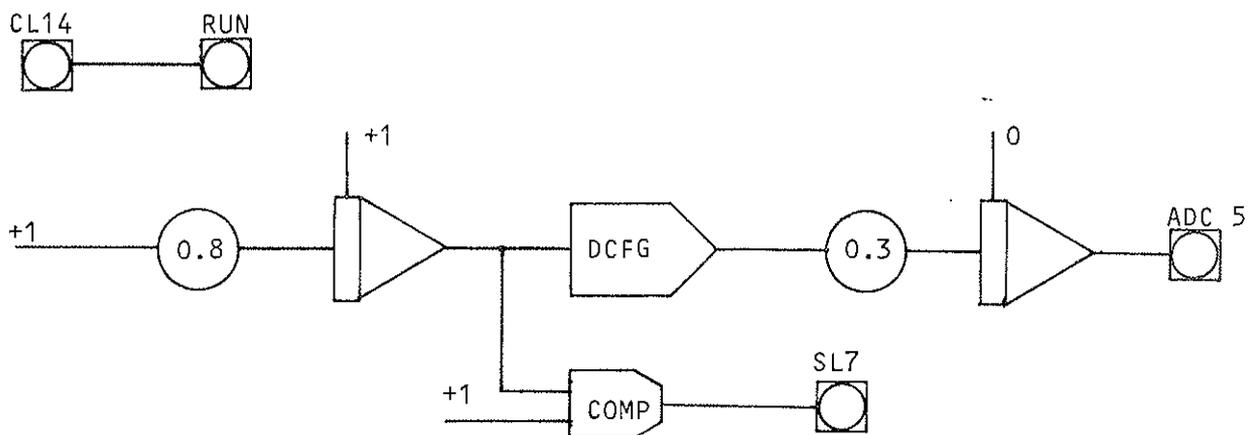
Programmstruktur:

Die Drahtkonfiguration wird durch die Stützen (Kabelstifte) angegeben; daraus läßt sich hernach die Konstruktion ableiten. Ein Programmteil des Hauptprogramms teilt die Oberfläche des Quaders in Zeilen und Spalten und berechnet für jeden Kreuzungspunkt den Strom nach Betrag und Richtung. Aufgrund dieser Daten, die auch auf einen Massenspeicher gelegt werden könnten, wird in einem Unterprogramm der DCFG (digital setzbarer Diodenfunktionsgeber) so angesteuert, daß auch die Aufpunktkoordinaten mitberücksichtigt werden. Es erfolgt eine Integration über alle Zeilen der Quaderoberfläche für jede Koordinatenrichtung.

Diese Programmfolge wird für jeden Aufpunkt durchlaufen und ermöglicht so die Darstellung des Feldverlaufes, wie er von der eingangs vorgegebenen Stromverteilung erzeugt wird und wie er letztendendes am Plotter ausgegeben wird.

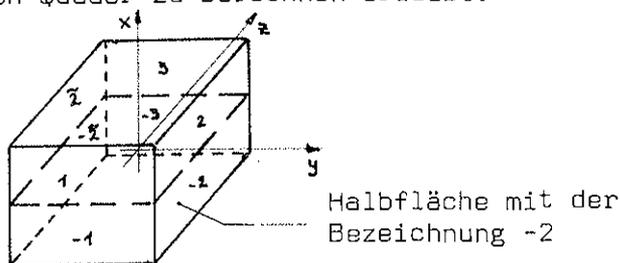
Eine Anpassung an einen vorgegebenen Feldverlauf müßte hier ansetzen, die Stromverteilung gezielt ändern, eine neuerliche Berechnung durchführen und wieder vergleichen.

Analogschaltplan:



Organisationsform der Integration:

Die Quaderoberfläche wird in acht Halbflächen zerteilt, sodaß aus Symmetriegründen nur drei verschiedene Stromverteilungen vorgegeben werden müssen. Der Ursprung des gewählten Koordinatensystems liegt im Raumzentrum. Grundsätzlich wird ein Würfel angenommen, der dann durch die implizite Verwendung von Maßstabsfaktoren auch Quader zu berechnen erlaubt.



Bei der Integration, die zeilenweise über jede der acht Halbflächen erfolgt - die Aufpunkte liegen ja manchmal nicht auf einer Symmetrieachse - wird so vorgegangen:

Die 16 Werte der Stromverteilung einer Zeile, die zuvor berechnet und eventuell auch auf Band abgespeichert wurden, werden durch den Kubus des Aufpunktabstandes im Digitalrechner dividiert und mit dem Richtungsbeitrag multipliziert.

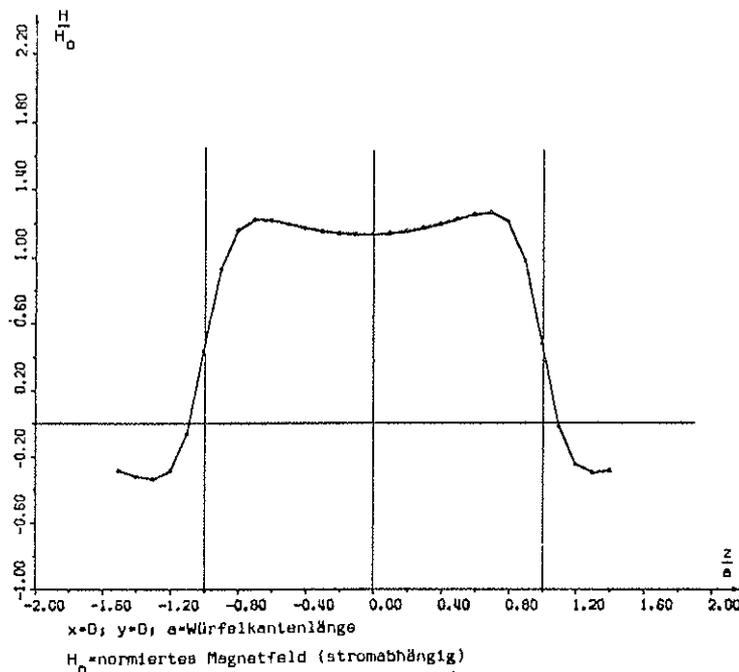
Hernach werden diese 16 Stromverteilungswerte auf 0.9 skaliert, damit unter allen Umständen eine Übersteuerung des DCFG verhindert wird.

Es folgt das Setzen des DCFG mit den 16 skalierten, durch die Aufkoordinaten korrigierten Stromverteilungswerten.

Die Integration (Analogrechner) wird nach einer kurzen IC-Phase gestartet und läuft bis das zeitproportionale Signal den Wert +1 erreicht (dies wird durch einen Komparator detektiert). Ein Interrupt auf einer Senseline ruft einen ADC auf, der den Wert der Integration speichert. Beim Betrieb im Milli-Second-Mode ist es notwendig, den ADC mit dem Komparatorsignal zu synchronisieren, damit auch der richtige Integrationswert abgelesen wird.

Über eine Controlline wird der Zeitgeber danach in den nächsten Mode geschaltet (IC) und verbleibt dort bis zu einem neuerlichen Anreiz des Systems.

Dieser Zyklus wird nun für jede Zeile einer Halbfläche, für alle Halbflächen und für jeden Aufpunkt durchlaufen und liefert schließlich die gewünschte Feldverteilung.



LÖSUNG DER WÄRMELEITUNGSGLEICHUNG MIT ORTSDISKRETISIERUNG UND TIME-SHARING DER ANALOGEN KOMPONENTEN

Gemeinschaftsprojekt der Mitarbeiter der Hybridrechenanlage
der Technischen Universität Wien

1. Einleitung:

Der vorliegende Bericht dokumentiert ein Gemeinschaftsprojekt, das von den Angehörigen der Abteilung Hybridrechenanlage des EDV-Zentrums der Technischen Universität Wien durchgeführt wurde. Ziel dieses Projektes war es, einerseits die Leistungsfähigkeit und die Grenzen des unter dem Betriebssystem JCS/TS 7 laufenden Hybridrechners EAI-PACER-600 auszuloten und aufzuzeigen, und andererseits durch arbeitsteilige Teamarbeit in einem abgegrenzten Zeitraum ein bestimmtes Problem, von der Analyse bis zur Realisierung am Rechner und der Dokumentation, zu lösen. Die Auswahl des Problems und der eingeschlagene Lösungsweg sind daher nur unter diesen Gesichtspunkten zu verstehen und bilden in einem gewissen Sinn nur den physikalischen und numerischen "Background" für die eigentliche Zielsetzung des Projekts.

2. Problemstellung und mathematisch numerische Analyse:

2.1 Ein Wärmeleitungsproblem

Zu lösen ist die folgende partielle Differentialgleichung samt den zugehörigen Anfangs- und Randbedingungen nach [1]:

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} + h(u, x, t) \quad a^2 = \frac{k}{\rho c}, \quad h(u, x, t) = \frac{q(u, x, t)}{\rho c} \quad (1)$$

$u = u(x, t)$

$\rho = \rho(x)$ Dichte, hier konstant

$x \in [0, 1]$

c Wärmekapazität

$t \in [0, T]$

k Wärmeleitkoeffizient

$q(u, x, t)$ die pro Volums- und Zeiteinheit im Material erzeugte Wärmemenge (Quelle oder Senke möglich)

a ist somit als Geschwindigkeit des Temperatúrausgleichs (Temperaturleitfähigkeit) definiert

$$u(0,t)=1 \quad u(1,t)=1 \quad u(x,0)=0 \quad \text{für } x \in (0,1) \text{ d.h.}$$

$$h(u,0,t) = h(u,1,t) = 0 \quad \text{für } x > 0$$

$$h(u,x,t) = -k_1 u^4 / k_2$$

d.h. ein dünner homogener Stab der Länge 1 mit der Temperatur 0 (Grad C) wird an beiden Enden mit einem Wärmebad der Temperatur von 1 Grad verbunden. Gesucht sind die mit der Zeit veränderlichen Temperaturprofile.



Abb. 1

2.2 Ortsdiskretisierung

Die Methode der Ortsdiskretisierung, die gerne zur rein analogen Lösung von parabolischen Differentialgleichungen vom Typ (1) verwendet wird, besteht nun darin, in (1) die 2. partielle Ableitung nach x durch den zentralen Differenzenquotienten

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{x=x_i} = \frac{u(x_{i-1},t) - 2u(x_i,t) + u(x_{i+1},t)}{(\Delta x)^2} \quad (2)$$

mit $0=x_0 < x_1 \dots < x_n = 1$, $\Delta x = x_i - x_{i-1}$

zu ersetzen, was mit

$$u_i(t) = u(x_i, t) \quad i=1(1)n-1 \quad \text{und} \quad h_i = h(u_i, x_i, t)$$

auf das System gewöhnlicher Differentialgleichungen

$$\frac{du_i(t)}{dt} = \frac{a^2}{(\Delta x)^2} (u_{i-1} - 2u_i + u_{i+1}) + h_i(u_i, t) \quad (3)$$

führt.

Zur direkten Nachbildung am Analogrechner werden daher $(n-1)$ Integrierer benötigt, d.h. im Falle des 680 Analogrechners wäre $n=30$ das Maximum für die Verfeinerung der Diskretisierung. Falls $h_i(u_i, t)$ nicht linear ist, so würden darüber hinaus $(n-1)$ Funktionsgeber benötigt werden, die bei uns nicht zur Verfügung stehen.

2.3 Hybrides Time-Sharing der analogen Komponenten

Bekey und Karplus [2] beschreiben die Möglichkeit, analoge Komponenten zu sparen, indem nur eine Zelle der Ortsdiskretisierung analog nachgebildet wird. Die benachbarten zeitlich veränderlichen Wärmepotentiale werden für jede Zelle simuliert, d.h. sie kommen als Tabellen vom Digitalrechner. Das Analogsystem wird dazu verwendet, sukzessiv die einzelnen Zellen der Ortsvariablen darzustellen.

Man erhält die Lösungen $u_i(t)$ iterativ, indem mit beliebigen Startfunktionen $s_i(t)$ für die Nachbarzellen u_{i-1} , u_{i+1} (die ja zeitabhängige "Randwerte" für $u_i(t)$ darstellen) begonnen wird ($u_i^{(0)}(t) = s_i(t)$).

Nach dem ersten Durchlauf werden nun die so erhaltenen Lösungen $u_i^{(1)}(t)$ als "Randwerte" für die nächsten Iterationen verwendet.

Unter bestimmten Voraussetzungen über $h_i(t)$ und $s_i(t)$ konvergieren die Näherungen $u_i^{(k)}(t)$ gegen die exakte Lösung $u_i(t)$ des diskretisierten Problems (3).

2.4 Konvergenzbetrachtungen des Iterationsverfahrens

Für die weitere Betrachtung führen wir zuerst eine Transformation

$$\tau = \frac{2ta^2}{(\Delta x)^2} \quad (d\tau = \frac{2a^2}{(\Delta x)^2} dt), \text{ gilt für konstante Dichte und Kapazität} \quad (4)$$

durch. Das liefert wegen $\frac{du_i}{dt} = \frac{du_i}{d\tau} \cdot \frac{d\tau}{dt}$

$$\frac{du_i}{d\tau} = -u_i + \frac{u_{i-1} + u_{i+1}}{2} + \frac{(\Delta x)^2}{2a^2} h_i(u_i, \tau) \quad (5)$$

und falls man $\frac{(\Delta x)^2}{2a^2} h_i(u_i, \tau) = g_i(u_i, \tau)$ setzt, führt dies zu

$$\frac{du_i}{d\tau} = -u_i + \frac{u_{i-1} + u_{i+1}}{2} + g_i(u_i, \tau) \quad i=1(1)n-1 \quad (6)$$

$$g_i(u_i, \tau) = \frac{(\Delta x)^2}{2a^2} h_i(u_i, \tau) = \frac{(\Delta x)^2 q(u_i, x_i, \tau)}{2a^2 c \rho} = \frac{(\Delta x)^2}{2k} q(u_i, x_i, \tau)$$

Das in 2.3 beschriebene Iterationsverfahren kann einfach interpretiert werden, wenn für die Startfunktionen $s_i(t)$ die konstanten Anfangsbedingungen $u_i(0)$ genommen werden.

$$u_i^{(0)}(t) = u_i(0) = s_i = \begin{cases} 1 & i=0, n \\ 0 & i=1(1)n-1 \end{cases} \quad (7)$$

Das Gleichungssystem (6) kann nun in folgender Weise angeschrieben werden:

$$\begin{aligned} \frac{du_0}{d\tau} &= 0 & = f_0(\tau, u_0, \dots, u_n) \\ \frac{du_i}{d\tau} &= -u_i + \frac{1}{2}(u_{i-1} + u_{i+1}) + g_i(u_i, \tau) = f_i(\tau, u_0, \dots, u_n) & i=1(1)n-1 \\ \frac{du_n}{d\tau} &= 0 & = f_n(\tau, u_0, \dots, u_n) \end{aligned} \quad (8)$$

mit den Anfangsbedingungen $u_i(0) = s_i$.

In diesem Fall ist unser Iterationsverfahren genau das Verfahren der sukzessiven Approximation, das nach einem Satz von Picard und Lindelöf zum Existenz- und Eindeutigkeitsbeweis einer Lösung des Differentialgleichungssystems herangezogen werden kann (siehe [3]).

Nach diesem Satz existiert genau ein System von Funktionen $u_i(\tau)$ in einer Umgebung G des Ursprungs, wenn die rechten Seiten von (8) eindeutig, beschränkt und stetig sind und eine Lipschitzbedingung in G erfüllen.

Eine hinreichende Bedingung für die Existenz und Eindeutigkeit einer Lösung von (8) - und damit auch für die Konvergenz der Iterationsvorschrift - ist die Existenz und Beschränktheit der ersten partiellen Ableitungen der $f_i(\tau, u_0, \dots, u_n)$ nach den u_l , also die Existenz einer Schranke K , sodaß

$$\left| \frac{\partial}{\partial u_l} f_i(\tau, u_0, \dots, u_n) \right| = |f_{il}| < K \quad i, l = 0, 1, \dots, n \quad (9)$$

$$f_{il} = \begin{cases} 0 & l < i-1, l > i+1, i=0,n \\ -1 + \frac{\partial g_i(u_i, \tau)}{\partial u_l} & l = i \\ \frac{1}{2} & l = i-1, l = i+1 \end{cases}$$

d.h. die Lipschitzbedingung ist für $\tau \in [0, T]$ sicher dann erfüllt, wenn im betrachteten Zeitintervall die partiellen Ableitungen $\frac{\partial g_i}{\partial u_l}$ existieren und beschränkt sind.

Hingewiesen werden soll darauf, daß der Bereich $[0, T]$ wegen der Transformation (4) quadratisch mit der Diskretisierung Δx wächst.

Setzt man z.B. $\Delta x = \frac{1}{50}$, $a^2 = 1$ so liefert (4)

$$\tau = \frac{1}{2 \cdot \frac{1}{2500}} \cdot t = 1250 \cdot t$$

d.h. für $t = 1$ sec müßte $\tau = 1250$ sec gesetzt werden!

Wendet man das sukzessive Iterationsverfahren auf (6) an, so erhält man die folgende Iterationsvorschrift

$$\frac{du_i(\tau)^{(k)}}{d\tau} = -u_i^{(k)} + \frac{u_{i-1}^{(k-1)} + u_{i+1}^{(k-1)}}{2} + g_i(u_i, \tau) \quad (10)$$

Wegen $u_0^{(k)} = 1$ und $u_i^{(0)} = 0$ erweist sich folgende Modifikation als zielführend und konvergenzbeschleunigend:

$$\frac{du_i^{(k)}}{d\tau} = -u_i^{(k)} + \frac{u_{i-1}^{(k)} + u_{i+1}^{(k-1)}}{2} + g_i(u_i, \tau), \quad u_i^{(k)}(0) = 0 \quad (11)$$

d.h. es werden die Näherungen $u_{i-1}^{(k)}$ sofort für die Berechnung von $u_i^{(k)}$ verwendet.

Ohne Beweis wird angeführt, daß sich die Iterationsgeschwindigkeit folgendermaßen abschätzen läßt

$$\left| u_i^{(k)} - u_i^{(k-1)} \right| < m \frac{(K(n+1)T)^k}{k!} \quad \begin{array}{l} K \text{ Lipschitz-} \\ \text{konstante} \end{array} \quad (12)$$

3. Realisierung am Hybridrechner

3.1 Allgemeine Festlegungen

Aus Kapitel 2 ergeben sich folgende Festlegungen für die Realisierung des Iterationsverfahrens am Hybridrechner:

- n : Anzahl der örtlichen "Diskretisierungszellen"
- T : Gesamtzeit, bis zu der jede Iteration integriert werden soll
- m : Anzahl der Zeitpunkte τ_j , $j = 1(1)m$, an denen die iterierten $u_i^{(k)}(\tau_j)$ im Digitalrechner abgespeichert werden sollen
- g_i : verschiedene Arten von Störfunktionen, werden in Tabellenform bei jeder Integration zur Verfügung gestellt.

3.2 Analyse der Realisierung am Rechner

Das Problem wurde in ein Vorlauf-, Haupt- und Nachlaufprogramm unterteilt, wobei sämtliche langsamen und programmbedingten Ein/Ausgabe-Operationen für Datenaufnahme und Ausgabe der Ergebnisse im Vor- bzw. Nachlaufprogramm durchgeführt werden.

So erstellt das Vorlaufprogramm einen Masterfile mit allen Parametern und Identifikationen für den Iterations- und Störfile. Der Iterationsfile wird mit den Anfangswerten $u_i^{(0)}$ belegt und der Störfile wird vollständig mit den Funktionen g_i erstellt.

Das Nachlaufprogramm erstellt einen Zusatzfile (transponierter Iterationsfile), auf dem die Iterationsvektoren auch nach dem Ort abrufbar sind. So kann die Lösungssiteration sowohl nach der Zeit als auch nach dem Ort und zwar sowohl am Scope als auch 3-dimensional am Plotter dargestellt werden. Das Hauptprogramm selbst arbeitet nur mit dem Analogrechner und den Files und bewerkstelligt die Iterationen k.

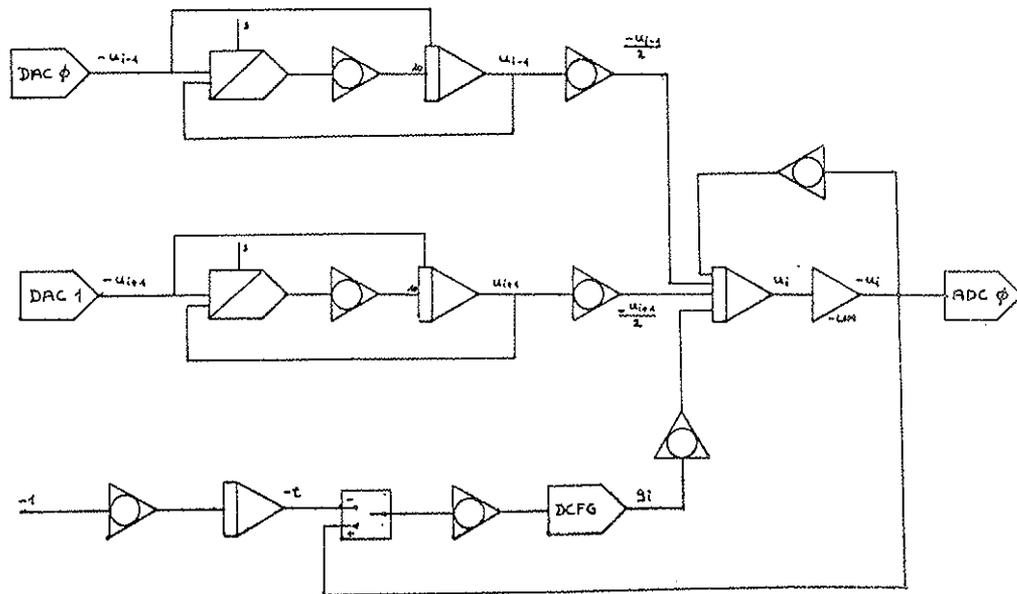


Abbildung 4
 qualitativer Schaltplan einer Ortszelle,
 die Ränder werden interpoliert zugeführt

Zur Steigerung der Geschwindigkeit wurden die variablen Vektoren $(g_i, u_{i+1}^{(k-1)}, u_i^{(k)})$ für eine parallele Auf- und Verarbeitung konzipiert. Alle drei Vektoren (nach τ) werden über eine Ortszellenintegration hinweg parallel von bzw. zu den Files transportiert. Bei der jeweiligen Integration stehen die Daten gesichert zur Verfügung. Dadurch wird eine Parallelität von 3 File-Übertragungen, einer A/D Übertragung vom Analogrechner der $u_i^{(k)}$ über einen Datenkanal, und einer D/A Übertragung der $u_{i+1}^{(k-1)}$ und $u_{i-1}^{(k)}$ zum Analogrechner über Interrupts sowie der Integration selbst erreicht.

MASCHINENAUSLASTUNG / PROGRAMMZUGRIFF

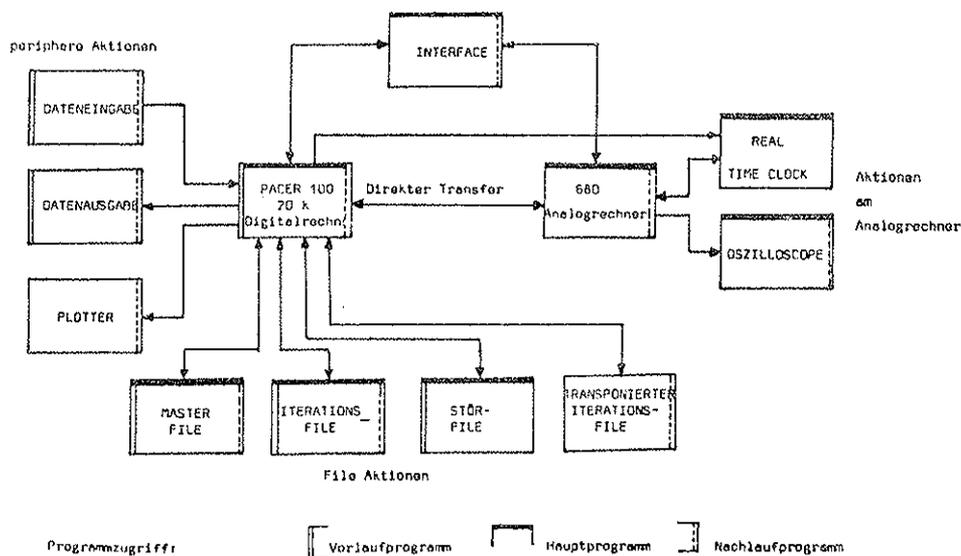


Abbildung 5

Die Arbeitsfelder der Iterationen, auf denen die benötigten Datenvektoren abgespeichert sind, sind strukturell in zwei Gruppen zu zerlegen. Eine Gruppe bezieht sich auf die laufende Iteration k , wobei 2 Felder benötigt werden, deren Bedeutung sich alternierend ändert. Die andere Gruppe besteht aus 3 Feldern, wobei alle durch Werte der letzten Iteration belegt werden. Diese werden zyklisch vertauscht. Diese Zyklen werden durch Wechsel der Indizes von mehrdimensionalen Feldern erreicht, wobei für die Arbeitsmoduln der Aufruf formal gleich bleibt, nur die Feldadressen vertauscht sind.

Zur Durchführung des Problems wurde eine Arbeitsgruppe mit der Koordinierung der Aufgaben, Programmaufteilung, Analyse bzw. mit dem Erstellen der übergeordneten Hauptprogramme betraut; eine zweite und dritte Arbeitsgruppe wurde jeweils für die Teilanalyse der ihr zugeordneten Moduln sowie deren Programmierung und Austestung eingesetzt.

Diese Gruppen trafen einander in regelmäßigen Abständen zur Diskussion der Moduln und deren Überarbeitung bzw. Koordinierung und Standardisierung. Durch ein möglichst modulares Konzept der Subroutinen, Common Blöcke, Parameter und Files als gemeinsame Datenstrukturen konnten die Gruppen unabhängig voneinander arbeiten.

4. Ergebnisse:

Die Problemdaten wurden im [MKS] System gewählt, als Temperaturskala wurde die Celsius-Skala verwendet ($^{\circ}\text{C} = ^{\circ}\text{K} - 273$)

Länge des Stabes : 1 m

Beobachtungszeit : 50 sec

Wärmeleitkoeffizient: $10 \text{ Wm}^{-1}\text{grad}^{-1}$ für

Chromnickelstahl (28% Cr, 5% Ni)

Die Enden des Stabes sind dauernd an Wärmebäder von 1° angeschlossen, der Rest des Stabes hat zum Zeitpunkt $t = 0$ die Temperatur 0° .

Der Stab strahlt nur nach dem T^4 Gesetz ab, d.h. also reine Temperaturstrahlung ohne Wärmeleitung und Konvektion, und damit auch unabhängig von der Umgebungstemperatur. Bei dieser Annahme hängt die Störfunktion $g_i(u_i, x, \tau) = g(u_i)$ nur mehr von u_i ab, die spezifische Wärme konnte im Beobachtungsintervall als temperaturunabhängig angenommen werden. Es wurden räumlich wirkende Plotterzeichnungen der Ergebnisfunktion über x und t angefertigt, wobei der Betrachtungswinkel der sich als Fläche darstellenden Temperaturverteilung variiert werden konnte.

Bei den gewählten Daten zeigte sich bereits nach 10 sec eine starke Abrundung und Verflachung des Trogges der Temperaturverteilung.

Nach etwa 20 sec hat die Mitte des Stabes eine Temperatur von ca 0.25° erreicht.

Ab etwa 30 sec Beobachtungsdauer beginnen infolge des Überwiegens der Wärmestrahlung über den Wärmezufuß die Temperaturwerte entlang des gesamten Stabes wieder leicht zu sinken, sodaß das x - t -Schaubild die Form eines verflachenden Trogtales mit Sattelpunkt bei etwa 30 sec Problemdauer und nachfolgender erneuter leichter Ausweitung bis zum Ende der Beobachtungsdauer annimmt.

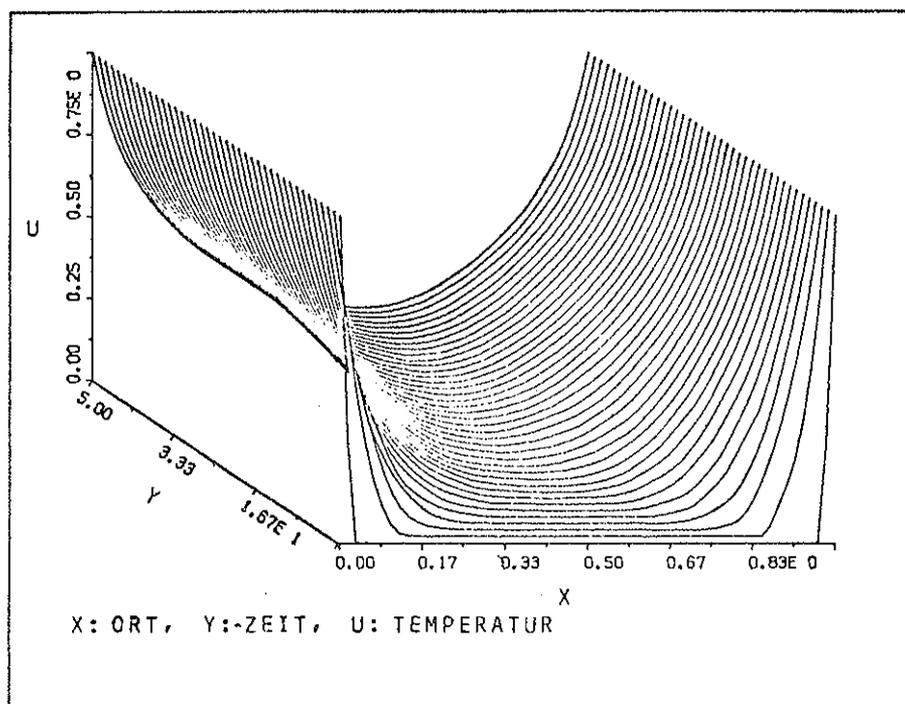


Abbildung 6
Temperaturverlauf der stationären Lösung

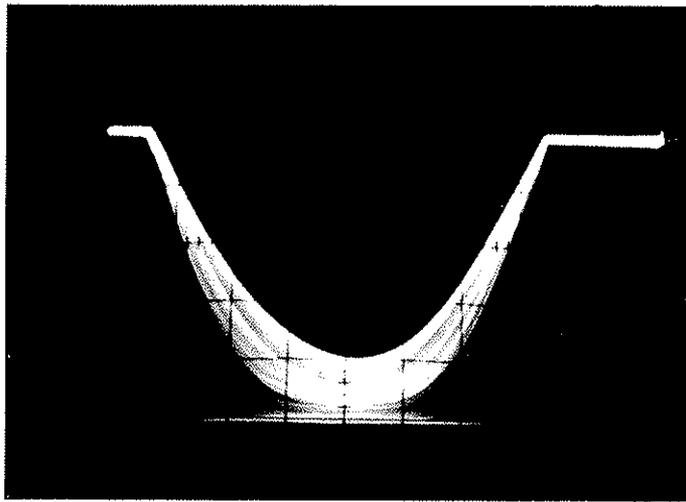


Abbildung 7
Dynamische Darstellung der Temperatur über dem Ort
im Laufe der Beobachtungszeit

Zusammenfassend ist zu bemerken, daß durch den eingeschlagenen Programmierungsweg auch andere Differentialgleichungen gelöst werden können, sofern sie nach einer unabhängigen Variablen diskretisierbar sind und zur Darstellung einer Zelle nicht mehr als zwei Randfunktionen und eine Störfunktion benötigen, Stabilität und Konvergenz vorausgesetzt.

Summary

A teamwork of the staff of the Hybrid Computation Centre of the Technical University of Vienna is reported. The aim was to show the capacity and to go to the limits of the hybrid system EAI PACER 600 under the operating system JCS/TS 7 on the one hand, and on the other hand to solve a given problem by teamwork including analysis, realization on the computer, and documentation. The problem given was a partial differential equation with boundary conditions (heat equation). The method of discrete-space-continuous-time leads to a system of ordinary differential equations. Only one cell of the discrete-space-continuous-time is built on the analog computer. The boundary conditions for each cell are simulated by tables from the digital part. The analog system integrates the discrete cells successively.

Literatur

- [1] Jaworsky-Detlaf: Physik griffbereit. Vieweg, Braunschweig (1972)
- [2] Bekey, G.,A., Karplus, W.,J.: Hybrid Computation. J.Wiley & Sons, Inc. New York (1968)
- [3] Gröbner, W., Lesky, P.: Mathematische Methoden der Physik I. Bibliographisches Institut, Mannheim (1964)

EINSATZMÖGLICHKEITEN EINES HYBRIDSYSTEMS

IN BIOLOGIE UND MEDIZIN

Vorführung eines Mini-Hybrid-Systems bei einem biometrischen Seminar in Krems

In Krems fand vom 26.9.1977 bis 29.9.1977 ein Seminar der Region Österreich-Schweiz der Internationalen Biometrischen Gesellschaft statt, bei dem von der Hybridrechenanlage an zwei Tagen zwischen den Vorträgen ein Mini-Hybrid-System der Firma EAI vorgeführt wurde. Dieses Mini-Hybrid-System besteht aus einem mikroprogrammierbaren EAI DATA-PACER Digitalrechner, der im wesentlichen den gleichen Instruktionssatz wie der Digitalrechner PACER 100 besitzt, mit zwei Floppy Disc Einheiten und einem EAI 185 Analogrechner, die über eine Interface-Einheit gekoppelt sind. Das System wird von einem Ein- und Ausgabe-Terminal mit Graphic Option bedient und stellt dem Benutzer eine Basissoftware zur Verfügung, die eine Programmierung in HOI (Hytran Operations Interpreter), FORTRAN IV oder Symbolic Assembler ermöglicht.

Ziel dieser Vorführung war es, auf die Einsatzmöglichkeiten eines Hybrid-systems in Biologie und Medizin hinzuweisen und diese an Hand von Beispielen zu demonstrieren. Zu diesem Zweck wurden aus "Analogcomputer in Biologie und Medizin" von H. Röpke und J. Riemann drei Probleme, nämlich

- o ein Grundmodell für eine zeitlich unterteilte Dosisverabreichung,
- o ein Modell für eine intravenöse Dauerinfusion und
- o die Metabolisierung und Elimination von Glycodiazin

als spezielle Beispiele ausgewählt und am Analogrechner mit digitaler Unterstützung durch ein HOI-Programm simuliert. Mit Hilfe dieses Programmes konnte das Verhalten der Modelle bei Parameteränderungen einfach untersucht werden, wobei die an der Hybridrechenanlage der TU durchgeführte Erweiterung des HOI-Prozessors eine graphische Ausgabe der Untersuchungsergebnisse am Terminal ermöglichte (siehe auch Seite 15). Der folgende Bericht stellt nun einen erweiterten Teil des Vorführungsinhaltes dar und ist als grobe Einleitung in die hybride Simulation in Biologie und Medizin zu verstehen, um auf diese Anwendungsmöglichkeiten aufmerksam zu machen.

Das wichtigste und umfangreichste Anwendungsgebiet einer hybriden Computersimulation innerhalb der Biologie und Medizin ist die Pharmakokinetik. Die Verteilung eines Pharmakons im Organismus folgt bekannten biochemischen und physikochemischen Gesetzen. Da im Organismus stets eine Vielzahl von Gleichgewichten und Regelkreisen ineinandergreifen, führt eine Modellbildung auf sehr komplexe Systeme, die sich mit den herkömmlichen Methoden praktisch kaum explizit lösen lassen.

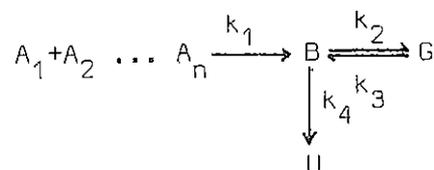
Mit Hilfe eines Analogrechners können jedoch sehr komplexe Systeme einfach und bequem gelöst werden. Die Arbeitsweise dieses Rechners ermöglicht ein direktes Arbeiten am Modell, sodaß Untersuchungen des Verhaltens und Änderungen des Modells, das ja nur eine Vereinfachung der Wirklichkeit dar-

stellt, einfach durchgeführt werden können. Anhand des Simulationsmodells, das aus bekannten biochemischen und pharmakokinetischen Gesetzen und den Grundgesetzen der Reaktionskinetik gewonnen wird, können die biologischen Vorstellungen über die Verteilung eines Pharmakons im Organismus überprüft werden, denn nur bei Übereinstimmung ist die Möglichkeit gegeben, daß die Annahmen wirklich zutreffen können. Im anderen Fall müssen die Modellvorstellungen solange variiert werden, bis eine Übereinstimmung mit den experimentellen Daten erreicht ist. Die auf diese Weise ermittelten Geschwindigkeitskonstanten für die einzelnen Transport- bzw. Diffusionsvorgänge gestatten eine Differenzierung des ganzen Prozesses in einzelne Schritte. Hat man ein brauchbares Modell für die Pharmakokinetik des betreffenden Pharmakons gefunden, so kann man leicht nachweisen, welche speziellen Einzelvorgänge für eventuelle Differenzen bei der unterschiedlichen Elimination verantwortlich zu machen sind. Damit besteht auch durchaus die Möglichkeit, für bestimmte Vorgänge Standardwerte zu erarbeiten und die Abweichungen in den Geschwindigkeitskonstanten mit bestimmten Krankheitsbildern oder Funktionsstörungen zu korrelieren. Ein weiterer wichtiger Vorteil der Analogsimulation besteht darin, daß man aufgrund des wahrscheinlich gemachten Modells alle interessierenden Bedingungen simulieren kann, ohne die hierzu notwendigen Versuche zuvor durchführen zu müssen. Dadurch werden kritische Situationen offenkundig, sodaß man sie dann leicht vermeiden kann. Weiters können die besten Applikationskombinationen, -arten und -frequenzen und die optimalen Dosierungsverhältnisse ermittelt werden. Natürlich sollte man die wichtigsten positiven Ergebnisse durch gezielte biologische Versuche zu erhärten trachten, ehe man sie als Fakten akzeptiert.

Das folgende Beispiel gibt einen kurzen Einblick in die Lösungsmethoden und -möglichkeiten eines pharmakokinetischen Modells mit Hilfe eines Analog- bzw. Hybridrechners.

Wenn eine Substanz in bestimmten Zeitabständen hintereinander verabreicht wird, so sind die Folgen nicht ohne weiteres übersehbar. Besteht aber ein gültiges Modell der Pharmakokinetik dieser Substanz, so kann leicht gezeigt werden, ob unter Umständen unerwartet eine sukzessive Anreicherung des Pharmakons im Körper erfolgt und nach dem Absetzen noch eine relativ lange Zeit verstreichen muß, ehe eine vollständige Elimination stattgefunden hat.

Für eine zeitlich unterteilte Dosisverabreichung läßt sich das folgende pharmakokinetische Grundmodell aufstellen:



wobei mit A_1, A_2, \dots, A_n die erste, zweite, etc. Verabreichung einer Dosis in den Magen-Darmkanal, bzw. die Muskeln oder die Haut ist (Absorptionskompartiment)

	B	Blut
	G	allgemeine Gewebe
	U	Urin
und	k_1, k_2, k_3, k_4	die Reaktionsgeschwindigkeitskonstanten der einzelnen Transporte erster Ordnung bezeichnet sind.

Die Konzentrationsvariablen genügen nun den Reaktionsgleichungen

$$\begin{aligned} \frac{dA}{dt} &= -k_1 \cdot A && \text{Absorptionskompartiment} \\ \frac{dB}{dt} &= k_1 \cdot A - (k_2 + k_4) \cdot B + k_3 \cdot G && \text{Blutspiegel} \\ \frac{dG}{dt} &= k_2 \cdot B - k_3 \cdot G && \text{Gewebespiegel} \\ \frac{dU}{dt} &= k_4 \cdot B && \text{Urinausscheidung} \end{aligned}$$

Der erste Schritt der hybriden Lösungsmethode besteht nun darin, dieses Gleichungssystem am Analogrechner nachzubilden. Dazu muß es in ein äquivalentes System derart transformiert werden, daß alle zeitabhängigen Variablen betragsmäßig nicht größer als 1 werden. Im vorliegenden Fall wird dies einfach durch eine Normierung aller Variablen mit der Summe der zu verabreichenden Dosen erreicht. Die folgende Abbildung zeigt die Analogschaltung des Modells bei Applikation einer einfachen Dosis A_0 .

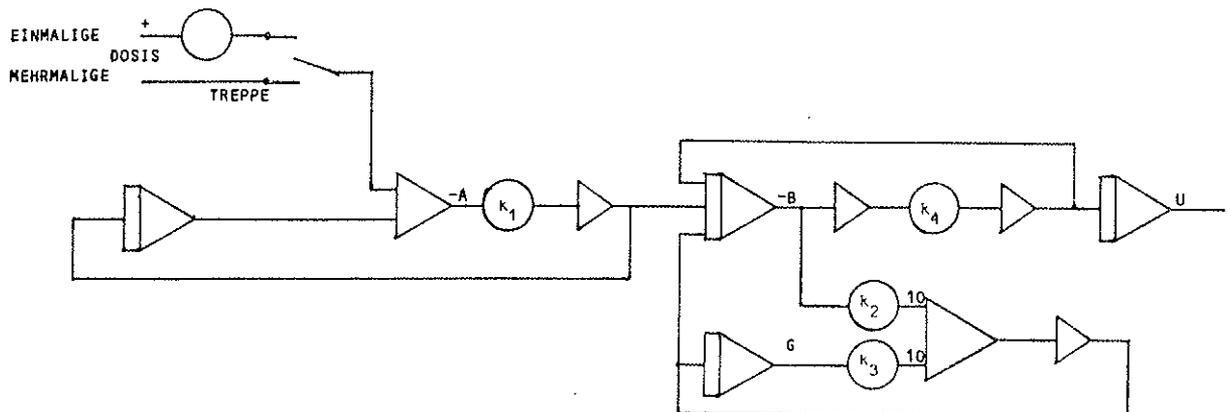


Abbildung 1
Analogschaltung für die Simulation
einer einfachen Dosisverabreichung

Nun können bereits rein analog Untersuchungen am Modell durchgeführt werden. Durch Änderung der Potentiometerwerte kann der Einfluß der einzelnen Reaktionskonstanten auf das Verhalten des Systems aufgrund der hohen Rechengeschwindigkeit direkt am Oszillographen beobachtet werden und so durch Vergleich mit experimentell ermittelten Werten die Reaktionskonstanten des Pharmakons oder ein für ein Optimierungsverfahren günstiger Bereich der Reaktionskonstanten ermittelt werden. In Abb. 2 ist der Verlauf der einzelnen Konzentrationsvariablen bei festen Reaktionskonstanten dargestellt, Abb. 3 zeigt den Blutspiegel des Pharmakons für mehrere Werte der Reaktionskonstanten k_1 des Absorptionskompartments.

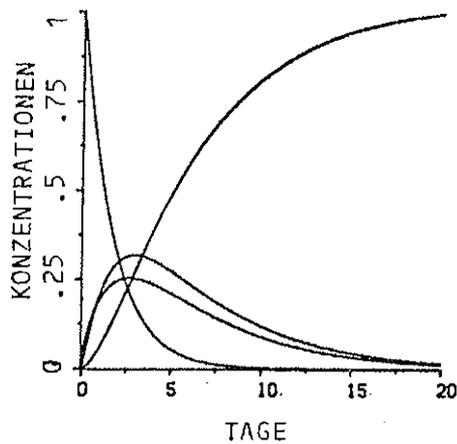


Abbildung 2

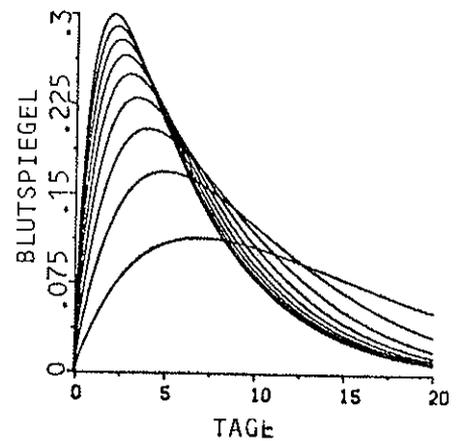


Abbildung 3

Sind die Reaktionsgeschwindigkeiten des Pharmakons bekannt, kann bereits das Verhalten bei mehrfacher Verabreichung einer Dosis A_0 untersucht werden. Die mehrfache Applikation einer gleichbleibenden Dosis eines Medikaments wird am Analogrechner mit Hilfe einer Treppenfunktion realisiert, die mit der in Abb. 4 angegebenen Schaltung erzeugt wird. Die Dosis A_0 und der zeitliche Abstand T der Applikation sind dabei in einem normierten Bereich beliebig variierbar.

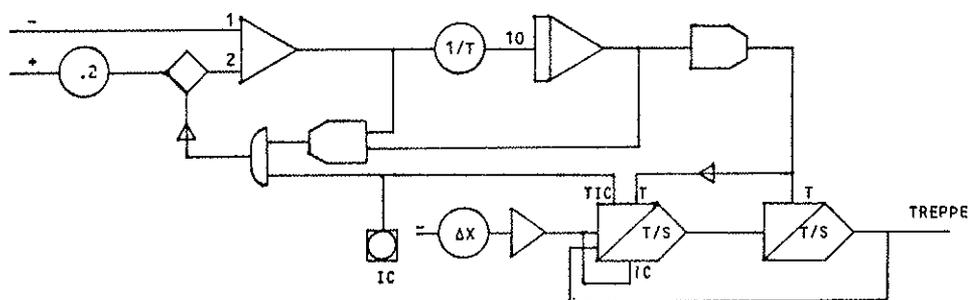


Abbildung 4

Diese Treppenfunktion wird nun anstelle der einfachen Dosis A_0 an die Schaltung des Modelles angeschlossen, sodaß nun die Kumulation des Pharmakons untersucht werden kann (Abb.5).

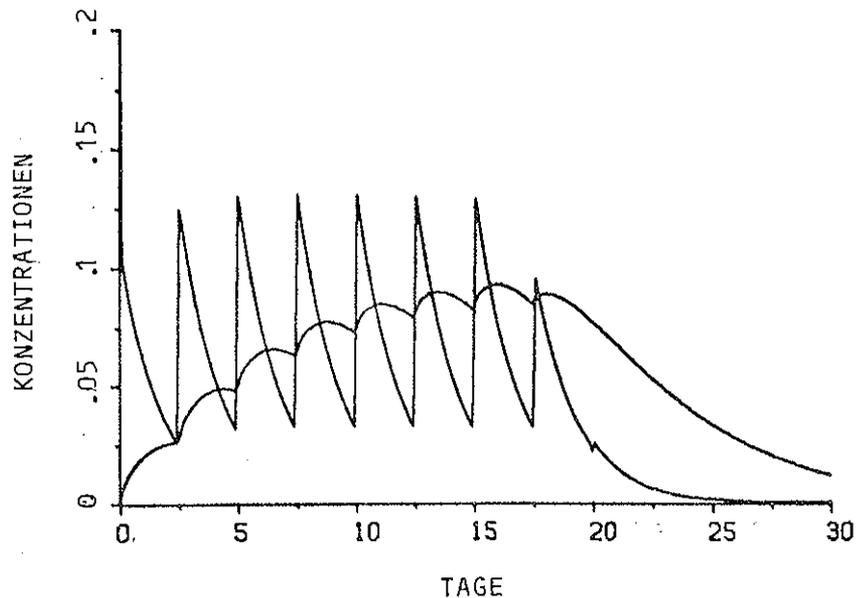


Abbildung 5
Verlauf der Absorption und des Blutspiegels

Bei größeren Modellen ist es jedoch meist unmöglich, die benötigten Reaktionskonstanten rein analog zu ermitteln, sodaß auf digitale Algorithmen Zugriff genommen werden muß. Ein weiteres Problem bildet die Dokumentation, die für ein problemorientiertes Arbeiten und den dabei dauernd benötigten Überblick über den momentanen Zustand des Systems unerlässlich ist.

Das hybride Dialogsystem HOI (Hytran Operations Interpreter) bietet nun nicht nur die Möglichkeit einer digitalen Steuerung der Anlogsimulation, sondern auch einer Verwendung von digitalen Algorithmen ohne Einschränkung der für eine genaue Untersuchung notwendigen Mensch-Maschine-Interaktivitäten. Ein für das jeweilige Problem erstelltes HOI-Programm kann während der Simulation einfach geändert oder erweitert werden und stellt nicht nur eine ständige Dokumentation des Modellzustandes dar, sondern ermöglicht auch eine numerische und graphische Dokumentation der Untersuchungsergebnisse.

Die folgende Abbildung zeigt ein Blockdiagramm und einen Ausschnitt des für das vorliegende Problem erstellten HOI-Programmes. Das Programm besteht im wesentlichen aus einer Reihe von Unterprogrammen (Blöcken) mit festgelegten Funktionen, die in beliebiger (natürlich sinnvoller) Reihenfolge exekutiert werden können, sowie einem Hauptblock, der diese Unterprogramme in einer für einen Standardfall definierten Reihenfolge exekutiert.

1	Initialisierung	2.010 2.015 2.016	1.01; 10; 12; 14; 17; KALT;	"KOMMENTAR" "PARAMETERBERECHNUNG"
2	Hauptblock	2.017 2.020	20; 15;	"ACHSENKREUZ" "DAM SETZEN"
10	Problemorientierter Kommentar	2.035 2.040 2.050	22; 21; KALT; 2.02.	"AUSLESEN, ZEICHNEN"
6	Berechnung der Reaktionskonstanten	10.010 10.011 10.020	"REAKTIONSGESCHWINDIGKEITSKONSTANTEN: " " " " K1 ABSORPTIONSPARTIMENT (A) - BLUT (B)";	
25	Setzen der Digital- Analog-Multiplizierer mit den Reaktions- konstanten	10.030 10.040 10.050 10.060	" K2 BLUT (B) - ALLG. GEWEBE (G)"; " K3 ALLG. GEWEBE (G) - BLUT (B)"; " K4 BLUT (B) - URIN (U)"; " A0 DOSIS (A)";	
22	Auslesen einer Kurve (Absorption im Magen, Blut- oder Gewebe- spiegel oder Urin- ausscheidung)	22.004 22.010 22.020 22.030 22.038 22.040 22.100	"AUSLESEN EINER FUNKTION" @IC, M; Y(I)=IAD00, I=1 END=1SLOO, @PP, M; VAL(@SLOO)? 22.1. I=I27? 22.1. I=I+1, Y(I)=IAD00, 22.03. @IC, M;	
21	Zeichnen der zuletzt ausgelesenen Kurve in einem Achsenkreuz	21.004 21.006 21.010 21.100 21.200 21.240 21.250 21.260 21.270	"ZEICHNEN EINER KURVE" 1, 1, 1, P; FACX, FACX, 5, P; .0, Y(I), 3, P; K=2, 1, I! X=K/I, X, Y(K), 2, P; IND=1? 21.27. 0, Y(I), 999, P; 1, 1, 1, P; 0, 20, 999, P;	" , \$NI, JP:
20	Zeichnung eines Achsenkreuzes	24.005 24.006 24.007 24.010 24.020 24.030 24.040	"AUTOMATISCHE PARAMETERAENDERUNG": KN=1, PA=.1, PE=.999, PST=.1 IND=1 KN, PA, PE, PST- 20; JP=PA, PST, PE! K(KN)=JP, 15; 22; 21; IND=0	
24	Verlauf einer Konzen- trationsvariablen für mehrere Werte einer vorzugebenden Reaktionskonstanten			

Abbildung 6

HCI stellt jedoch nicht die einzige Möglichkeit einer digital unterstützten Analogsimulation dar. Mit Hilfe des HYBRID LINKAGE Programmpakets kann der Analogrechner auf FORTRAN-Ebene bedient werden, wobei der vom Benutzer geforderte Programmieraufwand durch Verwendung von Programmen der Programmbibliothek zusätzlich reduziert wird. Die vorliegenden Zeichnungen zum Beispiel wurden mit dem Bibliotheksprogramm HYPLOT erzeugt.

Mit der Installation eines AutoPATCH und eines höheren hybriden Sprachprozessors vereinfacht sich die Verwendung eines Hybridrechners wesentlich, da dann eine Programmierung auf Problemebene mit minimaler Kenntnis der hybriden Simulationstechnik ermöglicht wird.

Literatur: H. Röpke, J. Riemann, "Analogcomputer in Biologie und Medizin", Springer-Verlag Berlin/Heidelberg 1969

D. Solar

EAI USERS GROUP MEETING 1977

Das EAI Users Group Meeting fand dieses Jahr an der Technischen Universität Braunschweig, Institut für Flugführung, statt. Es nahmen etwa 50 Benutzer von EAI Hybridrechenanlagen aus 7 Ländern Europas daran teil. Erstmals waren auch zwei Benutzer aus Asien anwesend (Institut für Technologie, Taiwan). Fachleute aus den europäischen und amerikanischen Büros der Firma EAI standen zur Diskussion der Probleme der Benutzer zur Verfügung. Der Präsident der amerikanischen Users Group berichtete über die Aktivitäten von EAI Benutzern in Amerika. Ein Mitglied der europäischen Users Group wird das amerikanische Users Group Meeting Anfang November in West Long Branch besuchen.

Die Vorträge der Benutzer enthielten Berichte über Software und Hardwareeigenentwicklungen. Interessant waren Vorträge über selbstentwickelte Verbindungen eines EAI 680 mit einer PDP 11/45, eines PACER 100 mit einem Versatec Printer/Plotter, sowie Digital-to-digital Verbindungen (Data Pacer mit PACER 100).

Von der Hybridrechenanlage der Technischen Universität Wien berichtete Frau Husinsky über das Job Accounting des Betriebssystems JCS/TS 7 (siehe auch Seite 19), das einzige automatische Job Accounting System an europäischen EAI Hybridrechnern. Herr Solar gestaltete einen Vortrag über ein von ihm entwickeltes Programmsystem zur automatischen on-line Skalierung von Anlogschaltungen, der großes Interesse fand.

EAI organisierte eine Vorführung des EAI 2000 Systems (ein Analogrechner mit einem eingebauten mikroprogrammierten Controller, der von einem Terminal bedient werden kann) und informierte über Auto-Patch Systeme und ECSSL (siehe auch Seite 4).

Die Aktivitäten der europäischen Users Group sollen in einem Users Group Newsletter publiziert werden. Den Abschluß des Meetings bildete eine Vorführung des Flugsimulators am Institut für Flugführung.

I. Husinsky

I M A C S (AICA) SYMPOSIUM

über

"SIMULATION VON REGELUNGSSYSTEMEN"

UNTER BESONDERER BERÜCKSICHTIGUNG

VON MODELLBILDUNGEN UND REDUNDANZ

27.,-29. SEPTEMBER 1978

TECHNISCHE UNIVERSITÄT WIEN

Das Symposium wird von I M A C S (International Association for Mathematics and Computers in Simulation) veranstaltet. Die Mitglieder des wissenschaftlichen Komitees sind:

a.o.Univ.Prof.Dr.I. Troch
o.Univ.Prof.Dr.A. Weinmann
Dipl.Ing.W. Kleinert

Es sind Beiträge erwünscht, die sich mit theoretischen und/oder praktischen Aspekten der Simulation beschäftigen. Dabei spielt es keine Rolle, ob die präsentierten Ergebnisse mit Hilfe von Digital-, Analog- oder Hybridrechnern oder mittels Mikroprozessoren erhalten wurden. Kontakte und Diskussionen zwischen den theoretisch interessierten Wissenschaftlern von Universitäten und Forschungsinstituten und jenen Tagungsteilnehmern, die in erster Linie an praktischen Problemen und industriellen Anwendungsgebieten interessiert sind, werden angestrebt. Arbeiten, die geeignet sind, solche Diskussionen anzuregen, werden bevorzugt.

Kurzfassungen

Um eine Auswahl der Beiträge zu ermöglichen, wird gebeten, bis zum 31. Dezember 1977 Kurzfassungen (Englisch oder Deutsch, 2-3 Seiten) an das Organisationskomitee zu senden.

Beiträge

Die ausführliche (endgültige) Fassung der Beiträge muß in englischer Sprache abgefaßt sein und bis zum 31. März 1978 an das Organisationskomitee gesandt werden. "Instruktionen für Autoren" werden nach der Annahme der Kurzfassung an jeden Autor versandt. Die Verständigung über die Annahme oder Ablehnung des eingereichten Beitrags erfolgt bis spätestens 15. Februar 1978. Jeder Autor muß seinen Beitrag persönlich präsentieren.

Publikation

Alle akzeptierten Beiträge, inklusive der Übersichtsvorträge, werden in den "Proceedings" des Symposiums im Druck erscheinen. Die "Proceedings" werden zu Beginn des Symposiums bereits vorliegen.

Konferenzsprache: Englisch

recon

INFORMATIONSSYSTEM

Wir freuen uns, den Lesern unserer Zeitschrift ein neues Service der Abteilung Hybridrechenanlage anbieten zu können: es handelt sich um den in kurzer Zeit bevorstehenden Anschluß an das automatische Informations- und Dokumentationssystem RECDN des Space Documentation Service ESRIN der Europäischen Weltraumorganisation ESA.

Da ein RECON Service bis jetzt in Österreich ausschließlich von der Österreichischen Gesellschaft für Sonnenenergie und Weltraumfragen (ASSA - Wien) und dem Institut für Maschinelle Dokumentation (IMD - Graz) geboten wurde, ist zu hoffen, daß durch den nun möglichen Zugriff direkt an der Technischen Universität die Ansprüche eines neuen Kreises von Interessenten, den wir speziell unter den Universitätsinstituten zu finden hoffen, erfüllt werden können.

Worum handelt es sich bei RECON im Detail?

Die Produktion aktueller Fachliteratur überfordert heute oft bereits die Fähigkeit des Wissenschaftlers oder Ingenieurs, sein Wissen auf dem aktuellen Stand zu halten. Entscheidende Hilfe bietet die Speicherung von Literatur, die durch Spezialisten aus internationalen Fachzeitschriften entnommen, klassifiziert und schließlich auf einem Datenträger interessierten Kunden als Abstract (Kurzfassung) zur Verfügung gestellt wird. Dabei werden die Literaturinformationen nach Wissensgebieten getrennt angeboten. Da der Bezug dieser Informationen für den Einzelbenutzer völlig unrentabel wäre, übernehmen einzelne Rechenzentren (z.B. CIG - Chemie Informationsdienst Graz) die "Aussiebung" von interessierender Literatur für viele Kunden, nachdem mit diesen ein "Profil" seiner Interessen erarbeitet worden ist. Da der Kunde laufend mit neuen Literaturinformationen versorgt wird, eignet sich dieser Literaturdienst speziell für Fachleute, die in ihrem Gebiet eingearbeitet sind und genau wissen, was sie suchen.

Die Situation ist aber völlig anders, wenn es darum geht, sich in ein Gebiet einzuarbeiten oder Literatur über ein Fachgebiet interdisziplinären Charakters zu suchen, weil in diesem Fall oftmals die Fragestellung selbst erst zu erarbeiten ist. Dazu ist ein Dialog Mensch - Datenbanksystem erforderlich, somit eine Rückkopplung von Informationen, die den Fragesteller zum kreativen Überdenken und zur Neuformulierung seines Anliegens zwingen. Diese Möglichkeit bietet RECDN.

Die vorhin erwähnten Datenträger mit aktueller Literatur werden laufend in Speicher eingebracht und bilden in ihren Gesamtheiten sogenannte Datenbanken, deren Einzugsbereich nicht völlig starr, sondern schwerpunktmäßig definiert ist.

Der Zentralrechner befindet sich in Frascati/Italien, in fast jedem anderen Land Europas gibt es bereits Konzentratoren, die über Standleitungen hoher Übertragungsrates mit Frascati verbunden sind. Die Verbindung mit RECON

wird über ein Terminal (und Drucker) mit Modem und telefonische Anwahl eines Konzentrators oder des Zentralrechners selbst hergestellt (Abb.1).

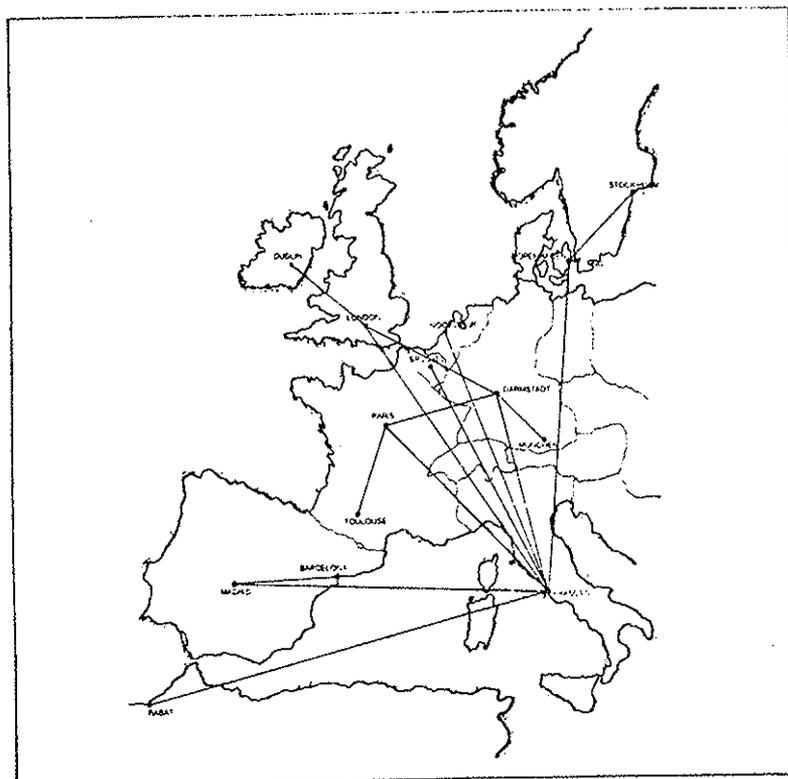


Abb. 1 ESANET-Standleitungen

RECON (REmote CONsole) selbst ist eine hochentwickelte Dialogsprache, die über ein Terminal interaktiv mit dem Benutzer verkehrt, wobei komplizierte Literatursuchstrategien angewendet werden können.

Die optimale Benützung des Systems erfordert eine vorherige Analyse der Fragestellung und fallweise auch Zusammenarbeit von RECON-Operator und Fragesteller am Terminal, wobei der Operator die erforderlichen Rechenzeiten aufgrund seiner Kenntnis der Dateien wesentlich senken kann. Dies ist notwendig, da sowohl Kosten für Dateibenutzung, Systembenutzung, Ausdruck von Ergebnissen sowie für die Benützung von ESANET und dazu noch Telefongebühren anfallen. Die Beschaffung der Literatur selbst erfolgt konventionell, doch können wir mehrere sehr prompt arbeitende Kontaktadressen anbieten (Microfiche-Kopien usw.).

Durch Mittel des Bundesministeriums für Wissenschaft und Forschung wird die Abteilung Hybridrechenanlage in die Lage versetzt werden, für Benützer aus Wissenschaft und Forschung kostenlose Suchläufe durchzuführen und das nötige Operating anzubieten.

Für Interessenten aus Wirtschaft und Industrie wäre (von kostenlosen Probeläufen abgesehen) eine Systembenützung gegen Ersatz der Selbstkosten denkbar.

Folgende Literaturdatenbanken können über RECON angesprochen werden:

Kollektion	Abstract Journal	Wissensgebiete	Gesamte Zahl d. Referenzen	Anzahl der monatlichen Referenzen	seit
NASA	Scientific and Technical Aerospace Reports (STAR) International Aerospace Abstracts (IAA)	Weltraum- und Luftfahrt, Biologie und Medizin bezogen auf Weltraumbedingungen, Elektronik, Chemie und Materialforschung	840 000	7 000	1962
CHEMABS	Chemical Abstracts	Biotechnie, angewandte, physikalische, analytische Chemie, Verfahrenstechnik	2.250.000	35 000	1969
METADEX	Metal Abstracts	theoretische und angewandte Metallurgie, Korrosion, Eigenschaften von Metallen und Legierungen, Pulvermetallurgie	220.000	2.000	1969
COMPENDEX	Engineering Index	Elektronik, Technik und Technologie auf dem Gebiet der Mechanik, Kraftwerksbau, chemische Verfahrenstechnik, Elektrizität, Metallurgie, Instrumente, Bauingenieurwesen, Materialeigenschaften	580 000	8 000	1969
ELECOMPS		Technische Daten und Produktinformation moderner elektronischer Bauteile wie Kapazitäten, Filter, Schalter, Dioden, Transistoren, integrierte Schaltungen usw. für Geräte mit hoher Zuverlässigkeit (hauptsächlich europäische Firmen)	33 000		1970
NTIS	Government Reports Announcements	Naturwissenschaften, techn. Gebiete, Sozialwissenschaft	360.000	4.000	1970
INSPEC	Physics Abstracts; Electrical Engineering and Electronic Abstracts; Computer and Control Abstracts	Physik, Elektrotechnik, Elektronik, Computerwissenschaft, Regelschaltung	770 000	9 000	1971
ALUMINIUM	World Aluminium Abstracts	Aluminium und Legierungen, Technologie, Anwendungen	50.000	500	1968
ENVIRON	Environment Abstracts	Umweltforschung, Energiequellen, Umweltverschmutzung, Fernerkundung	60 000	1.500	1971
PASCAL	Bulletin signalétique	Physik, Elektronik, Elektrotechnologie, Metallurgie, Energiequellen	490.000		1973
ISI	Science Citation Index	Physik, Chemie, Technologie	730.000		1972

Ferner werden (laut ESRIN-News) in diesem Jahr noch folgende Datenbanken installiert werden:

Name	Wissensgebiet	Gesamtzahl der Referenzen	seit
BIOSIS	Biologie	1 Million	1972
OCEANIC ABSTRACTS	Ozeanographie	100 000	1964
POLLUTION ABSTRACTS	Umweltforschung	50 000	1970

Zusätzlich ist die Errichtung eines Files geplant, in dem Informationen über das terrestrische Bildmaterial von Forschungssatelliten (z.B. LANOSAT) enthalten sind, das von europäischen Bodenstationen empfangen wurde. Für den Terminalbenutzer sind dabei sogenannte "QUICK LOOK" Schemazeichnungen am Bildschirm verfügbar.

(Abbildung und Tabelle wurden einer Broschüre des Bundesministeriums für Wissenschaft und Forschung entnommen).

H. Hummer

SICOS - EINE PROGRAMMIERSPRACHE ZUR SIMULATION DYNAMISCHER SYSTEME FÜR DIE RECHENANLAGE NOVA 2/10

Dieser Beitrag wurde der Redaktion vom Prozeßrechnerlabor der Fachhochschule Frankfurt am Main (Prof. P. Sokolowsky) freundlicherweise zur Verfügung gestellt.

Im Rahmen einer Graduierungsarbeit an der Fachhochschule Frankfurt am Main hat Herr Siegfried Hahn ein Simulationsprogramm für dynamische Systeme entwickelt. Der dabei grundsätzliche Gedanke war, wie weit man ein komfortables Simulationsprogramm auf einem Kleinrechner wie dem an der Fachhochschule Frankfurt am Main vorhandenen NOVA 2/10 realisieren könnte.

1. Einleitung:

Simulationen sind in der heutigen Forschung und Technik aus Gründen der Kostenersparnis und vielen anderen Gründen (z.B. Gefahrlosigkeit) unentbehrlich.

Die am Digitalrechner NOVA 2/10 installierte Sprache simuliert beliebige Systeme, die sich durch gewöhnliche Differentialgleichungen und algebraische Gleichungen hinreichend genau beschreiben lassen.

2. Eingabesprache:

SICOS (Simulation of Continuous System) benötigt dreierlei Arten von Informationen zur Iteration:

- a) Steueranweisungen: sie steuern den Simulator und definieren als Input: Teletype, Papertype, Card Reader, als Output: Teletype, Lineprinter oder Plotter.
- b) Eingabe der Konfiguration des zu simulierenden Systems über "Funktionsblöcke"
- c) Zahlenwerte der Parameter und Anfangsbedingungen in Form von Konstanten für die "Funktionsblöcke", Anfangsbedingungen.

Dazu ist im folgenden Näheres auszuführen:

- ad a) Die Eingabe kann wahlweise über Konsole interaktiv oder über Lochkarten bzw. Lochstreifen erfolgen. Das Eingabegerät wird vom Benutzer durch eine Steueranweisung definiert. Die Eingabereihenfolge der Funktionsblöcke ist beliebig.

Die Ausgabe kann auf Schnelldrucker, Konsole oder Plotter erfolgen, wobei die Ausgabegeräte während des Programmlaufes beliebig gewechselt werden können (durch Steueranweisungen). Bei Schnelldruckerausgabe wird eine Liste von vier Variablen, die vom Benutzer bestimmt werden, als Funktion der ebenfalls ausgedruckten unabhängigen Veränderlichen ausgegeben, sowie eine graphische Darstellung der vierten Variablen angefertigt, bei Konsolenausgabe wird nur eine Variable graphisch dargestellt.

Bei der Plotterausgabe wird eine Variable als Funktion der unabhängigen Veränderlichen in Form einer Kurve in einem Koordinatenkreuz gezeichnet. Die Einteilung der Achsen ist variabel und vom Benutzer bestimmbar.

ad b) Es gibt 26 Typen von Funktionsblöcken, die durch symbolische Buchstaben definiert werden, wobei dieselben Blocktypen natürlich mehrfach definiert werden können. Eine Auswahl der wichtigsten:

CODE	Funktion	Eingänge	Parameter
I	Integrator	3	3
D	Ableitung	1	0
T	Nadelimpulse	1	1
J	Zufallszahlengenerator	0	0
F	Funktionsgenerator	1	2
Z	T/S Einheit (Halteglied)	2	0
L	Begrenzer	1	2
S	Unempfindlichkeit (Dead Space)	1	2

Die Konfiguration des Systems wird nun durch Anweisungen der Form

```

BLOCK = XXX      XXX ... Blocknummer (1-200)
TYPE  = Y        Y ... Funktionscode
INPUT = ZZZ      ZZZ ... Nummer des Blocks, der an den
                  Eingang gelegt wird (maximal drei
                  Eingänge)
  
```

interaktiv festgelegt.

ad c) Die Zahlenwerte der Parameter und Anfangsbedingungen werden ebenfalls interaktiv definiert:

```

BLOCK = XXX      XXX ... Blocknummer, für welche die
PAR N = W        W ... Wert des Parameters
                  N ... Nummer des Parameters (max 3)
  
```

Die Funktionsblöcke können um (maximal 5) selbstdefinierte Funktionen erweitert werden.

3. Forderungen an das Programm SICOS:

Diese lassen sich in folgende Punkte gliedern:

- Aufstellen der internen Berechnungsfolge durch einen Sortiervorgang
- Bereitstellen der Funktionsblöcke
- Ausführung der Zeititeration, Integration und Ausgabe der Ergebnisse
- Interaktives Arbeiten und Unterbrechung des Simulationslaufes zu beliebigen Zeiten ohne Zerstörung des Modells.

4. Das Programmkonzept enthält also folgende drei wichtige Komponenten:

A. Übersetzer

er führt folgende Arbeiten durch:

- o Einlesen der Steueranweisungen
- o Einlesen der Konfigurations- und Parameteranweisungen und deren Prüfung auf Vollständigkeit und formale Richtigkeit
- o Sortieren der Strukturanweisungen

B. Funktionsbibliothek

Diese ist eine Sammlung von Unterprogrammen, welche die Funktion der Blöcke nachbilden.

C. Simulator

Er sorgt für die Durchführung der Iteration.

5. Besonderheiten des interaktiven Betriebes:

o Schalter:

Steueranweisungen werden über sogenannte "Programmschalter" definiert. Vor Beginn jedes Simulationslaufes hat der Benutzer die Möglichkeit, diese zu verändern.

Bedeutung der Schalter:

KONFIGURATIONS EINGABE	1
PARAMETER EINGABE	2
FUNKTIONS GENERATOR EINGABE	3
INTEGRATIONS INTERVALL UND GESAMTLAUFZEIT	4
DRUCK INTERVALL	5
DRUCK VARIABLE	6
ERSTELLE NEUES PLOTTERFELD	7
PLOTTER X - ACHSE	8
PLOTTER Y - ACHSE	9
UNTERRÜCKE DAS DRUCKEN DER BLOCK AUSGANGSWERTE	11
KONFIGURATIONS AUSGABE AN DRUCKER ODER LOCH-STREIFENSTANZER	12
ABFRAGE DER BLOCK AUSGANGSWERTE	13
NEUSTART AM UNTERBRECHUNGSPUNKT	14
UNTERRÜCKE DAS DRUCKEN DER EINGABEDATEN	15
WAHL DES SIMULATIONSAUSGABE GERÄTES	16
WAHL DES EINGABE GERÄTES	17
PROGRAMM ENDE	18

Konfigurationserweiterungen können also z.B. durch Setzen von Schalter 1 vorgenommen werden. Während das Setzen von Schalter 12 das Ausstanzen der momentanen Modellkonfiguration auf Lochstreifen bewirkt.

Es können ein oder mehrere Schalter gesetzt werden.

o Simulationslaufunterbrechung

Durch Eingabe eines CR (Carriage Return) am Terminal kann der Rechengvorgang jederzeit unterbrochen werden.

Das Modell kann nun nach Wunsch verändert und durch Initialisieren von Schalter 14 der Simulationsverlauf z.B. wieder fortgesetzt, oder, wenn gewünscht, bei $t=0$ neu gestartet werden.

Anwendungsbeispiel:

Als einfaches Beispiel dient ein Feder-Masse-Schwingungssystem mit einem Dämpfungsglied.

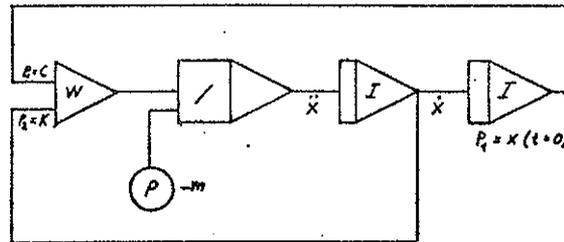
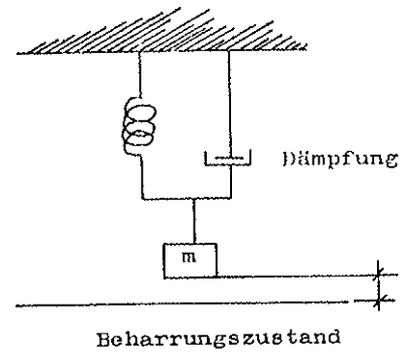
Die durch eine äußere Kraft einmal ausgelenkte und so in gedämpfte Schwingungen versetzte Masse schwingt, bis das System durch die Einwirkung des Dämpfungsgliedes wieder in Ruhelage kommt. Nun können die Auswirkungen verschiedener Feder- und Dämpfungskonstanten auf das System interaktiv untersucht werden. Diese Problemstellungen tauchen z.B. bei der Optimierung einer Kraftfahrzeugfederung (Stoßdämpfer und Spiralfeder) auf.

Das physikalische Originalmodell wird durch

$$\ddot{x} = -\frac{1}{m} \cdot (c\dot{x} + kx)$$

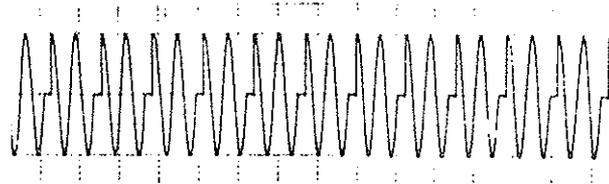
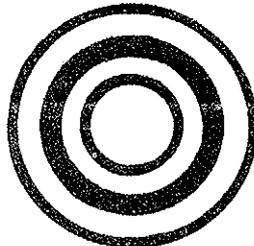
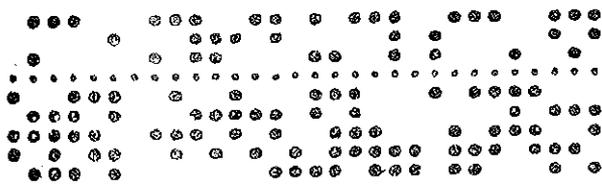
- c ... Dämpfungskonstante
- k ... Federkonstante
- m ... Masse
- x ... Auslenkung

beschrieben.



Beim Probelauf sind erste Reaktionen des Modells sofort erkennbar, und es kann korrigierend (optimierend) eingegriffen werden.

TIME	OUTPUT 3	OUTPUT 2	OUTPUT 1	OUTPUT 4
0.00	0.0000	1.0000	-10.0000	-10.0000
0.40	0.3835	0.9156	-9.1552	-9.9221
0.80	0.7315	0.8235	-8.2348	-9.6977
1.20	1.0416	0.7259	-7.2587	-9.3417
1.60	1.3118	0.6247	-6.2461	-8.8695
2.00	1.5418	0.5216	-5.2155	-8.2974
2.40	1.7290	0.4194	-4.1849	-7.6419
2.80	1.8759	0.3168	-3.1676	-6.9194
3.20	1.9828	0.2181	-2.1808	-6.1462
3.60	2.0509	0.1237	-1.2363	-5.3381
4.00	2.0823	0.0346	-0.3455	-4.5101
4.40	2.0794	-0.0483	0.4822	-3.6765
4.80	2.0447	-0.1239	1.2387	-2.8506
5.20	1.9812	-0.1918	1.9188	-2.0444
5.60	1.8922	-0.2516	2.5156	-1.2689
6.00	1.7810	-0.3029	3.0286	-0.5335
6.40	1.6510	-0.3456	3.4557	0.1537
6.80	1.5057	-0.3797	3.7968	0.7855
7.20	1.3494	-0.4054	4.0533	1.3567
7.60	1.1824	-0.4228	4.2279	1.8631
8.00	1.0111	-0.4325	4.3241	2.3019
8.40	0.8374	-0.4347	4.3464	2.6717
8.80	0.6643	-0.4301	4.3003	2.9719
9.20	0.4942	-0.4192	4.1917	3.2034
9.60	0.3296	-0.4028	4.0271	3.3679
10.00	0.1726	-0.3814	3.8132	3.4680
10.40	-0.0251	-0.3558	3.5573	3.5072
10.80	-0.1116	-0.3267	3.2664	3.4894
11.20	-0.2360	-0.2948	2.9476	3.4195
11.60	-0.3471	-0.2609	2.6082	3.3024
12.00	-0.4444	-0.2255	2.2548	3.1435
12.40	-0.5274	-0.1894	1.8939	2.9466
12.80	-0.5959	-0.1532	1.5317	2.7234
13.20	-0.6500	-0.1174	1.1738	2.4737
13.60	-0.6900	-0.0825	0.8254	2.2092
14.00	-0.7162	-0.0492	0.4912	1.9235
14.40	-0.7295	-0.0176	0.1751	1.6339
14.80	-0.7305	0.0120	-0.1195	1.3415
15.20	-0.7202	0.0390	-0.3695	1.0518
15.60	-0.6997	0.0633	-0.6327	0.7667
16.00	-0.6700	0.0848	-0.8475	0.4924
16.40	-0.6323	0.1033	-1.0328	0.2317
16.80	-0.5877	0.1188	-1.1879	-0.0126
17.20	-0.5376	0.1313	-1.3129	-0.2378
17.60	-0.4831	0.1409	-1.4061	-0.4421
18.00	-0.4253	0.1475	-1.4744	-0.6238
18.40	-0.3655	0.1513	-1.5129	-0.7820
18.80	-0.3046	0.1526	-1.5252	-0.9168
19.20	-0.2438	0.1514	-1.5131	-1.0257
19.60	-0.1839	0.1479	-1.4707	-1.1111
20.00	-0.1257	0.1425	-1.4243	-1.1729
20.40	-0.0701	0.1353	-1.3522	-1.2120
20.80	-0.0177	0.1265	-1.2648	-1.2294
21.20	0.0310	0.1165	-1.1647	-1.2266
21.60	0.0754	0.1055	-1.0545	-1.2052
22.00	0.1152	0.0937	-0.9365	-1.1669
22.40	0.1503	0.0813	-0.8132	-1.1136
22.80	0.1803	0.0687	-0.6869	-1.0474
23.20	0.2052	0.0560	-0.5598	-0.9701
23.60	0.2251	0.0434	-0.4339	-0.8839
24.00	0.2399	0.0311	-0.3109	-0.7987
24.40	0.2500	0.0193	-0.1927	-0.6926
24.80	0.2554	0.0081	-0.0885	-0.5913



benützerforum

PSEUDOZUFÄLLIGE BINÄRSIGNALE UND IHRE EIGNUNG ZUR IDENTIFIKATION LINEARER EINGRÖSSENSYSTEME

P.Kopacek u. P.Nowotny
Institut für Wasserkraftmaschinen und Pumpen,
Abteilung Regelungstechnik

1. Problemstellung:

Ziel der vorliegenden Untersuchung war es, verschiedene pseudo-zufällige Binärsignale (PZBS) auf ihre Eignung zur experimentellen Identifikation linearer Eingrößensysteme zu prüfen.

Eine pseudozufällige Binärfolge ist eine zweiwertige Zahlenfolge, deren Zahlen (z.B. 0 und 1) sich nach einer bestimmten Periode wiederholen. Ein Binärsignal stellt die zu einer Binärfolge äquivalente Aufeinanderfolge von positiven und negativen Signalamplituden (im Fall der Einheitssignale von +/- Einsen) dar, zwischen denen ein Wechsel nur zu bestimmten äquidistanten Zeitpunkten (Δt) erfolgen kann. Ein PZBS erscheint als reine Zufallsfolge, wenn die Beobachtungszeit kleiner als die Periodendauer ($N\Delta t$) gewählt wird. Die exakte Reproduzierbarkeit und die einfache Erzeugung dieser Signale sind große Vorteile gegenüber den analogen Zufallssignalen. Bei der experimentellen Identifikation wird das zu untersuchende System mit einem Eingangs-PZBS beaufschlagt und aus Meßwerten des Ausgangssignals das Systemverhalten (mathematisches Modell) bestimmt.

Die Versuche wurden an der Hybridrechenanlage der TU Wien durchgeführt. Am Analogteil (AR) des Rechners wurde stellvertretend für lineare Eingrößensysteme ein Verzögerungsglied 2. Ordnung simuliert und mit verschiedenen PZBS, die im Logikteil erzeugt wurden, beaufschlagt. Der Digitalteil verarbeitete Eingangs- und Ausgangssignalmeßwerte des Verzögerungsgliedes und berechnete die Korrelationsfunktionen sowie Ortskurven und Frequenzgänge. Anhand des Vergleiches der aus dem Versuch gewonnenen Daten und Kurven und der vorausberechneten theoretischen Werte konnte eine Beurteilung der einzelnen PZBS vorgenommen werden.

2. Grundlagen des Identifikationsverfahrens

Die Systemidentifikation erfolgte mittels der Korrelationsanalyse [1]. Bei Verwendung von PZBS als Eingangssignale sind die berechneten Kreuzkorrelationsfunktionen (KKF)

annähernd proportional zur Gewichtsfunktion. Eine exakte Übereinstimmung beider Kurven (bis auf einen konstanten Faktor) wäre nur bei Verwendung von Zufallssignalen mit unendlich großer Bandbreite ("weißes Rauschen") als Eingangssignale möglich. Die besonders gute Eignung von PZBS zur Systemidentifikation mittels Korrelationsanalyse resultiert aus ihrer engen statistischen Verwandtschaft zum weißen Rauschen, was daraus ersichtlich ist, daß sich die Autokorrelationsfunktionen (AKF) nur wenig von der Deltafunktion, der AKF des weißen Rauschens, unterscheiden. Für die Systemidentifikation im Frequenzbereich gilt, daß die Werte für den Frequenzgang F als Quotient aus Kreuzleistungsspektrum und Leistungsspektrum zur jeweiligen Frequenz berechenbar sind; letztere Funktionen werden durch WIENER-KHINTCHINE-Transformation der Kreuzkorrelationsfunktion und der Autokorrelationsfunktion gebildet.

3. Versuche [2]

Die verwendeten PZBS sowie ihre wichtigsten Eigenschaften sind in Tab. 1 zusammengestellt. Die Erzeugung der PZBS erfolgte im Logikteil des AR, weil dadurch eine Verkürzung der Versuchsdauer durch eine Zeittransformation (im Verhältnis 1:1000) möglich war. Da die Elemente der parallelen Logik mit 10^6 getaktet sind, wird durch die Erzeugung der PZBS über eine Rückkopplungslogik gewährleistet, daß zu Beginn jedes Bit-Intervalls (dessen Dauer mit 10^{-3} bzw. $2 \cdot 10^{-3}$ sec festgelegt wurde) das richtige Signal am Eingang des Verzögerungsgliedes anliegt. Ausgangspunkt für die Erzeugung der PZBS ist ein n -stufiges Schieberegister mit Rückkopplungslogik. Die Rückkopplungslogik ist so zu berechnen, daß sie aufgrund der Kombination der letzten n Werte der Folge den nachfolgenden in eindeutiger Weise festlegt. Dieser neue Wert wird beim nächsten Taktimpuls in die erste Stufe des Schieberegisters eingeschrieben. Nach den Gesetzen der Schaltalgebra stehen bei einer Folge mit N Werten zur Ermittlung der Schaltfunktion der Rückkopplungslogik N Gleichungen in n Variablen zur Verfügung. Mit

Nr.	Bezeichnung, Kurzbezeichnung,	mögliche Periodenlängen $N =$	gewählte Per. Länge N	Verlauf der Folge	Verlauf der AKF
1	M-Folge (MS)	$2^n - 1, n \in \mathbb{N}$ 3, 7, 15, 31, 63, ...	31	11111 00011 01110 10100 00100 10110 0	
2	Doppelprim-Folge (Twinprime Sequence) (DPS)	$p_1 \cdot p_2$ $p_1 \in \mathbb{N}^2, p_2 = p_1 + 2$ p_1, p_2 Primzahlen 15, 35, 143, ...	35	11111 01110 01000 01010 11001 00101 11000	
3	Quadrat.Res.-Folge (Quadratic Residue/Legendre Sequence) (QRS)	$4 \cdot k - 1, k \in \mathbb{N}$ N Primzahl 3, 7, 11, 19, 23, 31, 43, ...	31	00001 11010 10001 11101 10111 00100 1	
4	Hall-Folge (HS)	$4 \cdot k^2 + 27, k \in \mathbb{N}$ 31, 43, 63, 127, ...	43	00000 11011 00111 01100 00111 10111 10010 10101 001	
5	Volladjustierte M-Folge (VAMS)		wie MS		
6	N-Folge (Inverse-Repeat Sequence) (NS)	$2 \cdot (2^n - 1), n \in \mathbb{N}$ 6, 14, 30, 62, ...	30	11111 01001 00011 00000 10110 11100	
7	Barker-Folge (BS)	9, 12, 15, 21, 33, 39	39	11111 00110 10111 11100 11010 10000 01100 1010	
8	Folge Maximaler Orthogonalität (MOS)	$8 \cdot n, 8 \cdot n + 4, n \in \mathbb{N}$ $N \neq 32, 44, 68, 72, 80$ $N \leq 80$ 8, 12, 16, 20, 24, ...	28	11111 01100 01011 00001 00111 010	
9	D-Folge (Pluckett-Burman Sequence) (DS)	$4 + 4 \cdot n, n \in \mathbb{N}$ $n \leq 24$ $N \neq 92$ 8, 12, 16, 20, 24, ...	32	11111 00110 10010 00001 01011 10110 00	

Tabelle 1

Hilfe einer graphischen Darstellung in erweiterten KARNAUGH-Diagrammen ist es möglich, diese disjunktive Normalform der Schaltfunktion zu minimieren, wobei eventuell redundante Terme über das Verfahren von QUINE und McCLUSKEY eliminiert werden können.

Abb. 1 zeigt die Schaltung des Analogteiles. Ein Teil der Schaltung im analogen Programmierfeld links von der Linie A-A diente zur Erzeugung des Signals aus der im Logikteil erzeugten Folge von Nullen und Einsen. Der D/A

Schalter 39 wird von der Originalfolge gesteuert, Schalter 34 hingegen von der negierten Folge. D/A Schalter 44 wird geöffnet, wenn die Gefahr des Übersteuerns der Integrierer durch zu lange gleichbleibenden Eingangswert besteht.

Der rechts von der Linie A-A befindliche Teil der Schaltung stellt die eigentliche Differentialgleichung des Verzögerungsgliedes 2. Ordnung dar. Mit den Bezeichnungen x für das Eingangssignal und y für das Ausgangssignal lautet diese:

$$T^2 \ddot{y} + 2DT \dot{y} + y = K_0 \cdot x$$

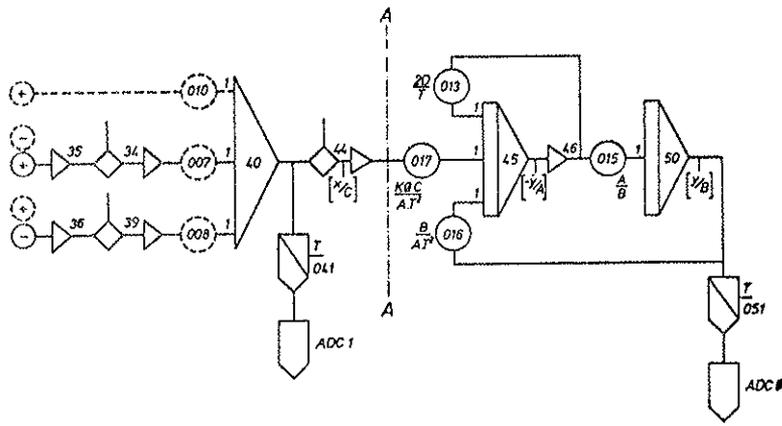


Abbildung 1

4. Versuchsergebnisse

Als Vergleichsgrundlage für die untersuchten PZBS diente eine Folge maximaler Länge mit der Periodenlänge $N \approx 31$. Nachdem sich die Bewertung der einzelnen Versuchsergebnisse mit Hilfe einer zu berechnenden Kennzahl teilweise als sehr schwierig herausstellte (z.B. die Beurteilung der Güte der Identifikation im Frequenzbereich) wurden nach ausgewählten Kriterien "Bewertungsnoten" vergeben, und zwar auf der Grundlage der Abweichungen der nach der Korrelationstheorie berechneten Werte im Zeit- und Frequenzbereich von den auf Grund der bekannten Daten des Verzögerungsgliedes vorausgerechneten Werten. In einer abschließenden Gesamtbeurteilungstabelle konnten zur Gewährleistung einer möglichst umfassenden Bewertung auf diese Weise sehr verschiedene, schwer objektivierbare Gesichtspunkte (wie z.B. die Störung des Systemausgangs durch das PZBS und die Schwierigkeit der Erzeugung des Signals) gleichermaßen berücksichtigt werden.

Wie auch aus dem Logik-Schaltplan (Abb.2) ersichtlich wurden im Rahmen von sieben Einzelversuchen die Ergebnisse jeweils zweier PZBS-folgen miteinander sowie mit den Gewichtsfunktionswerten verglichen. Eine Umschaltmöglichkeit zwischen den beiden PZBS war durch Controlline (1) gegeben. Zur Erzeugung bestimmter PZBS wurde der Schaltungsblock II bzw. III an Stelle von I gesetzt; ebenso dient der Schaltungsblock A zur Realisierung des Bitintervalls $\Delta t = 2$ sec, Schaltungsblock B hingegen wird für das Bitintervall $\Delta t = 1$ sec eingesetzt. Die Übergabe der Werte vom AR an den DR erfolgte über die T/S Einheiten 041 und 051; die Ausgabe der Kurven und Tabellen erfolgte am Schnelldrucker.

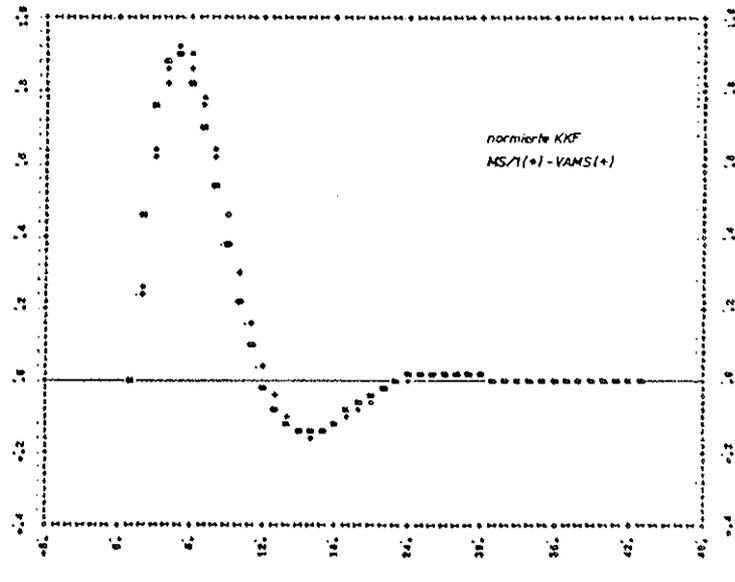
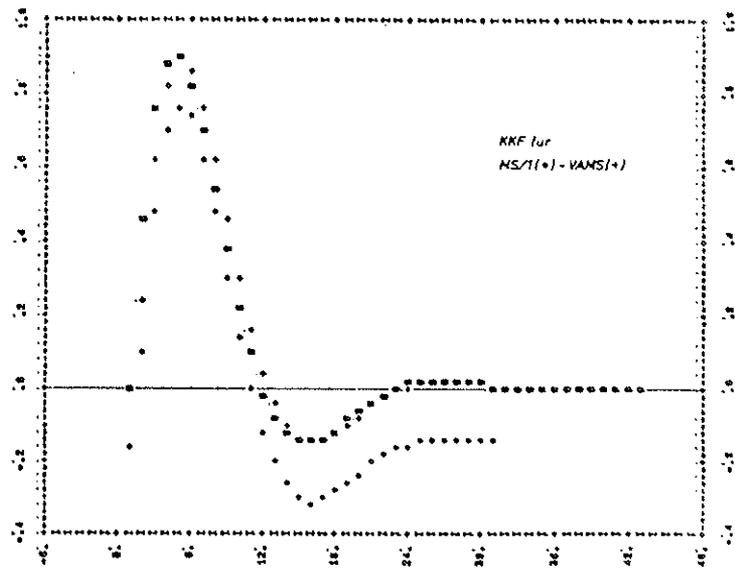


Abbildung 3

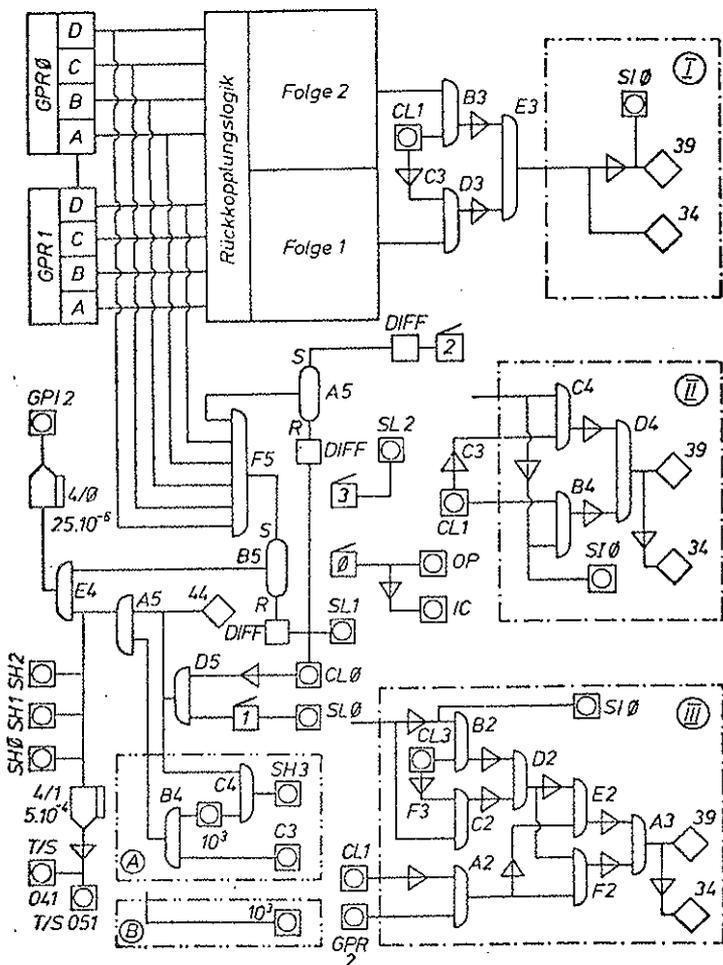


Abbildung 2

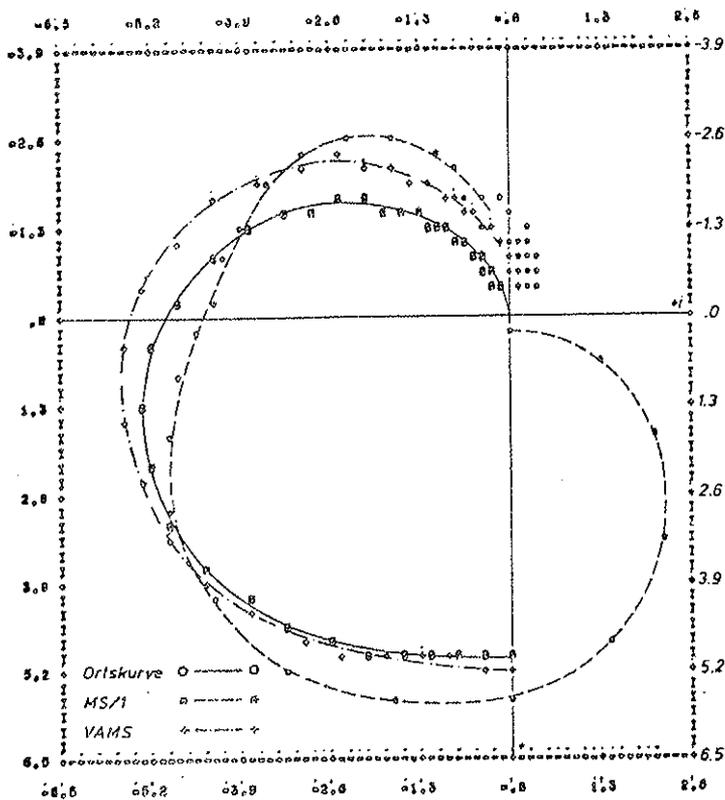


Abbildung 4

Abb.3 und 4 zeigen typische Versuchsergebnisse. In Abb.3 sind die Kreise die genauen Werte der Gewichtsfunktion, die Sterne (Einheits-M-Signal) und Kreuze (Volladjustiertes M-Signal) die der gemessenen KKF's. In Abb.4 sind die entsprechenden Ortskurven dargestellt. Es zeigt sich die vorteilhafte Anwendung eines volladjustierten Signals an Stelle eines Einheits-M-Signals, wenn auch die Nachteile der komplizierteren Erzeugung und des relativ großen linearen Signalmittelwerts in Betracht gezogen werden müssen. Die Signale der N-Folge sowie der

Folge maximaler Orthogonalität entsprechende Signale erwiesen sich im Gegensatz zur D-Folge und zur Barker-Folge ebenfalls als dem M-Einheitssignal überlegen. Dabei ist zu beachten, daß sich die Erzeugung der Folge maximaler Orthogonalität bei größeren Periodenlängen ungleich schwieriger gestaltet als jene der N-Folge. Die Eignung der verschiedenen Nullfolgen (die AKF der zugehörigen Signale weist für die Verschiebung Δ einen größtmöglichen und für jede andere Verschiebung einen kleinstmöglichen Wert auf) zur Systemidentifikation hängt lediglich von den Signalparametern, der Periodenlänge und dem Bitintervall ab, ist aber vom Typ der Folge bzw. deren Bildungsgesetzen unabhängig.

Summary

Several tests aimed at the comparison of the qualification of various pseudo-random binary signals (P.R.B.S.) for system identification by means of the correlation theory with the results achieved by use of a signal of maximal length (representing the most frequently used P.R.B.S.) as comparative values. For this purpose, a SISO-system was simulated on the hybrid computing system of the TU Vienna. Among the P.R.B.S. in regard, especially the fully adjusted m-signal proved superior to usual signals of maximal length.

Literatur:

- [1] Kopacek, P.: Korrelationsverfahren zur Untersuchung linearer Regelsysteme. In Korrelationstechnik, Lexika Verlag 1977, S 63-80
- [2] Nowotny, P.: Vergleichende Untersuchung über die Eignung verschiedener pseudozufälliger Binärsignale zur Identifikation linearer Eingrößensysteme. Diplomarbeit, TU-Wien, 1977.

AUTOMATISCHE AUSWERTUNG VON VERKEHRS- ZÄHLUNGEN AM HYBRIDRECHNER

Helmut Hummer
Hybridrechenanlage der
Technischen Universität Wien

Institut für Verkehrsplanung
der Technischen Universität Wien

Vom Institut für Verkehrsplanung der TUW werden halbautomatische Messungen des Verkehrsflusses an der Südautobahn durchgeführt. Die teilweise von einer Lichtschranke, teilweise von einer Bedienperson erzeugten Meßwerte werden auf Tonband aufgezeichnet. Die Lichtschranke liefert bei Durchfahrt eines Fahrzeuges jeweils zwei Impulsflanken (Einfahrt/Ausfahrt), der Beobachter bestimmt den Typ des Fahrzeuges und gibt ihn auf der Tastatur eines Zusatzgerätes ein, welches daraus eine Impulskette zur Aufnahme auf der anderen Spur des Tonbandgerätes formt.

Durch die physikalischen Eigenheiten der Magnetaufzeichnung werden die ursprünglich exakten Impulsformen deformiert und erhalten etwa folgendes Aussehen bei der Wiedergabe:

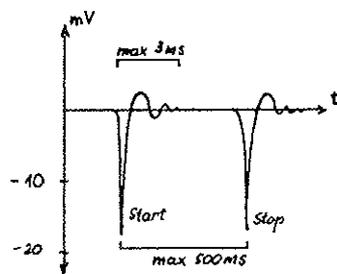


Abb.1 Lichtschranken-Signal

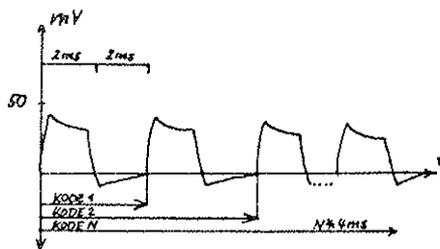


Abb.2 Kodeketten-Signal

Es sollte nun ein Programm am Hybridrechner entwickelt werden, das beim Auswerten der Tonbänder folgende Werte zur Weiterverarbeitung in digitaler Form liefert:

- o Geschwindigkeit des vorbeifahrenden Fahrzeuges
- o Zeiträume zwischen aufeinanderfolgenden Fahrzeugen
- o Dekodierung des Fahrzeugtyps
- o zusätzliche Möglichkeit der Auswertung eines Korrekturkodes, welcher z.B. Falschmessungen oder Irrtum der Bedienperson anzeigt

Die Messungen wurden bis jetzt ausschließlich an einer Fahrbahnspur durchgeführt. Zur Verringerung der Rechenzeiten am System wurden die bespielten Tonbänder mit zumindest verdoppelten Bandgeschwindigkeiten wiedergegeben.

o Geschwindigkeitsmessung

Bei genauer Kenntnis der Länge der Lichtschranke läßt sich die Geschwindigkeitsmessung leicht auf die Messung des Zeitraumes für das Passieren der Lichtschranke zurückführen.

Abb. 3 zeigt die erste Behandlung des von der "Lichtschranken-Spur" des Tonbandes abgenommenen Signals.

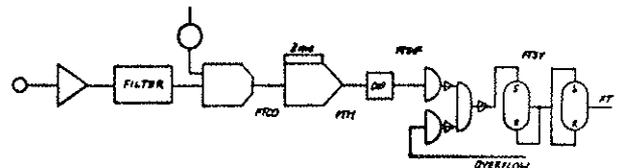


Abb.3 Lichtschranken-Signal, Impulsformung

Ein Filter beseitigt höherfrequente Störungen des Signals, die monostabile Kippstufe verhindert, daß durch das Nachschwingen des Signals Störimpulse erzeugt werden.

Das synchronisierte Signal FTSY steuert ein triggerndes Flip-Flop so, daß es vom Start-Impuls der Lichtschranke gesetzt, vom Stop-Impuls zurückgesetzt wird (Signal FT). Bleibt der Rücksetzimpuls aus, wird ein von einem BCD Counter erzeugtes OVERFLOW RESET Signal zur Simulierung des Rücksetzsignals verwendet.

Mit der steigenden Flanke von FT wird Interrupt \emptyset ausgelöst, der die in der Real Time Clock aufsummierten Impulse von FPCNT (Takt 10^2) zwischen zwei Fahrzeugen ausliest, ins Digitalprogramm übergibt und den Zähler mit einem Wert neu lädt, in dem die Interruptoverhead- und Rechenzeit FTIRZ bereits berücksichtigt ist.

Während dieser Zeit muß Signal FT (Takt 10^5) noch von FTIRZ gesperrt werden, damit der Zählerstand aus FPCNT (Takt 10^2) nicht um Größenordnungen verfälscht wird, bevor er ausgelesen werden kann.

Die Sperre wird durch einen Impuls von Controlline 8 aufgehoben, wenn Interrupt \emptyset fertig gerechnet hat.

Die bis jetzt besprochene Logikschaltung zeigt Abb.4

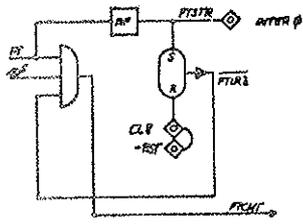


Abb.4 Fahrzeug-Transit-Logik

Hat das Fahrzeug die Lichtschranke passiert, löst das Signal FT Interrupt 1 aus. Darin wird die Zahl der während des Fahrzeugtransits gezählten Impulse FTICNT ausgelesen, in das digitale Programm übergeben und die Real Time Clock neu initialisiert. Das durch FT bereits erlaubte Zählen des Pausentakts (10^2) wird durch FTIRZ allerdings noch während der Interruptrechenzeit gesperrt. FPCNT muß deshalb aber nicht korrigiert werden, da diese Rechenzeit sicher viel kleiner als 10 ms (Taktabstand FPCNT) ist. Signal CDIRZ sperrt den Pausentakt während der Registrierung einer Kodekette.

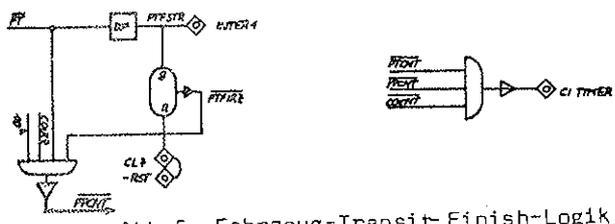


Abb.5 Fahrzeug-Transit-Finish-Logik

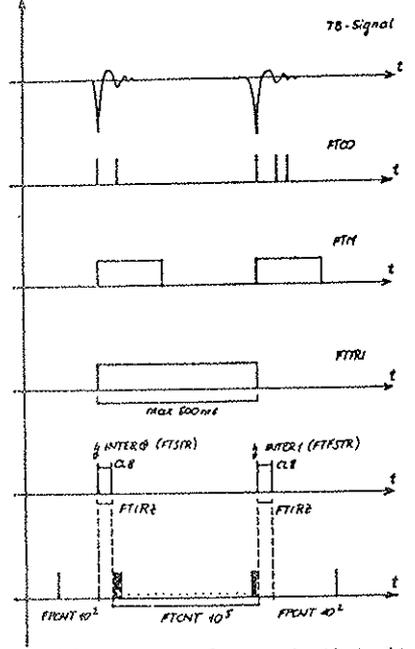


Diagramm 1 : Timing der Geschwindigkeitsmessung

o Dekodierung

Die Zuordnung der Kodeimpulsketten zu den Geschwindigkeitsmeßsignalen wurde so definiert, daß sich alle zwischen zwei Fahrzeugtransits aufgezeichneten Codes auf das nächstfolgende Fahrzeug beziehen. Damit ist in der Praxis eine Fehlerkorrekturmöglichkeit durch beliebig oftmaliges Überspeichern des bereits registrierten Codes bis zum Zeitpunkt des Fahrzeugtransits durch die Lichtschranke gegeben. Ein sogenannter "fehlercode" verändert rückwirkend den Code der vorherigen Messung so, daß diese Messung als fehlerhaft vom Programm erkannt werden kann.

Wird auf der zweiten Tonbandspur eine Kodekette registriert, so wird die laufende Zählung der Pausenimpulse durch Interrupt 2 (ausgelöst von COIRZ) unterbrochen, der Zwischenzählerstand wird gerettet und der Zähler neu initialisiert. Um während der Rechenzeit von Interrupt 2 nicht den ersten Kodeimpuls für die Kodezählung zu verlieren, wird mit dem verzögerten und invertierten Signal von COOIF, das ist COCNT, gezählt. Die Verzögerungszeit von 2 ms reicht als Rechenzeit für Interrupt 2 sicher aus. Der von COIRZ erzeugte Interrupt 3 liest den Wert des Codes aus dem Zähler aus und speichert den vorher geretteten, korrigierten Pausencount ein. Dieser muß um jene Zahl von Taktimpulsen höher gesetzt werden, die während COIRZ und der Rechenzeit von Interrupt 3 vom 10^2 Taktgeber abgegeben, aber nicht registriert wurden (konstant 6 für die 60 ms Dauer von COIRZ). Abb.6 zeigt Erzeugung und Verarbeitung der Kode-Logiksignale.

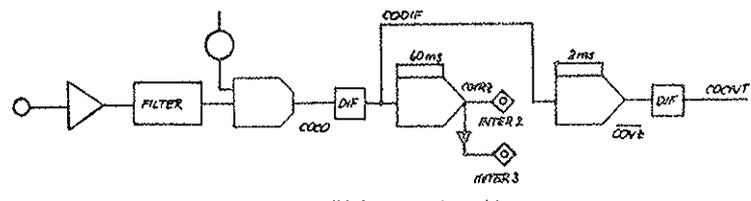


Abb.6 Kodezählungs-Logik

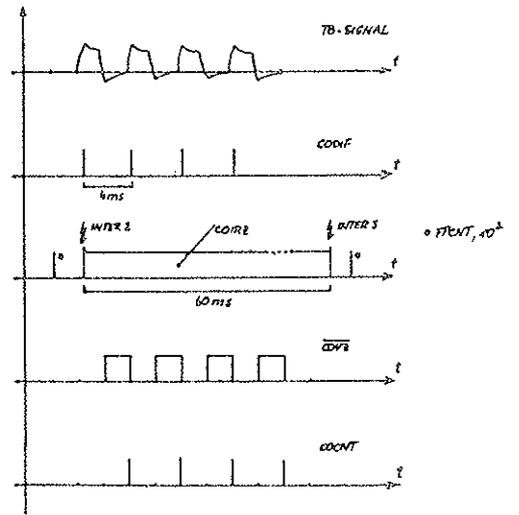


Diagramm 2 : timing der Kodezählung

Die ins Digitalprogramm übergebenen Zählerstände werden on-line in Fahrzeuggeschwindigkeiten, Pausendauern und Fahrzeugtypen umgerechnet und auf Magnetplatte zur weiteren Bearbeitung abgespeichert.

Die für die Auswertung benötigte CPU Zeit ist relativ gering, und die mit höchster Priorität gerechneten Interrupts benötigen ebenfalls wenig Rechenzeit.

Es ist daher ohne weiteres möglich, ein zweites Programm in der anderen Partition von JCS/TS zu exekutieren und die Wartezeiten des Auswerteprogramms intensiv für andere Aktivitäten zu nutzen.

o Summary

Signals from an automatic highway traffic measuring system are recorded on standard tape recorder synchronously with code chains from a code generator operated by a person distinguishing different types of cars. The played back signals (stereo 2-track) are processed first logically and using real time clock and 4 types of interrupts afterwards. Car speeds, dead spaces between cars and types of cars are evaluated on-line in multiprogramming system JCS/TS and stored in digital computer for further statistical processing then.

EIN VERGLEICH EINIGER VERFAHREN DER MERKMALBESTIMMUNG FÜR DIE SPRACHERKENNUNG

II. Die Anwendung der Prädiktorkoeffizienten für die Spracherkennung *)

Romualdas Gudonawitschius
Politechnische Hochschule in Kaunas, UdSSR

Irmgard Husinsky
Hybridrechenanlage
Technische Universität Wien

Einleitung

Die Aufgabe der automatischen Spracherkennung ist es, das physikalische Sprachsignal in eine diskrete Folge von sprachlichen Zeichen umzusetzen. Es ist jedoch heute leider nicht bekannt, welche Verarbeitungsweisen zur Erkennung von sprachlichen Zeichen benötigt werden. Ebenso unbekannt ist es auch, welche sprachlichen Einheiten primär erkannt werden und welche als Ausgangspunkt für das Verstehen einer Nachricht dienen.

Sprachliche Zeichen kann man in verschiedenen Ebenen einordnen. Auf einer unteren Ebene steht die Darstellung durch eine Folge von Lauten, auf einer höheren die Wortfolge und noch höher die Satzfolge. Laute tragen selbst keine Bedeutung, sie sind aber eine Art kleinste Entscheidungseinheit und können dazu dienen, Worte verschiedener Bedeutung zu unterscheiden. Die Verwendung von Lauten als Entscheidungseinheit hat den Vorteil, daß in den meisten Sprachen nur 30 bis 60 verschiedene Laute existieren.

In der vorliegenden Arbeit, die an der Hybridrechenanlage des Rechenzentrums der TU Wien durchgeführt wurde, wird ein System vorgestellt, das in seiner gegenwärtigen Form für die Sprachanalyse isoliert gesprochener Wörter geeignet ist. Es wurde eine Feinanalyse für eine optimale Intervalllänge und optimale Zahl von Prädiktorkoeffizienten für stimmhafte und stimmlose Lautfolgen und Laute, die von zwei Sprechern (männlich und weiblich) gesprochen werden, und eine weitere Reduzierung der Koeffizienten für die Erkennung durchgeführt.

Wesentlich neu daran ist, daß

1. die Prädiktorkoeffizienten als Merkmale für die Sprachanalyse angewandt werden. Die Prädiktorkoeffizienten werden meistens für die Sprachübertragung verwendet.

2. die Berechnung der Prädiktorkoeffizienten nicht für ein ganzes Wort, wie es üblich ist, sondern nur für bestimmte Sprachteile durchgeführt wird. Eine Auflösung des Sprachsignals in Laute ist technisch kaum möglich. Dann wird im Analogrechner eine Aufteilung in Sprachsignale und Pausen, weiters in stimmhafte und stimmlose Segmente und in stationäre und dynamische Segmente versucht. Die Einführung der ersten analogen Stufe der Segmentation schuf die Möglichkeit, die Datenrate und damit den Verarbeitungsaufwand für die nachfolgenden Stufen im Digitalrechner zu reduzieren.

Die Anwendungsmöglichkeiten des Verfahrens sind mit der Analyse der isoliert gesprochenen Wörter nicht erschöpft. Es gibt die Möglichkeit, mit einem leistungsfähigeren Digitalrechner das Verfahren für eine Analyse der fließenden Sprache zu erweitern.

Beschreibung des Systems

Bild 1 zeigt das Blockdiagramm des Systems zur Feinanalyse des Sprachsignals. Die schaltungstechnische Bestimmung der Merkmale für die Sprachanalyse (Unterscheidung von Sprachsignalen und Sprachpausen, von stimmlosen und stimmhaften Lauten) wurde in "Interface" Nr. 9 bereits beschrieben [1].

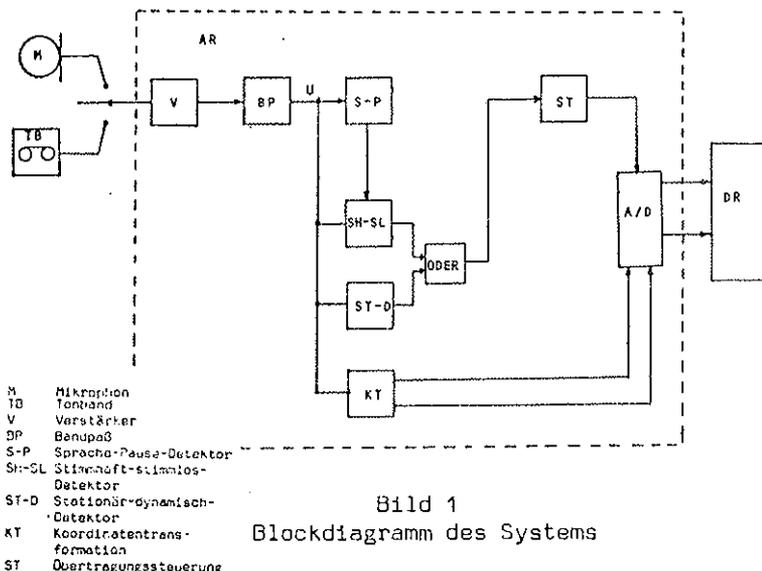


Bild 1
Blockdiagramm des Systems

*) Über diese Arbeit wurde beim IX. Österreichischen Mathematikkongress berichtet, der von 26. - 30. September 1977 in Salzburg stattfand.

Zusätzlich wurde eine Schaltung zur Bestimmung von stationären und dynamischen Sprachbereichen eingebaut. Diese Segmentation des Sprachsignals U am Analogrechner bedeutet eine erhebliche Einsparung von Speicher- und Analyseaufwand für die weitere Verarbeitung.

Datenübertragung

Die Analog/Digital-Übertragung der Sprachsignalanteile erfolgte mit einer Abtastfrequenz von 12 kHz. Die Übertragung wird über zwei Senselines gesteuert (Übertragungsimpuls und Übertragungsende). Es wurde ein Assembler-Unterprogramm geschrieben, das bei jedem Übertragungsimpuls einen ADC ausliest und den Wert auf einem Kernspeicherfeld abspeichert (es stehen maximal 83 μ sec zur Verfügung). Der Hintergrundbetrieb des Betriebssystem JCS/TS 7 muß während der Übertragung gesperrt sein. Der Beginn einer Übertragung ist dem digitalen Übertragungsprogramm nicht bekannt, die Übertragung muß jedoch sofort beim Eintreffen des ersten Transferimpulses einsetzen. Das Versuchsmaterial besteht aus kurzen Worten (z.B. Zahlworte), die entweder in ein Mikrofon gesprochen oder von einem Tonband aufgenommen werden. Die maximale Dauer eines solchen Testwortes ist aus Speicherungsgründen auf etwa 1 sec beschränkt. Nach der Übertragung eines Wortes in den Kernspeicher werden die Daten zur weiteren Verarbeitung auf Platte abgespeichert und es kann das nächste Wort übertragen werden.

Aus dem Impulsdiagramm (Bild 2) sind die analog bestimmten Merkmale, sowie die Steuerungssignale für die Datenübertragung ersichtlich.

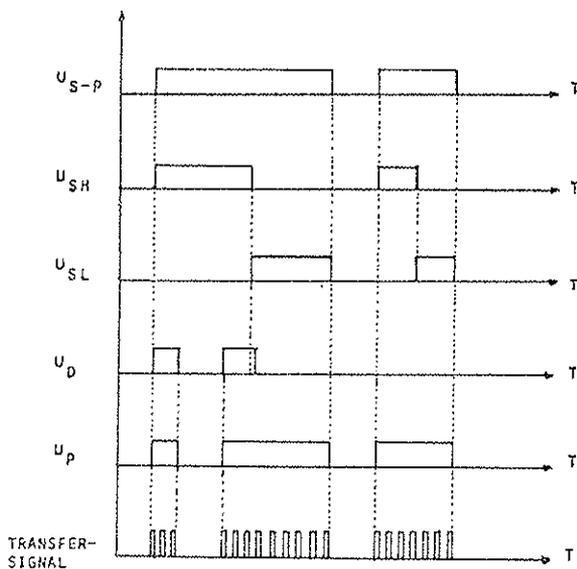


Bild 2
Impulsdiagramm

Berechnung der Prädiktorkoeffizienten

Das Verfahren der linearen Prädiktion wurde bisher mit Erfolg bei Sprachcodierung verwendet [2]. Der extrahierte Parameter wird bei Codierung mit möglichst geringer Bandbreite übertragen und daraus ein möglichst verständliches Sprachsignal generiert. Diese Parameter sollen nun für die Spracherkennung angewendet werden. Bei der Spracherkennung besteht die Forderung, daß sich die extrahierten Merkmale zur Trennung verschiedener Sprachelemente durch möglichst einfache lineare Klassifizierer eignen sollen. Es gibt nur wenige Arbeiten, in denen Prädiktorkoeffizienten für Spracherkennung angewendet werden.

Das Prinzip der linearen Prädiktion soll folgendermaßen formuliert werden:

Ein zukünftiger Wert $x(n)$ soll aus den vergangenen N Werten $x(n-1)$, $x(n-2)$, ..., $x(n-N)$ durch eine lineare Schätzung

$$\hat{X}(n) = \sum_{k=1}^N a_k x(n-k) \quad (1)$$

vorhergesagt werden. Das Ziel ist, für eine Folge von N Abtastwerten den Koeffizientenvektor $\{a_k\}$ so zu bestimmen, daß die Leistung des Differenzsignals für diese N Abtastwerte zum Minimum wird.

$$e_n^2 = [X(n) - \hat{X}(n)]^2 = \frac{1}{N} \sum_{i=1}^N (x_i - a^T \underline{x}_i)^2 = \text{Min} \quad (2)$$

Für jedes Intervall i mit N Abtastwerten wird die Optimierung durchgeführt und damit ein neuer Koeffizientenvektor a_k bestimmt.

Aus dieser Forderung ergibt sich das Gleichungssystem

$$\begin{pmatrix} r_0 & r_1 & \dots & r_{k-1} \\ r_1 & r_0 & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ r_{k-1} & r_0 & \dots & r_0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_k \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ \cdot \\ r_1 \\ r_k \end{pmatrix} \quad (3)$$

(r_1 ... normierte Autokorrelationskoeffizienten)

aus dem die optimalen Prädiktorkoeffizienten a_k gewonnen werden. Die Auflösung des linearen Gleichungssystems (3) wird unter Ausnutzung der speziellen Eigenschaften der Töplitzstruktur der Autokorrelationsmatrix mit einem rekursiven Verfahren durchgeführt. Es gibt viele Verfahren, die das Gleichungssystem (3) direkt und iterativ lösen können. Mit dem rekursiven Verfahren aber kann man eine bedeutende Ersparnis an Rechenoperationen gegenüber anderen Standardverfahren der Matrix erzielen.

Man erhält drei einfach aufgebaute Rekursionsgleichungen zur Berechnung der Prädiktorkoeffizienten [3]:

$$a_i^{(1)} = -[r_i + \sum_{j=1}^{i-1} a_j^{(1-1)} r_{i-j}] e_{i-1}^2 \quad (4)$$

$$a_j^{(1)} = a_j^{(1-1)} + a_1^{(1)} a_{i-j}^{(1-1)}, \quad 1 \leq j \leq i-1$$

$$e_i^2 = (1 - a_1^{(1)})^2 e_{i-1}^2, \quad 1 \leq i \leq k$$

Startwerte für die Rekursion sind $e_0^2 = 1$,
 $a^{(0)} = 0$.

Im Verlauf der Rechnung erhält man einen immer länger werdenden Prädiktor, wobei jeder erzeugte Prädiktor $a^{(i)}$, $i=1(1)k$ gültig ist.

Eine i -te Zwischenlösung soll dann als "sinnvoll" bezeichnet werden, wenn die auftretende Differenzsignalleistung e_i^2 kleiner (oder höchstens gleich) der im vorherigen Rechenschritt bestimmten Differenzsignalleistung e_{i-1}^2 ist. Ist das nicht der Fall oder tritt ein negativer Wert auf, wird der Algorithmus abgebrochen. Die Folge $e_0^2, e_1^2, \dots, e_i^2$ muß halbmonoton fallend sein. Sie gibt an,

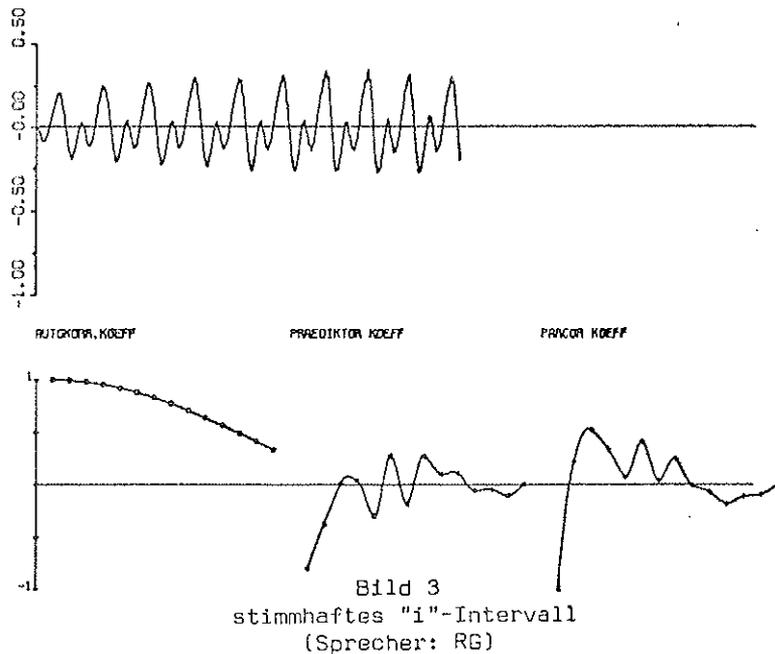
welcher Gewinn bei Verlängerung des Prädiktors um ein Element erreicht wird.

Zur Berechnung der Prädiktorkoeffizienten bzw. Parcorkoeffizienten aus vorgegebenen Autokorrelationskoeffizienten nach den Rekursionsgleichungen (4) wurde ein FORTRAN-Unterprogramm PRED erstellt, das nun in der Programm-bibliothek der Hybridrechenanlage zur Verfügung steht.

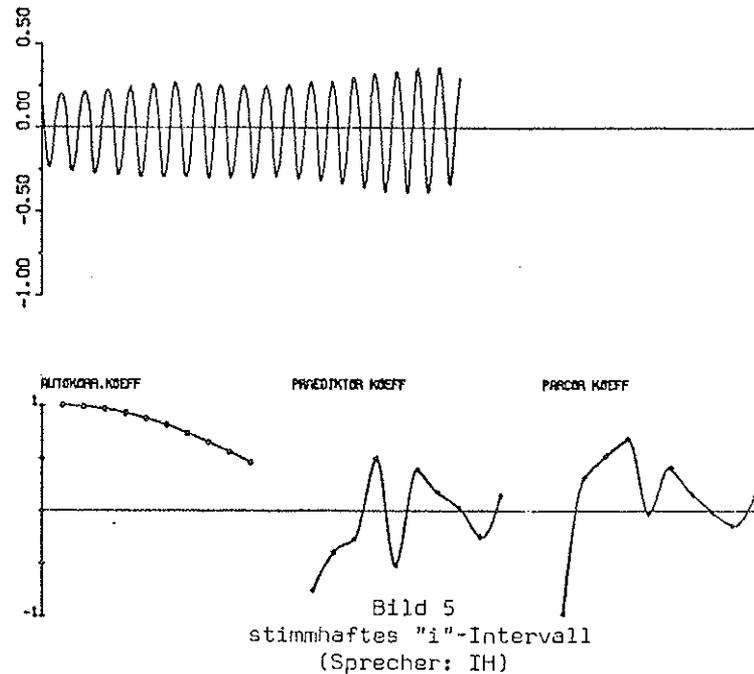
In Tabelle 1 ist die Folge e_i^2 für stimmhafte (z.B. "i") und stimmlose Intervalle (z.B. "s") angeführt.

Die Bilder 3 und 4 zeigen stimmhafte "i" und stimmlose "s" - Intervalle für einen männlichen Sprecher, die berechneten Autokorrelationskoeffizienten r_k , die Prädiktorkoeffizienten a_k und die Parcorkoeffizienten p_k für diese Intervalle. Die Bilder 5 und 6 stellen dasselbe für einen weiblichen Sprecher dar.

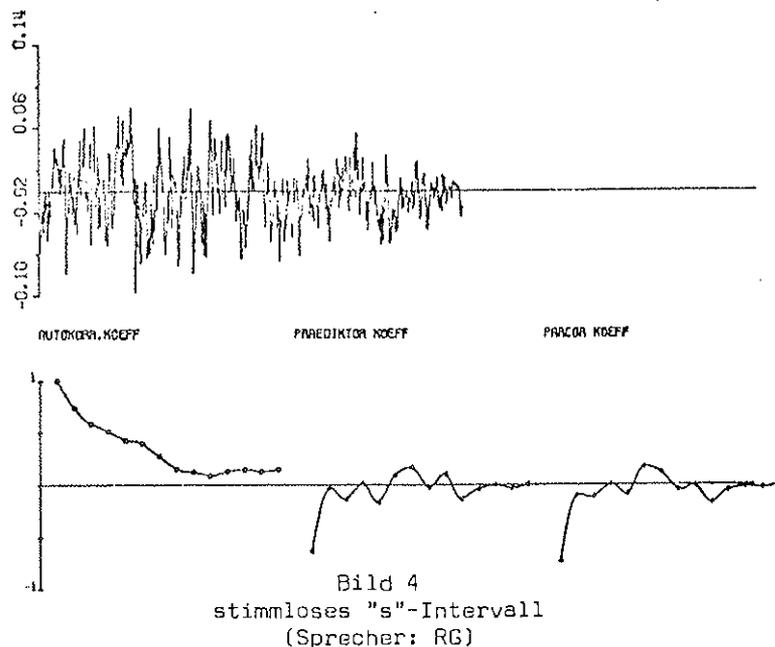
SPRACHSIGNAL N = 1000 NP = 14 INTERVALL 2 FILE M01007



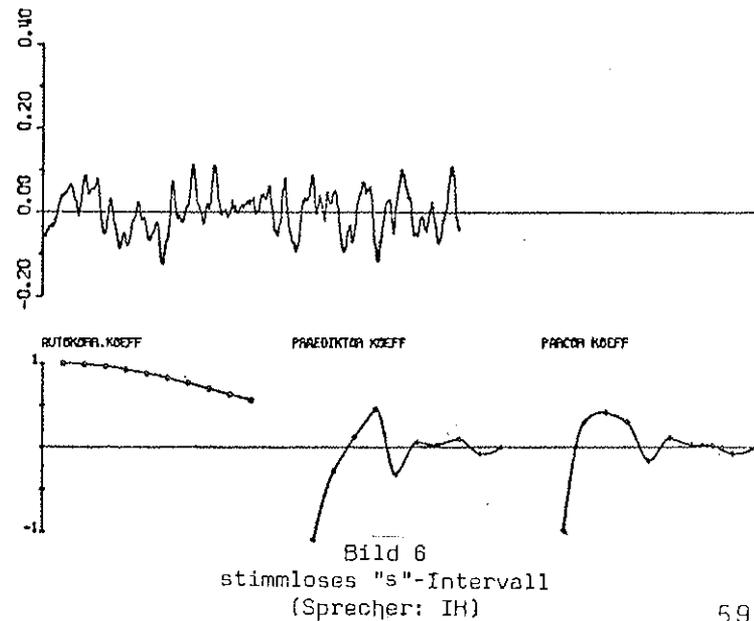
ANALOGSIGNAL N = 1000 NP = 10 INTERVALL 2 FILE M01017



SPRACHSIGNAL N = 1000 NP = 14 INTERVALL 7 FILE M01007



ANALOGSIGNAL N = 1000 NP = 10 INTERVALL 8 FILE M01017



INTERFACE November 1977