
Interface

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

Nummer 13
April 1979



Eigenentwicklung: Control Device

INHALTSVERZEICHNIS

	Seite
AutoPATCH-Erweiterung an der Hybridrechenanlage	3
Systemkonfiguration Hybridrechenanlage EAI PACER 600	6
Aktuelle Mitteilungen	8
Kurse	10
Control Device	11
Datenübertragung vom Hybridrechner in den Hörsaal X	14
Datenübertragung von CDC CYBER 74 auf EAI PACER 600 über Magnetband	15
Dayfile-Accounting in JCS/TS 7	17
Die Verwendung von digital setzbaren Funktionsgebern	22
Gasfluß durch ein langes Rohr	31
Experimentelle Überprüfung eines Parameterschätzverfahrens für zeitvariante Systeme	37
Klassifikation von akustischen Signalen durch Phasenbildverfahren	40
Analoge Simulation von Stoffwechselabläufen anhand von Stoffwechselfluxgrößen	43
Squash - Einmanntennis am Analogrechner	47
Schmiedeeisenkunst mit dem Hybridrechner	49

Redaktion: Irmgard Husinsky

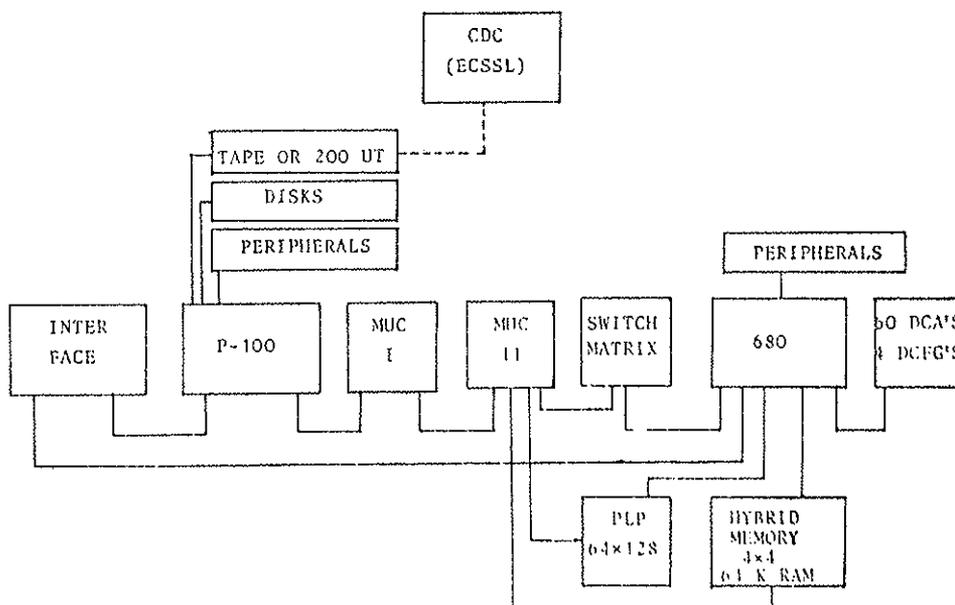
Eigentümer, Herausgeber, Verleger: EDV-Zentrum der Technischen Universität Wien, Abteilung Hybridrechenanlage, Vervielfältigung: Österreichische Hochschülerschaft Technik, für den Inhalt verantwortlich: Dipl. Ing. W. Kleinert, alle: Gußhausstraße 27-29, 1040 Wien
Tel: (0222) 65 37 85/803, Telex: 136875 rzthw a

AUTOPATCH-ERWEITERUNG AN DER HYBRIDRECHENANLAGE

Wie in INTERFACE Nr. 12 berichtet wurde, konnte nach langen Vorbereitungen Anfang Oktober 1978 vom Bundesministerium für Wissenschaft und Forschung die Genehmigung zur Vertragsunterzeichnung über eine AutoPATCH-Erweiterung der Hybridrechenanlage erteilt werden. Am 8. November 1978 wurde der Vertrag unterzeichnet.

Unser wichtigstes Ziel beim Ausbau der Hybridrechenanlage ist es, durch die Anschaffung einer analogen Schaltmatrix und eines hybriden Compilers die Voraussetzungen für einen hybriden Time-sharingbetrieb zu schaffen, der den Benutzern den Zugriff zum Hybridrechner auf der Ebene einer komfortablen, gleichungsorientierten Software ermöglicht.

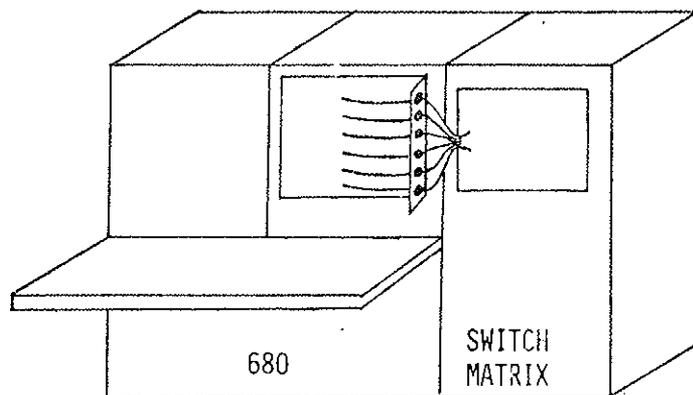
EAI wird an Hardware eine dreistufige analoge Schaltmatrix samt den dazugehörigen Controllern und der Testsoftware liefern und installieren. Ferner wird EAI den hybriden Compiler ECSSL II liefern, der - als EAI Standardprodukt - auf der CYBER laufen soll.



TU - AUTOPATCH HARDWARE

Abbildung 1

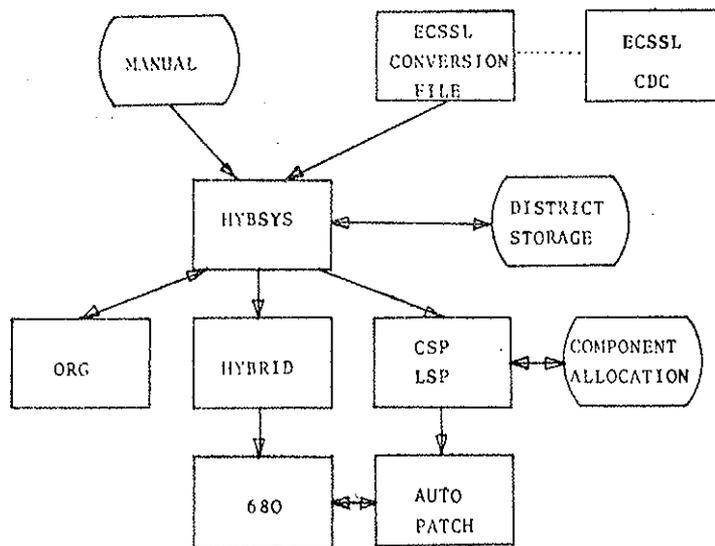
Abbildung 1 zeigt die geplante Hardwarekonfiguration. Für den Benutzer wird es erfreulich sein, daß der bisher nicht vorgesehene Ersatz der parallelen Logik durch einen digital programmierbaren PLP (Parallel Logic Processor) nun doch durch eine Eigenentwicklung der TU realisiert werden wird (worüber wir in einem der nächsten INTERFACE genau berichten werden). Ebenfalls neu ist das zusammen mit dem Institut für elektrische Regelungs-, Steuerungs- und Antriebstechnik geplante HYBRID Memory, bei dem es sich um ein vom PACER kontrolliertes Mikroprozessorsystem zur Verwaltung von in Tabellenform gespeicherten "hybriden Variablen" und zur Erzeugung nichtlinearer Funktionen von bis zu vier Veränderlichen handelt. Darüber hinaus hat sich die Firma EAI bereit erklärt, zu einem äußerst geringen Preis den Analogrechner durch Aufstockung und Umrüstung mit nun insgesamt 60 4-Quadranten-DCA's auszustatten.



680 AUTOPATCH CONSOLE

Abbildung 2

Abbildung 2 zeigt die Art und Weise, in der die Verbindung zwischen der Schaltmatrix (bzw. PLP) und dem Analogrechner geplant ist. Ein spezielles AutoPATCH-Panel wird über seitlich angebrachte Stecker mit der Matrix verbunden. Das ermöglicht optimal kurze Kabelverbindungen und läßt auch weiterhin den bisher gewohnten Betrieb als Alternative zum AutoPATCH zu.



TU - AUTOPATCH SOFTWARE

Abbildung 3

Das Softwarekonzept des TU AutoPATCH-Systems basiert neben dem ECSSL-Compiler auf dem hybriden Sprachprozessor HYBSYS, einer Eigenentwicklung der Hybridrechenanlage, die seit einiger Zeit bereits erfolgreich in Verwendung steht (siehe INTERFACE 11). HYBSYS wird durch einige neue Moduln erweitert werden, die das Komponentenverbindungsproblem lösen sowie das Laden und Setzen der Schaltmatrix und des PLP ermöglichen. Dem Benutzer werden analoge AutoPATCH "Macros" (Zusammenfassung mehrerer Verstärker, Potentiometer und Logikelemente zu einem neuen Rechenelement) zur Verfügung stehen. Die Zuordnung der Gleichungsstruktur zu den Macros wird in einem HYBSYS-Modul erfolgen. Die Verbindung zu ECSSL soll durch einen eigens (im Rahmen einer Diplomarbeit) zu schreibenden Interpreter erfolgen, der den ECSSL Standard Output (in HYTRAN Source) in eine HYBSYS-kompatible Form übersetzt.

Mit der Installation des ECSSL-Compilers und der AutoPATCH Hardware wird noch 1979 gerechnet.

At Nov. 8, 1978 the contract concerning the AutoPATCH expansion of the EAI PACER 600 installation at TU Vienna was signed. This article gives a short review of the TU AutoPATCH hardware and software projects. EAI will deliver the switch matrix and ECSSL II hybrid compiler. There are projects running at TU Vienna to build up a parallel logic processor and a so-called "HYBRID Memory" which is a microprocessor implementation of hybrid variables and nonlinear function generation. The AutoPATCH software concept is based on the hybrid processor HYBSYS, a powerful simulation tool developed at TU Vienna.

W. Kleinert

SYSTEMKONFIGURATION HYBRIDRECHENANLAGE EAI PACER 600

STAND VOM 79/03/01

EAI PACER 100 Digitalrechner

64 k Worte à 16 Bit
1 µsec Zykluszeit
64 Interruptebenen
52-Bit Floating Point Processor
Multiprogramming
Intervallzeitgeber
1 Standard Input/Output Kanal
2 direkte Speicherkanäle (max.
1 MHz Transferrate)

EAI 680 Analogrechner

Parallele Logik
30 Integrierer
204 Verstärker
24 Multiplizierer
EXP-, LOG-, SIN-, COS-Funktionsgeber
Rauschgenerator
80 Servopotentiometer

EAI 693 Interface

24 Analog-Digital-Wandler
16 Digital-Analog-Wandler
4 Digital-Analog-Multiplizierer
40 Digitale Potentiometer
4 Digitale Funktionsgeneratoren
8 Interruptebenen
8 Senselines
16 Controllines
1 Digitaler ABCD-Timer

Analoge Peripherie

4-Strahl-Oszillograph
6-Kanal-Recorder
Revox-A77 Tonbandgerät
AKG K 158 Kopfhörer-
Mikrophon-Kombination
Steuerknüppel

Massenspeicher

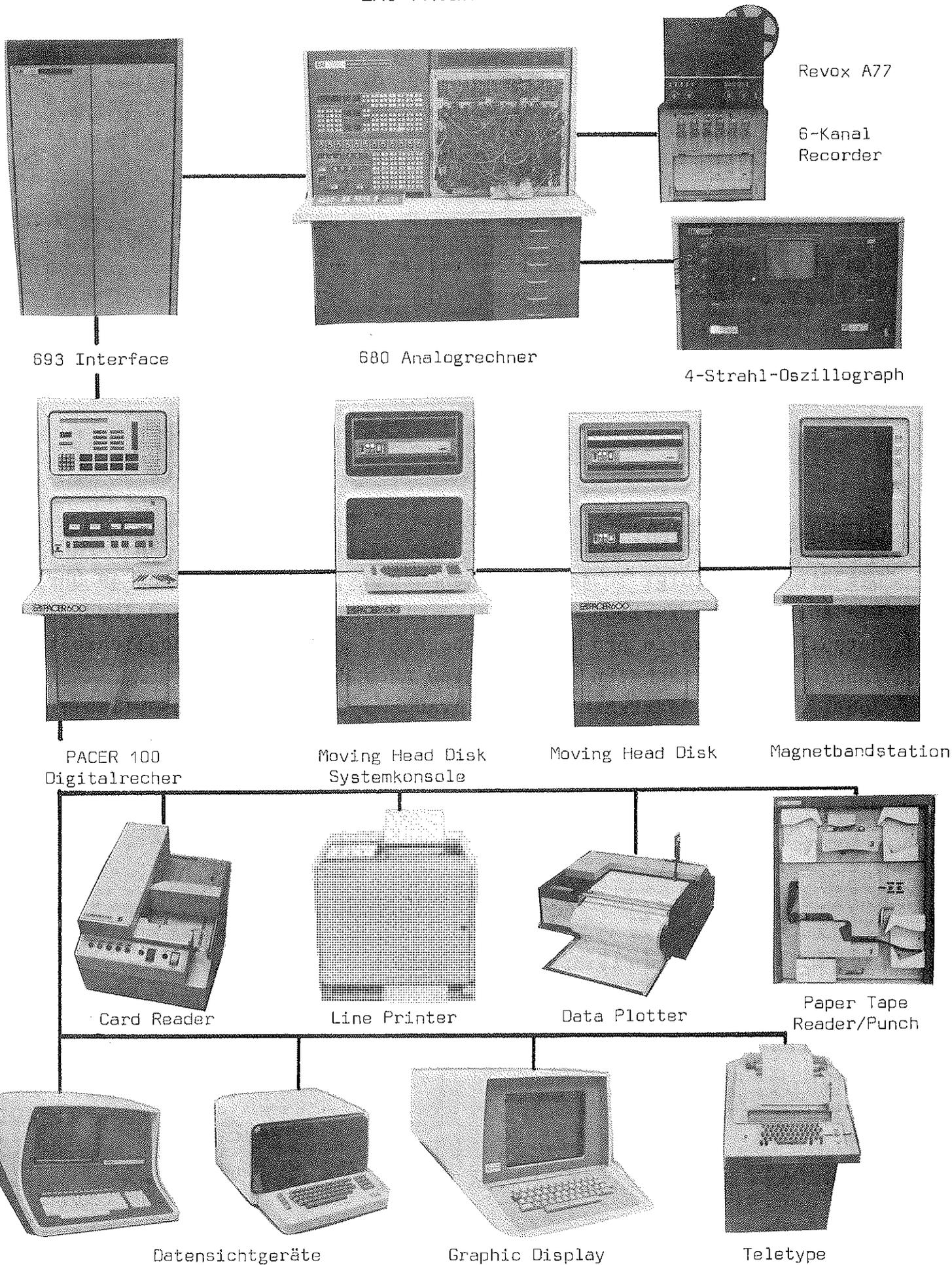
Magnetbandstation 1724:
9 Spuren, 800 BPI, 45 IPS
Pertec Disks 1272:
3 Wechselplatteneinheiten à 2.2 mill Worte,
Transferrate 74 k Worte/sec

Digitale Peripherie

Card Reader EAI 300:
300 Karten/min
Line Printer EAI 600:
80 Spalten, 350-1200 Zeilen/min
Data Plotter EAI 145:
Inkrementalplotter, 0.005 Inch/Step, 450 Steps/sec,
programmierbarer Wiederholungsfaktor
2 Datensichtgeräte EAI C101:
64 Zeichen, 21 Zeilen à 80 Zeichen, 4800 baud asynchron,
Datensichtgerät ADDS Regent 100:
mikroprozessorgesteuert, 128 Zeichen, 24 Zeilen à 80 Zeichen,
graphische Zeichen, 4800 baud asynchron
Graphic Display Tektronix 4006-1:
Alpha Mode: 64 Zeichen, 35 Zeilen à 74 Zeichen,
Graphic Mode: 1024x780 Punkte, 4800 baud asynchron
Teletype 35KSR
High-Speed-Paper-Tape-Reader/Punch: 5,7 und 8 Kanäle,
Reader: 300 Frames/sec
Punch: 120 Frames/sec
Control Device:
9x16 Dioden Anzeigen, 9 Digital-Anzeigen,
8 Control Switches, 9 Selektoren, Keyboard
Communication Control Modul:
zur synchronen Datenübertragung für 200 UT Emulation
Panel Device:
9 Indikatoren, 8 Panel Switches
Switch Panel:
zum Schalten verschiedener Rechnerverbindungen
Z80 Interface:
zur Datenfernübertragung für ein Z80 Mikroprozessorsystem

SYSTEMKONFIGURATION HYBRIDRECHENANLAGE

EAI PACER 600



aktuelle mitteilungen

OFFNUNGSZEITEN:

Die Hybridrechenanlage ist von
Montag bis Freitag
von
8 Uhr bis 20 Uhr
geöffnet.

CLOSED-SHOP-BETRIEB:

Die Eingabe erfolgt zu jeder vollen Stunde (letzte Eingabe: 19 Uhr).
Die Ausgabe von Kurzjobs (maximal 5 min CPU-Zeit, standard Input/
Output = 51 k Worte pro Ein/Ausgabegerät) erfolgt nach Möglichkeit
innerhalb der nächsten zwei Stunden nach der Eingabe.
Jobs, die mehr Betriebsmittel verbrauchen (TIME- oder IO-Option),
insbesondere große Plotterjobs, werden nach Möglichkeit im Laufe
von 24 Stunden gerechnet, die Ausgabe erfolgt in Abhängigkeit von
der Plotterauslastung.

PERSONELLES:

Frau Eva Quantschnigg ist seit Anfang 1979 anstelle von Frau Monika
Sagerschnigg als Programmierassistentin an der Hybridrechenanlage
tätig.

NEU: EIGENENTWICKLUNG CONTROL DEVICE

Die Control Device ist eine Anzeigetafel, die zur Steuerung und Kontrolle des Rechenbetriebes verwendet werden kann (siehe auch Seite 11).

NEU: 4.DIGITAL SETZBARE FUNKTIONSGEBER

Seit Mitte Februar 1979 ist der Analogrechner mit zwei weiteren digital setzbaren Funktionsgebern (DCFG) bestückt (DCFG Ø - 3). Siehe dazu auch Seite 22.

HYBSYS-DOKUMENTATION:

An einer ausführlichen Beschreibung des an der Hybridrechenanlage entwickelten hybriden Sprachprozessors HYBSYS wird gerade gearbeitet. Das Manual wird nach Ostern erhältlich sein.

INTERFACE-INHALTSÜBERSICHT:

Es wurde ein thematisch geordnetes Verzeichnis aller bisher erschie-
nener INTERFACE-Artikel erstellt. Das Verzeichnis kann bei der Re-
daktion angefordert werden.

NEUE DRUCKSORTEN:

Seit Anfang des Jahres ist eine neue Beschreibung der graphischen Grundsoftware erhältlich, die eine Überarbeitung des bisherigen Data Plotter Manuals darstellt und die Programme des Files GRAFIC beschreibt.

kurse

Folgende Kurse werden im Sommersemester 1979 an der Hybridrechenanlage abgehalten werden:

RH9 EINFÜHRUNG IN DIE HYBRIDE PROGRAMMIERUNG

Dieser Kurs wird nach Bedarf für wissenschaftliche Benutzer, die keinerlei Vorkenntnisse auf dem Gebiet der hybriden Programmierung besitzen und an einer möglichst raschen Problemlösung interessiert sind, abgehalten.

Dauer: 2 Tage

Vortragender: D. Solar

Nähere Auskünfte bei Hr. D. Solar (1040 Wien, Gußhausstraße 27-29, 4. Stock, Zimmer 1401, Tel. 65-37-85/747 Dw).

RH11 ASSEMBLER-PROGRAMMIERUNG FÜR FORTGESCHRITTENE MIT ÜBUNGEN

Dauer: 5 Tage

Termin: 1979/05/07 bis 1979/05/11, 9-12 Uhr

Vortragender: Dipl. Ing. A. Blauensteiner

RH14 BEDIENUNG DES PROGRAMMSYSTEMS HYBSYS ZUR DIGITALEN UNTERSTÜTZUNG DES ANALOGRECHNERS 2.TEIL MIT ÜBUNGEN

Dauer: 2 Tage

Termin: 1979/04/24 bis 1979/04/25

9-12 Uhr und 13-17 Uhr

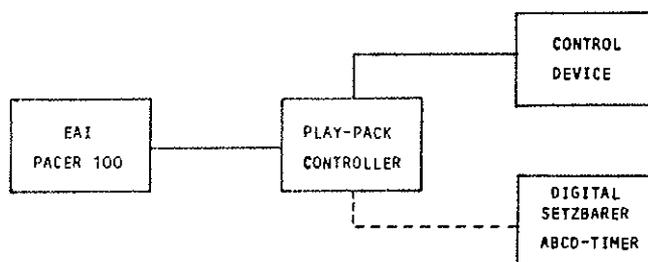
Vortragende: Dipl.Ing. W. Kleinert und D. Solar

Kursort: 1040 Wien, Gußhausstraße 27-29, 4.Stock, Zimmer 1401

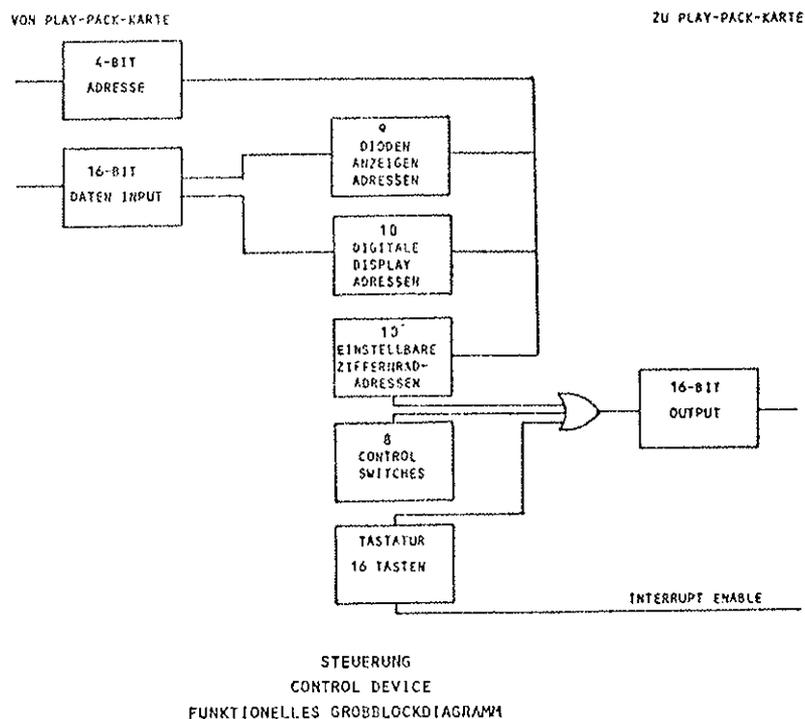
Nähere Auskünfte und Anmeldung zu den Kursen telefonisch oder persönlich bei Hr. M. Schandl (1040 Wien, Gußhausstraße 27-29, 4. Stock, Zimmer 1404/05, Tel. 65-37-85/803 Dw).

CONTROL DEVICE

An der Hybridrechenanlage wurde Anfang Februar ein neues Gerät, die Control Device, installiert. Sie stellt im wesentlichen eine Anzeigetafel dar, die die Steuerung und Kontrolle des in den letzten Monaten immer stärker werdenden Rechenbetriebes vereinfachen soll. Mit der Planung und dem Design wurde vor etwa einem Jahr begonnen. Die Steuerung der Control Device selbst, die Montage, sowie die Anordnung der Platinen wurden von Herrn V. Franz durchgeführt. Design, Layout, sowie Anschluß an den Digitalrechner EAI PACER 100 durch den Bau des entsprechenden Controllers wurden von der Hybridrechenanlage erstellt. Dabei wurde eine Play-Pack-Karte verwendet, auf der nun zwei Funktionen parallel bedient werden (siehe auch INTERFACE Nr. 12, "Arbeitsweise und Verwendung der Rep-op-Timer an der Hybridrechenanlage").

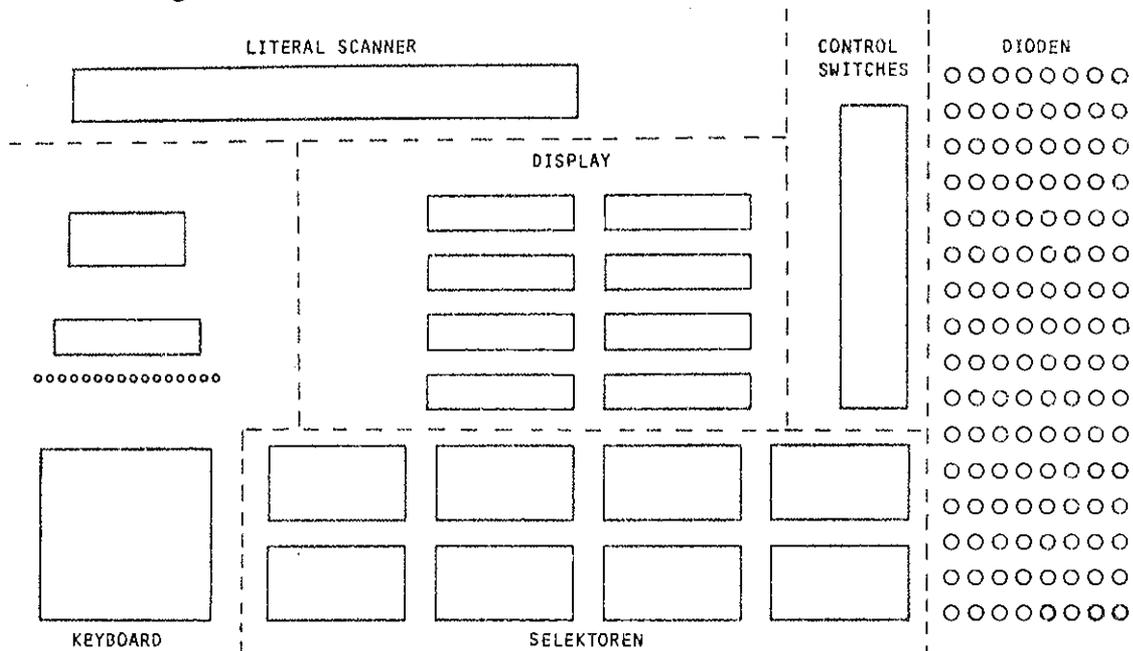


Die Unterscheidung der beiden Devices auf diesem Controller erfolgt logisch durch entsprechend codierte Befehle.

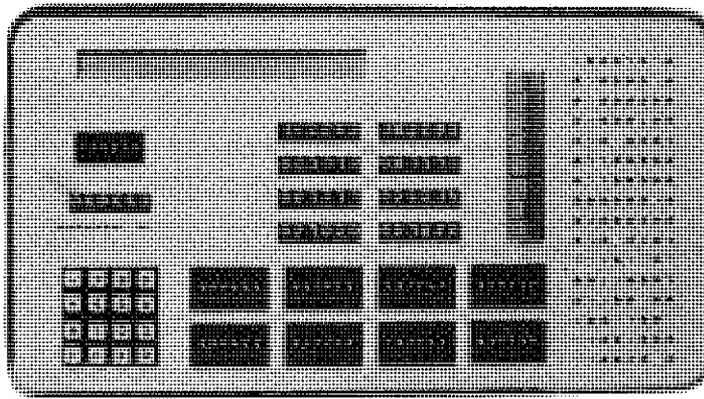


Das Anzeigefeld umfaßt folgende Elemente:

- 9 mal 16 in Gruppen steuerbare Dioden-Anzeigen.
Diese können für die Anzeige von Indikatoren, Weichen, logischen Variablen sowie für die Anzeige gewisser Bitmuster verwendet werden.
- 9 Digital-Anzeigen. Hier können 8 mal oktale und einmal hexadezimale Werte ausgegeben werden.
Diese Anzeigen eignen sich zur Darstellung von Adressen, wichtigen Systemparametern, Daten oder Iterationswerten in Programmen.
- Mit 8 Control Switches können Pausen überwunden bzw. Programmverzweigungen angesteuert werden.
- 9 Selektoren, oktal bzw. einmal dezimal einstellbare Ziffernräder, ermöglichen die Eingabe von Adressen, Parametern bzw. Schleifengrenzen, Zeitschranken, Gerätenummern, etc.
- Schließlich können über ein Keyboard mit 16 Tasten Werte eingegeben werden. Diese werden über Interrupts verarbeitet und sind dem Benutzer im allgemeinen nicht zugänglich. Sie stehen vor allem dem Systemprogrammierer für On-line-Debug-Aufgaben zur Verfügung.
- Geplant ist die Aktivierung eines Literal Scanners, der Textausgabe als Laufschrift ermöglicht.



CONTROL DEVICE - GRUPPIERUNG DER ELEMENTE



CONTROL DEVICE - FRONTANSICHT

Die Control Device wurde auf dem freien Feld an der PACER-Tür, direkt über der PACER-Konsole, montiert.

Alle Elemente sind direkt programmgesteuert ansprechbar. Duale bzw. gekoppelte Funktionen werden also über Software bewerkstelligt.

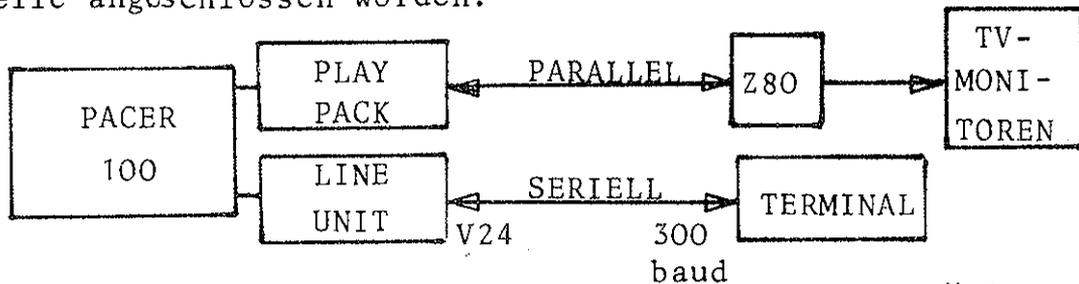
Derzeit gibt es zwei Einsatzarten der Control Device: Einmal kann sich der Systemoperator durch geeignete Befehle Zutritt zu allen Adressen und deren Inhalt verschaffen, wobei er diese variabel an den Selektoren ansteuert. Weiters hat er die Möglichkeit, über das Keyboard Tischrechnerfunktionen in Oktal- oder Dezimalarithmetik durchzuführen.

Andererseits kann der Benutzer über leicht aufrufbare FORTRAN-Unterprogramme alle Elemente bis auf das Keyboard für seine Zwecke ansprechen. So werden mit CALL DISPLAY die Digitalanzeigen angesprochen, mit CALL SELECT die Selektoren und mit CALL SWITCH die Control Switches abgefragt und mit CALL SLIGHT bzw. CALL RLIGHT die Dioden-Anzeigen gesetzt bzw. zurückgesetzt. Um den Benützern beider Partitions die Benützung der Control Device zu ermöglichen, spricht jede der beiden, intern exakt unterschieden, nur die Hälfte der Elemente an. Programmkurzbeschreibungen sind an der Hybridrechenanlage erhältlich.

A. Blauensteiner

DATENÜBERTRAGUNG VOM HYBRIDRECHNER IN DEN HÖRSAAL X

Anfang März wurde eine bidirektionale parallele asynchrone Datenverbindung von der Hybridrechenanlage in den Hörsaal X fertiggestellt. Gleichzeitig ist ein Terminal im Hörsaal über eine V24-Seriell-Schnittstelle angeschlossen worden.



DIGITALRECHNER CONTROLLER

HÖRSAAL X

Dieses Projekt ist von der Hybridrechenanlage in Zusammenarbeit mit dem Institut für elektrische Regelungs-, Steuerungs- und Antriebstechnik (Vorstand Prof. Weinmann) und Diplomanden realisiert worden, wobei von den Mitarbeitern der Hybridrechenanlage die gesamte Software für die Steuerung der Datenleitung sowie deren Einsatz im Rahmen der CalComp-standardisierten Plotter Software für die TV-Monitore entwickelt wurde, sowie der Einsatz des Terminals als Ein/Ausgabegerät für Programme, die an der Hybridrechenanlage laufen, ermöglicht wurde.

Die Software ist nun vollkommen kompatibel mit den bisherigen graphischen Geräten (Tektronix, Data Plotter) an der Hybridrechenanlage, so daß diese ohne Aufwand für Test- bzw. Dokumentationszwecke eingesetzt werden können.

Ebenso ist das Terminal im Rahmen des Betriebssystems voll kompatibel mit allen anderen I/O-Geräten und von der gesamten Systemsoftware ansprechbar.

Über die Entwicklung des Z80-Mikroprozessorsystems zur graphischen und alphanumerischen Ausgabe auf TV-Geräten, den Aufbau der Datenleitungen sowie deren Gesamteinsatz während Vorlesungen wird in einer der nächsten Nummern von INTERFACE berichtet werden.

A. Blauensteiner

DATENÜBERTRAGUNG VON CDC CYBER 74 AUF EAI PACER 600 ÜBER MAGNETBAND

Die seit 1975 bestehende Möglichkeit der Datenübertragung von der Digitalrechenanlage CDC CYBER 74 auf die Hybridrechenanlage EAI PACER 600 mit Hilfe von binären Lochkarten (Subroutinen "HYPNCH" und "HYREAD") hat sich für kleinere Datenmengen bei zahlreichen Benutzern bewährt. Für größere Datenmengen hat sich diese Methode aber als nicht zufriedenstellend erwiesen. Aus diesem Grund wurde schon vor langer Zeit die Möglichkeit der Datenübertragung über ein Magnetband projektiert. Dabei sollten neben REAL- auch INTEGER- und Character-Daten übertragen werden können; außerdem sollte die Übertragung in beiden Richtungen (also auch von der Hybridrechenanlage zur Digitalrechenanlage) möglich sein. In Gemeinschaft mit der Abteilung Digitalrechenanlage (Herr H. Krautschneider) wurden jetzt die Routinen für die Datenübertragung von der CYBER 74 auf den PACER 600 fertiggestellt.

Daten, die zur Hybridrechenanlage übertragen werden sollen, müssen dazu zuerst auf der CYBER 74 auf einen permanenten File geschrieben werden. Zum Beschreiben dieses Files stehen fünf FORTRAN-aufrufbare Routinen zur Verfügung:

Zu Beginn eines Datensatzes muß mit

```
CALL HYOPEN ( NUNIT, FNAME, JOBNR )
```

ein Leader-Record geschrieben werden, wobei NUNIT die logische Einheitennummer des Files, FNAME einen maximal 8 Zeichen langen Namen für den Datensatz und JOBNR die Jobnummer des Benutzers auf der Hybridrechenanlage angeben. Die Angaben FNAME und JOBNR sind zur Identifizierung des Datensatzes beim Einlesen am PACER 600 erforderlich.

Danach können mit

```
CALL HYPUTI ( IWERT )
```

```
CALL HYPUTR ( WERT )
```

```
CALL HYPUTC ( NCHAR, ITEXT )
```

einzelne INTEGER-Zahlen (IWERT), REAL-Zahlen (WERT) oder Zeichenketten (ITEXT, bestehend aus NCHAR Zeichen) in beliebiger Reihenfolge ausgegeben werden.

Die Daten werden intern zu Records fester Wortlänge gepuffert, sodaß die noch im Puffer befindlichen Daten am Ende eines Datensatzes mit

```
CALL HYCLOS
```

auf den File ausgegeben werden müssen. Danach kann mit "HYOPEN" ein neuer Datensatz auf dem gleichen File eröffnet werden.

Nach dem Beschreiben mit einem oder mehreren Datensätzen muß der File katalogisiert werden. Durch Aufruf einer Prozedur (im selben oder einem weiteren Job) kann der permanente File mit allen Datensätzen dann auf ein Magnetband kopiert werden. Um

die Daten an der Hybridrechenanlage weiterverarbeiten zu können, muß sich der Benutzer das Magnetband an der Digitalrechenanlage ausfolgen lassen und es beim Operator der Hybridrechenanlage abgeben. Das Band muß mit einem Namen und der Jobnummer des Benutzers gekennzeichnet sein. Durch Aufruf der Routine NOTE oder mit der /NOTE-Option auf einer Steuerkarte kann es dann in einem Job angefordert werden.

Für das Einlesen der Daten auf der Hybridrechenanlage stehen vier FORTRAN-auf-rufbare Routinen zur Verfügung:

Am Beginn der Eingabe muß mit

```
CALL DIPOSI ( FNAME, JOBNR )
```

der einzulesende Datensatz auf dem Magnetband positioniert werden. Dabei muß die im Leader-Record stehende Jobnummer mit der angegebenen Jobnummer übereinstimmen.

Durch die Aufrufe

```
CALL DIGETI ( IWERT )
```

```
CALL DIGETR ( WERT )
```

```
CALL DIGETC ( NCHAR, ITEXT )
```

können dann Integerzahlen, Realzahlen und Zeichenketten in der gleichen Reihenfolge, in der sie an der Digitalrechenanlage auf das Band geschrieben wurden, eingelesen werden.

Kurzbeschreibungen für die angeführten Routinen sind in Ausarbeitung, Interessenten für die Datenübertragung über Magnetband mögen sich an Herrn Krautschneider (Abt. Digitalrechenanlage) oder Herrn Blöser (Abt. Hybridrechenanlage) wenden.

Der Datentransport in der umgekehrten Richtung vom PACER 600 zur CYBER 74 wird in gleicher Weise realisiert werden, die entsprechenden Routinen DIOPEN, ... , HYGETC sind jedoch noch in Arbeit.

F. Blöser

DAYFILE-ACCOUNTING IN JCS/TS 7

Das Betriebssystem JCS/TS 7 erstellt laufend ein Protokoll des Rechenbetriebs auf dem Dayfile. Diese Protokolle werden nun ausgewertet und geben detaillierten Aufschluß über den täglichen Rechenbetrieb an der Hybridrechenanlage.

WAS WIRD IM DAYFILE EINGETRAGEN ?

- Die Exekution jeder /-Steuerkarte (außer der /DELETE-Karte) bewirkt eine Eintragung im Dayfile, d.h. es wird der Start jedes Tasks (Source Processor, Assembler, FORTRAN-Compiler, Core Image Generator, HOI oder User Run) sowie der Beginn und das Ende eines Jobs vermerkt.
- Ferner werden sämtliche während des Rechenbetriebs auftretende System- und Programmereignisse eingetragen, die die Ausgabe einer Meldung bewirken:
 - Meldungen für den Operator (z.B. Device not ready, District full, etc.),
 - Meldungen aus Benutzerprogrammen (z.B. FORTRAN-Fehlermeldungen, Plotter-Zeichenfeldüberschreitungen, etc.),
 - Fehler, die zum Abbruch eines Benutzerprogramms führen (z.B. Sprung aus der Partition, Rechenzeitüberlauf, etc.).
- Außerdem können Eintragungen in den Dayfile vom Benutzer programmiert werden.

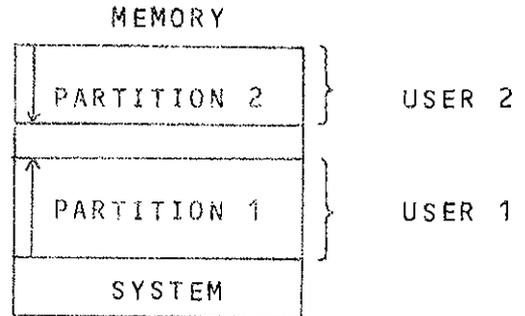
WIE SIEHT EINE DAYFILE-EINTRAGUNG AUS ?

Jede Eintragung enthält den genauen Zeitpunkt der Eintragung, die Jobnummer desjenigen, der die Eintragung verursacht hat, den Level, wo sie aufgetreten ist, einen Indikator und eine Nummer, die auf einen Meldungstext verweist, der auf einem weiteren Systemfile gespeichert ist. Diese kurzen englischen Texte geben die Art des Ereignisses an. Die Eintragungen werden in möglichst gepackter Form abgespeichert (vier 16-Bit-Worte pro Eintragung).

JOB-PROTOKOLL

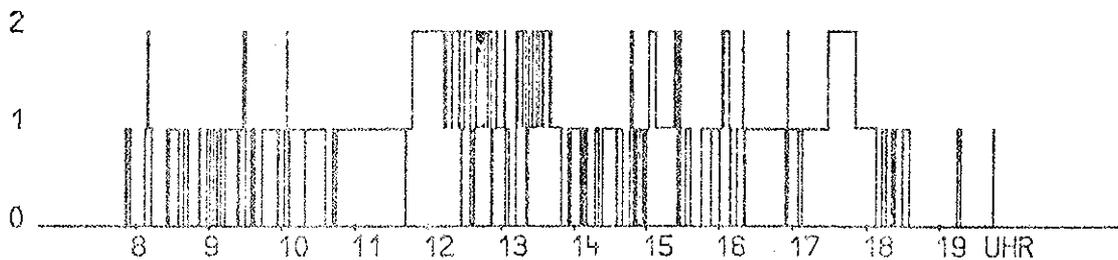
Das Betriebssystem JCS/TS 7 bedient im Multiprogramming vier Level parallel: zwei Benützer und zwei Systemaktivitäten. Die Systemaktivitäten (Supervisor und System Manager) werden nicht im Dayfile vermerkt, nur im Falle einer Fehlermeldung erfolgt eine Eintragung. Die Eintragungen für die beiden Benützerlevel (Partition 1 und 2) ermöglichen nun eine genaue Auswertung der Belegung der Level und Partitions.

Partition 1 und Partition 2 sind die Kernspeicherzonen, in die die Benützerprogramme geladen werden, und zwar wird zuerst Partition 1 belegt, und wenn Partition 1 besetzt ist, wird Partition 2 belegt.



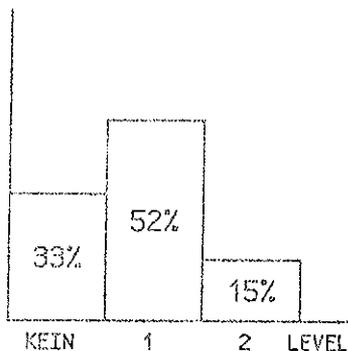
Aus den Dayfile-Eintragungen läßt sich nun, wie das folgende Beispiel zeigt, pro Tag ein Diagramm erstellen, das, über der Einschaltzeit des Rechners aufgetragen, angibt, ob kein Level, ein Level oder zwei Level belegt sind, unabhängig von der Partition.

ANZAHL DER BELEGTEN LEVEL UEBER DER ZEIT

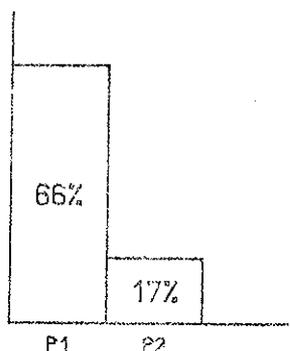


Ausgedrückt in Prozent der Einschaltzeit sind nun, für dasselbe Beispiel wie oben, die Levelbelegung und die Partitionbelegung dargestellt.

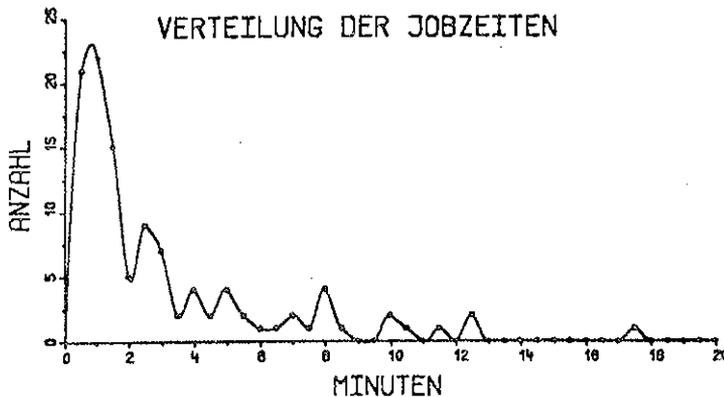
LEVELBELEGUNG
IN % DER ZEIT



PARTITIONBELEGUNG
IN % DER ZEIT



Im Unterschied zum Job Accounting, das auf einem anderen Systemfile, dem Jobfile, durchgeführt wird und wo für jeden Benutzer pro Task die reine CPU-Zeit und die Input/Output-Operationen gezählt werden, geben die Eintragungen im Dayfile ein Protokoll des Rechenbetriebs in Echtzeit wieder. Die so ermittelten Jobzeiten (Echtzeit) lassen sich für einen Tag, in Intervallen von halben Minuten zusammengefaßt, in folgendem Diagramm darstellen:



AUSWERTUNG DER FEHLERMELDUNGEN

Pro Tag werden die auftretenden Fehler- und Systemmeldungen nach ihrer Häufigkeit geordnet. Man kann feststellen, auf welchem Level sie aufgetreten sind (Level 0 - System Manager, Level 1 - User 1, Level 2 - User 2, Level 3 - Supervisor).

HAUFIGKEITEN DER MELDUNGEN

NR	LEV0	LEV1	LEV2	LEV3	TOT	%	
311	0	80	38	0	118	18,50	BATCH JOB STARTS EXECUTION
312	0	78	39	0	117	18,34	BATCH JOB FINISHED
320	0	66	33	0	99	15,52	USER PROGRAM RUN
321	0	0	0	67	67	10,50	DAYFILE END MARK
313	0	37	29	0	66	10,34	FORTRAN COMPILATION
315	0	31	27	0	58	9,09	PROGRAM BUILDING
264	0	16	0	0	16	2,51	REAL TO SCALED FRACT. OVERFLOW, SET TO MAX
023	0	5	7	0	12	1,88	ATTEMPT TO LOAD NOT EXISTANT CI-FILE
027	0	9	0	0	9	1,41	ACCESS TO MEMORY OUT OF PARTITION
100	0	0	1	0	7	1,10	TASK TERMINATED
060	0	3	4	0	7	1,10	ERROR IN OBJECT

⋮

Ferner kann man einen Überblick bekommen, wieviele und welche Jobnummern an diesem Tag gerechnet haben, wieviele Jobs, wieviele Tasks und wieviele Programmabbrüche sie hatten.

MELDUNGEN NACH JOBNUMMERN

JOBNUMMER	JOBS	TASKS	EXITS
903	6	6	1
370	25	49	7
353	4	12	1
19	1	1	0
160	20	26	15
906	5	5	0
12	9	23	4
133	10	21	1
149	9	27	7
302	4	4	0
340	9	28	11
376	5	15	0
85	1	3	0
14	3	0	0

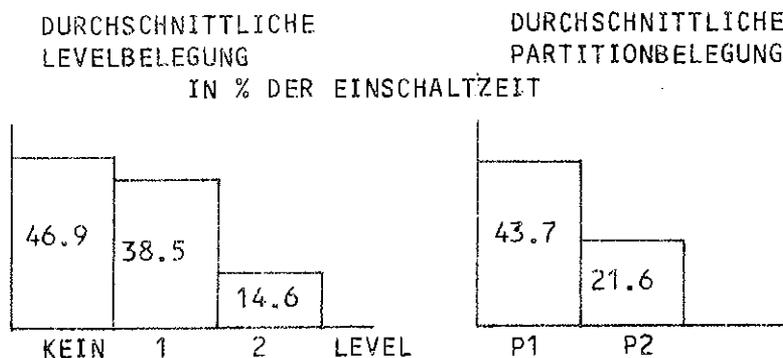
⋮

Es lassen sich natürlich noch eine Menge anderer Aussagen aufgrund der Dayfile-Eintragungen machen, da die Information sehr umfassend ist. Die angegebenen Auswertungen schienen uns jedoch den Aufwand zu rechtfertigen und zu Studien des Rechenbetriebs innerhalb des an der Hybridrechenanlage entwickelten Betriebssystems JCS/TS 7 ausreichend.

DISKUSSION DER ERGEBNISSE

Die Auswertung des Dayfiles wird etwa seit Anfang Oktober 1978 regelmäßig vorgenommen. Da der Dayfile nur eine beschränkte Anzahl von Eintragungen aufnehmen kann, muß der Inhalt des Dayfiles öfters gerettet werden, spätestens jedoch unmittelbar vor dem Systemausstieg. Die so erzeugten Files werden pro Tag zusammengefaßt und gelangen zur Auswertung.

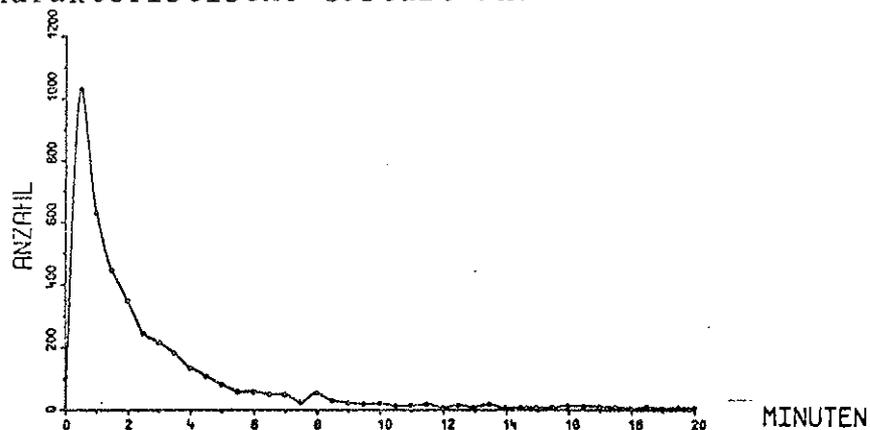
Über den bisherigen Beobachtungszeitraum konnten folgende Feststellungen getroffen werden:



Der hohe Prozentsatz der Zeit, in der keine Partition (kein Level) belegt ist, kann nicht so interpretiert werden, daß etwa doppelt soviele Jobs gerechnet werden könnten. Zur Durchführung von Benutzerjobs sind eine Reihe von Tätigkeiten erforderlich, die hier nicht erfaßt sind und die vom Operator notwendigerweise durchgeführt werden müssen, bevor weitere Jobs verarbeitet werden können. Zu diesen Tätigkeiten gehören: Vorbereiten und Aufarbeiten von Jobs (Kartenverarbeitung, Aufarbeiten von Line Printer- und Data Plotter-Ausgabe), Überwachung der Ein/Ausgabegeräte, Organisation der Files auf den Districts, Beschaffung von Betriebsmitteln (Papier, etc.), etc.

Da im Betriebssystem JCS/TS 7 Line Printer- und Data Plotter- Ausgabe im Scratch/Spooling-System organisiert werden, erfolgt die Ausgabe auf diesen Geräten erst nach der Beendigung der Exekution des Jobs und scheint in der angeführten Levelbelegung nicht auf. Der Line Printer-Ausdruck eines durchschnittlichen Jobs ist innerhalb von weniger als einer Minute gedruckt. Der Data Plotter-Output hingegen dauert meist etwa 5 Minuten, im Extremfall bis zu einigen Stunden. Bei einem täglichen Betrieb von 8 bis 20 Uhr mit nur zwei Operatoren scheint daher nur eine geringfügige Steigerung der Maschinenauslastung möglich.

Die Verteilung der Jobzeiten nimmt über einen längeren Beobachtungszeitraum eine charakteristische Gestalt an:



Die Auswertung der Fehlermeldungen zeigt folgende Ergebnisse:

- häufigste Fehlermeldungen, die zum Abbruch eines Programms führen:

ATTEMPT TO LOAD NOT EXISTANT CI-FILE (aufgrund von FORTRAN-Fehlern oder Bindefehlern kann ein Programm nicht exekutiert werden),
 TASK TERMINATED (wobei nicht festgestellt werden kann, ob der Task Zeit-Überlauf hatte oder vom Operator abgebrochen werden mußte),
 END OF CARD READER INPUT FILE (zu wenig Datenkarten)
 ACCESS TO MEMORY OUT OF PARTITION (Fehlsprung im Programm),
 I/O LIMIT REACHED (Überschreitung des Input/Output Limits).

- Fehlermeldungen ohne Programmabbruch:

INPUT ERROR (Fehler in Datenkarten oder Tippfehler),
 REAL TO SCALED FRACTION OVERFLOW (Wertüberlauf in Hybridprogrammen),
 PLOT RANGE VIOLATION: Y>9.5 INCH (Plotter-Zeichenfeldüberschreitung).

- häufigste FORTRAN-Fehlermeldungen:

STATEMENT OUT OF ORDER } (meist fehlende Dimensionierung oder Lochfehler)
 ILLEGAL CHARACTER }

- durchschnittliche Anzahl von Tasks pro Job: 3 (meist /FOR, /BUILD, /RUN)

- durchschnittliche Anzahl von Fehlermeldungen pro Job: 1.18

- durchschnittliche Anzahl von verschiedenen Jobnummern, die an einem Tag rechnen: 14

I. Husinsky

DIE VERWENDUNG VON DIGITAL SETZBAREN FUNKTIONSGEBERN

Die digital setzbaren Funktionsgeber (DCFG) werden im Digitalprogramm gesetzt und liefern die analoge Funktion einer analogen Variablen am Analogrechner. Es sind vier Funktionsgeber vorhanden: DCFG 0, DCFG 1, DCFG 2 und DCFG 3 (Feld zwischen der zweiten und dritten Reihe der analogen Komponenten am Steckbrett). Die Funktion, die am Analogrechner zur Verfügung steht, wird durch maximal 16 Stützpunkte definiert, wobei die Werte durch lineare Interpolation approximiert werden.

Zur Berechnung der Stützstellen, zum Skalieren und zum Setzen der DCFG's stehen dem Benutzer die Programme DCFG, LFGS, LFGR und LFGFD aus der HYBRID LINKAGE (LOAD,D9 HYBRID) zur Verfügung. Diese Programme wurden erst kürzlich verbessert und erweitert. In der alten Version wurden nicht alle Möglichkeiten, die in der Hardware vorgesehen sind, realisiert. So konnte man z. B. mit dem Aufruf von DCFG nicht die Stützpunkte für Hysteresen berechnen. Außerdem wurden nicht alle Beschränkungen, die von EAI vorgegeben sind, überprüft; wie z. B., daß für je zwei Stützpunkte gelten muß:

$$|y_{i+1} - y_i| < 1.3 \quad \text{für } x_{i+1} \neq x_i \quad \text{mit } y = \hat{y}(x)$$

(x und y schon skaliert im Intervall (-1,1))

Die Erweiterung der Software ermöglicht es dem Hybridbenutzer nun, neben stetigen Funktionen auch Hysteresen und Funktionen, die bis zu drei Sprungstellen mit Sprunghöhe > 1.3 (schon skaliert) haben, zu definieren. Außerdem können jetzt die Skalierungsfaktoren für x und y(x) vom Benutzer angegeben werden.

Das Programm DCFG berechnet für eine gewünschte Funktion, die durch eine FORTRAN FUNCTION FUN(X) im Intervall [X1,X2] definiert ist, die optimale Lage von maximal 16 Stützpunkten, bringt die Koordinaten der Stützpunkte auf DCFG-Format und speichert die Werte in einem Feld ab. Die DCFG's können wahlweise auf diese Werte gesetzt werden. Über den Commonblock /CDCFG/XSCALE,YSCALE,ISCALE stehen die Skalierungsfaktoren zur Verfügung. Falls ISCALE beim Aufruf den Wert 1 hat, müssen die Skalierungsfaktoren vom Benutzer vorgegeben sein. Mit ISCALE = 2 (standardmäßig gesetzt) wird automatisch skaliert. In diesem Falle können die Skalierungsfaktoren über XSCALE und YSCALE ausgelesen werden.

BEISPIEL 1: DURCH FUNKTIONSUNTERPROGRAMME DEFINIERTE FUNKTIONEN

Berechnung der Stützpunkte für stetige Funktionen bzw. Funktionen mit Sprungstellen.

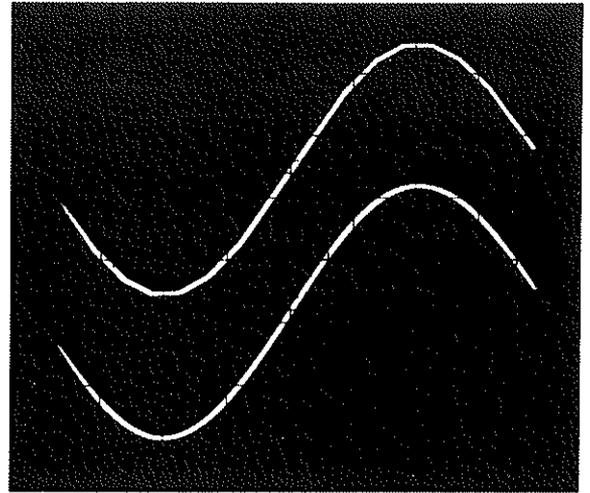
Gewählt wurden die Funktionen:

FUN1 = SIN(X)	X ∈ [-3,3]
FUN2 = $\begin{cases} \text{COS}(X) \\ 1 \end{cases}$	X ∈ (-1.58, 1.58) X ∈ [-2.5, -1.58] und X ∈ [1.58, 2.5]
FUN3 = [X]	X ∈ [-2.9, 2.9]

Digitalprogramme zum Setzen der DCFG's:
für FUN1 :

```
DIMENSION H(16)
EXTERNAL FUN1
CALL HYBRID
CALL DCFG (1,FUN1,-3.,3.,1,H,IER)
CALL EXIT
END
```

```
FUNCTION FUN1(X)
FUN1 = SIN(X)
RETURN
END
```



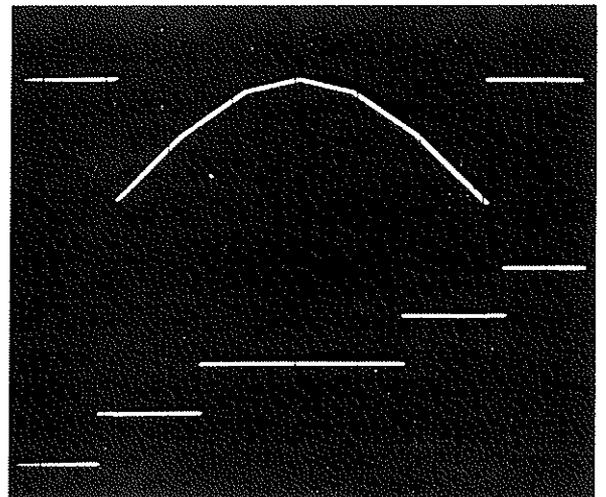
Die Funktion FUN1 wurde einmal mit dem DCFG (16 Stützpunkte) realisiert und zum Vergleich auch direkt am Analogrechner gesteckt (im Bild die untere Funktion).

für FUN2 und FUN3:

```
DIMENSION H2(16),H3(16)
EXTERNAL FUN2,FUN3
CALL HYBRID
CALL DCFG (1,FUN2,-2.5,2.5,12,H2,IER)
CALL DCFG (2,FUN3,-2.9,2.9,13,H3,IER)
CALL EXIT
END
```

```
FUNCTION FUN2(X)
FUN2 = 1.
IF(X.GT.-1.58.AND.X.LT.1.58) FUN2=COS(X)
RETURN
END
```

```
FUNCTION FUN3(X)
FUN3 = FLOAT(INT(X))
RETURN
END
```



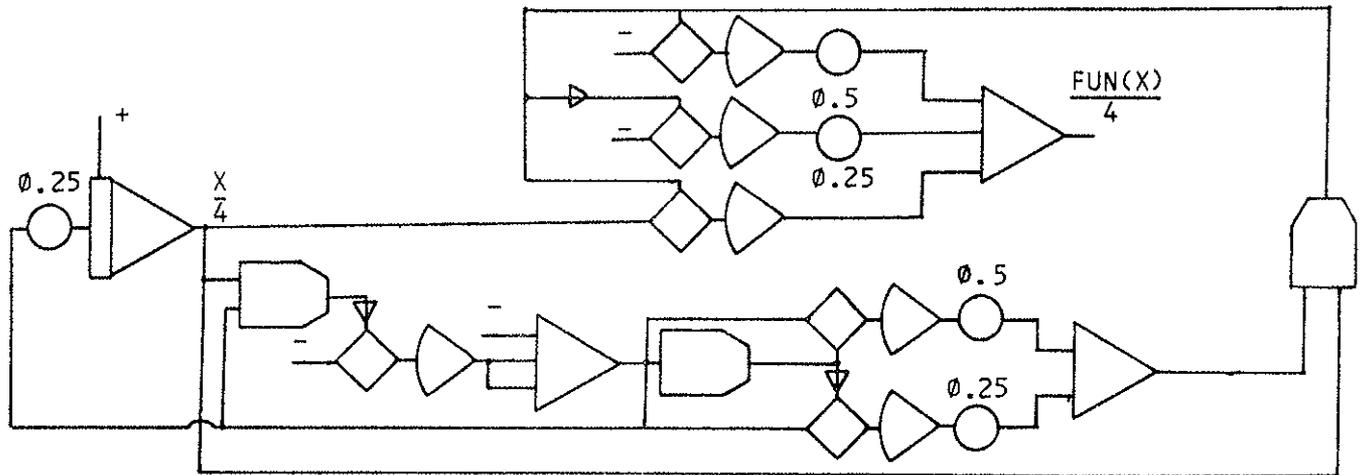
BEISPIEL 2: VERGLEICH DES SCHALTAUFWANDES ANHAND EINER FUNKTION MIT HYSTERESE

Der Commonblock /HYST/X1HY,X2HY,IHY,IR enthält die Variablen, die zur Berechnung der Stützpunkte benötigt werden, wobei die Variablen folgende Bedeutung haben.

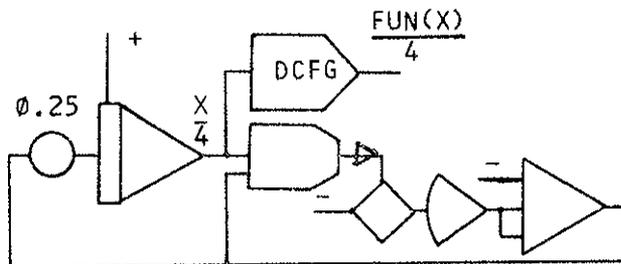
- X1HY gibt den Beginn der Hystereseschleife an
 X2HY gibt das Ende der Hystereseschleife an
 IHY kennzeichnet die Art der Hysterese, die vorliegt. Sind bei X1HY und bei X2HY keine Sprungstellen, so muß IHY=1 gesetzt werden; sonst muß IHY den Wert 3 erhalten.
 IR wird im Unterprogramm gesetzt und ermöglicht dem Benutzer, auf einfache Weise die Funktionswerte für die 2 verschiedenen Hystereseäste in der FUNCTION FUN(X) anzugeben. Ist IR=1 müssen die Funktionswerte für eine aufsteigende Folge von X-Werten angegeben werden, bei -1 die Funktionswerte für eine fallende Folge von X-Werten.
- Der Commonblock muß sowohl im Hauptprogramm als auch in FUN deklariert werden.

$$\text{FUN}(X) = \left\{ \begin{array}{ll} -1 & X \in [-4, 1] \\ X-2 & X \in [1, 4] \end{array} \right\} \quad X \uparrow \\
 \left\{ \begin{array}{ll} -1 & X \in [-4, -2] \\ X-2 & X \in [-2, 4] \end{array} \right\} \quad X \downarrow$$

Falls die Funktion ohne DCFG dargestellt werden soll, braucht man folgende Analogschaltung:



mit DCFG:



Digitalprogramm zum Setzen des DCFG:

```

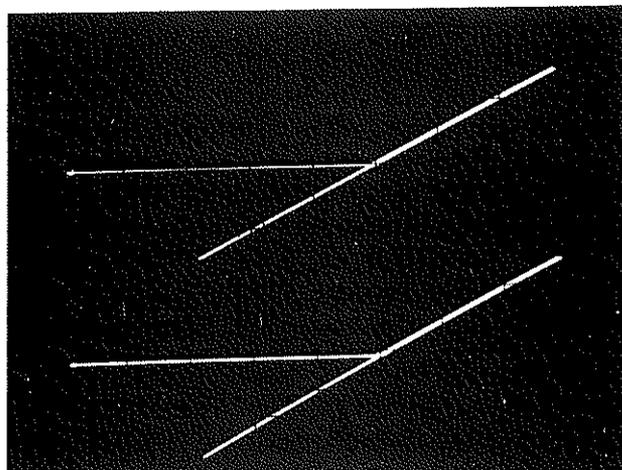
DIMENSION H(16)
COMMON/HYST/X1HY,X2HY,IHY,IR
EXTERNAL FUN
CALL HYBRID
X1HY = -2.
X2HY = 1.
IHY = 3
CALL DCFG (1,FUN,-4.,4.,I,H,IER)
CALL EXIT
END

```

```

FUNCTION FUN(X)
COMMON/HYST/X1HY,X2HY,IHY,IR
FUN = -1.
IF(IR.EQ.-1.AND.X.GT.-2.) FUN=X-2.
IF(IR.EQ.1.AND.X.GT.1.) FUN=X-2.
RETURN
END

```



Analog erzeugte Funktion (oben),
und mit DCFG erzeugte Funktion
(unten)

Das Programm DCFG berechnet also die Stützpunkte für eine analytisch gegebene Funktion. Falls man die Stützpunkte jedoch vorgeben will, transformiert das Programm LFGS die Koordinaten der Stützpunkte in das DCFG-Format. Dieses Programm kommt in den folgenden Beispielen zur Anwendung. Durch das Setzen eines Eingangsparameters kann angegeben werden, ob die Werte schon skaliert vorliegen, oder ob erst eine Skalierung durchgeführt werden muß.

BEISPIEL 3: EINFACHE PARAMETERVARIATION MIT HILFE EINES DCFG

Gegeben sei ein mathematisches Modell in Form eines Differentialgleichungssystems. Im Modell sind einige Parameter enthalten, deren Werte nicht bekannt sind. Die Lösungskurven sind in Form von Meßwerten an bestimmten Punkten bekannt. Gesucht sind die Werte der Parameter, sodaß die Lösungskurven der Differentialgleichungen an den Stützpunkten mit den Meßwerten übereinstimmen. Das Problem kann gelöst werden, indem man die DCFG's auf die gegebenen Stützpunkte setzt und die Differentialgleichungen im vorgegebenen Intervall löst. Durch Vergleich der Lösungskurven mit den von den DCFG's approximierten Funktionen erhält man Fehlerfunktionen. Mit Hilfe dieser Fehlerfunktionen ändert man die Parameter, sodaß man iterativ bis zu einer vorgegebenen Genauigkeit zur Lösung des Problems kommt.

Als einfaches Beispiel sei die Differentialgleichung

$$\ddot{y}'' = y \quad \text{Funktion: } y = e^t + a \cdot \cos t, \quad a = 5$$

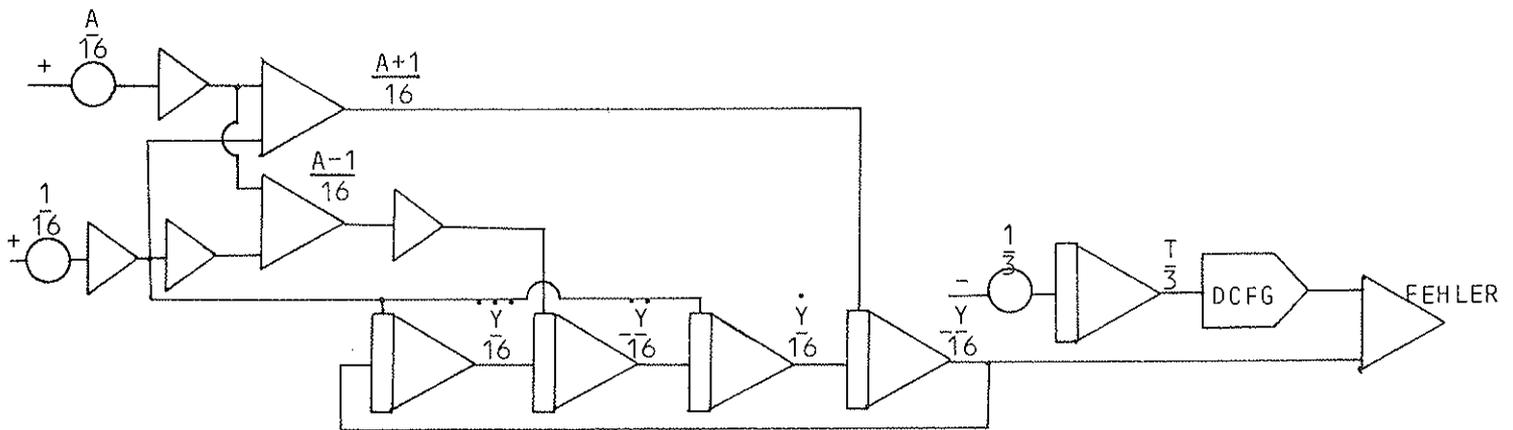
mit den Anfangswerten $y(0) = a + 1, \quad \dot{y}(0) = 1$

$$\dot{y}(0) = 1 - a, \quad \ddot{y}(0) = 1 \quad \text{gegeben.}$$

Meßwerte zu äquidistanten Stützpunkten:

t	y	t	y	t	y	t	y	t	y
0	6	0.75	5.77544	1.5	4.83538	2.25	6.34687	3	15.13557
0.25	6.12859	1	5.41979	1.75	4.86337	2.5	8.17678		
0.5	6.03663	1.25	5.06695	2	5.30832	2.75	11.02112		

Analogschaltung:



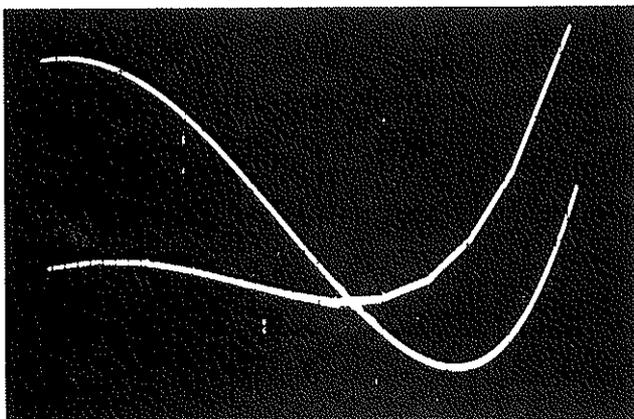
Für die Variation von a wurde ein Handpotentiometer verwendet.

Digitalprogramm zum Setzen des DCFG (X und Y Werte schon skaliert):

```

DIMENSION X(13),Y(13),H(16)
CALL HYBRID
DO 1 I = 1,12
J = I-1
X(I) = FLOAT(J)*.25/3.
1 CONTINUE
X(13) = .9999
Y(1) = .375
Y(2) = .38304
Y(3) = .37729
Y(4) = .36897
Y(5) = .35874
Y(6) = .34668
Y(7) = .33221
Y(8) = .30396
Y(9) = .33177
Y(10) = .39668
Y(11) = .51105
Y(12) = .68882
Y(13) = .94597
CALL LFGS (1,2,2,13,XSCALE,YSCALE,IER,X,Y,H,1)
CALL EXIT
END

```



Lösungskurve bei einem beliebigen Wert für a



Lösungskurve beim ausgelesenen Parameterwert von 5.008

Das Programm LFGR transformiert die Y-Koordinaten der Stützpunkte in das DCFG-Format. Die Werte können skaliert oder unskaliert angegeben werden. (Die X-Koordinaten sind die gleichen wie nach dem letzten Aufruf von DCFG bzw. LFGR.)

Das Programm LFGFD setzt einen DCFG. Die Koordinaten der Stützpunkte müssen schon im DCFG-Format in einem Feld abgespeichert sein. Das Programm braucht ungefähr 70 µsec zum Setzen eines DCFG's.

Da die Möglichkeit besteht, in relativ kurzer Zeit einen DCFG neu zu laden, kann man, falls mehrere verschiedene Funktionen während eines OP-Intervalls am Analogrechner benötigt werden, zuerst durch Aufruf von DCFG bzw. LFGR die Stützpunkte der Funktionen in verschiedene Felder abspeichern und zu einem bestimmten Zeitpunkt (Setzen einer Senseline bzw. Aktivieren eines Interruptunterprogrammes) durch LFGFD den DCFG neu setzen. Obwohl der DCFG immer wieder neu gesetzt wird, kann am Analogrechner in NMS gerechnet werden.

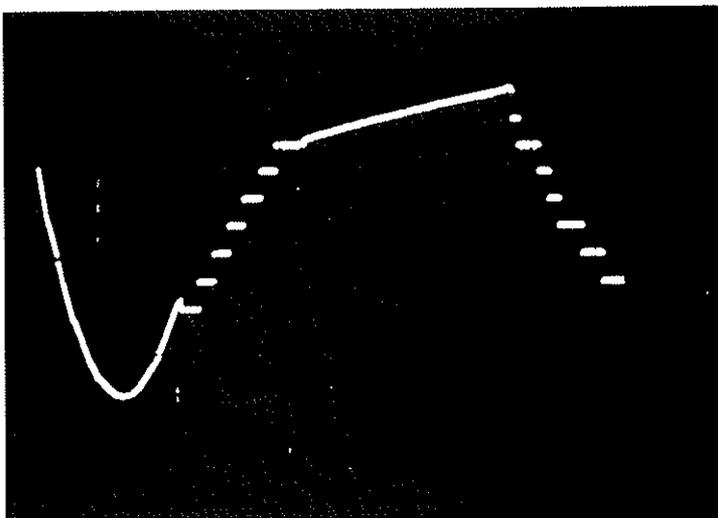
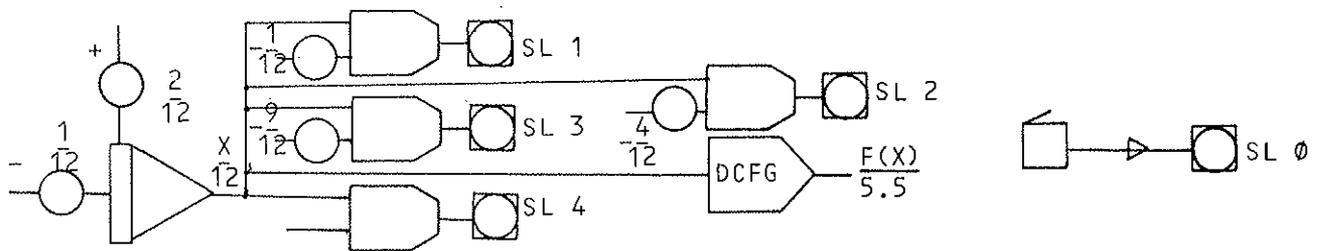
BEISPIEL 4: ERZEUGUNG ZUSAMMENGESETZTER FUNKTIONEN

Eine Funktion sei im Intervall [-2,12] durch 61 Stützpunkte gegeben und soll am Analogrechner realisiert werden.

Da ein DCFG auf maximal 16 Stützpunkte gesetzt werden kann, muß man das Intervall in 4 Teilintervalle zerlegen. (Die Stützpunkte am Rand der Intervalle müssen doppelt genommen werden.) Zerlegung: [-2,1], [1,4], [4,9], [9,12]. Die Stützpunkte müssen nun im DCFG-Format in 4 Felder (Länge 16) abgespeichert werden. In einer Schleife kann durch Abfrage einer Senseline jeweils am Ende eines Teilintervalls der DCFG neu gesetzt werden.

(Da bei der Zerlegung nicht mehr als 4 Teilintervalle notwendig sind, könnten auch, anstatt einen DCFG immer wieder neu zu setzen, alle vier DCFG's verwendet werden.)

Analogschaltung:



- Y1: x^2 [-2,1]
- Y2: TREPPE [1,4]
- Y3: $\sqrt{x} + 2.5$ [4,9]
- Y4: TREPPE [9,12]

Y1	Y2	Y3	Y4
----	----	----	----

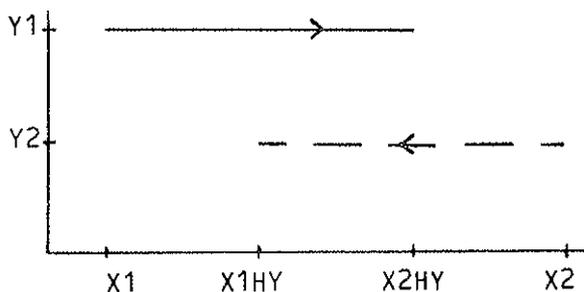
Digitalprogramm:

```

        DIMENSION X(16),Y(16),H(16,4)
        LOGICAL ILOG
        CALL HYBRID
        CALL AMIC
        CALL TSNMS
        DO 1 I = 1,4
        READ (5,11) X
        READ (5,11) Y
        CALL LFGS (-1,2.2,16,XSCALE,YSCALE,IER,X,Y,H(1,I),1)
1      CONTINUE
4      CALL LOCK(32000)
        CALL AMPP
        DO 2 I = 1,4
        CALL LFGFD (1,16,IER,H(1,I))
        CALL WAITSL(I)
2      CONTINUE
        CALL RELOCK
        CALL LFGFD (1,16,IER,H(1,1))
        CALL AMIC
        CALL SL (0,ILOG)
        IF (ILOG) GOTO 4
11     FORMAT (16F5.2)
        CALL EXIT
        END
    
```

Um den Fehler zwischen der eigentlichen Funktion und der durch den DCFG approximierten Funktion möglichst klein zu halten, kann man oben beschriebenes Verfahren auch für Funktionen, die durch Gleichungen gegeben sind, anwenden. Nur muß man statt LFGS das Unterprogramm DCFG für jedes Teilintervall aufrufen. Gibt man die Funktion in Form von Stützpunkten an, so muß man beachten, daß bei Sprungstellen ein- und dieselbe X-Koordinate mit zwei verschiedenen Funktionswerten angeführt werden muß. Bei Hystereseen muß man auf die richtige Reihenfolge der Stützpunkte achten. Zuerst wird der erste Hystereseast vom Intervallanfang bis zum Ende der Hysterese-schleife, dann der zweite Hystereseast vom Anfang der Hysterese-schleife bis zum Intervallende angegeben.

BEISPIEL 5: BERECHNUNG DER STÜTZPUNKTE FÜR HYSTERESEN



$[X_1, X_2]$

x	f(x)
X1	Y1
X1HY	Y1
X2HY	Y1
X1HY	Y2
X2HY	Y2
X2	Y2

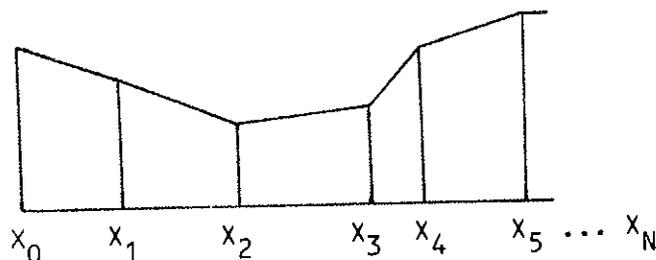
Prinzipiell sind die X-Koordinaten in steigender Reihenfolge anzugeben. Bei der Programmierung von Hysterese ist darauf zu achten, daß außerhalb der Hysterese-schleife die Funktion möglichst einfach ist, da die Stützpunkte auf die zwei verschiedenen Funktionen aufgeteilt werden müssen; d.h. pro Hystereseast stehen maximal 6 Stützpunkte zur Verfügung (4 Stützpunkte werden für die Intervallgrenzen und Beginn und Ende der Hysterese-schleife benötigt).

Zu beachten:

Bei der Verwendung von DCFG's kann am Analogrechner nicht in FMS gerechnet werden; außerdem sollte man Sprünge beim Eingangssignal für die DCFG's vermeiden, da es dadurch kurzzeitig zu Übersteuerungen kommen kann, die wiederum zu einer Verfälschung der Resultate führen.

BEISPIEL 6: BERECHNUNG EINES BESTIMMTEN INTEGRALS MIT HILFE VON DCFG'S

Nachfolgendes Beispiel zeigt auf einfache Weise, wie man unter anderem durch Verwendung von DCFG's den Schwierigkeiten der Datenübertragung entgegen gehen kann. Gegeben ist ein Polygonzug mit n Stützpunkten ($n \leq 16$):



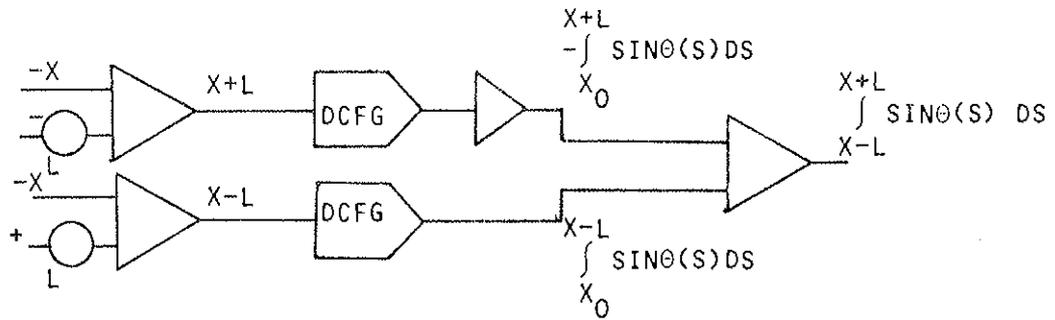
Am Analogrechner wird zu jedem Punkt x $\int_{x-L}^{x+L} \sin\theta(s) ds$ benötigt, wobei L variabel sein soll und $\tan\theta(x)$ der Anstieg der Funktion im Punkt x ist und $x \in [x_0, x_n]$.

$$\text{Berechnung: } \int_{x-L}^{x+L} \sin\theta(s) ds = \int_{x_0}^{x+L} \sin\theta(s) ds - \int_{x_0}^{x-L} \sin\theta(s) ds$$

Die Integrale $\int_{x_0}^{x_i} \sin\theta(s) ds$ lassen sich digital sehr leicht berechnen, da θ in jedem Intervall $[x_i, x_{i+1}]$ konstant ist. Sei θ_i der Winkel im Intervall $[x_i, x_{i+1}]$

$$\begin{aligned} \int_{x_0}^{x_i} \sin\theta(s) ds &= \int_{x_0}^{x_1} \sin\theta(s) ds + \int_{x_1}^{x_2} \sin\theta(s) ds + \dots + \int_{x_{i-1}}^{x_i} \sin\theta(s) ds = \\ &= (x_1 - x_0) \sin\theta_0 + (x_2 - x_1) \sin\theta_1 + \dots + (x_i - x_{i-1}) \sin\theta_{i-1} \end{aligned}$$

Es gilt weiters, daß der Wert des Integrals zu einem Punkt $x \in [x_i, x_{i+1}]$ die Linearkombination der Werte der Integrale zum Punkt x_i und zum Punkt x_{i+1} ist. Speichert man nun in ein Feld die Werte der Integrale für jeden Stützpunkt ab, transformiert diese durch den Aufruf von LFGS in DCFG-Format und setzt 2 DCFG's auf diese Werte, so hat man zu jedem Punkt x den Wert des gewünschten Integrals.



Würde man keine DCFG's verwenden, müßte man jeweils durch Abfrage des momentanen X-Wertes und des momentanen Wertes von L den richtigen Funktionswert zu einem bestimmten Zeitpunkt überspielen. Außer den Schwierigkeiten bei der Synchronisation erhält man zusätzlich einen Fehler, der dadurch entsteht, daß man ein gewisses Zeitintervall mit dem gerade überspielten, konstanten Funktionswert arbeiten muß.

SUMMARY

The 680 Analog Computer at the Hybrid Computation Centre of the Technical University of Vienna is now equipped with four Digital Controlled Function Generators (DCFG). A new software has been developed for calculating the breakpoints, scaling and setting the DCFGs (Subroutines DCFG, LFGR, LFSG, LFGFD of the Hybrid Linkage software). The subroutines LFGR, LFSG, and LFGFD have been modified for our system, in regard to a comfortable usage and to be compatible with the new subroutine DCFG.

The user may define any function in a FORTRAN FUNCTION. This function does not have to be continuous but may contain up to three discontinuities or may be a hysteresis. The subroutine DCFG then computes and scales optimal breakpoints and sets the DCFG. The scaling factors may be supplied by the user or be computed automatically in which case the scaling factors are available for the calling program. Several examples are given which demonstrate the advantages and various applications of the DCFGs using the Hybrid Linkage software.

E. Berger, F. Berger

GASFLUSS DURCH EIN LANGES ROHR

Beispiel für die Lösung eines Systems nichtlinearer partieller Differentialgleichungen durch die Methode der Zeitdiskretisierung.

PROBLEMSTELLUNG

Betrachtet man den Gasfluß in einer langen geraden zylindrischen Rohrleitung mit konstantem Durchmesser D , so sei der Druck $P=P(x,t)$ und die Flußrate $Q=Q(x,t)$. Am einen Ende, $x=0$, wird das Gas aus einem Reservoir eingespeist, am anderen Ende, $x=X$, wird das Gas verbraucht (Abb. 1).

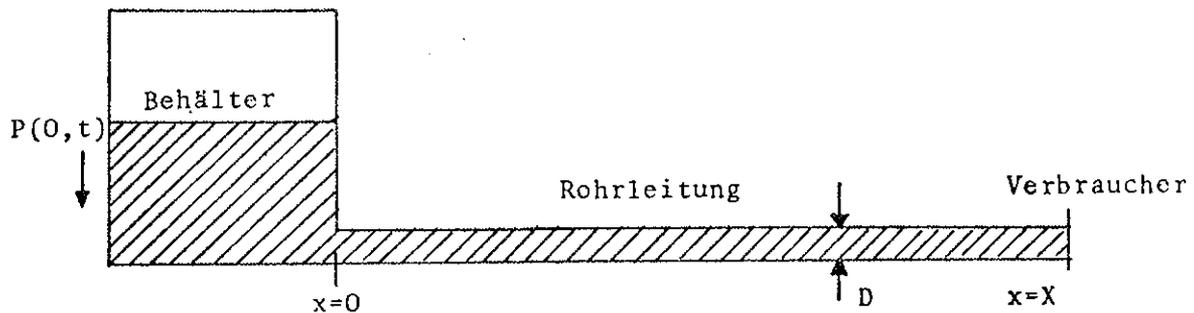


Abbildung 1

Nach [1] erhält man die folgende Grundgleichung für den Gasfluß für Zeiten $t > 0$ aus den Gesetzen der Masseerhaltung und des Energiegleichgewichts.

$$\frac{\partial Q}{\partial x} = -A \cdot D^2 \frac{\partial P}{\partial t}$$

(1)

$$\frac{\partial P^2}{\partial x} = -\frac{B}{D^5} Q^2$$

wobei A und B Funktionen der Temperatur sind. Vereinfachend wird vorausgesetzt, daß die Reibung an der Rohrwand und andere ähnliche Einflüsse konstant gehalten werden. Zur Zeit $t=0$ sei $P(X,0)=P_0$ und $Q(X,0)=Q_0$, d.h. am Ende der Rohrleitung erfolgt keine Entnahme.

Ferner seien die folgenden Randbedingungen vorgegeben:

$$P(0,t) = P_0 \qquad Q(X,t) = (1 - \cos t)/2 \quad (2a)$$

$$P(0,t) = P_0 \qquad Q(X,t) = Q_1 \quad t > 0 \quad (2b)$$

d.h. der Druck im Behälter wird konstant gehalten und die durch den Verbrauch bei $x=X$ bestimmte Flußrate ändert sich einmal periodisch mit der Zeit, bzw. es tritt eine sprunghafte konstante Entnahme ein. In beiden Fällen ist die zeitliche Änderung des Drucks und der Flußrate über der ganzen Rohrlänge gesucht.

ZEITDISKRETISIERUNG FÜHRT ZU RANDWERTPROBLEM

Zum Aufbau einer Simulation dieses Systems am Hybridrechner wurde die Methode der Zeitdiskretisierung von (1) gewählt (siehe [2]), was zu

$$\frac{dQ_i}{dx} = -\frac{AD^2}{\Delta t} (P_i - P_{i-1}) \quad i=1,2, \dots$$

$$\frac{dP_i^2}{dx} = -\frac{B}{D^5} Q_i^2 \quad (3)$$

führt, wobei $Q_i = Q_i(x, t_i) = Q_i(x)$ und $P_i = P_i(x, t_i) = P_i(x)$ zu jedem festen Zeitpunkt $t_i = i \cdot \Delta t$ Funktionen von x darstellen.

Die Anfangs- und Randbedingungen von (1) werden jetzt zu einem Zwei-Punkt-Randwertproblem von (3)

$$P_i(0) = P_0 \quad i=1,2, \dots \quad (4)$$

$$Q_i(X) = (1 - \cos t_i)/2 \text{ bzw. } Q_1(X) = Q_1$$

mit $P_0(X) = P_0$ und $Q_0(X) = 0$

Zu jedem Zeitschritt t_i muß die im Digitalteil tabellierte Funktion $P_{i-1}(x_j)$, $j=1(1)N$, $x_1=0$, $x_N=X$, dem Analogteil überspielt und iterativ das Randwertproblem (4) für $Q_i(X)$ durch Verändern von $Q_i(0)$ gelöst werden.

REALISIERUNG AM HYBRIDRECHNER

Das folgende Blockschaltbild (Abb. 2) zeigt den prinzipiellen Aufbau der verwendeten Anlogschaltung, die Abgrenzung der analogen und digitalen Rechnung, sowie die wichtigsten Funktionen des Digitalprogramms für die Simulation.

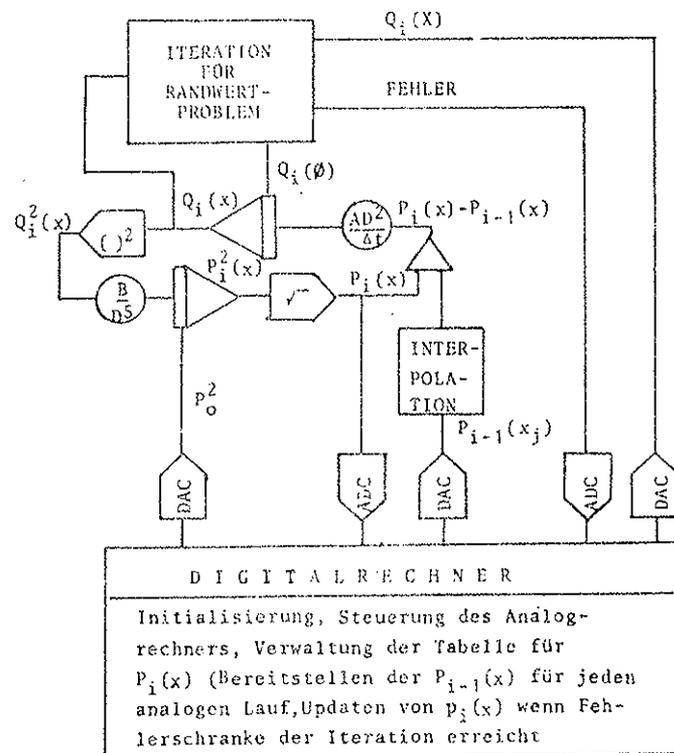


Abbildung 2
Blockschaltbild

Der Endpunkt X und die Anzahl der zu Übertragenden Punkte werden am Beginn des Programmablaufs eingelesen. Aus diesen Werten wird, zusammen mit der festen Abtastfrequenz von 1kHz, die eine maximale Rechengeschwindigkeit ermöglicht, eine optimale Zeittransformation automatisch ermittelt und der digitale ABCD-Timer entsprechend geladen.

In der Initialisierungsphase, die jederzeit durch ein Senseline-Signal angesteuert werden kann, werden die über Handpots einstellbaren Werte für A, B, D und P_0^2 über ADC's und die digital berechneten Potentiometer-Werte über DAC's ausgegeben, die am DVM ausgelesen werden können. Um über einen größeren Wertebereich die Parameter variieren zu können und gleichzeitig die Skalierung des Problems zu verbessern, werden sowohl die Eingänge der beiden Integrierer für $p^2(x)$ und $Q(x)$ durch Relays zwischen 1 und 10 als auch die individuellen Zeitkonstanten zwischen FS und NMS automatisch umgeschaltet.

Die theoretischen Potentiometerwerte können so 4 Dekaden umfassen. Abbildung 3 zeigt auch die verwendeten Schaltungen zur analogen linearen Interpolation der Datenpunkte. Das Auslesen und Überspielen von $Q_1(x)$ dient nur zur gemeinsamen Darstellung von Druck und Flußrate am Oszillographen.

Besonders sei noch auf die verwendete Iterationsschaltung zur Lösung des Randwertproblems hingewiesen. Der verwendete Typus der Nichtlinearität des Problems führt dazu, daß es bei der üblichen Vorgangsweise - die Änderung des gesuchten Anfangswertes $Q_1(0)$ proportional dem Fehler $[Q_1(X) - Q_{soll}(X)]$ anzusetzen - entweder zu extrem hohen Iterationszahlen bei zu kleinem oder zu Instabilität bei zu großem Proportionalitätsfaktor kommt. Dagegen hat es sich sehr bewährt, diese Änderung proportional dem relativen Fehler $(Q_1(X) - Q_{soll}(X)) / \text{Min}(Q_{soll} + 0.1, 1)$, anzusetzen.

Die Synchronisation der Datenübertragung erfolgt über eine Senseline, die Datenübertragung in beiden Richtungen über den Standard-Kanal, wobei nur nach Beendigung jeder Iteration (Fehler kleiner EPS) die analogen Variablen $Q_1(x)$ und $P_1(x)$ ausgelesen werden.

```

C      IC-STATE
20 CONTINUE
   CALL TSTOP
   CALL RELOCK
   CALL SL(1,SL1)
   CALL SL(1,SL1)
   IF(,NOT,SL1) GOTO 1
   CALL AOC(3,S)
   FEHL=SABS(S)
   IF(LEPS) IND=IND+1
   IF(,NOT,IANF) LEPS=FEHL.LE.EPS
   IANF=.FALSE.
   CHANGE = IND.EQ.0 .AND. LEPS
   IF(IND.EQ.0.OR.,NOT,LEPS) GOTO 60
   T=T+DELT
   IND=0
   LEPS=.FALSE.
60      I=0
      S=Q(T)
      CALL LOCK(300)
      P(ICOUNT+1)=P(ICOUNT)
      CALL DAC1(S)
      S=P(1)
      S1=QH(1)
      CALL DAC0(S)
      CALL DAC2(S1)
      CALL TRADAC
      CALL TSTART

C      ITERATE
100     05  I,1
        07  I
        LA  P,1,1
        OCT 3140
        LA  QH,1
        OCT 3142
        OCT 4040
        OCT 4040
        SKN
        J   =2
        LA  CHANGE
        SKN
        J   ,110
        OCT 26740
        OCT 5064
        OCT 2064
        STA P,1
        OCT 2065
        STA QH,1
110     LA  I
        C   ICOUNT
        SG
        J   ,100
        J   ,20
900     CALL AMPC
        CALL EXIT
        END

```

Haupt-
schleife

Abbildung 4
Auszug aus dem FORTRAN-
Programm mit Inline-
Assemblercodierung

Die Programmierung erfolgte in FORTRAN, nur die zeitkritischen Abschnitte wurden direkt in Inline-Assembler geschrieben (siehe Abb. 4).

ERGEBNISSE

Die folgenden Abbildungen 5 und 6 zeigen den Verlauf von Druck (oben) und Flußrate (unten) zu verschiedenen Zeiten bei periodischer Entnahme am Ende der Rohrleitung. Man sieht deutlich, wie bei steigender Entnahme auch die Flußrate über der Länge ansteigt, während sie bei wieder sinkender Entnahme über der Länge fällt. Der Druck wird am linken Ende konstant gehalten und nimmt bei jeder Entnahme gegen das Rohrende zu ab. Auch hier ist bei wieder sinkender Entnahme der "Umkehreffekt" zu beobachten.

Abbildung 7 zeigt die steigende Entnahme bei verkleinertem Rohrquerschnitt. Man sieht deutlich, daß der Druck am Rohrende bereits nahezu auf Null abfällt. Abbildung 8 zeigt das Einschwingen nach plötzlicher konstant gleichbleibender Entnahme.

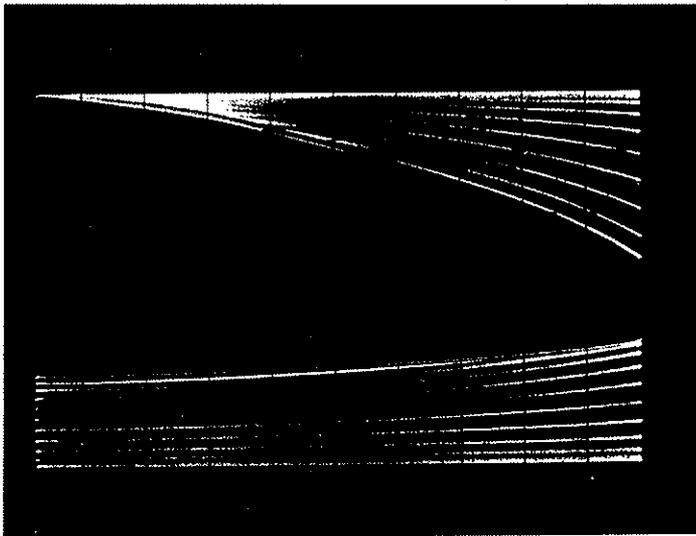


Abbildung 5
Steigende Entnahme

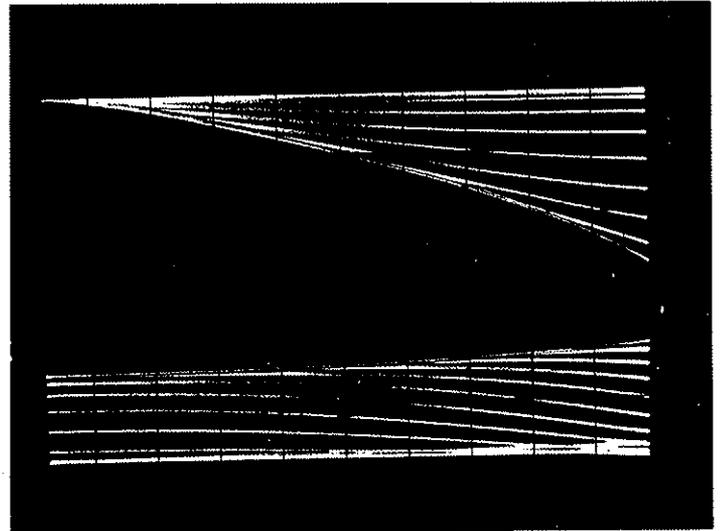


Abbildung 6
Sinkende Entnahme

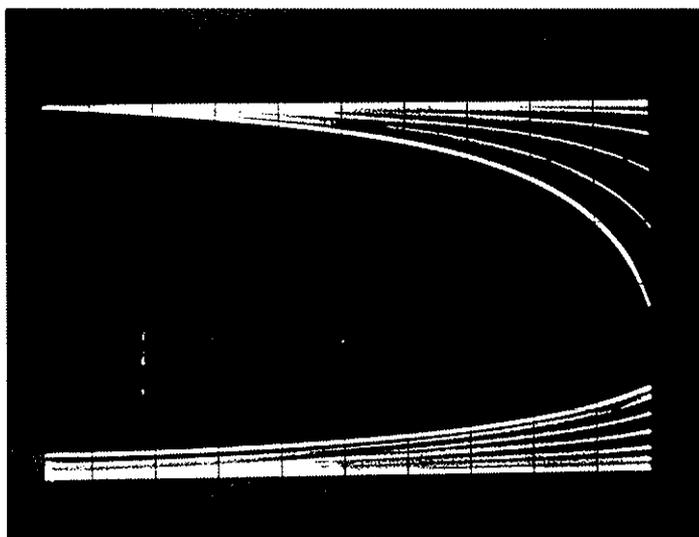


Abbildung 7
Steigende Entnahme
bei verkleinertem Rohrquerschnitt

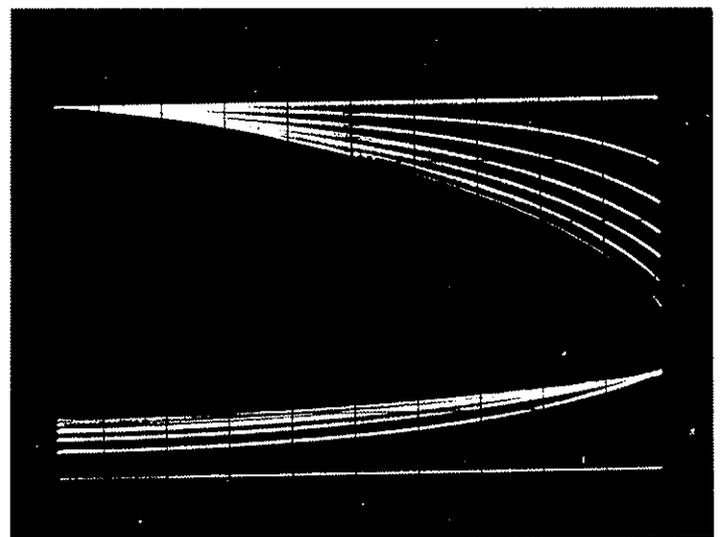


Abbildung 8
Plötzliche Entnahme

ZUSAMMENFASSUNG

Es wurde die hybride Simulation des Gasflusses durch eine lange Rohrleitung mit Hilfe der Zeitdiskretisierung eines Systems nichtlinearer partieller Differentialgleichungen vorgestellt. Das Lösungsverfahren ist prinzipiell leicht auf andere Systeme partieller Differentialgleichungen zu übertragen, die durch Zeitdiskretisierung zu lösen sind. Auf Wunsch kann das Quellprogramm interessierten Benutzern zur Verfügung gestellt werden.

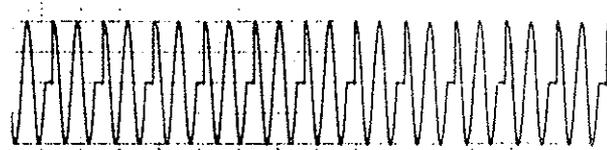
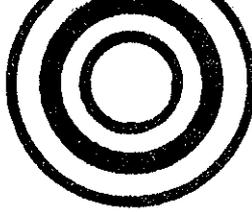
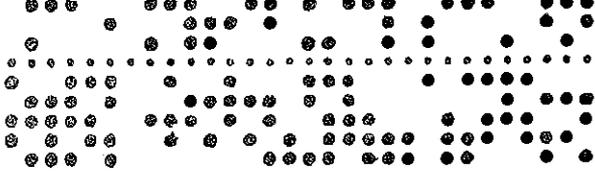
SUMMARY

This article describes a hybrid simulation of the gas flow through a long pipe. The basic equations are nonlinear partial differential equations for the pressure and the volumetric flux rate of the gas which are solved by the method of discrete-time-continuous-space. There is a new analog algorithm implemented for fast solution of the resulting two point boundary value problem. The digital part of the simulation program is written in FORTRAN with inline assembler coding in the time critical loops. The program can very easily be modified to solve other systems of nonlinear partial equations.

LITERATUR

- [1] Electronic Associates, Inc., Course Notes, "Hybrid Computation", Princeton, N.J. (1964)
- [2] G.A. Bekey, W.J. Karplus: Hybrid Computation. J. Wiley & Sons, Inc. New York (1968)

W. Kleinert



benutzerforum

EXPERIMENTELLE ÜBERPRÜFUNG EINES PARAMETERSCHÄTZVERFAHRENS FÜR ZEITVARIANTE SYSTEME

P. Kopacek, H. Traxler
 Institut für Wasserkraftmaschinen und Pumpen
 Abteilung Regelungstechnik der TU Wien

1. EINLEITUNG

Ziel dieser Arbeit ist die Überprüfung eines Parameterschätzverfahrens für lineare Regelsysteme, deren Parameter sich in Form von stationären stochastischen Prozessen ändern (zeitvariante stochastische Systeme) [1]. Diese Systeme sind für die Praxis von großer Bedeutung, da die Parameter vieler industrieller Prozesse zufällig variieren und der Einsatz fortgeschrittener Regelungsverfahren eine möglichst genaue Kenntnis der Parametermomentanwerte erfordert.

2. PARAMETERSCHÄTZVERFAHREN

Das untersuchte Parameterschätzverfahren arbeitet im wesentlichen rekursiv und kann als modifiziertes Kalman-Filter bezeichnet werden. Zunächst werden die n zeitveränderlichen Parameter des Systems zu einem Parametervektor

$$\underline{\theta}(t) = [\delta_1(t), \dots, \delta_n(t)]^T \quad (1)$$

zusammengefaßt und angenommen, daß ihre Zeitvarianz durch das Parametermodell

$$\underline{\theta}(t+1) = \underline{\phi}(t)\underline{\theta}(t) + \underline{w}_1(t) \quad (2)$$

mit $\underline{\phi}(t)$ als Matrix, deren Elemente determinierte Zeitfunktionen und $\underline{w}_1(t)$ als Vektor, dessen Komponenten stochastische Prozesse sind, hinreichend genau beschrieben werden kann.

Das Systemmodell hat die Form

$$\underline{y}_B(t) = \underline{H}(t)\underline{\theta}(t) + \underline{w}_2(t) \quad (3)$$

mit $\underline{y}_B(t)$ als Vektor der Systemausgangsgrößen. $\underline{H}(t)$ enthält im wesentlichen die Systemeingangsgrößen oder deren Ableitungen. Die Komponenten der Vektoren $\underline{w}_1(t)$ und $\underline{w}_2(t)$ werden als weiße Gaußprozesse vorausgesetzt

$$\begin{aligned} E\{\underline{w}_1(t)\underline{w}_1^T(s)\} &= \underline{R}_1(t)\delta(t-s) \\ E\{\underline{w}_2(t)\underline{w}_2^T(s)\} &= \underline{R}_2(t)\delta(t-s) \end{aligned} \quad (4)$$

$\underline{R}_1(t)$ und $\underline{R}_2(t)$ sind die entsprechenden Varianzmatrizen und $\delta(t-s)$ die Deltafunktion. Weiters wird angenommen, daß der deterministische Anteil der Parameter konstant ist, $\underline{\phi} = \underline{I}$, und die Parameter unabhängig voneinander sind, $\underline{R}_1 = \text{diag}\{\mu_1^2\}$. Für den Fall, daß $\underline{R}_1(t)$ und $\underline{R}_2(t)$ zeitinvariant sind, kann rekursiv mit Hilfe der Gleichungen

$$\underline{R}_{yy}(t/t-1) = \underline{H}(t) \underline{R}_{\theta\theta}(t/t-1) \underline{H}^T(t) + \underline{R}_2$$

$$\underline{K}(t) = \underline{R}_{\theta\theta}(t/t-1) \underline{H}^T(t) \underline{R}_{yy}^{-1}(t/t-1)$$

$$\underline{\tilde{y}}(t/t-1) = \underline{y}_B(t) - \underline{H}(t)\hat{\underline{\theta}}(t/t-1) \quad (5)$$

$$\hat{\underline{\theta}}(t+1/t) = \hat{\underline{\theta}}(t/t-1) + \underline{K}(t)\underline{\tilde{y}}(t/t-1)$$

$$\underline{R}_{\theta\theta}(t+1/t) = \underline{R}_1 + \underline{R}_{\theta\theta}(t/t-1) - \underline{K}(t)\underline{R}_{yy}(t/t-1)\underline{K}^T(t)$$

ein Schätzwert $\hat{\underline{\theta}}$ für den Parametervektor $\underline{\theta}$ ermittelt werden. $\underline{R}_{yy}(t/t-1)$ und $\underline{R}_{\theta\theta}(t/t-1)$ sind die Kovarianzmatrizen von Ausgangssignalen und Parametern zum Zeitpunkt t aus den Messungen zu den Zeitpunkten $1, \dots, t$. $\underline{\tilde{y}}(t/t-1)$ ist der Fehler zwischen den tatsächlichen Ausgangssignalen $\underline{y}_B(t)$ und den vorausberechneten $\underline{H}(t)\hat{\underline{\theta}}(t)K(t)$ ist eine Matrix von Gewichtungsfaktoren. Mit den Werten u_t und y_t von Ein- und Ausgangssignalen zu diskreten Zeitpunkten $1, \dots, t$ gilt für die Prädiktion

$$\hat{\underline{\theta}}(t+1/t) = E\{\underline{\theta}(t+1)/y_t, u_t\} \quad (6)$$

$$\underline{R}_{\theta\theta}(t+1/t) = \text{Var}\{\underline{\theta}(t+1)/y_t, u_t\}$$

Zur Lösung der Gleichungen (5) und (6) müssen \underline{R}_1 und \underline{R}_2 sowie die Anfangswerte $\underline{\theta}(1/0)$ und $\underline{R}_{\theta\theta}(1/0)$ geeignet gewählt werden [1], [2]. Für \underline{R}_1 und \underline{R}_2 kann ein Optimierungsalgorithmus, basierend auf der Methode der Maximum-Likelihood Funktion Verwendung finden.

3. HYBRIDRECHNERSIMULATION

Stellvertretend für zeitvariante Systeme wurden am Analogteil des Hybridrechners ein Verzögerungsglied erster Ordnung mit stochastisch variierender Verstärkung

$$T_1 \dot{y}(t) + y(t) = K(t)u(t)$$

$$K(t) = K_0 + K^*(t)$$

und ein Verzögerungsglied zweiter Ordnung mit stochastisch variierender Dämpfung simuliert.

$$T_2^2 \ddot{y}(t) + 2D(t)T_2 \dot{y}(t) + y(t) = Ku(t)$$

$$D(t) = D_0 + D^*(t)$$

$K^*(t)$ und $D^*(t)$ sind die stochastisch variierenden Anteile des Parameters. Für das Verzögerungsglied zweiter Ordnung wurde am Analogrechner die in Abb. 1 dargestellte Schaltung aufgebaut. Der stochastisch variierende Anteil der Dämpfung $D^*(t)$ wurde mit dem Rauschgenerator erzeugt, der an Potentiometer P 73 angeschlossen wurde. Die beiden Varianten a) und b) dienen dazu, um bei verschiedenen Mittelwerten der Dämpfung D_0 eine möglichst gute Aussteuerung des Multiplizierers zu erreichen.

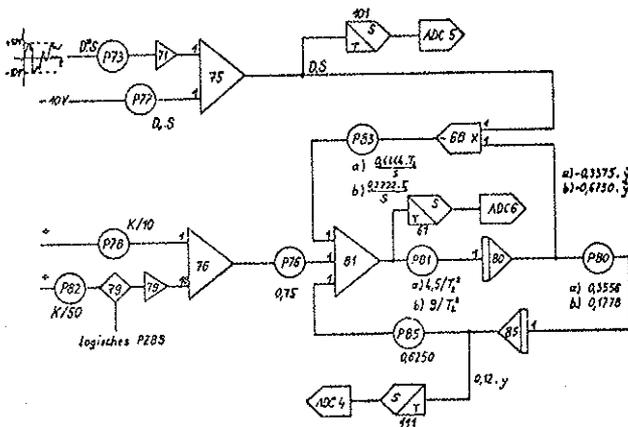


Abbildung 1

Auf den Digitalrechner wurden die Werte des Ausgangssignals, seiner 2. Ableitung und des Parameters übertragen. Dazu dienen die Analog-Digital-Wandler ADC 4, ADC 6 und ADC 5 mit den vorgeschalteten Punktspeichereinheiten. Als Eingangssignal des Systems fand ein pseudozufälliges Binärsignal mit der Länge von 31 Bit Verwendung. Die Taktzeit betrug 1 ms. Die Erzeugung des Signals erfolgte im Logikteil des Analogrechners mit Hilfe zweier in Serie geschalteter rückgekoppelter Schieberegister und wurde über einen O/A Schalter in den Analogteil übertragen. Die Logikschaltung zeigt Abb. 2. Die restliche Logik dient dazu, den Analogrechner bei einem gewünschten Analoglauf auf "Rechnen" zu schalten sowie die Punktspeichereinheiten und das Auslesen der Datenwerte zu steuern. Mit der monostabilen Kippstufe 2, der Senseline SL 0 und der Controlline CL 2 wurde erreicht, daß die Über-

tragung der Meßwerte erst nach dem Einschwingvorgang erfolgen kann. Der Beginn des Auslesens der Datenwerte erfolgt durch die logische Steuerung synchron zum Eingangssignal, sodaß dieses nicht gesondert auf den Digitalrechner übertragen werden muß. Sein Verlauf liegt damit eindeutig fest. Das Digitalprogramm ist so aufgebaut, daß dem Digitalrechner mit dem Analoglauf die notwendigen Datenwerte zur Verfügung gestellt wurden.

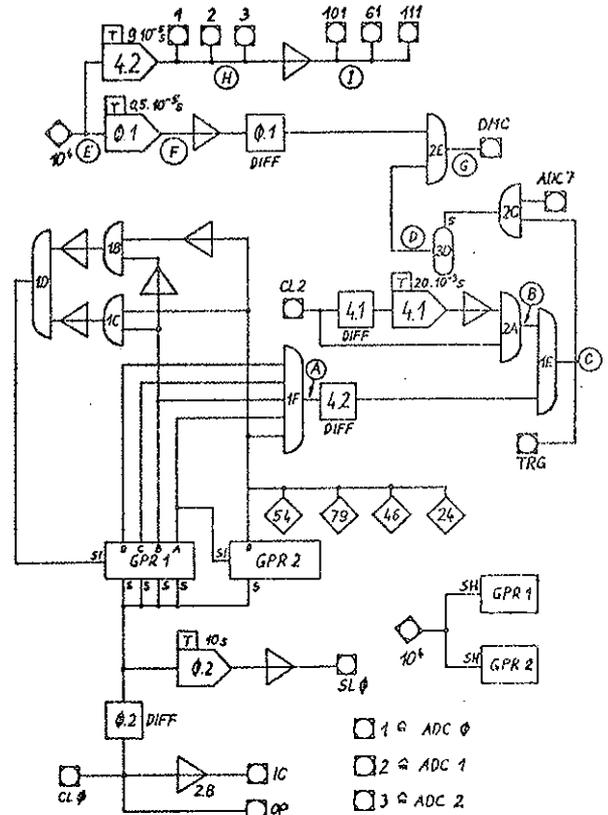


Abbildung 2

Anschließend kann gewählt werden, ob der Schätzalgorithmus mit einem Intervallabstand 0,1 ms oder 1 ms auszuführen ist. Die Anfangswerte sowie die für den Schätzalgorithmus notwendigen Werte sind einzugeben. Mit diesen Daten erfolgt die Berechnung der Schätzwerte, wobei die Werte für die Varianz des stochastischen Anteils des Parameters und des Fehlersignals des Ausgangssignals wahlweise optimiert werden können. Um einen besseren Überblick über die Schätzergebnisse zu erhalten, werden anschließend der Verlauf der tatsächlichen Parameterwerte und der Schätzwerte über der Zeit abgebildet, sowie der quadratische Mittelwert der Parameterwerte berechnet. Damit ist ein Rechendurchlauf beendet; es besteht dann die Möglichkeit, mit den gleichen ausgelesenen Werten aber anderen Anfangswerten bzw. anderem Abtastintervall einen weiteren Rechendurchlauf durchzuführen oder einen neuen Analoglauf zu starten. Ein schematisches Flußdiagramm des Digitalprogramms zeigt Abb. 3.

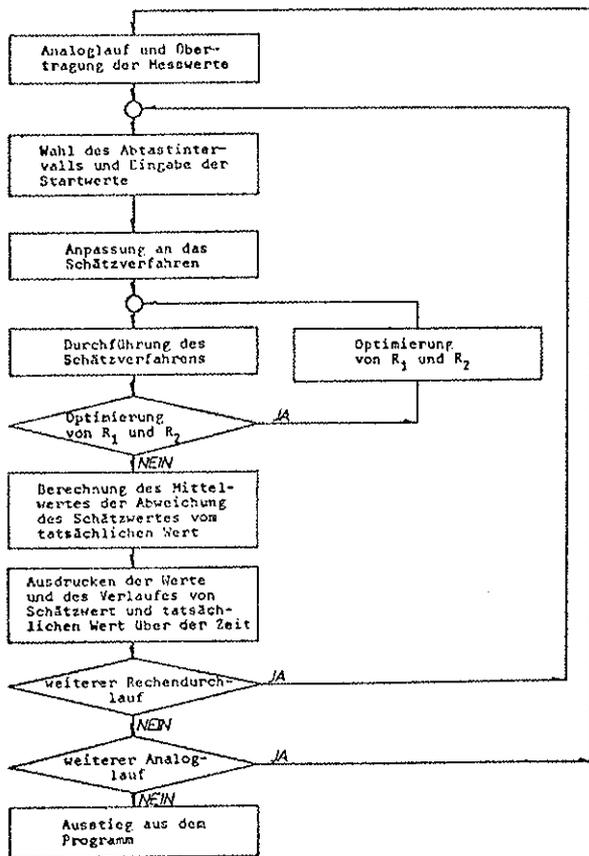


Abbildung 3

4. VERSUCHSERGEBNISSE

Es zeigt sich, daß die Schätzergebnisse hauptsächlich von der passenden Wahl der für den Schätzalgorithmus notwendigen Werte R_1 , R_2 , R_{00} ($1/0$) abhängen. Maßgebend ist dabei das Verhältnis der Varianz des stochastischen Anteils des Parameters R_1 zur Varianz des Fehlers des Ausgangssignals R_2 . Die absoluten Werte von R_1 und R_2 sind dabei gleichgültig. Je größer die Streuung der von den Parameterschwankungen hervorgerufenen Ausgangssignalschwankungen gegenüber denen des Fehlersignals ist, umso größer muß das Verhältnis R_1/R_2 gewählt werden. Außerdem muß es mit steigender Frequenz der Parameterschwankung erhöht werden. Bei ungünstiger Annahme des Verhältnisses R_1/R_2 werden die Abweichungen zwischen den Schätzwerten und den tatsächlichen Parameterwerten sehr groß. Man braucht also gewisse a-priori-Kenntnisse über die Varianz des Fehlers des Ausgangssignals und des Parameters, um das Verfahren zielführend anwenden zu können. Es besteht aber die Möglichkeit, die Werte von R_1 und R_2 zu optimieren. Gute Ergebnisse zeigen sich aber nur dann, wenn die Grenzen, zwischen denen R_1 optimiert werden soll, näherungsweise bekannt sind.

Eine günstige Wahl der Werte für R_1 und R_2 führt zu Ergebnissen, wie sie in Abb. 4 für die Verstärkung $K(t)$ des Verzögerungsgliedes erster Ordnung und in Abb. 5 für die Dämpfung $D(t)$ des Verzögerungsgliedes zweiter Ordnung dargestellt sind. Die tatsächlichen Parameterwerte sind durch Kreise und die Schätzwerte durch Sterne markiert.

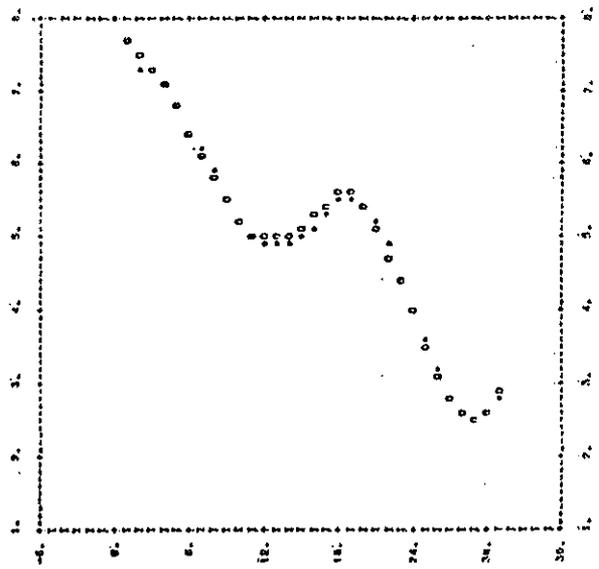


Abbildung 4

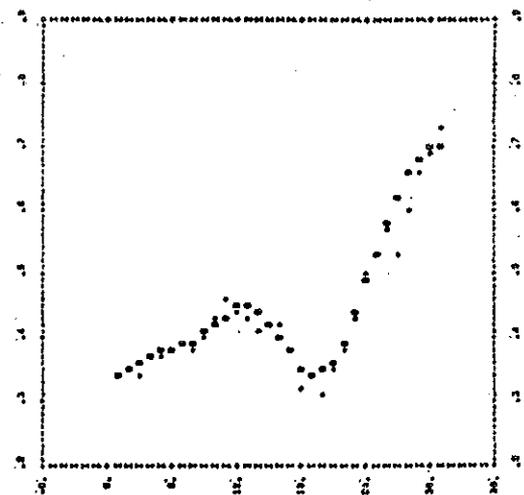


Abbildung 5

Summary

Systems with stochastically varying parameters (time varying stochastic systems) are of great importance for practical applications, because the parameters of many industrial processes are varying with time in a stochastic manner. For the applications of advanced control algorithms the actual values of the time varying parameters must be determined as accurately as possible. Therefore the paper treats of tests with a well known parameter estimation method for such control systems. The method leads to satisfying results if the starting values will be suitably chosen.

LITERATUR

- [1] BOHLIN, T.: Four cases of identification of changing systems. In: Mehra, R.; Lainiotis, O.: System Identification, Advances and Case Studies. Academic Press, New York (1976), pp. 442-515
- [2] TRAXLER, H.: Experimentelle Überprüfung eines Parameterschätzverfahrens. Diplomarbeit, TU Wien (1978)

KLASSIFIKATION VON AKUSTISCHEN SIGNALEN DURCH PHASENBILDVERFAHREN

R. Gudonawitschius,
Politechnische Hochschule in Kaunas, UdSSR

I. Husinsky
Hybridrechenanlage,
Technische Universität Wien

Einführung

Von großem Interesse für viele Anwendungen bei der Verarbeitung von zufälligen akustischen Signalen, so z.B. für die automatische Sprach- oder Geräuscherkennung, ist die Frage, welche Merkmale für eine Klassifikation wesentlich sind. In der Literatur sind zahlreiche Untersuchungen über die Bestimmung und Anwendung von Merkmalen beschrieben, in letzter Zeit besonders über lineare Prädiktorkoeffizienten [1]. Dieses Verfahren benötigt jedoch viel Rechenaufwand, besonders, wenn die Ordnung des Modells groß ist. Daher befaßt sich diese Arbeit mit der Beschreibung von zufälligen akustischen Signalen (Sprach- und Geräuschsignale) durch sogenannte "Phasenbild-Verfahren" [2] unter Ausnützung der Vorteile der linearen Prädiktoren [3].

Die Bestimmung der Kreuzpunktdichten im Phasenbild $\{x(t)^{(k)}, x(t)^{(l)}\}$

Voraussetzungen für die theoretischen Betrachtungen sind, daß das zu analysierende akustische Signal $x(t)$ ein normalverteilter stationärer Prozeß ist und eine Beschreibung dieses Prozesses durch ein Phasenbild der Ordnung $\{x(t)^{(k)}, x(t)^{(l)}\}, 1 \leq k, l$, möglich ist. $x(t)^{(k)}$ und $x(t)^{(l)}$ bezeichnen die k -te bzw. l -te Ableitung von $x(t)$. Jetzt wird die Zahl der Kreuzungspunkte n_i des Phasenbildes mit i beliebigen Strahlen während der Zeit (t_0, t_0+T) bestimmt.

$$n_i = \int_{t_0}^{t_0+T} |\dot{z}_i| \delta(z_i) h(y_i) dt \quad (1)$$

wobei

$$z_i = a_1 x(t)^{(k)} + b_1 x(t)^{(l)}$$

$$y_i = c_1 x(t)^{(k)} + d_1 x(t)^{(l)}$$

$$\dot{z}_i = \frac{dz_i}{dt}$$

δ Deltafunktion

h Funktion mit der Amplitude 1

a_1, b_1, c_1, d_1 sind die Koeffizienten, die die Lage des i -ten Strahles im Phasenbild $\{x(t)^{(k)}, x(t)^{(l)}\}$ definieren (Abb. 1).

Die mittlere Zahl \bar{n}_i der Kreuzungspunkte wird als Mittelwert der Werte n_i bestimmt

$$\bar{n}_i = \int_{t_0}^{t_0+T} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |\dot{z}_i| \delta(z_i) h(y_i) W_3(z_1, \dot{z}_1, y_1) dz_1 d\dot{z}_1 dy_1 dt \quad (2)$$

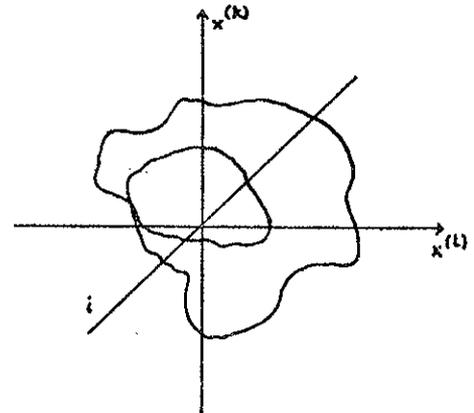


Abbildung 1

Nach Berücksichtigung der Eigenschaften der Funktionen $\delta(z_i)$ und $h(y_i)$ erhält man für stationäre Prozesse

$$\bar{n}_i = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [W_3(0, \dot{z}_1, y_1) + W_3(0, -\dot{z}_1, y_1)] d\dot{z}_1 dy_1 \quad (3)$$

wobei W_3 die dreidimensionale Verteilungsdichte ist, die auch ein normalverteilter Prozeß ist. Die dieser Verteilungsdichte entsprechende Matrix hat folgende Form:

$$\begin{pmatrix} R_{z_1 z_1} & R_{z_1 \dot{z}_1} & R_{z_1 y_1} \\ R_{\dot{z}_1 z_1} & R_{\dot{z}_1 \dot{z}_1} & R_{\dot{z}_1 y_1} \\ R_{y_1 z_1} & R_{y_1 \dot{z}_1} & R_{y_1 y_1} \end{pmatrix} \quad (4)$$

wobei die Elemente der Matrix (4) die Werte der Autokorrelations- und Kreuzkorrelationsfunktion bei Null sind.

Unter Anwendung der Gleichung (1) kann man diese Koeffizienten nach folgender Formel berechnen:

$$\begin{aligned} R_{z_1 z_1} &= (-1)^k a_1^2 R^{(2k)} + (-1)^l b_1^2 R^{(2l)} + [(-1)^k + (-1)^l] a_1 b_1 R^{(k+l)} \\ &= a_1^2 M^{(2k)} + b_1^2 M^{(2l)} + [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] a_1 b_1 M^{(k+1)} \\ R_{z_1 \dot{z}_1} &= R_{\dot{z}_1 z_1} = 0 \\ R_{z_1 y_1} &= a_1 c_1 M^{(2k)} + b_1 d_1 M^{(2l)} + [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] \frac{1}{2} (a_1 d_1 + b_1 c_1) M^{(k+1)} \\ R_{\dot{z}_1 \dot{z}_1} &= a_1^2 M^{(2k+2)} + b_1^2 M^{(2l+2)} + [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] a_1 b_1 M^{(k+1+2)} \\ R_{\dot{z}_1 y_1} &= R_{y_1 \dot{z}_1} = [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] \frac{1}{2} (a_1 d_1 - b_1 c_1) M^{(k+1+1)} \\ R_{y_1 y_1} &= c_1^2 M^{(2k)} + d_1^2 M^{(2l)} + [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] c_1 d_1 M^{(k+1)} \\ R_{y_1 z_1} &= R_{z_1 y_1} \end{aligned} \quad (5)$$

$R^{(k)}, R^{(1)}$ k-te bzw. 1-te Ableitung der Autokorrelationsfunktion von $x(t)$ bei Null

$M_{(k)}$ $\frac{1}{2\pi} \int_0^{\infty} \omega^k s(\omega) d\omega$ Spektralmomente der Ordnung k des Prozesses $x(t)$, $s(\omega)$ ist das Leistungsspektrum von $x(t)$

Auf der Basis der Gleichung (6) erhält man die nächste Formel für die Berechnung der Kreuzpunkte des Phasenbildes $\{\dot{x}(t), x(t)\}$ des stationären Prozesses mit den entsprechenden Strahlen:

$$\begin{aligned} n_1 &= n_5 = \frac{1}{2\pi} \sqrt{\frac{M_{(4)}}{M_{(2)}}} \\ n_3 &= n_7 = \frac{1}{2\pi} \sqrt{\frac{M_{(2)}}{M_{(0)}}} \\ n_2 &= n_4 = n_6 = n_8 = \frac{1}{2\pi} \sqrt{\frac{M_{(4)} + m^2 M_{(2)}}{M_{(2)} + m^2 M_{(0)}}} \end{aligned} \quad (7)$$

Nach der Bestimmung der dreidimensionalen Verteilungsdichte W_3 erhält man schließlich die Formel für die Berechnung der Zahl der Kreuzpunkte im Phasenbild $\{\dot{x}(t)^{(k)}, x(t)^{(l)}\}$ für einen beliebigen Strahl i (ausführliche Betrachtungen befinden sich in der Arbeit [4]):

$$n_i = \frac{1}{2\pi} \left(\frac{a_i^2 M_{(2k+2)} + b_i^2 M_{(2l+2)} + [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] a_i b_i M_{(k+l+2)}}{a_i^2 M_{(2k)} + b_i^2 M_{(2l)} + [(-1)^{\frac{k-1}{2}} + (-1)^{\frac{l-1}{2}}] a_i b_i M_{(k+l)}} \right) \quad (6)$$

Schaltungstechnische Bestimmung der Kreuzpunktdichten

In der Praxis sind akustische Signale nicht stationär, und darum kann man die mittleren Werte der entsprechenden Kreuzpunkte nur schätzungsweise z.B. durch Analogschaltungen (Abb. 3) bestimmen. Für das Phasenbild $\{\dot{x}(t), x(t)\}$, $k=1, l=0$, erhält man mit Hilfe von analogen und logischen Schaltungen die mittlere Zahl \bar{n}_i der Kreuzpunkte mit vier Strahlen während der Zeit (t_0, t_0+T) , wobei T im allgemeinen das Analyseintervall ist (z.B. kann T auch die Periode der Grundfrequenz beim Sprachsignal sein [5]).

Die Bestimmung der Kreuzpunktdichten im Phasenbild $\{\dot{x}(t), x(t)\}$

Sei als Beispiel das Phasenbild $\{\dot{x}(t), x(t)\}$ $l=0, k=1$, mit 8 Strahlen betrachtet (Abb. 2), dann erhält man die folgenden Bedingungen:

Strahl i	z_i	y_i	a_i	b_i	c_i	d_i
1	\dot{x}	x	1	0	0	1
2	$\dot{x}-m\dot{x}$	x	1	-m	0	1
3	x	\dot{x}	0	1	1	0
4	$\dot{x}+m\dot{x}$	-x	1	m	0	-1
5	\dot{x}	-x	1	0	0	-1
6	$\dot{x}-m\dot{x}$	-x	1	-m	0	-1
7	x	$-\dot{x}$	0	1	-1	0
8	$\dot{x}+m\dot{x}$	x	1	m	0	1

(hier gilt $m=tg\psi$, wobei ψ der Winkel des entsprechenden Strahles mit der X-Achse ist).

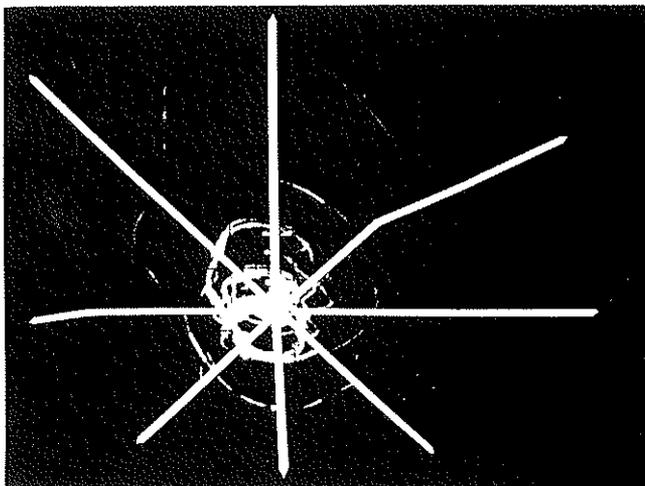
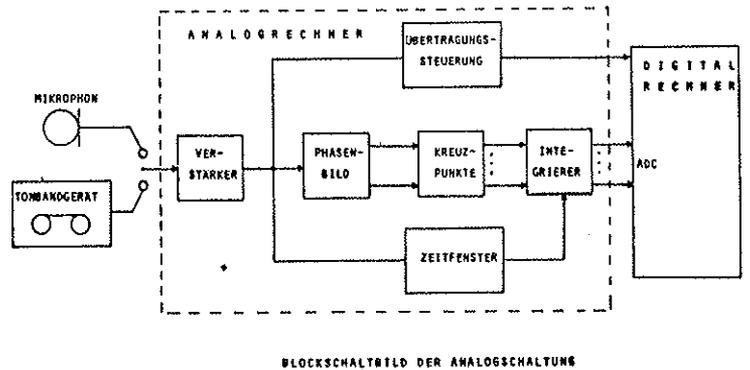


Abbildung 2



BLOCKSCHALTBIELD DER ANALOGSCHALTUNG

Abbildung 3

Als Testsignale wurden auf einem Tonband gespeicherte akustische Geräusche verwendet. Die am Analogrechner bestimmten Kreuzpunktdichten für die vier Strahlen wurden in Intervallen von 20 msec zum Digitalrechner übertragen und auf Platte abgespeichert. Es wurde ein Digitalprogramm zur Berechnung der Merkmalkoeffizienten (siehe auch [3]) und zur graphischen Darstellung der Ergebnisse erstellt.

In Abbildung 4 sind einige Beispiele dieser Kreuzpunktdichten über der Zeit, sowie die entsprechenden Autokorrelations-, Prädiktor- und Parcorckoeffizienten dargestellt.

In der nächsten Nummer von INTERFACE werden Vergleiche der Klassifikationsergebnisse bei akustischen Signalen durch die Anwendung einiger Entscheidungsregeln beschrieben werden.

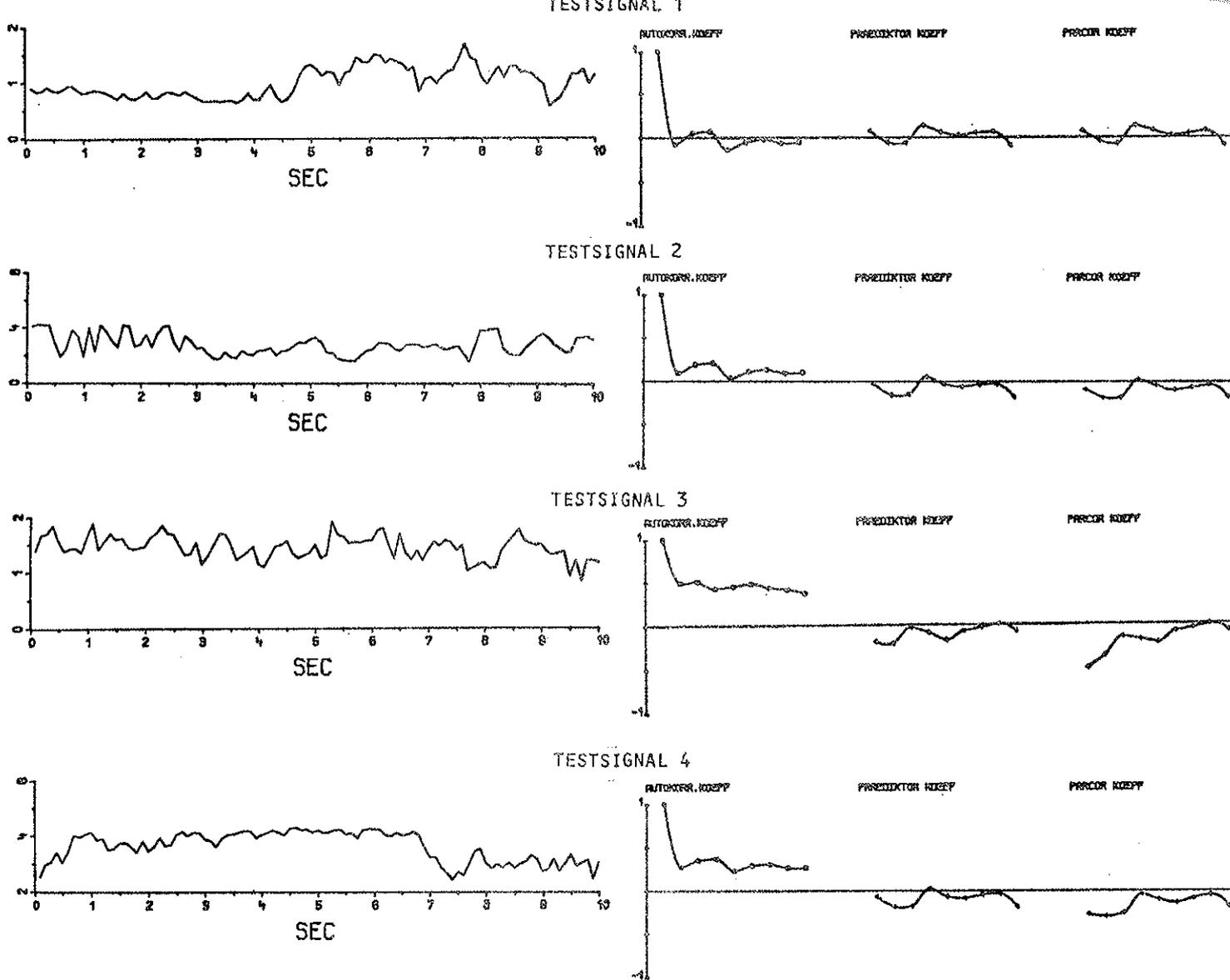


Abbildung 4

Summary

This paper deals with the description of random acoustic signals by phase-path-methods, using the advantages of linear prediction. The number of intersections of the phase path with the straight line starting from the origin of coordinates is derived. For the phase path $\{\hat{x}(t), x(t)\}$ the intersection number is computed on the analog computer and then is transferred to the digital computer where autocorrelation-, predictor-, and parcor-coefficients are calculated. Further classification of acoustic signals based on this method will be done.

Literatur

- [1] ATAL, B.S., HANAUER, S.L.: Speech Analysis and Synthesis by Linear Prediction of the Speech Wave. JASA, vol. 50 No. 2 (1971)
- [2] CHITAVICHIUS, A., GUDONAVICHIUS, R., ZAYEZONI, A.: Word recognition by using explicit structural relations. Technical Cybernetics, Proceedings of the Academy of Science of USSR, No 2 (1973) (russisch)
- [3] GUDONAWITSCHUIS, R., HUSINSKY, I.: Die Anwendung der Prädiktorkoeffizienten für die Spracherkennung. IX. Österreichischer Mathematikerkongress, Salzburg (1977)
- [4] GUDONAWITSCHUIS, R. et al.: Die Erkennung der akustischen Signale durch Anwendung des Phasenbild-Verfahrens. Bericht der VII. Akustischen Konferenz, Moskau (1977) (russisch)
- [5] GSANDTNER, A., GUDONAWITSCHUIS R.: Die Bestimmung der Merkmale für Vokalerkennung INTERFACE 3 (1975)

ANALOGUE SIMULATION VON STOFFWECHSELABLAUFEN

AN HAND VON STOFFWECHSELFLUXGRÖSSEN

Kubicek, C. P.
Institut für Biochemische Technologie und Mikrobiologie

Husinsky, I., Solar, D.,
Hybridrechenanlage des EDV-Zentrums
Technische Universität Wien

1. EINLEITUNG

Die Regulation zellulärer Stoffwechselfvorgänge bildet die Basis für die Koordination biologischer Prozesse. Biochemische Studien zur Stoffwechselregulation müssen eine Vielzahl von Faktoren berücksichtigen, wie Konzentration an Enzymen, Konzentration der Metaboliten und Regulationsverhalten der Enzyme. Bei genauer Kenntnis all dieser Parameter ist es möglich, den Stoffwechselfvorgang mathematisch zu simulieren. Derartige Systeme sind dann aber zumeist so aufwendig und komplex, daß nur kurze Stoffwechselfpassagen behandelt werden können oder entscheidende Vereinfachungen angenommen werden müssen (vgl. [1]).

Die Alternative dazu wäre die Beschreibung des Stoffwechsels auf Grund seiner 'in vivo' Flux-Größen. Diese sind -zumindest bei Untersuchungen an Mikroorganismen- zumeist durch Messung von Umweltparametern bzw. der Zellzusammensetzung zugänglich [2].

In der vorliegenden Arbeit sollte der umgekehrte Weg gegangen werden. In der industriellen Mikrobiologie spielen die Anhäufungen von Stoffwechselprodukten des Energiestoffwechsels, besonders des Tricarbonsäurecyclus, eine wichtige Rolle ([3]). Die Anhäufung solcher Intermediate kommt wahrscheinlich durch charakteristische Veränderungen der Flux-Werte verschiedener Enzyme zustande ([4], [5]). Es sollte daher festgestellt werden, welche Variation in den einzelnen Flux-Größen zur Anhäufung von Intermediaten führt.

2. MODELL

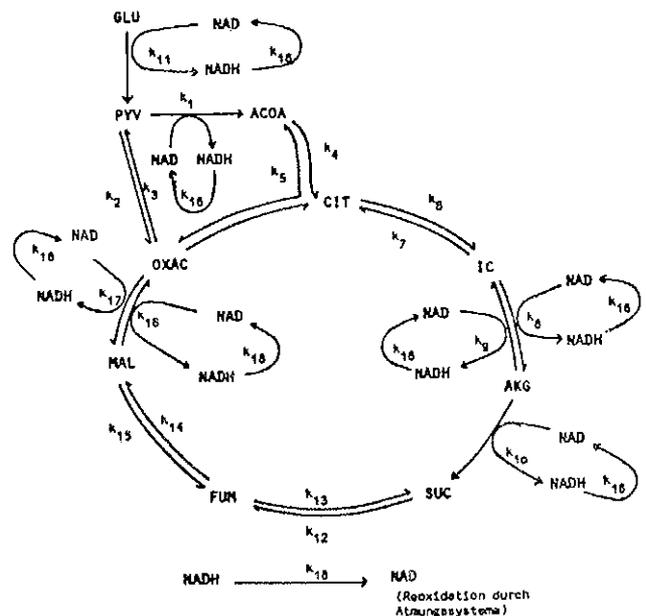
Terminologie und Symbole:

GLU	Glukose
PYV	Pyruvat
ACOA	Acetyl-CoA
DXAC	Dxalacetat
CIT	Citrat
IC	Isocitrat
AKG	α -Ketoglutarat
SUC	Succinat
FUM	Fumarat
MAL	Malat
NAO(H)	Nicotinamidadeninindinukleotid (reduziert)

Vereinfachende Annahmen:

Das System wurde isoliert zwischen Pyruvat und dem Tricarbonsäurecyclus betrachtet. Die aufgenommene Glucose wurde direkt in Pyruvat umgesetzt. Die enzymatischen Schritte wurden analog den bisher bei Pilzen nachgewiesenen Reaktionen angenommen. Regulation durch Intermediate wurde in den Flux-Größen festgelegt. Anabolische Reaktionen des Cyclus wurden nicht berücksichtigt. Die erhaltenen Werte für AKG und DXAC beinhalten daher auch Werte für primäre Aminosäuren.

Allgemeines Schema:



Differentialgleichungen:

Um dieses System am Analogrechner nachbilden zu können, muß es durch ein Differentialgleichungssystem beschrieben werden.

$$\frac{dGLU}{dt} = -k_{11} (GLU) (NAD)$$

$$\frac{dPYV}{dt} = -k_1 (PYV) (NAD) - k_2 (PYV) + k_3 (DXAC) + k_{11} (GLU) (NAD)$$

$$\frac{dACOA}{dt} = -k_4 (ACOA) (DXAC) + k_5 (CIT) + k_1 (PYV) (NAD)$$

$$\frac{dDXAC}{dt} = -k_3 (DXAC) + k_2 (PYV) + k_{16} (MAL) (NAD) - k_4 (DXAC) (ACOA) - k_{17} (DXAC) (NADH) + k_5 (CIT)$$

$$\begin{aligned} \frac{dCIT}{dt} &= -k_6(CIT) + k_7(IC) - k_5(CIT) + k_4(OXAC)(ACOA) \\ \frac{dIC}{dt} &= +k_6(CIT) - k_7(IC) - k_8(IC)(NAD) - \\ &+ k_9(AKG)(NADH) \\ \frac{dAKG}{dt} &= -k_{10}(AKG)(NAD) + k_8(IC)(NAD) - k_9(AKG)(NADH) \\ \frac{dSUC}{dt} &= +k_{10}(AKG)(NAD) + k_{13}(FUM) - k_{12}(SUC) \\ \frac{dFUM}{dt} &= +k_{12}(SUC) - k_{13}(FUM) - k_{14}(FUM) + k_{15}(MAL) \\ \frac{dMAL}{dt} &= -k_{16}(MAL)(NAD) + k_{17}(OXAC)(NADH) - k_{15}(MAL) \\ &+ k_{14}(FUM) \\ \frac{dNAD}{dt} &= +k_{18}(NADH) + k_{17}(OXAC)(NADH) - k_{16}(MAL)(NAD) \\ &- k_8(IC)(NAD) + k_9(AKG)(NADH) - k_{10}(AKG)(NAD) \\ &- k_1(PYV)(NAD) - k_{11}(GLU)(NAD) \\ \frac{dNADH}{dt} &= -k_{18}(NADH) - k_{17}(OXAC)(NAD) + k_{16}(MAL)(NAD) \\ &+ k_8(IC)(NAD) - k_9(AKG)(NADH) + k_{10}(AKG)(NAD) \\ &+ k_1(PYV)(NAD) + k_{11}(NAD)(GLU) \end{aligned}$$

Da die Lösungen des Gleichungssystems in der Größenordnung sehr unterschiedlich sind, ist eine gute Skalierung der Verstärkerausgänge unbedingt erforderlich, um eine Stabilität der analogen Schaltung zu gewährleisten. Ein sinnvolles Arbeiten mit dem Modell ist daher nur unter Verwendung eines automatischen Skalierungsprogrammes möglich, da fast bei jeder Parameteränderung neu skaliert werden muß. Der hybride Sprachprozessor HYBSYS ermöglicht über ein Terminal interaktiv die Kommunikation mit dem Analogrechner, führt die automatische Skalierung sowie statische Tests durch und ermöglicht eine graphische Darstellung der Lösung.

Rate Constants (=Flux-Größen) und Anfangswerte:

$k_3 = 0.$	$k_{14} = 15.$	GLU(0) = .3
$k_5 = 0.$	$k_{15} = 5.$	NAD(0) = .075
$k_6 = 2.$	$k_{16} = 0.1$	alle anderen
$k_7 = 10.$	$k_{17} = 50.$	Anfangswerte
$k_8 = 10.$	$k_{18} = 100.$	sind Null
$k_{11} = 3.$		
$k_{12} = 3.$		
$k_{13} = 3.$		

Die angegebenen Werte sind dimensionslos und nur in ihren Verhältnissen zueinander maßgebend. Die Werte k_1, k_2, k_4, k_8, k_9 und k_{10} wurden variiert und sind daher hier nicht angegeben.

Realisierung des Modells am Hybridrechner:

Die Arbeitsweise des Analogrechners ermöglicht ein direktes Arbeiten am Modell, sodaß Untersuchungen des Modellverhaltens und Änderungen von Modellparametern relativ einfach durchgeführt werden können, wobei der Untersucher in unmittelbarem Kontakt mit der Maschine ist.

Das Differentialgleichungssystem wurde am Analogrechner nachgebildet. Die Kontrolle und Auswertung des analogen Modells wurden von der an der Hybridrechenanlage vorhandenen digitalen Unterstützungssoftware für hybride Simulationen (HYBSYS) durchgeführt.

Konstanz des Stoffwechselfools:

Um den Fließgewichtszuständen der Zelle nahe zu kommen, wurde versucht, die Glukose-Konzentration konstant zu halten. Dies wurde durch eine konstante Zugaberate dgz erreicht. Dabei wurde die Funktion GLU durch GLUS ersetzt, das der folgenden Differentialgleichung genügt:

$$\frac{dGLUS}{dt} = dgz - k_{11}(GLUS)(NAD)$$

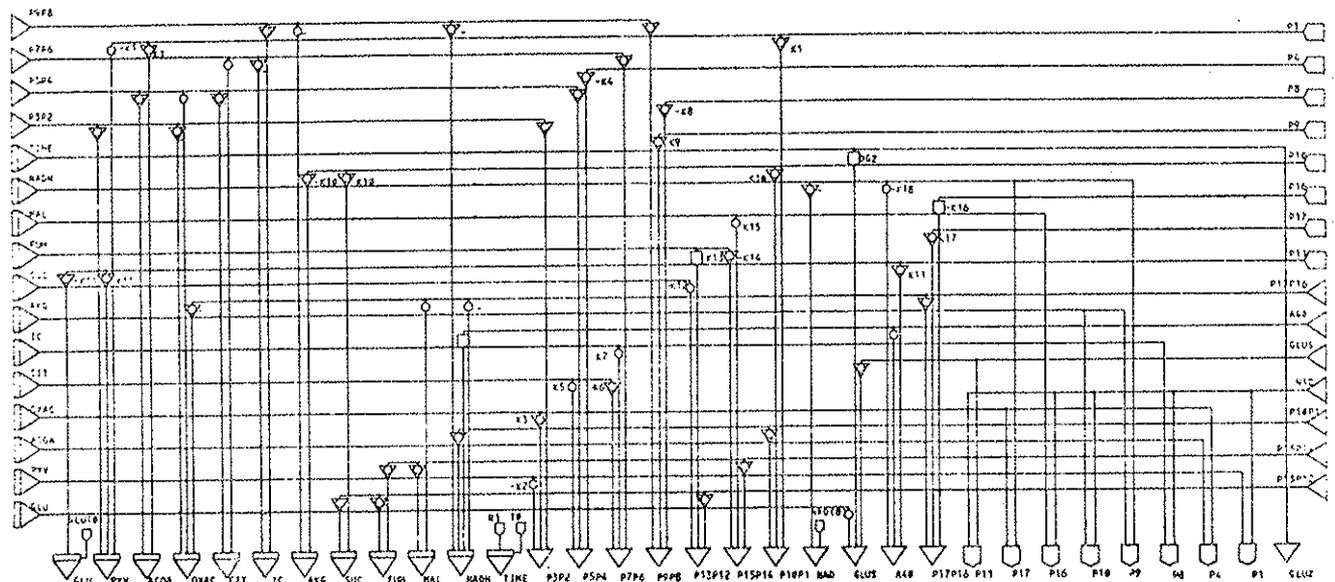
Die ursprüngliche Variable GLU tritt im weiteren als

$$\begin{aligned} GLU &:= GLUS - dgz.t \\ &= GLUS - GLUZ \end{aligned}$$

auf, wobei GLUZ die Glukosezugabe darstellt.

Analogschaltung:

(von HYBSYS automatisch am Data Plotter erstellt)



3. ERGEBNISSE

Unter Benützung des angegebenen Modells wurde nun versucht, festzustellen, welche Variation in den Flux-Größen verschiedener Enzyme zu einer Steigerung der Anhäufung von Intermediaten führt. Zwei grundlegende Annahmen wurden hierbei untersucht: der Einfluß der Pyruvat-Carboxylase (k_2) sowie der Einfluß einer möglichen Hemmung des Tricarbonsäurecyclus auf der Stufe der α -Ketoglutarat-Dehydrogenase (k_{10}). Annahmen von $k_2 = 0$ lieferten Akkumulation von Pyruvat, was etwa der Anhäufung von Äthanol durch Hefe ("Alkoholische Gärung") entsprechen würde, mit nur geringer Beiproduktion von Intermediaten des Citronensäurecyclus. Eine Hemmung des Cyclus auf der Stufe der α -Ketoglutarat-Dehydrogenase ($k_{10} = 0$) führt je nach Wahl der k -Werte zur Anhäufung verschiedener Intermediate. Dies stimmt mit Ergebnissen des Autors (C.P. Kubicek) überein, daß eine aktive Pyruvat-Carboxylase und eine Hemmung des Cyclus auf der Stufe der α -Ketoglutarat-Dehydrogenase charakteristische Bedingungen dieser Säureanhäufungen sind.

In den folgenden Abbildungen sind die Ergebnisse eines typischen Versuchs gezeigt: es konnten hierbei drei Typen von Produktbildungskurven gefunden werden:

- (a) Sättigungskurven,
- (b) hyperbolische Kurven und
- (c) solche, die ein frühes Maximum durchlaufen und später einen tieferen Wert anstreben.

Diese Effekte sind der Vereinfachung des Systems zuzuschreiben und besitzen keinerlei physiologische Signifikanz. Vom Standpunkt der Physiologie können daher nur Werte im Sättigungsbereich interpretiert werden, welche in ihren Verhältnissen zueinander das Verhältnis der "steady-state"-Konzentrationen dieser Intermediate widerspiegeln.

In den vorliegenden Abbildungen sind typische Kurvenverläufe wiedergegeben. Die Resultate beziehen sich nicht auf identische Versuchsbedingungen, sondern sollen nur die angeführten Typen besonders deutlich herausstellen. Konzentration und Zeit sind in willkürlichen Einheiten angegeben.

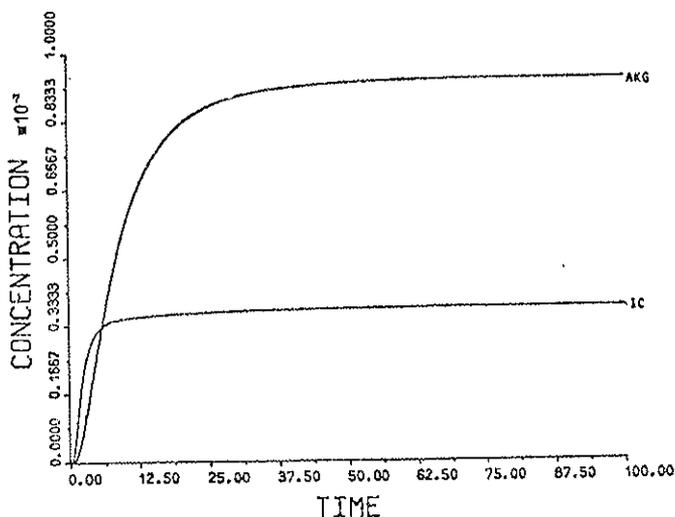


Abbildung 1

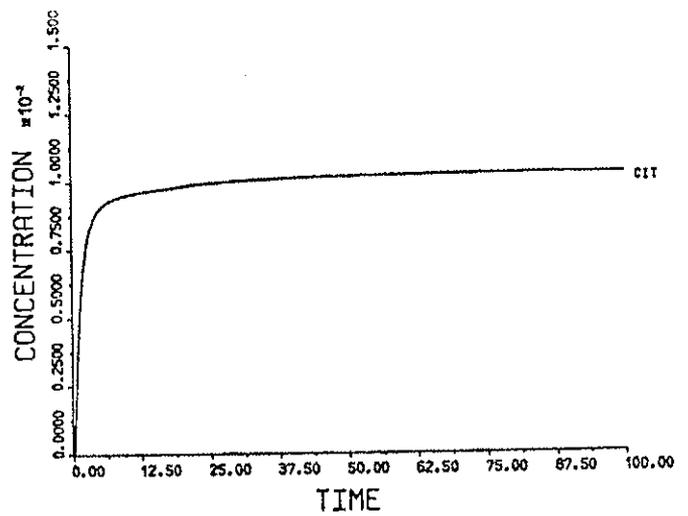


Abbildung 2

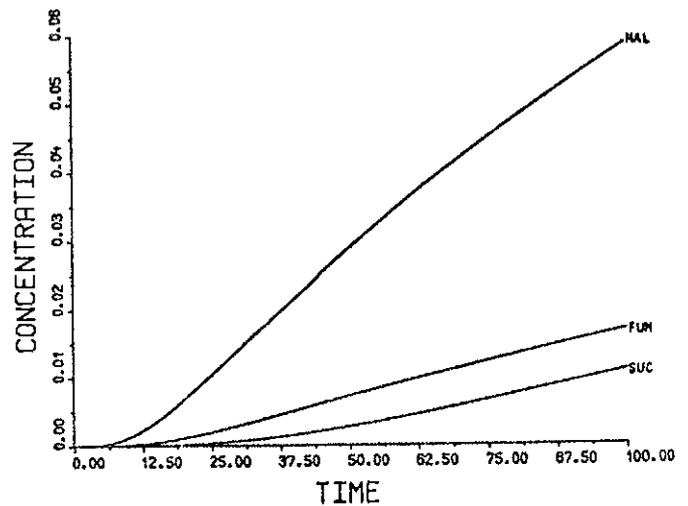


Abbildung 3

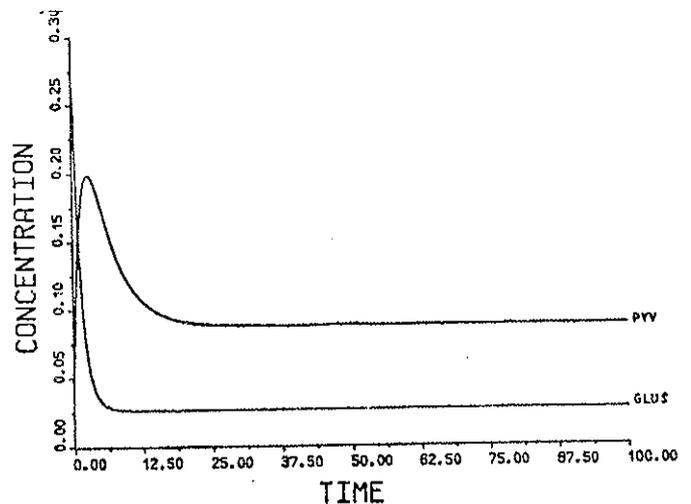
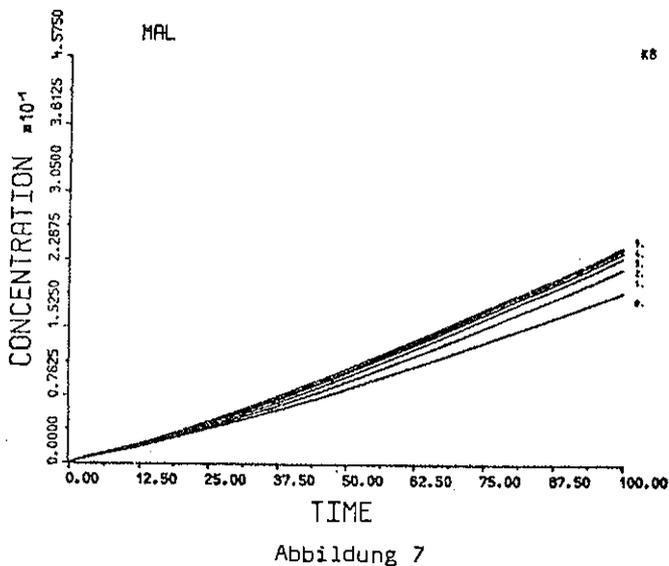
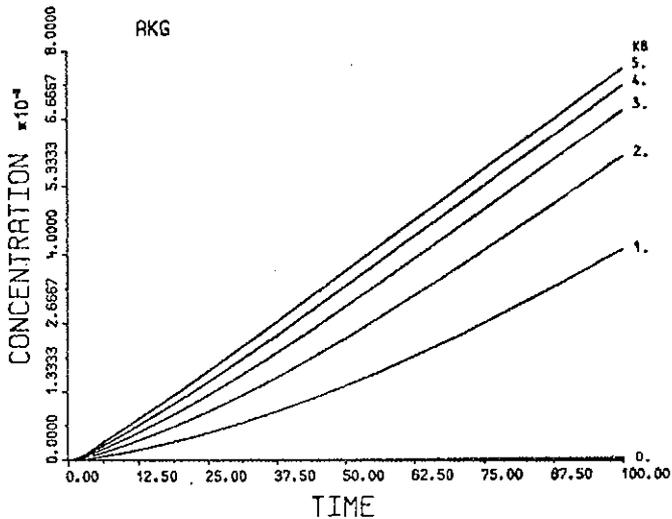
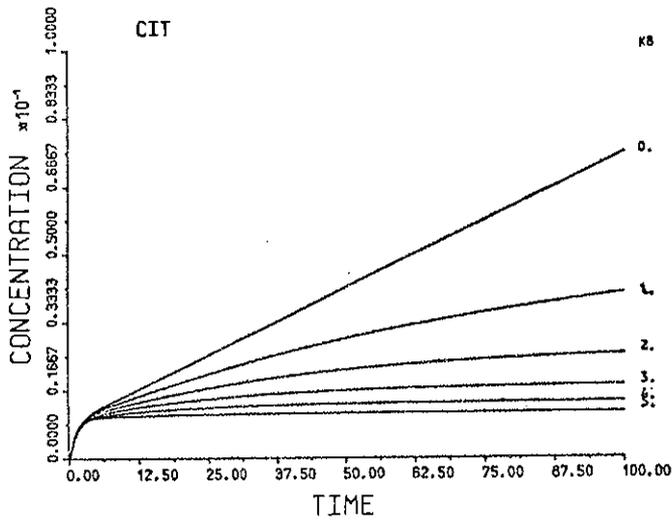


Abbildung 4

Von mehreren Autoren wurde postuliert, daß eine Hemmung der Isocitrat-Dehydrogenase zu einer Akkumulation von Citrat führt. Die Aktivität dieses Enzyms wurde durch Variation des Wertes von k_8 manipuliert. In den Abbildungen 5 - 7 wird gezeigt, wie die Konzentrationen an Citrat,

α -Ketoglutarat und Malat hierdurch beeinflusst werden. Tatsächlich kann das Verhältnis CIT/AKG durch eine Hemmung der Isocitrat-Dehydrogenase gesteigert werden, was z.B. den physiologischen Verhältnissen bei Anhäufung von AKG (Endprodukt) oder Mangel an AMP (hohe Energy Charge) entspricht.



($k_1=5.$, $k_2=10.$, $k_4=50.$, $k_9=0.1$, $k_{10}=2.$)

4. DISKUSSION

Wenngleich für die Erstellung dieses Modells einige einschneidende Vereinfachungen postuliert werden mußten, - so z.B. mußte eine ruhende Zelle ohne nennenswerte Beeinflussung durch den Stickstoffmetabolismus angenommen werden, - konnte mit dem Modell dennoch eine Annäherung an die biologischen Phänomene erreicht werden. In dieser Hinsicht wird derzeit versucht, die Feinregulation der beteiligten Enzymreaktionen durch ständige Variation der Flux-Werte zu simulieren.

SUMMARY

A model is proposed which enables the simulation of organic acid formation of fungi. It is based on a description of the enzymatic reactions of the tricarboxylic acid cycle by means of metabolic flux values. The model is described by a system of differential equations and is simulated on the analog computer with the support of the hybrid processor HYBSYS. The model is applicable for the study of the interrelationship between enzymatic activities and product accumulation in fungi.

LITERATUR

- [1] GARFINKEL, A.: Simulation of glycolytic systems. In: Concepts and Models of Biomathematics (HEINMETS, F., ed.) Marcel Dekker Inc. New York (1969)
- [2] HEINMETS, F.: Analysis of cellular growth process. In: Concepts and Models of Biomathematics (HEINMETS, F., ed.) Marcel Dekker Inc. New York (1969)
- [3] LOCKWOOD, L.B.: Organic acid production. In: The filamentous fungi, Vol I, p. 140 ff (J.E. SMITH, O.E. BERRY, eds.) Edward Arnold London (1974)
- [4] KUBICEK, C.P., RÖHR, M.: Influence of Manganese on Enzyme Synthesis and Citric Acid Accumulation in *Aspergillus niger*. European J. Appl. Microbiol. 4, 167-175 (1977)
- [5] KUBICEK, C.P., RÖHR, M.: The Role of the Tricarboxylic Acid Cycle in Citric Acid Accumulation by *Aspergillus niger*. European J. Appl. Microbiol. Biotechnol. 5, 263-271 (1978)

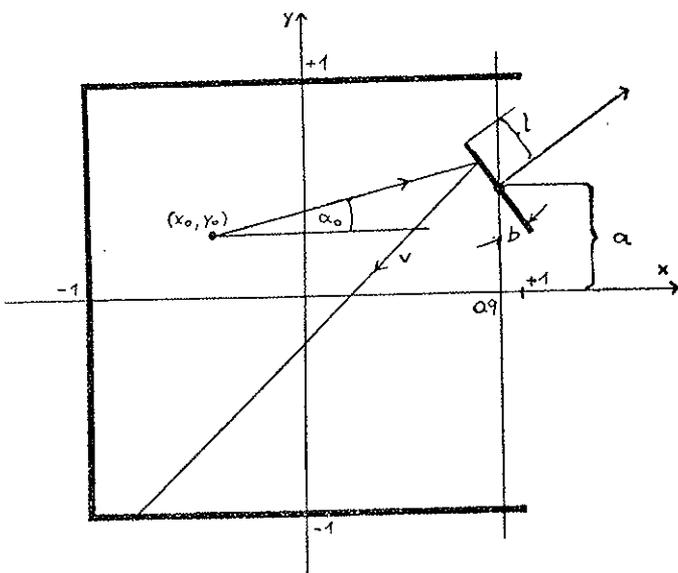
aus dem praktikum

SQUASH - EINMANN TENNIS AM ANALOGRECHNER

Beispiel aus dem Fortgeschrittenenpraktikum für Analogrechner.
 (Gruppe E. Königstein, M. Schiefert, W. Thaler)
 F. Breitenecker
 Institut für Technische Mathematik
 Technische Universität Wien

Im Fortgeschrittenenpraktikum für Analogrechner wurde im Wintersemester 1978/79 ein "TV-Spiel", nämlich Squash, am Analogrechner nachgebildet. Ein Ball wird an drei festen Wänden und an einem beweglichen Schläger reflektiert, wobei die Lage und der Winkel des Schlägers kontinuierlich verändert werden können.

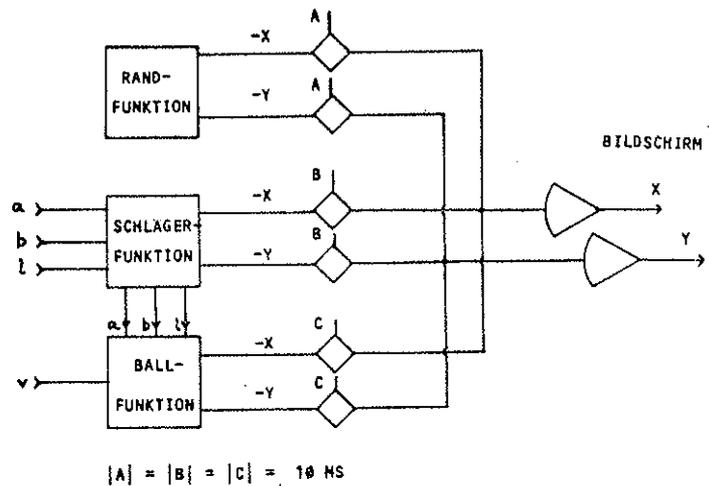
Folgendes Modell wurde verwendet:



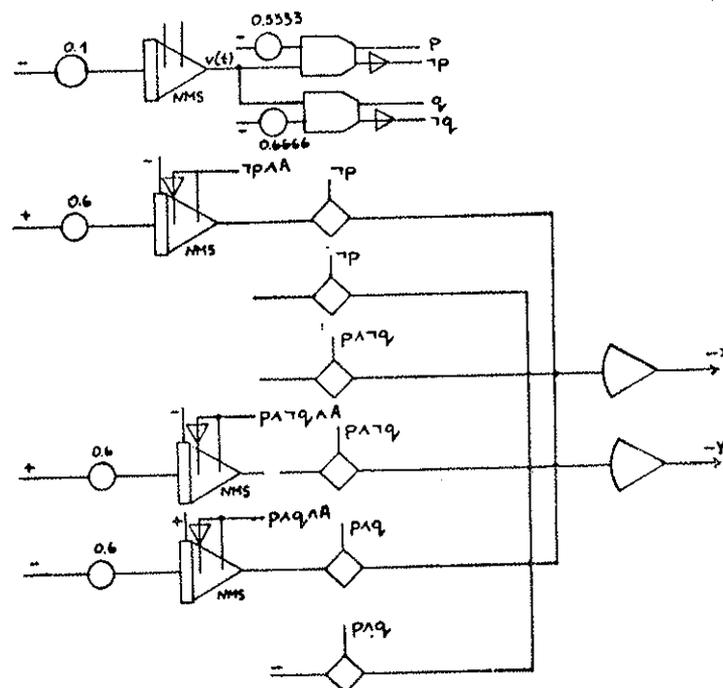
Die Größen bedeuten:

- (x_0, y_0) Startpunkt des Balles
- α_0 Startwinkel des Balles
- l Schlägerlänge
- b Schlägerwinkel
- v Ballgeschwindigkeit
- α Winkel des Balles
- a Schlägerhöhe

Die Darstellung am Oszillographen wurde in drei Teildarstellungen zerlegt, die hintereinander auf die x- und die y-Ablenkung des Oszillographen geschaltet und mit dem Zeitgeber synchronisiert wurden:



Während der Dauer von 10 ms wird der Rand, also die drei festen Wände, durch folgende Schaltung dargestellt:

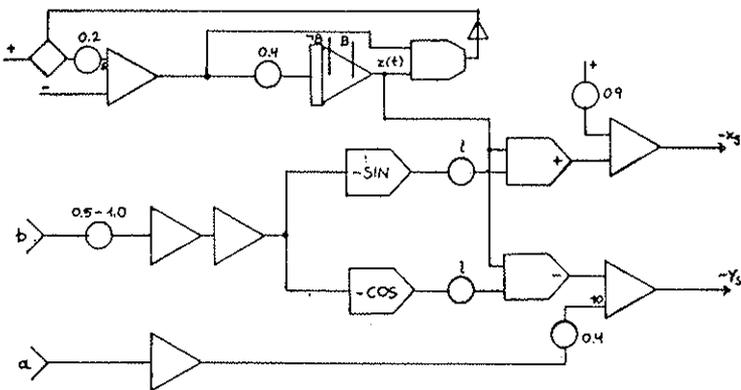


Für den Schläger wurde folgende Darstellung gewählt:

$$x_s = 1 \cdot \sin b \cdot z(t) + 0.9$$

$$y_s = 1 \cdot \cos b \cdot z(t) + a$$

$z(t)$ ist dabei eine Dreiecksschwingung



Der Ball ist durch Koordinaten in der Form

$$\dot{x} = v \cdot \cos \alpha$$

$$\dot{y} = v \cdot \sin \alpha$$

gegeben.

Trifft der Ball auf eine Wand oder auf den Schläger, so wird die Reflexion auf folgende Art berechnet:

- 1) Reflexion an senkrechter Wand:

$$\alpha_{NEU} = -\alpha \pm 180^\circ$$

- 2) Reflexion an waagrechter Wand:

$$\alpha_{NEU} = -\alpha$$

- 3) Reflexion am Schläger:

$$\alpha_{NEU} = -\alpha - 2b \pm 180^\circ$$

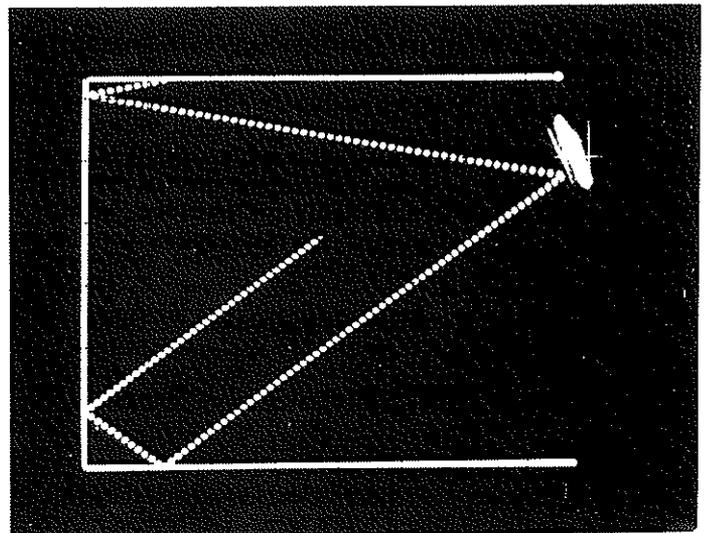
Ob $+180^\circ$ oder -180° zu wählen ist, wird durch $\alpha_{NEU} \in [-180^\circ, +180^\circ]$ festgelegt.

Die Schaltung arbeitet nach folgendem Grundprinzip:

Bewegt sich der Ball innerhalb des Spielfeldes, so werden alle möglichen Reflexionen berechnet (TS-Einheiten A,B,C); trifft der Ball den Rand oder den Schläger (Komparatoren A,B,C₁,C₂,C₃,C₄), so steht daher der gewünschte Reflexionswinkel α_{NEU} sofort zur Verfügung (Übernahme durch TS-Einheit D)

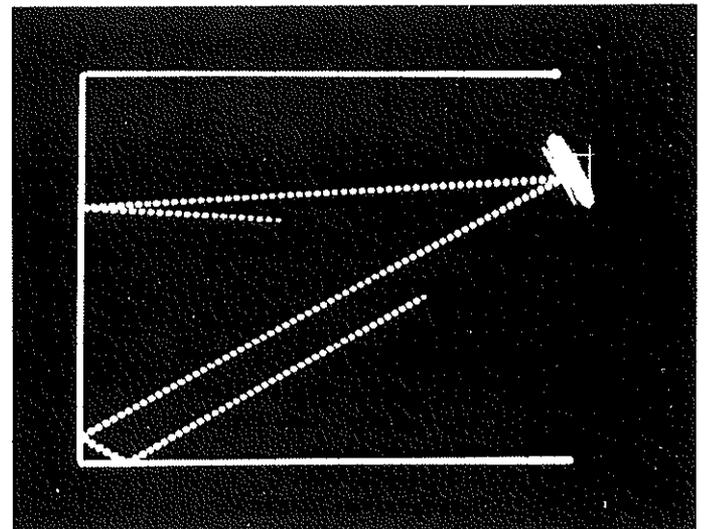
Die Größen x_0 , y_0 , α_0 , l und v können durch Handpotentiometer verändert werden.

Die Änderung von Schlägerhöhe und Schlägerwinkel erfolgt kontinuierlich durch den Steuerknüppel, wobei die Vertikalbewegung des Knüppels der Schlägerhöhe, die Horizontalbewegung dem Schlägerwinkel entspricht.



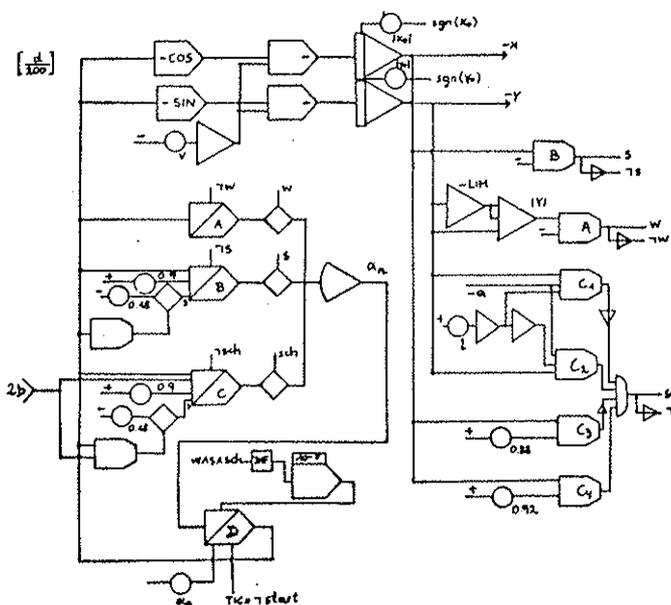
Oszillographenbild von Squash

$$\alpha_0 = 215^\circ$$



Oszillographenbild von Squash

$$\alpha_0 = 175^\circ$$



SCHMIEDEEISENKUNST MIT DEM HYBRIDRECHNER

Beispiel aus dem Praktikum für Hybridrechnen I
 (Gruppe Czedik-Eysenberg, Robl)
 F. Rattay
 Institut für Technische Mathematik
 Technische Universität Wien

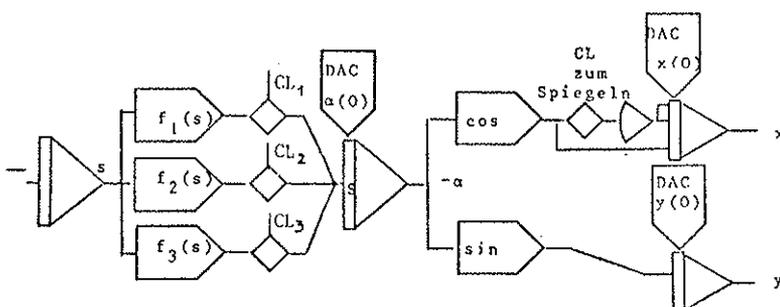
Aus Schmiedeeisen werden oft ebene Kunstwerke wie etwa Geländer, Raumteiler oder Konsolen durch Zusammensetzen einzelner Grundformen hergestellt. Jede einzelne Grundform kann durch Biegen von Flacheisen über eine speziell dafür angefertigte Biegeform erzeugt werden. Der Entwurf von Mustern mit neu anzufertigenden Grundformen ohne Verwendung von Zeichenschablonen ist aber ungenau oder sehr zeitaufwendig, sodaß die Kreativität darunter leidet.

Im Rahmen des Praktikums wurde ein Programm erstellt, das den Entwurf von Mustern mit leicht veränderbaren Grundformen in recht handlicher Form ermöglicht. Das einfach gehaltene Programm, das in zwei Doppelstunden entwickelt wurde, erlaubt die Darstellung ebener Gebilde am Oszillographen, die sich aus drei Grundformen zusammensetzen. Durch Eingabe an der Konsole des Tektronix Terminals werden die Anzahl der darzustellenden Grundformen (Teilkurven) und die Nummer der zu verändernden Grundform sowie ihr Typ festgelegt. Durch Verdrehen von drei Handpotentiometern werden Anfangspunkt und Anfangswinkel der ausgewählten Teilkurven verstellt. Die aktuellen Anfangswerte jeder darzustellenden Teilkurve, sowie die Angabe, ob die Kurve normal oder gespiegelt wiedergegeben werden soll, werden digital abgespeichert. Alle Teilbilder werden repetierend immer wieder dargestellt, wobei durch die große Arbeitsgeschwindigkeit von fast 1000 Bildern pro Sekunde (weitere Steigerung möglich) der Eindruck entsteht, daß man die zu verändernde Grundform praktisch kontinuierlich an eine beliebige Stelle des Bildschirms bewegen kann.

Die Gleichungen

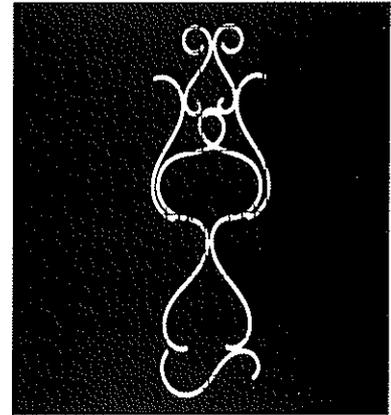
$$\frac{dx}{ds} = \cos \alpha, \quad \frac{dy}{ds} = \sin \alpha$$

beschreiben eine Kurve im R^2 , deren Krümmung durch die Ableitung $d\alpha/ds$ gegeben ist. Setzt man $s=t$, so läßt sich mittels Funktionsgebern, mit denen die Krümmung als Funktion der Bogenlänge gegeben wird, durch den abgebildeten Anlogschaltplan mit wenigen Elementen die Darstellung einer Grundform in beliebiger Lage erreichen.

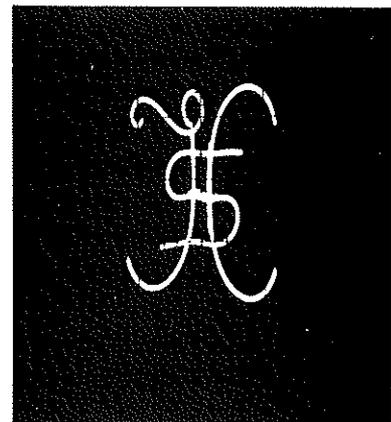


Analogschaltplan

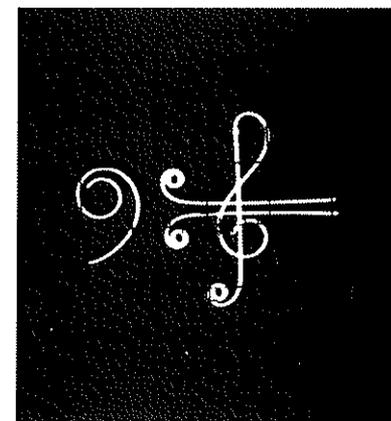
Die Bilder zeigen einige der während des Praktikums entworfenen Motive.



Fantasieform



Stilisiertes Kurzzeichen für das Betriebssystem JCS/TS 7



Musikzeichen

INTERFACE April 1979