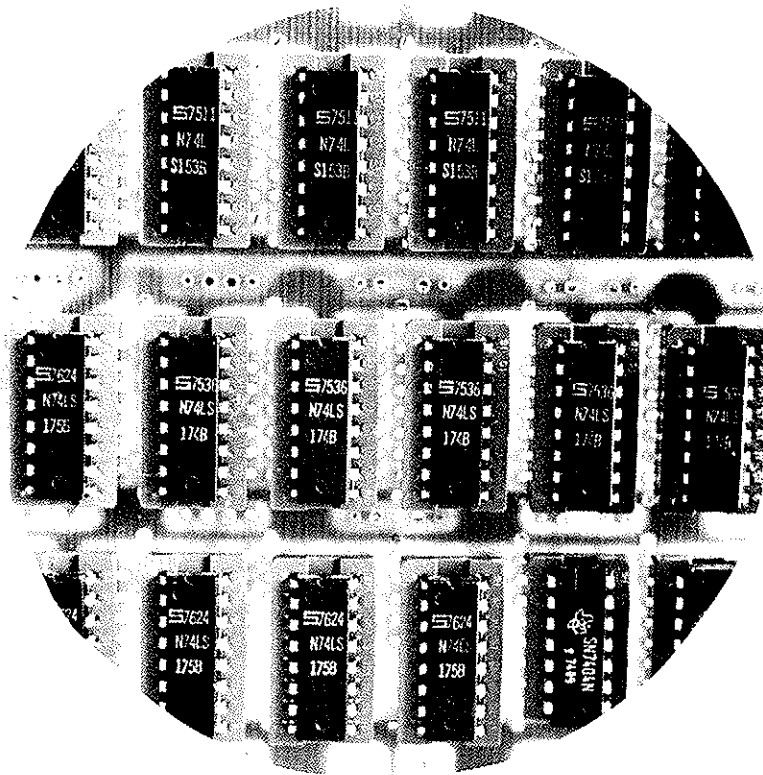

Interface

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

Nummer 9
Juni 1977



Teil der Hardware-Eigenentwicklungen:
Verdopplung des Registersatzes
des Floating Point Processors
für Multi-User-Betrieb

INHALTSVERZEICHNIS

	Seite
PACER 100 Autopatch-Erweiterung	3
Technischer Betrieb	6
Memory Erweiterung	7
Kurse	8
Benutzung des Programmpakets IPP1	10
District Transfer in HOI	11
Neuer Bootstrap	12
Hybride FORTRAN-Programmierung im Betriebssystem JCS/TS 7 mit der HYBRID LINKAGE Software	13
Spezielle Probleme der hybriden Programmierung im Multiprogramming des Betriebssystems JCS/TS 7	20
Hardware-Ecke	26
Die Optimierung der elastischen Lagerung eines Verbrennungsmotors mit Hilfe der Hybridrechen- anlage an der TU-Wien	30
Modell zur Simulation der Altersabhängigkeit der Dosis im Atemtrakt bei der Inhalation natürlich radioaktiver Nuklide	37
Ein Vergleich einiger Verfahren der Merkmal- bestimmung für Spracherkennung	47

Redaktion: Irmgard Husinsky

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

PACER 600 AUTOPATCH - ERWEITERUNG

Im folgenden drucken wir mit freundlicher Genehmigung der Firma EAI eine eigene Übersetzung eines Aufsatzes von J. Bruce Mawson ab. In weiteren Nummern der Zeitschrift INTERFACE werden wir genauer auf die Hard- und Software des Autopatch PACER eingehen, sowie unseren Lesern im Detail ein Bild von den Ausbauplänen der Hybridrechenanlage mit einem Autopatch System mit hybrider Time-Sharing Möglichkeit geben.

Seit kurzer Zeit ist eine Autopatch-Erweiterung zum PACER 600 System erhältlich. Sie ermöglicht die Lösung von Simulationsaufgaben am PACER 600 mithilfe hochentwickelter Benutzer-Software, ohne jeden unnötigen manuellen Eingriff am Analogrechner. Es gibt kein Stecken am Steckbrett mehr. Die Autopatch-Erweiterung ergänzt die große Rechengeschwindigkeit des Analogrechners, indem sie einen raschen Problemdurchsatz, automatische Programmvorbereitung und Fehlerbeseitigung, sowie vor allem effektive Programmdokumentation vorsieht. All dies wird sich in Kostenreduktion, effektiveren Simulationen, größerer Produktivität des Rechnersystems und vor allem effektiverem Einsatz von Arbeitskräften äußern. Der Autopatch PACER 600 ist der erste und effektivste Hybridrechner, der in Hardware und Software voll integriert ist.

Dem Benutzer des PACER 600 mit Autopatch steht ein hochentwickeltes Software-system (ECSSL) zur Verfügung, das eine einfache Beschreibung eines Simulationsmodells in FORTRAN-ähnlichen Gleichungen ermöglicht. Das ECSSL-System reduziert, ordnet und skaliert die Simulationsvariablen und ordnet sie den Autopatch-Komponenten des Analogrechners zu. ECSSL erstellt eine komplette Simulationsdokumentation und generiert HYTRAN-Files. Diese HYTRAN-Files ermöglichen dem Benutzer eine komplette interaktive Kommunikation mit dem PACER 600 Autopatch System. Durch das HYTRAN-System werden die Steckverbindungen in der Switch Matrix automatisch vorgenommen. Der Logic Processor wird geladen und mit den Daten, die die parallelen logischen Funktionen aus der Source-Definition der Simulation repräsentieren, initialisiert.

Die Erstellung und das Austesten des Koppelplans wird von Standard-HYTRAN-Programmen durchgeführt. Spezielle Eigenschaften des Systems erlauben jede Art von interaktiver Benutzeraktivität, um Teile des Problems nachzuprüfen und neu zuzuordnen oder in andere Hardware-Untersysteme einzugliedern. Der Lauf eines Programms und seine Verbindung zu Real-Time Digitalprogrammen geschieht unter der Kontrolle eines Standard-HYTRAN-Run-Executive.

Der Autopatch PACER 600 ist eine Erweiterung des EAI PACER 600 Systems mit Hardware und Software Untersystemen. ECSSL, HYTRAN und die Hybrid-Run-Time-Libraries sind am PACER 100 System exekutierbar. Das 600 Autopatch Hardware System besteht aus fünf größeren Subsystemen: Analog Switch Matrix, Logic Processor, programmierbare DCAs, programmierbare Time-Scale Kontrolle und programmierbarer Gain Select für die Summierer. Da alle Autopatch-Verbindungen zu Rechnerkomponenten hinter dem Steckbrett gemacht werden, verwandelt ein bestimmtes, speziell gestecktes Brett den PACER 600 sofort in ein PACER 600 Autopatch System. Wenn man dieses Brett entfernt und Standard-Software verwendet, wird das PACER 600 Autopatch System wieder ein manuell steckbarer PACER 600 mit kompletten manuellen Eigenschaften.

Hochentwickelte Software (ECSSL) reduziert die Turnaround-Zeit für ein Problem und stellt eine standard CSSL-kompatible Programmiersprache für Hybridbenutzer zur Verfügung. Zusätzlich wird das komplette hybride Problem hardwaremäßig digital gespeichert.

Automatische Verbindung der Komponenten

Durch die Switch Matrix und die Software zur Kontrolle der Switch Matrix wurde die Steckarbeit eliminiert und durch die stark verkürzte Zeit für die Erstellung eines Schaltplans, sowie für das manuelle Stecken und Überprüfen, reduziert sich die Programmentwicklung um einige Stunden.

Stored Program Capability

Die große Anzahl von Steckbrettern und Steckkabel, die früher Ausgaben in der Größenordnung von \$ 30000,- für ein aktives Hybridlabor ausgemacht haben und von anderen anfallenden Kosten (Abstellplatz, Abnutzung von Kabeln, Verwaltung der Steckbretter) begleitet waren, kann nun eliminiert werden. Selbst wenn man sogar von den Kosten für die Manpower eines hochqualifizierten Personals absieht, die mehr oder weniger unproduktiv zur wiederholten Überprüfung und Fehlerbehebung fertiggesteckter Problembretter vor jedem Lauf eingesetzt werden mußte, machen die sonst anfallenden Kosten bis zu einigen tausend Dollar jedes Jahr aus.

Konfigurationskontrolle

Die tatsächliche Konfiguration einer Konsole wird auf einem Software-Datenfile festgehalten. Dadurch können verschiedene analoge Konsolen verschiedene Konfigurationen haben. Das Software System kann selbsttätig die Konfiguration und seine Eignung für ein Programm feststellen.

Transfer von Programmen

Ein Programm kann von einem Rechenzentrum zu einem anderen, oder sogar von einer Konsole zu einer anderen transferiert werden, vorausgesetzt, daß jedes Hybridrechenzentrum oder jede Konsole die Basiskapazität für die Lösung des Programms besitzt.

Unterprogrammentwicklungen

Die hochentwickelte Software und die Autopatch Eigenschaften erlauben die Entwicklung und das Testen von Modell-Subsystemen und zwingen nicht zur Vorbereitung aller Modell-Subsysteme gleichzeitig am Rechner. Das führt bald zu Bibliotheken von Modell-Subsystemen und weiter zu rascherer Systementwicklung und ebenfalls rascher Rekonfiguration von dynamischen Modellen.

Geschwindigkeit der Fehlerbehebung

Wegen der hochentwickelten Software und der Autopatch Hardware wird der gesamte Fehlerbeseitigungsprozeß beschleunigt. Wie die oben erwähnte Entwicklung und das Austesten von Subsystemen und die Übertragbarkeit von Programmen sind alle Gesichtspunkte der Fehlerbehebung vereinfacht. Der Prozeß der analogen Skalierung z.B. war früher sehr arbeitsintensiv und zeitaufwendig, er wird nun, im Autopatch, algorithmisch und automatisch durchgeführt.

Verbesserte Wartung

Sowohl die Autopatch Software als auch die Autopatch Hardware wirken sich direkt in verbesserter präventiver Wartung aus, und wenn ein Fehler festgestellt wird, sind die Korrekturtechniken ebenfalls verbessert. Falls eine fehlerhafte Komponente am Rechner festgestellt wird, kann das Autopatch System dazu verwendet werden, das Problem im Bereich um die defekte Komponente neu zu konfigurieren. Es wird jedoch weiterhin empfohlen, fehlerhafte Komponenten zu reparieren anstelle sie durch Autopatch-Techniken zu umgehen.

Programmdokumentationskontrolle und Management

Ein bedeutender Vorteil des Autopatch ist die große Verbesserung in der Programmkontrolle, die der Benützer gewinnt. Zur Zeit, ohne Autopatch, muß der Benützer entweder ein erfahrener Programmierer sein oder sich auf einen Programmierer verlassen, sei er Experte oder nicht. In beiden Fällen machen der Benützer und /oder der Programmierer oft nicht-dokumentierte Programmänderungen, deren spätere Auffindung oder Rückgängigmachung einigen Aufwand erfordert. Mit Autopatch ist das Programm jederzeit, in allen Stadien, voll dokumentiert, und der Benützer kann eher seine eigene Programmierung durchführen und dadurch das Hybridprogramm zum ersten Mal voll unter seine Kontrolle bringen.

PACER 600 Erweiterungsprogramm

Da es sich um eine ergänzende Erweiterung zum PACER 600 handelt, sind alle Programme, die zur Zeit im manuellen Betrieb am PACER 600 laufen, am erweiterten PACER 600 exekutierbar. Wenn die Autopatch-Erweiterung entfernt ist, bleiben alle manuellen Möglichkeiten wie früher zur Verfügung. Das bedeutet, daß man für ein individuelles Programm wählen kann, ob man ein bereits existierendes Programm zum Autopatch umwandelt. Natürlich werden alle neuen Programme die Autopatch Eigenschaften verwenden. Der manuelle Betrieb soll jedoch in keiner Weise herabgesetzt werden.

Zusammenfassung der Vorteile der Autopatch-Erweiterung

- o rasches und leichtes Erstellen und Austesten von Schaltplänen;
- o große, dynamische Simulationsprobleme können eher und zu niedrigen Kosten bewältigt werden;
- o Automatisierung des analogen Programmierens und merkliche Reduzierung der Programmierzeit und -kosten;
- o komplette Programmdokumentation der Simulation zum späteren Nachschlagen und/oder zur sofortigen Kritik und Analyse;
- o Erschließung der Analog/Hybridrechentechnik für Nicht-Spezialisten, Erschließung neuer Anwendungen und Ermöglichung von billigerem Experimentieren;
- o mehr Zeit für kreative Arbeit anstelle von ermüdendem Programmieraufwand;
- o bessere Qualität der Simulationen;
- o effektiverer Einsatz von Arbeitskräften und Material;
- o Resultate innerhalb kürzerer Zeit;
- o Zusammenfassung der Programmiererfahrung vieler Analog/Hybridprogrammierer;
- o der Benützer kann das allgemeine Simulationsprogramm mit seinen eigenen speziellen Anforderungen verwenden;
- o standardmäßig interaktiver Betrieb bei der gesamten hybriden Simulation, einschließlich der Verwendung von FORTRAN- und Assemblerprogrammen;
- o Programme können für eventuelle Wiederverwendung erzeugt werden;
- o verbesserte Transportabilität analog/hybrider Programme;
- o automatische Erzeugung hybrider Programmhilfen aus der Problemdefinition;
- o automatische Generierung von großen Simulationsprogrammen (mehrere Konsolen);
- o Einrichtung zur Ausbildung neuer Analytiker.

Technischer Betrieb

ÖFFNUNGSZEITEN WÄHREND DER SOMMERFERIEN:

Die Hybridrechenanlage ist in der Zeit von

4.7.1977 bis 2.9.1977

Montag bis Freitag

von

8 Uhr bis 16 Uhr

geöffnet. Die CLOSED-SHOP-EINGABE während dieser Zeit erfolgt stündlich, bis 15 Uhr.

TAPE DISTRICTISIERT

Das Magnetband kann standardmäßig als District 16 angesprochen werden.

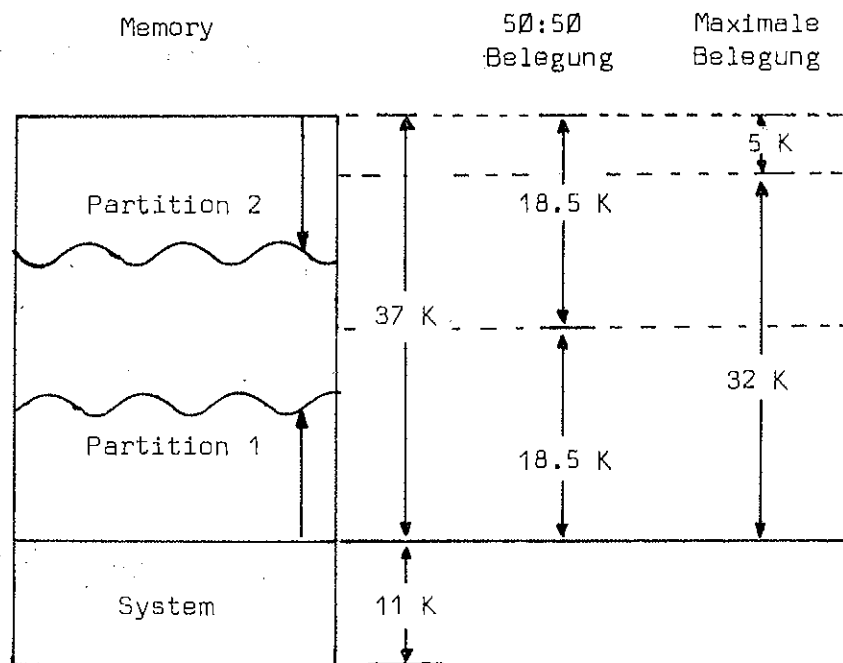
KERNSPEICHERERWEITERUNG

Der Kernspeicher des PACER 100 Digitalrechners wurde um 16 K erweitert und umfaßt somit 48 K 16-Bit Worte (siehe auch Memory Erweiterung, Seite 7, und Hardware-Ecke, Seite 26).

MEMORY ERWEITERUNG

Der "Kernspeicher" des PACER 100 Digitalrechners wurde um 16 K auf 48 K Memory erweitert (siehe auch Hardware-Ecke, Seite 26). Dadurch stehen den beiden Partitions für User-Tasks im Betriebssystem JCS/TS 7 nun insgesamt 37 K Memory zur Verfügung. Da ein User-Task maschinenbedingt höchstens 32 K belegen kann, wird eine wesentliche Durchsatzsteigerung erwartet, da sich größere Tasks nicht mehr gegenseitig blockieren.

Für den Closed Shop Betrieb ergibt sich daraus, daß Jobs bis zu 18.5 K stündlich eingegeben werden können. Die Maximallänge eines Tasks beträgt nun 32 K statt 21 K. Auf der Map Liste des Core Image Generators (/BUILD) wird durch FREE der Kernspeicherplatz angezeigt, der auf 32 K noch frei wäre. Standardmäßig wird dem Task der durch CORE angegebene Kernspeicher automatisch zugeteilt. Dieser Wert kann jedoch durch die CORE-Option verändert werden.



A. Blauensteiner

KURSE

Folgende Kurse werden im kommenden Wintersemester 1977/78 an der Hybridrechenanlage abgehalten werden:

RH1 GERÄTETECHNIK EAI PACER 600

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

Dauer: 3 Tage

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

Vortragender: Dipl.Ing. W. Kleinert

RH2 BENÜTZUNG DES BETRIEBSSYSTEMS JCS/TS 7

Optimale Ausnützung der Möglichkeiten des Systems, Systemstruktur

Dauer: 1 Woche

Termin: 1977/10/17 bis 1977/10/21

Vortragender: Dipl.Ing. A. Blauensteiner

RH3 HINWEISE FÜR FORTRAN-PROGRAMMIERER AN DER HYBRIDRECHENANLAGE

1 Vortrag

Termin: 1977/10/13

Vortragender: Dipl.Ing. F. Blöser

RH4 HYBRIDE FORTRAN-PROGRAMMIERUNG MIT ÜBUNGEN

Einführung in das Hybrid Linkage Softwarepaket

Dauer: 1 Woche

Termin: 1977/11/14 bis 1977/11/18

RH7 SOFTWAREUNTERSTÜTZUNG FÜR DIE BENÜTZUNG DES PACER 600 ALS PLOTTER SYSTEM

Dauer: 2 Tage (mit Vorführung am Rechner)

Termin: 1977/11/08 und 1977/11/09

Vortragender: Dipl.Ing. F. Blöser

RH9 EINFÜHRUNGSKURS IN DIE HYBRIDE PROGRAMMIERUNG

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

Dauer: 2 Tage

Vortragender: D. Solar

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

RH10 DATENVERBINDUNG MIT CYBER 74

1 Vortrag

Termin: 1978/01/19

Vortragender: Dipl.Ing. A. Blauensteiner

RH11 ASSEMBLER PROGRAMMIERUNG FÜR FORTGESCHRITTENE MIT ÜBUNGEN

Dauer: 1 Woche

Termin: 1978/03/06 bis 1978/03/10

Vortragender: Dipl.Ing. A. Blauensteiner

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

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

BENUTZUNG DES PROGRAMMPAKETS IPP1

Die Unterprogramme der neuen Version des Interpolations- und Plotterpakets IPP1 (vgl. Interface Nr. 8) sind für eine bequeme Verwendung in sieben Bibliothek-Sammelfiles zusammengefaßt. Auf fünf Sammelfiles befinden sich die mathematischen Unterprogramme für die Interpolation und Glättung von Funktionen und ebenen Kurven, getrennt nach Methoden. Jeder File enthält auch die Hilfsprogramme für die Auswertung und tabellarische Ausgabe der berechneten Kurven. Die für die graphische Darstellung benötigten Programme sind in einem weiteren File zusammengefaßt. Auf einem eigenen Sammelfile befinden sich auch alle Unterprogramme für die Spline-Approximation und Darstellung einer "analytisch" gegebenen Funktion.

Die IPP1-Sammelfiles sind:

AKIMA	- Akima-Interpolation
SPLINE	- Spline-Interpolation
NUSPL	- v-Spline-Interpolation
GLATTR	- Spline-Glättung nach Reinsch
GLATT	- Spline-Glättung nach Späth
RAHMEN	- Darstellung von Funktionen und ebenen Kurven
APPROX	- Darstellung einer "analytisch" geg. Funktion

Der genaue Inhalt der Sammelfiles und die Nummer des Districts, auf dem die Sammelfiles gespeichert sind, sind den Anschlägen bei der Rechenanlage zu entnehmen.

ACHTUNG - Es wird nochmals darauf hingewiesen, daß die bisher verwendeten Interpolations-, Glättungs- und Plotter-Bibliotheksprogramme (mit Ausnahme der Plotter-Grundsoftware) nur mehr bis Ende Juni zur Verfügung stehen werden (betrifft z.B. die Files DAPLOT, ASPLOT, FUPLOT, etc.). Die Programmberatung ist gerne bei der Umstellung bestehender Programme behilflich. Ab 1. Juli 1977 können nur mehr die neuen Versionen der IPP1-Programme verwendet werden.

F. Blöser

DISTRICT TRANSFER IN HOI

Der Hytran Operations Interpreter JCShOI bringt in seiner neuesten Version eine wesentliche Erweiterung für alle Benutzer: Bisher konnten Steps und Variable nur von der Konsole bzw. vom Lochstreifen oder Kartenleser eingegeben werden. Dieses Einlesen mußte mit jedem Run neu erfolgen, da diese Daten nach Beendigung des JCShOI-Jobs nicht mehr zur Verfügung standen. Es mußten daher die sich im On-Line Betrieb ergebenden Verbesserungen nachträglich auf Lochkarten durchgeführt werden. Nun können aber Steps und Variable auf Districts abgespeichert werden und von dort wieder zurückgelesen werden:

Durch die Steuerkarte

```
/HOI NAMEXX,DISTRICT=d
```

wird HDI aktiviert und zusätzlich der Datenfile NAMEXX auf District d zur Verfügung gestellt. Ohne die Options NAMEXX und d wird ein Standardfile generiert, der nach Beendigung von HOI nicht mehr zur Verfügung steht. Im allgemeinen können Files mit spezifiziertem Namen NAMEXX auf den Districts 19,20 und 21 angesprochen werden.

Beim ersten HOI-Programm mit NAMEXX wird dieser Datenfile errichtet. Erst mit einer

```
/DELETE NAMEXX,DISTRICT=d
```

wird dieser Datenfile wieder vom District d gelöscht.

Im HOI-Programm selbst kann nun durch

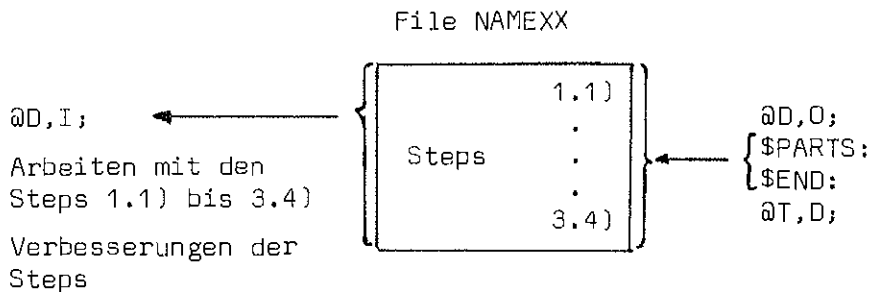
```
@D,I; Eingabe von einem File und
```

```
@D,O; Ausgabe zu einem File auf
```

dem bestimmten District durchgeführt werden.

Die ein- bzw. auszugebenden Steps oder Variable folgen diesen Befehlen. Ausgabe auf Files muß mit einem \$END: abgeschlossen werden. Anschließend an die Ausgabe muß durch @T,O; die Ausgabe auf Files abgeschlossen werden. Bei jedem Run von HOI wird sowohl die Ein- als auch die Ausgabe von Beginn des Files an angesprochen. Es können aber mehrere Blöcke hintereinander abgespeichert werden.

Beispiel:

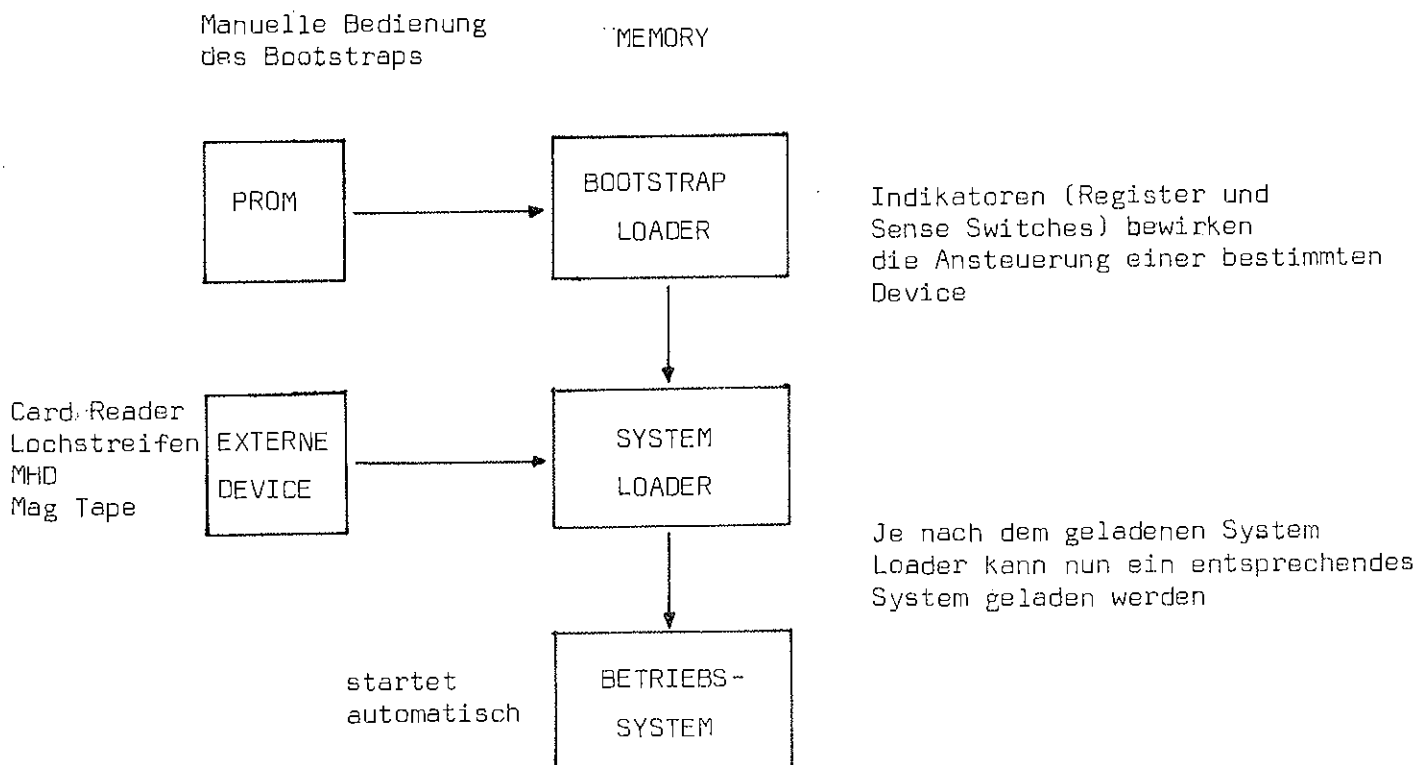


A. Blauensteiner

NEUER BOOTSTRAP

Der bisherige Bootstrap des PACER 100 Digitalrechners erlaubte lediglich das Laden eines Monitors von Lochstreifen bzw. Platten in Standardkonfiguration. Erst dieser Monitor konnte unser Betriebssystem laden.

Der neu entwickelte und installierte Bootstrap hingegen kann direkt das Betriebssystem laden und zwar von Massenspeichern beliebiger Konfiguration sowie von Lochstreifen und Lochkarten. Da die Art des Betriebssystems unabhängig vom Bootstrap ist, kann jedes andere System, also auch der ursprüngliche Monitor, geladen werden. Durch den neuen Bootstrap ist man vor allem bei Ausfall einer Device oder eines Massenspeichers nicht blockiert, das Betriebssystem zu laden. Weiters können bei Abstürzen direkt Notstarts durchgeführt werden bzw. verschiedene Versionen abgelegter Systeme geladen werden.



A. Blauensteiner

HYBRIDE FORTRAN-PROGRAMMIERUNG IM BETRIEBSSYSTEM JCS/TS 7

MIT DER HYBRID LINKAGE SOFTWARE

1. EINLEITUNG

Die Unterprogramme des HYBRID LINKAGE Softwarepakets, das als System-Software innerhalb des Betriebssystems JCS/TS 7 entwickelt wurde, dienen zur Kommunikation zwischen dem Digitalrechner und dem Analogrechner im Hybridsystem PACER 600.

Im folgenden werden alle Unterprogramme kurz beschrieben.

Detaillierte Beschreibungen (mit Aufruf und Parameterliste) sind in der Programmberatung erhältlich.

Was der Benützer beim Rechnen eines Hybridprogrammes im Multi-programming des Betriebssystem JCS/TS 7 beachten muß, wird in einem eigenen Artikel behandelt (siehe Seite 20). Werden in einem Hybridprogramm Programme des Hybrid Linkage Pakets verwendet, muß zuerst das Unterprogramm HYBRID aufgerufen werden. Ohne diesen Aufruf führt der Aufruf anderer hybrider Unterprogramme zum Programmabbruch.

2. DATENTRANSFER

Die Übertragung von Daten kann vom Analogrechner zum Digitalrechner oder umgekehrt erfolgen. Die zu übertragenden Werte liegen im Intervall $(-1, .9999)$, entsprechend der Normierung der Rechengrößen (Spannungen) am Analogrechner. Ihre digitale Entsprechung haben diese analogen Werte in den SCALED FRACTION Variablen^{+) der Digitalprogramme.}

Der Transferzeitpunkt von Daten kann sowohl vom digitalen Programm als auch durch logische Signale des Analogrechners gesteuert werden.

^{+) SCALED FRACTION Variable sind Ein-Wort Variable (16 Bit). Zuweisungen von REAL auf SCALED FRACTION und umgekehrt sind erlaubt. Mixed-Modes, d.h. arithmetische Ausdrücke mit Variablen anderen Typs sind möglich, jedoch ist zu beachten, daß bei der Auswertung eines Mixed-Mode Ausdrucks, in dem eine SCALED FRACTION Größe vorkommt, zuerst alle Variablen auf SCALED FRACTION gewandelt werden, sodaß es durch das Abschneiden auf $(-1, .9999)$ zu Fehlern bei der Wandlung kommen kann, obwohl das Ergebnis völlig richtig im Bereich der SCALED FRACTION Größen liegen würde. Diese Fehler können durch Zuweisen aller SCALED FRACTION Größen auf Hilfsvariable (vor dem Anführen in einem Ausdruck) vermieden werden.}

Digital-Analog-Wandler (Digital-Analog-Converter, DAC):

sie dienen zur Übergabe von Werten aus dem Digitalprogramm zum Analogrechner. Es sind 16 DAC vorhanden (rote Felder zwischen der letzten und der vorletzten Reihe der analogen Komponenten am Steckbrett). Die Ausgänge am Steckbrett haben die Bezeichnung DAC 0 bis DAC 7 und DAC 12 bis DAC 19. DAC 8,9,10 und 11 sind 4-Quadranten Digital-Analog-Multiplizierer (DAM), d.h. der Wert, der übergeben wird, kann noch mit einem analogen Wert, der unmittelbar unterhalb des DAM-Ausgangs gesteckt wird, multipliziert werden.

Die Übertragung kann generell auf 2 Arten erfolgen: falls das Jam Flip-Flop gesetzt ist (durch Aufruf des Unterprogramms SJAM) werden die Werte beim entsprechenden Übertragungsbefehl sofort am Analogrechner gesetzt. Ist das Jam Flip-Flop zurückgesetzt, (Unterprogramm RJAM) werden die Werte beim Übertragungsbefehl vorerst in ein Register geladen und erst durch einen Impuls auf die entsprechende Transferbuchse des Analogrechners oder durch Aufruf des Unterprogramms TRADAC übertragen. Die DAC's invertieren die zu übertragenden Werte. Die Genauigkeit beträgt 14 Bit plus Vorzeichen. Zu beachten ist, daß die Übertragung auf alle Fälle durch Signale auf den Transferbuchsen gesteuert wird (falls diese besteckt sind), auch wenn das Jam Flip-Flop gesetzt ist.

Analog-Digital-Wandler (Analog-Digital-Converter, ADC):

sie dienen zur Übergabe von Werten vom Analogrechner in das digitale Programm. Die Genauigkeit beträgt 14 Bit plus Vorzeichen. Es sind 24 ADC vorhanden (grüne Felder zwischen der ersten und zweiten Reihe der analogen Komponenten am Steckbrett). Die Eingänge am Steckbrett haben die Bezeichnungen ADC 0 bis ADC 23. ADC 0, 1, 2 und 3 haben Track-Stores eingebaut, die von den A/D Sample Control Buchsen 1-4 gesteuert werden können (log. Steuerung: 0=Track, 1=Store). Diese Steuerung ist genau invers zu jener der normalen Track/Store-Einheiten am Analogrechner. Diese vier ADC's invertieren den zu wandelnden Wert.

2.1. DIGITAL SYNCHRONISIERTER TRANSFER

Der Datentransfer wird durch den Aufruf des entsprechenden Unterprogramms gestartet und ist nach der Beendigung der Exekution dieses Unterprogramms beendet.

Die Digital-Analog-Wandler und DAM werden im digitalen Programm durch den Aufruf des Unterprogramms DAC geladen.

Die Unterprogramme DAC0, DAC1, DAC2, DAC3 und DAC4 ermöglichen ein im Aufruf einfaches Arbeiten mit den ersten 5 DAC's.

Die Übertragungsdauer ist im Vergleich zum Unterprogramm DAC kürzer. Mit dem Unterprogramm SBDAC können mehrere aufeinanderfolgende DAC mit einem Aufruf geladen werden. Das Unterprogramm TBDAC entspricht SBDAC mit TRADAC, gibt also zusätzlich noch das Übertragungssignal. Das Auslesen von ADC's erfolgt durch Aufruf des Unterprogramms ADC. Das Unterprogramm RBADC liest mehrere aufeinanderfolgende ADC's mit einem Aufruf aus.

2.2. ANALOG SYNCHRONISIERTER TRANSFER

Es können Datenblöcke (Blocklänge entspricht der Zahl der zugleich angesprochenen ADC's oder DAC's) zwischen dem Analogrechner und dem Digitalrechner weitgehend CPU-unabhängig übertragen werden. Der Transferimpuls für jeden Datenblock erfolgt durch ein Blip vom Analogrechner (DMC-Buchse). Zur Initialisierung der Übertragung über Digital-Analog-Wandler dient das Unterprogramm T0680, zur Initialisierung der Übertragung über Analog-Digital-Wandler das Unterprogramm OF680.

Nach erfolgter Initialisierung wird der Datentransfer bei Eintreffen der Übertragungssignale, von denen jedes die Übertragung eines Blocks auslöst, weitgehend CPU-unabhängig, parallel zu den Aktivitäten des digitalen Programmes, durchgeführt. Mithilfe des Unterprogrammes TRASTA kann man abfragen, ob der Datentransfer bereits gestartet oder schon beendet ist. Mit Aufruf des Unterprogrammes TRAFIN kann im Digitalprogramm gewartet werden, bis der Transfer beendet ist.

Die Übertragung der Datenwerte erfolgt unter Verwendung eines Arbeitsfeldes. Das Unterprogramm TRADDR liefert den Index des Feldelementes, das als letztes übertragen wurde.

3. LOGISCHE SIGNALE

Controllines sind Flip-Flops, die vom digitalen Programm gesetzt oder zurückgesetzt werden können und deren Ausgang am Analogrechner zur Verfügung steht. Es sind 16 Controllines vorhanden

und sie tragen die Bezeichnungen CL 0 bis CL 15 (rotes Feld zwischen dem Analogteil und dem Logikteil auf dem Steckbrett). Die Controllines 0 bis 9 sind auch vom Analogrechner her rücksetzbar (Reset Buchse RST unmittelbar unter den Ausgängen der Controllines am Steckbrett).

Im digitalen Programm erfolgt das Ansprechen durch Aufruf der Unterprogramme SCL und RCL.

Das Unterprogramm CL16 ladet alle Controllines gleichzeitig mit dem Bitmuster eines Digitalrechnerwortes.

Senselines sind Flip-Flops, die vom Analogrechner gesetzt, aber nur vom Digitalrechner zurückgesetzt werden können und deren Ausgang im digitalen Programm abgefragt werden kann.

Durch Abfrage im digitalen Programm werden diese Senselines zurückgesetzt. Es gibt 8 Senselines, sie haben die Bezeichnungen SL 0 bis SL 7.

Im digitalen Programm erfolgt das Auslesen durch Aufruf des Unterprogramms SL.

Durch das Unterprogramm SL08 können alle Senselines mit einem Befehl gleichzeitig ausgelesen werden. In diesem Fall werden sie jedoch nicht zurückgesetzt.

Soll mit Senselines der dynamische Zustand logischer Größen ausgelesen werden, ist ein vorheriges DUMMY-Auslesen erforderlich, um frühere Informationen zu löschen.

Mit dem Unterprogramm WAITSL kann im digitalen Programm gewartet werden, bis die entsprechende Senseline gesetzt wird.

Setzen und Zurücksetzen von Function Relays geschieht durch Aufruf des Unterprogrammes FRELAY.

Auslesen von Komparatoren im digitalen Programm geschieht durch Aufruf des Unterprogrammes COMPAR.

Real-Time-Clock: Die Real-Time-Clock ist ein 15 Bit Binärzähler, der vom digitalen Programm oder von einem Oktal-Switch-Panel im Interface auf einen bestimmten Wert geladen werden kann und diesen auf Impulse (fallende Flanke oder Blips) des Analogrechners hin verringert. Wenn der Zähler den Wert Null erreicht hat, erfolgt ein Carry-Out Signal auf einer entsprechenden Buchse des Analogrechners, und es kann ein Real-Time-Clock-Interrupt-Unterprogramm exekutiert werden, falls ein solches initialisiert wurde (siehe auch Interrupts (5)).

Gleichzeitig mit dem Carry Out wird der Zähler automatisch wieder aus einem Register mit dem vorherigen Wert geladen, sofern dieser inzwischen nicht digital durch ein Lade-Unterprogramm neu definiert wurde. Die Programmierung der Real-Time-Clock im digitalen Programm erfolgt durch die Lade-Unterprogramme LCLOCK (sofortiges Neuladen des Zählers unabhängig vom CO-Zustand, und PCLOCK (das Register wird gesetzt, der Zähler aber erst auf den neuen Wert geladen, nachdem er Null erreicht hat), durch RCLOCK (Auslesen des momentanen Zählerstandes) und ICLOCK (Initialisierung des Real-Time-Clock-Interruptunterprogrammes).

Setzen eines BCD-Zählers: Das Unterprogramm SETBCD setzt einen BCD-Zähler auf den Wert einer Integer Zahl aus dem Digitalprogramm, wobei 8 Controllines verwendet werden und eine neunte ein Lade-Signal für den BCD-Zähler liefert.

4. STEUERUNG DER ZUSTÄNDE UND BEDIENUNG DES ANALOGRECHNERS VOM DIGITALEN PROGRAMM

Setzen und Auslesen der Potentiometer: Das Unterprogramm SETPOT setzt ein Potentiometer (Potpositionen 20 - 119).

Durch Aufruf des Unterprogramms POT wird der Ausgang eines Potentiometers ausgelesen, im Analog SP oder PC Mode dessen Pot-Coefficient (eingestellter Wert ohne Vorzeichen). Das Unterprogramm QPOT liest den Ausgang eines Handpots aus, im Analog SP oder PC Mode dessen eingestellten Wert ohne Vorzeichen.

Ferner steht dem Benutzer ein Hauptprogramm SETPTS zur Verfügung, das zum Setzen mehrerer Potentiometer und DCA's dient. Die Potentiometernummern und Werte werden auf Datenkarten eingegeben, wobei pro Potentiometer eine Karte verwendet wird, welche die Nummer und den Wert des Pots im Format I3,S6 enthalten muß.

Digital-Controlled-Attenuators (DCA):

sind digital setzbare, invertierende 2-Quadranten Potentiometer in den Potpositionen 0 - 19. Das Setzen erfolgt durch Aufruf des Unterprogramms SETDCA. Das Auslesen kann am Analogrechner (Anwählen wie ein Potentiometer) oder im digitalen Programm (Aufruf des Unterprogramms READCA) erfolgen. READCA liefert den eingestellten Wert, der momentane Wert des DCA im ST oder PP-Mode

muß mit POT ausgelesen werden. Die DCA's können nur auf positive Werte gesetzt werden. Sie invertieren! Das Setzen und Auslesen mehrerer DCA's im digitalen Programm kann auch blockweise erfolgen. Dazu sind die Unterprogramme SBDCA bzw. RBDCA aufzurufen.

Anwählen der Logic-Modes: die entsprechenden Unterprogramme sind LMRUN, LMSTOP und LMCLR.

Anwählen der Analog-Modes: die entsprechenden Unterprogramme sind AMSP, AMST, AMPC, AMPP, AMH, AMIC und AMOP.

Time Scale: die entsprechenden Unterprogramme sind TSNS, TSFS, TSNMS und TSFMS.

Das Auslesen eines Verstärkerausganges geschieht durch Aufruf des Unterprogramms AMPL, die Unterprogramme DERIVA und DERIVZ liefern die Summer aller Eingänge (ohne IC) eines Verstärkers bzw. das Zehntel dieser Summe.

Das Digitalvoltmeter wird durch Aufruf des Unterprogramms READVM ausgelesen.

Der Aufruf des Unterprogramms TRUNK liest den Wert eines Trunks aus.

5. INTERRUPTS

Ein Interruptunterprogramm ist ein digitales Unterprogramm, das exekutiert wird, wenn ein entsprechendes Signal vom Analogrechner (Interrupt) kommt. Dabei wird das laufende Hauptprogramm bis zur Beendigung des Interruptunterprogramms unterbrochen. Es gibt Interrupts verschiedener Prioritäten. Der Real-Time-Clock-Interrupt hat höchste Priorität, dann folgen die General-Purpose-Interrupts 0, 1, 2, 3, 4, 5, 6 und 7.

In einem Programm sind maximal 8 General-Purpose-Interruptunterprogramme möglich. Die Signale dafür sind am Steckbrett (Feld zwischen dem Analogteil und dem Logikteil, neben den Senselines) in die Eingänge GPI 0 bis GPI 7 zu stecken (GPI = General-Purpose-Interrupt). Im digitalen Programm muß eine Initialisierung aller gewünschten Interrupts und die Zuordnung der GPI's zu den Interruptunterprogrammen erfolgen. Das geschieht durch Aufruf des Unterprogrammes INTER.

Ein Signal auf der entsprechenden Interruptbuchse bewirkt nur dann die Exekution eines initialisierten Interruptunterprogrammes, wenn vorher Interrupts durch Aufruf des Unterprogramms SI

erlaubt wurden. Ein Aufruf des Unterprogramms RI sperrt die Durchführung von initialisierten Interrupts bis zum nächsten SI-Aufruf. SI darf jedoch niemals in einem Interruptunterprogramm aufgerufen werden.

Interruptunterprogramme sind nicht rekursiv aufrufbar, und können sich nicht wechselseitig unterbrechen. Parameter müssen über COMMON-Blöcke übergeben werden.

Interruptunterprogramme müssen mit CALL RETURN beendet werden. Sollen sie auch vom Digitalprogramm aufgerufen werden können, ist ein zusätzlicher Rücksprung (nur RETURN) vorzusehen.

Mithilfe des Unterprogramms GPITST kann festgestellt werden, ob ein General-Purpose-Interrupt gerade anliegt oder nicht (nur sinnvoll, wenn RI aufgerufen wurde). Das Unterprogramm GPIØ8 gibt ebenfalls Auskunft, welche der Interrupts gerade anliegen.

6. DIGITAL-CONTROLLED-FUNCTION-GENERATORS (DCFG):

Die digital setzbaren Funktionsgeber werden im Digitalprogramm gesetzt und liefern die analoge Funktion einer Variablen am Analogrechner. Es sind zwei Funktionsgeber vorhanden: DCFG 1 und DCFG 2 (Feld zwischen der zweiten und dritten Reihe der analogen Komponenten am Steckbrett). In einem Intervall (max. (-1,1)) werden vom digitalen Programm maximal 16 Stützstellen mit den dazugehörigen Stützwerten definiert. Die so festgelegte Funktion steht auf dem Analogrechner zur Verfügung.

Der Eingang wird unmittelbar über dem Ausgang des DCFG gesteckt.

Das Setzen des DCFG im digitalen Programm (Definieren von Stützstellen und Stützwerten) erfolgt z.B. durch Aufruf des Unterprogramms DCFG. Dabei muß die zu setzende Funktion in einer FORTRAN FUNCTION definiert werden. Das Unterprogramm DCFG trifft eine optimale Stützstellenwahl und skaliert die Funktion entsprechend. Wahlweise kann dann der DCFG nach den berechneten Stützstellen gesetzt werden.

Die Unterprogramme LFGS und LFGR wandeln vorgegebene Stützpunkte in das DCFG-Format und setzen wahlweise den DCFG. Das Unterprogramm LFGFD benötigt zum Setzen eines DCFG etwa 70 µsec.

SPEZIELLE PROBLEME DER HYBRIDEN PROGRAMMIERUNG

IM MULTIPROGRAMMING DES BETRIEBSSYSTEMS

JCS/TS 7

Bei der Exekution von Programmen treten bedingt durch die unterschiedliche Arbeitsgeschwindigkeit peripherer Geräte (Line Printer, etc.) Wartezeiten auf, in denen die Central Processing Unit (CPU) auf die Beendigung eines Datentransfers warten muß. Um die CPU optimal ausnützen zu können, ist es sinnvoll, mehrere Programme logisch parallel zu exekutieren (Multiprogramming). Dies wird durch die Definition mehrerer Exekutionslevel erreicht, wobei die CPU jedem Level für eine fest vorgegebene Zeit (Cycle Time) zur Verfügung steht. Wenn der momentan aktive Level auf das Ende einer Hardware-Aktivität wartet, die CPU also für ihn inaktiv ist, wird dieser Exekutionslevel vorzeitig abgebrochen und die CPU für den nächsten Level freigegeben. Das auf dem vorhergehenden Level laufende Programm wird erst wieder beim nächsten Zyklus weiterexekutiert.

Die daraus resultierenden zeitlichen Unterbrechungen des Programmablaufes sowie die zeitliche Trennung von logischer und physikalischer Aktivität spielen zwar bei der Exekution von digitalen Programmen keine Rolle, bei hybriden Programmen müssen sie jedoch wegen der notwendigen Synchronisation mit dem Analogrechner berücksichtigt werden.

Im Betriebssystem JCS/TS 7 werden zwei Benutzer- und zwei Systemlevel bedient, die Standard Cycle Time beträgt 6 Millisekunden. Das System stellt FDRTRAN-aufrufbare Unterprogramme zur logischen Steuerung des Analogrechners, für den hybriden Datentransfer über einen direkten Datenkanal und einen Standard I/O Kanal und zur Bedienung von Interrupts zur Verfügung.

Bei der Exekution eines Programmes können im Normalfall folgende Unterbrechungen auftreten (die angegebenen Zeiten sind Maximalzeiten):

- Unterbrechung bis zur dreifachen Cycle Time durch das Multiprogramming (standard bis zu 18 msec)
- Unterbrechung bis zu 500 µsec durch Systeminterrupts, die vom eigenen oder von den anderen Exekutionsleveln initialisiert, aber erst zu einem späteren Zeitpunkt exekutiert werden
- Unterbrechung bis zu 100 µsec durch Systeminterrupts, die unabhängig von den laufenden Exekutionsleveln in gleichbleibenden Abständen exekutiert werden (z.B. zum Update der Tageszeit)
- Warten auf das Ende eines eventuell momentan laufenden Datentransfers eines anderen Exekutionslevels über den direkten Datenkanal (bis zu 800 µsec).

Da diese Unterbrechungen eine Synchronisation von Analog- und Digitalrechner erschweren bzw. bei zeitkritischen Problemen unmöglich machen, muß das System die Möglichkeit bieten, in zeitkritischen Phasen alle störenden Aktivitäten zu sperren.

JCS/TS 7 sieht für den Benutzer folgende Möglichkeiten vor:

- Sperren aller anderen Exekutionslevel für eine vorzugebende Zeit durch Aufruf eines Unterprogrammes. Die von den Exekutionsleveln unabhängigen Systeminterrupts werden jedoch nicht gesperrt.
- Sperren des direkten Datenkanals (OMAC - Direct Memory Access Channel) für alle anderen Exekutionslevel für eine vorzugebende Zeit.

Diese Sperren werden nach Ablauf der angegebenen Zeitschranke automatisch aufgehoben, sie können jedoch auch vom Benutzer vorzeitig durch

Aufruf eines entsprechenden Unterprogrammes beendet werden. Die Sperre des DMAC wird sofort nach Beendigung der initialisierten Datenübertragung aufgehoben. Durch die zeitliche Beschränkung wird gewährleistet, daß auch bei fehlerhaften Programmen (z.B. Endlosschleife) die Kontrolle wieder vom System übernommen werden kann. Bei extrem zeitkritischen Problemen besteht jedoch die Möglichkeit, alle anderen Aktivitäten auf unbegrenzte Zeit zu sperren.

- Unterbrechen des momentan aktiven Exekutionslevels durch vom Benutzer definierte Interrupts (General-Purpose und Real-Time-Clock Interrupt). Während der Exekution des entsprechenden Interruptunterprogrammes sind alle anderen Exekutionslevel gesperrt. Systeminterrupts können jedoch nicht von Userinterrupts unterbrochen werden.

Bei der Entwicklung eines hybriden Programmes sind also die angeführten Unterbrechungen zu berücksichtigen, bzw. bei zeitkritischen Problemen mit den gegebenen Möglichkeiten zu steuern. Die Programmierung sollte dabei so erfolgen, daß die anderen Exekutionslevel nicht unnötig behindert werden. Im folgenden werden die wichtigsten Arten der Synchronisation und Datenübertragung in Hinblick auf dieses Problem analysiert und Anleitungen zur Programmierung gegeben. Die verwendeten Unterprogrammaufrufe beziehen sich auf die Programme des Programmpakets HYBRID LINKAGE (siehe auch Seite 13).

1. LOGISCHE SYNCHRONISATION VON ANALOG- UND DIGITALRECHNER

Es können im wesentlichen drei Arten der Synchronisation unterschieden werden, deren Verwendung von der Art des gestellten Problems abhängt.

1.1 DIGITALRECHNER - ZENTRAL
ANALOGRECHNER - PERIPHER

Der Einsatzzeitpunkt des Analogrechners wird vom Digitalrechner gesteuert.

Standardsteuerung:

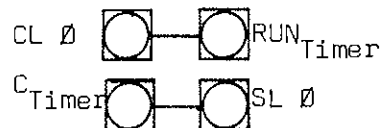


Im folgenden gibt Controlline \emptyset (CL \emptyset) das Startsignal zum Analogrechner, Senseline \emptyset (SL \emptyset) erhält vom Analogrechner das 'Ende'-Signal:

CL \emptyset - OP_{Start}
OP_{Ende} - SL \emptyset

Beispiel bei Verwendung des 68 \emptyset Rep-Op Timers:

A = IC-Zeit
B = DP-Zeit
C = .1



1.1.1. Nichtzeitkritische Synchronisation:

Das vom digitalen Programm erkannte DP_{Ende} kann beliebig lang hinter dem tatsächlichen OP_{Ende} liegen. Der Multiprogrammbetrieb wird nicht eingeschränkt.

a) CALL SCL (Ø)	b) CALL SL (Ø,SLØ)	c) CALL SL (Ø,SLØ)
CALL WAITSL (Ø)	CALL SCL (Ø)	CALL SCL (Ø)
CALL RCL (Ø)	1 CALL SL (Ø,SLØ)	1 CALL CYCLE
	IF (.NOT.SLØ) GOTO 1	CALL SL (Ø,SLØ)
	CALL RCL (Ø)	IF (.NOT.SLØ) GOTO 1
		CALL RCL (Ø)

In allen drei Fällen muß das OP_{Ende} -Signal vor Aufruf des ersten Unterprogrammes logisch Ø sein, das DP_{Ende} wird vom digitalen Programm mit einer Verzögerung von maximal 19 msec erkannt.

ad a) Da in WAITSL die Senseline zuerst zurückgesetzt und dann repetierend abgefragt wird, ob sie gesetzt ist, muß der Abstand $OP_{\text{Start}} - DP_{\text{Ende}}$ mindestens 19 msec betragen. (Falls der Einsprung in WAITSL wegen des Cycles erst nach DP_{Ende} zu einem Zeitpunkt erfolgt, da das Signal auf SL Ø bereits wieder zurückgesetzt ist, kann das OP_{Ende} nicht mehr erkannt werden.)

ad b) Keine Einschränkungen

ad c) Keine Einschränkungen. Durch Aufruf von CYCLE wird die Kontrolle immer an den nächsten Exekutionslevel weitergegeben, sodaß die Zeit, in der das Programm auf das Setzen von SL Ø wartet, von den anderen Exekutionsleveln ausgenutzt werden kann.

Achtung: Es ist zu berücksichtigen, daß der Analogrechner wegen des verzögerten Zurücksetzens von CL Ø eventuell mehrere Läufe durchführt, falls er im REP-DP Betrieb (PP-Mode) verwendet wird. Soll dies vermieden werden, muß CL Ø von der parallelen Logik direkt mit dem OP_{Ende} -Signal zurückgesetzt werden. Der Aufruf von RCL kann dann entfallen.

1.1.2. Zeitkritische Synchronisation:

Das vom digitalen Programm erkannte DP_{Ende} muß möglichst genau mit dem tatsächlichen OP_{Ende} übereinstimmen. Alle anderen Exekutionslevel müssen durch Aufruf von LOCK für die benötigte Zeit ITIME (in Millisekunden) gesperrt werden.

CALL LOCK (ITIME)	CALL LOCK (ITIME)
CALL SCL (Ø)	CALL SL (Ø,SLØ)
CALL WAITSL (Ø)	CALL SCL (Ø)
CALL RCL (Ø)	1 CALL SL (Ø,SLØ)
CALL RELOCK	IF (.NOT.SLØ) GOTO 1
	CALL RCL (Ø)
	CALL RELOCK

Das Eingangssignal auf SL Ø muß bei Aufruf des ersten Unterprogrammes zurückgesetzt sein. Die Differenz zwischen analogem und digital erkanntem OP_{Ende} beträgt maximal 100 µsec (wegen möglicher Unterbrechungen durch Systeminterrupts). Bei Verwendung von WAITSL muß daher der Abstand $OP_{\text{Start}} - OP_{\text{Ende}}$ oder das Signal auf SL Ø mindestens genau so lang sein. Diese Art der Synchronisation wird jedoch im allgemeinen nur im Zusammenhang mit hybridem Datentransfer benötigt.

1.2 ANALOGRECHNER - ZENTRAL DIGITALRECHNER - PERIPHER

Der Einsatzzeitpunkt des Digitalrechners wird vom Analogrechner gesteuert. Bei Problemen dieser Art wird die Synchronisation am günstigsten mit Interrupts durchgeführt. Bei Eintreffen eines Interruptsignals wird der momentan laufende Exekutionslevel unterbrochen und das entsprechende Interruptunterprogramm ohne Unterbrechung vollständig exekutiert.

Verzögerung: Overhead (Abstand Startsignal bis zum tatsächlichen Beginn der Exekution)

bis 500 μ sec bei unmaskiertem Betrieb (kein Aufruf von LOCK im Hauptprogramm, daher keine Behinderung des Multiprogramming). Diese Verzögerung entsteht durch eventuell momentan aktive Systeminterrupts, die vom Userinterrupt nicht unterbrochen werden können.

80 μ sec bei maskiertem Betrieb (die anderen Exekutionslevel sind durch Aufruf von LOCK gesperrt worden).

Der wesentliche Vorteil der Interrupttechnik liegt darin, daß die Zeit zwischen den einzelnen Interrupts sowohl vom eigenen als auch von den anderen Exekutionsleveln für andere Aktivitäten ausgenützt werden kann. Dabei müssen jedoch die eventuell vom eigenen Level ausgelösten Systeminterrupts berücksichtigt werden. Falls die Synchronisation wegen der Overhead-Zeit nicht auf diese Weise durchgeführt werden kann, muß auf die Verwendung von Senselines und Sperre aller anderen Exekutionslevel zurückgegriffen werden.

1.3 ANALOGRECHNER - ABWECHSELND ZENTRAL UND PERIPHER DIGITALRECHNER

Steuerung durch Kombination von 1.1 und 1.2, hauptsächlich mit Interrupts, die den Analogrechner und somit sich selbst steuern. Diese Art der Synchronisation wird jedoch nur bei komplexeren Problemen benötigt.

2. DATENTRANSFER

Der Datentransfer kann wahlweise über den direkten Datenkanal (DMAC) oder über den Standard I/O Kanal erfolgen, jedoch jeweils nur in eine Richtung, falls der DMAC verwendet wird.

Standard I/O Kanal: Der Transfer erfolgt bei Aufruf eines entsprechenden Unterprogrammes im digitalen Programm.

Direkter Datenkanal: Der Transfer wird nach erfolgter Initialisierung unabhängig vom laufenden Programm und vom laufenden Exekutionslevel durch Signale vom Analogrechner gesteuert.

2.1. Datentransfer vor oder nach einem analogen Problemlauf

2.1.1. Vor dem Problemlauf:

Digital-Analog-Übertragung von nur einem Wert je DAC, nicht zeitkritisch (asynchron). In diesem Fall genügt die Verwendung des Standard I/O Kanals mit Einzel- oder Blockübertragung. Der Transferzeitpunkt ist identisch mit dem Aufruf des entsprechenden Unterprogrammes, kann aber auch vom Analogrechner zu einem späteren Zeitpunkt unabhängig vom laufenden Digitalprogramm bestimmt werden. Eine Übertragung über den direkten Datenkanal ist in diesem Fall nicht sinnvoll, da weder die Rechenzeit verkürzt noch das Programm vereinfacht werden kann.

2.1.2. Nach dem Problemlauf:

Übertragung von nur einem Wert je ADC. Zeitkritische Phase: Zeitraum zwischen OP_{Ende} und dem Auslesen der ADC. Die den ADC vorgeschalteten Punktspeicher müssen solange in den Store-Mode versetzt werden, bis die ADC vom digitalen Programm ausgelesen wurden.

Unmaskierte Durchführung (ohne LOCK): Synchronisation nach 1.1.1.

- Über den Standard I/O Kanal:
Keine Behinderung des Multiprogramming, jedoch an eine sehr lange Speicherdauer von 19 msec gebunden.
- Über den direkten Datenkanal (DMAC):
Kurze Speicherdauer (30 µsec/ADC), der DMAC ist jedoch während des Analoglaufes für andere Aktivitäten unnötig gesperrt.
- Über den Standard I/O Kanal auf Interruptbasis:
Kurze Speicherzeit (500 µsec plus 60 µsec/ADC), keine Behinderung des Multiprogramming. Diese Art der Durchführung ist sowohl wegen der kurzen Speicherdauer als auch im Sinne eines ungestörten Multiprogramming als optimal anzusehen.

Maskierte Durchführung (mit LOCK): Synchronisation nach 1.1.2.

Alle anderen Exekutionslevel sind gesperrt.

- Über den Standard I/O Kanal:
Notwendige Speicherdauer: 60 µsec/ADC plus Rechenzeit für die Synchronisation.
- Über den direkten Datenkanal:
Siehe oben.
- Über den Standard I/O Kanal auf Interruptbasis:
Benötigte Speicherdauer: 80 µsec Overhead plus 60 µsec/ADC.

2.2. Datentransfer parallel zu Analoglauf

Ausschlaggebend für die Wahl der Übertragungsart ist die benötigte Übertragungsfrequenz. Die Zeit zwischen zwei Transferimpulsen muß ausreichen, um

- bei einer D/A Übertragung alle benötigten DAC zu laden, der Transferzeitpunkt selbst kann dann vom Analogrechner auf 1 µsec genau gesteuert werden,
- bei einer A/D Übertragung alle verwendeten ADC seriell auszulesen. Das 'Laden' der ADC (= Bereitstellen der zu übertragenden Werte) kann mit Punktspeichern vom Analogrechner auf 1 µsec genau gesteuert werden.

Dabei ist zu beachten, daß das Laden von DAC vor und das Auslesen von ADC nach dem Transferimpuls zu erfolgen hat.

Für die Durchführung stehen zwei grundsätzlich verschiedene Möglichkeiten zur Verfügung:

- Die digital synchronisierte Datenübertragung:
Der Datentransfer wird durch Aufruf entsprechender Unterprogramme aktiviert und ist beim Rücksprung in das rufende Programm bereits beendet. Das digitale Programm führt also während des Analoglaufes die Datentransfers durch und ist mit den Transferimpulsen voll synchronisiert. Zur Übertragung wird der Standard I/O Kanal verwendet, die Richtung der Datenübertragung (D/A oder A/D) kann während des Analoglaufes beliebig oft gewechselt werden.
- Die analog synchronisierte Übertragung:
Die Übertragung der Daten wird im digitalen Programm vor dem Analoglauf initialisiert und während des Analoglaufes unabhängig vom laufenden Programm durch Transferimpulse vom Analogrechner gesteuert. Für diese Art der Datenübertragung wird der direkte Datenkanal verwendet, die Richtung muß bei der Initialisierung angegeben werden und kann während der Übertragung nicht mehr

geändert werden. Die Verwendung eines direkten Datenkanals erlaubt gegenüber dem Standard I/D Kanal eine weit höhere Übertragungsfrequenz.

- Die Datenübertragung mit Interrupts über den Standard I/D Kanal als dritte Möglichkeit kann als Kompromiß aus den beiden anderen Übertragungsarten angesehen werden, da sie einen beliebigen Wechsel der Übertragungsrichtung ermöglicht und unabhängig vom laufenden Programm durchgeführt werden kann.

Im folgenden hat $CL \emptyset$ dieselbe Bedeutung wie in Kapitel 1., die (N) Transferimpulse seien je nach Übertragungsart entweder auf SL 1, DMC oder einen Interrupt gesteckt.

2.2.1. Digital synchronisierte Übertragung (über den Standard I/D Kanal)

```
CALL SL (1,SL1)
CALL SCL (0)
DO 2 I=1,N
CALL DAC ( )           Laden der DAC
1 CALL SL (1,SL1)      Warten auf den Transferimpuls
IF (.NOT.SL1) GOTO 1
CALL ADC ( )           Auslesen der ADC
2 CONTINUE
```

Bei unmaskiertem Betrieb (keine Sperre der anderen Exekutionslevel) ist diese Art der Synchronisation nur dann funktionsfähig, wenn der Abstand zweier Transferimpulse sowie der Abstand $DP_{\text{Ende}} - 1$. Impuls mindestens 18 msec (Cycle) plus 25 $\mu\text{sec}/\text{DAC}$ plus 60 $\mu\text{sec}/\text{ADC}$ plus Rechenzeit für die Synchronisation beträgt. Bei maskiertem Betrieb tritt anstelle der durch den Cycle verursachten Verzögerung eine durch Systeminterrupts bedingte maximale Verzögerung von 100 μsec .

2.2.2. Analog synchronisierte Datenübertragung (über den DMAC)

Die Initialisierung dieser Übertragung erfolgt vor dem Analoglauf, der DMAC wird für eine vorzugebende Zeitschranke für alle anderen Aktivitäten gesperrt. Dabei müssen jedoch folgende mögliche Verzögerungen einkalkuliert werden:

- Falls der DMAC zum Zeitpunkt der Initialisierung noch aktiv ist, muß das Ende dieser Aktivität abgewartet werden. Dadurch kann eine maximale Verzögerung von 800 μsec entstehen (bei nicht hybrider Aktivität), bei hybrider Aktivität kann diese Verzögerung der beim letzten Transfer angegebenen LOCK-Zeit entsprechen.
- Bei der Vorgabe einer Zeitschranke zur Sperre des DMAC ist zu berücksichtigen, daß ein sofort nach der Initialisierung des Datentransfers auftretender Wechsel des Exekutionslevels den Start des Analoglaufes um bis zu 18 msec verschieben kann, falls die anderen Exekutionslevel nicht gesperrt wurden.

Der Mindestabstand zweier Transferimpulse ist bei einer D/A-Übertragung mit 2 $\mu\text{sec}/\text{DAC}$ plus 10 μsec Settling Time, bei einer A/D-Übertragung mit 40 $\mu\text{sec}/\text{ADC}$ gegeben.

2.2.3. Datentransfer mit Interrupts (über den Standard I/D Kanal)

Der Mindestabstand zweier Transferimpulse ergibt sich in diesem Fall aus 60 $\mu\text{sec}/\text{ADC}$ plus 25 $\mu\text{sec}/\text{DAC}$ plus Dverhead für die Interruptbedienung (bis 500 μsec bei unmaskiertem, 80 μsec bei maskiertem Betrieb).

D. Solar

HARDWARE - ECKE

1. EINLEITUNG

An der Hybridrechenanlage wurden im letzten Jahr als Eigenentwicklungen eine Reihe von Hardwaremodifikationen an der Zentraleinheit, am Interface und den verschiedenen I/D-Geräten vorgenommen. Alle diese Entwicklungen hatten den Sinn, die Schnelligkeit des Systems zu erhöhen, mit sinnvollen Erweiterungen neue Möglichkeiten zu schaffen und Geldmittel zu sparen.

In der folgenden "Hardware-Ecke" wollen wir unseren Lesern einen kleinen zusammenfassenden Überblick über unsere Aktivitäten auf dem Gebiet der "Hardwareprogrammierung", wie wir es selber oft nennen, geben.

2. INTERFACE MODIFIKATIONEN

a) Trennung des hybriden DMAC von den Massenspeichern

Der Digitalteil des Hybridrechners EAI PACER 100 besitzt zwei voneinander unabhängig arbeitende direkte Speicherkanäle (DMAC's), die einen unmittelbaren Zugriff zum Hauptspeicher erlauben, ohne den Instruktionsablauf der CPU zu unterbrechen. Falls der Hauptspeicher zur gleichen Zeit von der CPU und den DMAC's angesprochen wird, so hat der DMAC1 Priorität gegenüber dem DMAC2, und dieser wieder gegenüber der CPU. Bei unserer Anlage werden die Massenspeicher (Festkopffplatte, Wechselplatte und Magnetband) und das Interface über die DMAC's angesprochen. Aus historischen Gründen war bisher die Festkopffplatte dem DMAC1 zugeordnet, das Interface und die restlichen Massenspeicher dem DMAC2. Dies führte zu einer erheblichen Durchsatzverminderung am Digitalrechner, falls im Multiprogrammbetrieb hybrider Datentransfer längere Zeit den DMAC blockierte und es gab keine Möglichkeit, größere Datenmengen vom Analogrechner z.B. zum Mag. Tape zu übertragen. Nach eingehenden theoretischen und praktischen Untersuchungen wurde die folgende wichtige Änderung durchgeführt:

Der hybride Datenkanal hängt nun gemeinsam mit der Fixed Head Disk auf DMAC1, die restlichen Massenspeicher auf DMAC2. Dadurch beträgt die maximale Wartezeit für die Initialisierung eines hybriden DMA-Block-Transfers 280 μ sec (1 Sektor-Übertragung zur Fixed Head Disk), die beim Aufruf des Unterprogrammes T0680 oder DF680 automatisch abgewartet werden, während durch lange Blöcke eines hybriden Transfers der Datenaustausch zu den großen Massenspeichern nicht mehr behindert wird.

b) Vereinfachung der Synchronisation für den hybriden direkten Datenkanal

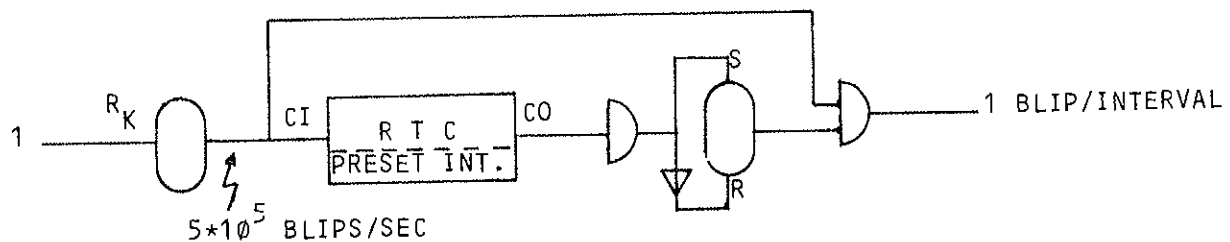
Der hybride direkte Datenkanal dient üblicherweise dazu, entweder eine Gruppe von analogen zeitabhängigen Variablen mit einer bestimmten Frequenz (bis zu 30 kHz) abzutasten oder um einen Block von digital gespeicherten Feldern über DA-Wandler zur Erzeugung von synchronisierten analogen Funktionen dem Analogrechner zu überspielen (bis zu 200 kHz Durchsatzrate).

Um den Datentransfer mit dem Analogrechner über den direkten Speicherkanal richtig zu synchronisieren, war für den Fall der gleichzeitigen Übertragung mehrerer Kanäle bisher ein relativ aufwendiges Programm am Logikteil des 680 Analogrechners

zu stecken (2 BCD-Zähler, 2 Flip-Flops, 3 And-Gates). Gerade bei komplexeren Hybridproblemen stellte dieser Komponentenaufwand oft eine echte Limitierung der Programmiermöglichkeiten dar. Noch dazu waren die BCD-Zähler, je nachdem ob ADC's oder DAC's über den direkten Datenkanal verwendet wurden, verschieden einzustellen. Diese Einschränkungen waren der Grund für eine Hardwaremodifikation im Interface. Jetzt wird zur Synchronisation nur noch eine steigende Flanke oder ein Blip zur Initialisierung der Übertragung einer Gruppe von Funktionen zu jedem Synchronisationszeitpunkt benötigt. Die Wandlung bzw. das Setzen aufeinanderfolgender Kanäle erfolgt mit der hardwaremäßig größtmöglichen Geschwindigkeit.

c) Real-Time-Clock

Die am Hybridrechner vorhandene Real-Time-Clock (RTC) besteht aus einem 15-Bit Binärzähler (maximaler Einstellwert 32768), der entweder softwareweise von einem Digitalprogramm her oder von einem Octal Preset Panel im Interface geladen werden kann. Dekrementiert wird die RTC durch eine steigende Flanke bzw. ein Blip an der CI-Buchse am 680-Steckbrett. Hat der Counter den Wert Null erreicht, so wird die Carry Out (CO) Buchse am 680 logisch 1 gesetzt und wahlweise ein Interrupt erzeugt. An dieser Stelle soll darauf hingewiesen werden, daß das CO-Signal mit der internen Clock der CPU und nicht mit der der 680 logisch getaktet ist. Es ist daher z.B. wie folgt zu synchronisieren:



LOGIC BUFFERING OF REAL-TIME-CLOCK

Das softwaremäßige Laden der RTC (CALL LCLOCK (I)) bewirkte bisher, daß unmittelbar bei der Ausführung des Ladebefehls die RTC auf den Wert I gesetzt wurde. Nun sind bei uns aber spezielle Anwendungen aufgetreten - wie die Erzeugung von nicht-äquidistanten Impulsfolgen - bei denen es wünschenswert ist, während eines Count-Down-Zyklus der RTC einen neuen Wert für das Laden nach dem nächsten CO "vorzubereiten". Die Hardware wurde so modifiziert, daß diese Möglichkeit nun besteht. Das Aufrufen der Subroutine PCLOCK ermöglicht nun die Vorbereitung eines neuen Wertes, LCLOCK setzt diesen Wert wie bisher sofort.

3. MEMORY ERWEITERUNG

Bisher standen an der Hybridrechenanlage nur 32 K Hauptspeicher à 16 Bit zur Verfügung. Dies führte bereits zu spürbaren Einschränkungen des Multiprogrammbetriebs, falls größere Programme gerechnet wurden. Darüber hinaus waren wegen der 11 K, die das Betriebssystem benötigt, auch 21 K die obere Schranke für die Größe eines Benutzerprogrammes (bzw. die Maximalgröße des größten Dverlays).

Dieser Engpaß führte zu Überlegungen, den Hauptspeicher auf vorerst 48 K Worte auszubauen. Dabei war in allererster Linie die Kostenfrage zu berücksichtigen.

Der bisherige Hauptspeicher war als Kernspeicher mit 8 Blöcken à 4 K Worte ausgelegt. Eine Kernspeichererweiterung hätte den Umtausch aller Blöcke und die Verwendung von 3 Blöcken à 16 K Worte nach sich gezogen (Kostenpunkt ca. S 500.000.-). Das war finanziell so gut wie keine Lösung. So beschritten wir einen anderen Weg: Hauptspeichererweiterung mit einem Random-Access-Memory (RAM, Halbleiterspeicher). Preis für 16 K Worte: ca. S 100.000.-, d.h. 1/5.

In Zusammenarbeit mit der Firma Steiner in Wien konnte auch das schwierige Timing-Problem gelöst werden, Kernspeicher und Halbleiterspeicher mit den gleichen logischen Signalen anzusprechen und im wahrsten Sinn des Wortes zu einer funktionierenden "Mixed Hardware" zu gelangen. Die Schwierigkeiten ergeben sich daraus, daß das Core Memory beim Lesen die Information im Speicher zerstört, d.h. daß nach jedem Lesezyklus ein Schreibzyklus folgen muß, während beim RAM ein zerstörungsfreies Lesen möglich ist. Diese unterschiedliche Arbeitsweise mußte im Prinzip mit den gleichen Steuersignalen bewerkstelligt werden, um große Umbauten der Zentraleinheit zu vermeiden. Dies führte dazu, daß in unserer Konfiguration das RAM nur mit einer Zykluszeit von 1.2 µsec arbeiten kann (obwohl es nur 500 nsec interne Zugriffszeit besitzt), während die Core-Zykluszeit weiterhin 1 µsec beträgt.

Der derzeitige Hauptspeicher sieht also so aus:

1 - 32 K	Core
33 - 48 K	RAM

Bei der gleichzeitig erfolgten Modifikation des Betriebssystems stellte sich heraus, daß für ein optimal schnelles Laden von Zellen im neuen RAM ein neuer Operationscode im PACER 100 definiert und implementiert werden mußte, die RIL (Reset Multi-Indirect Level) Instruktion. Diese erlaubt eine direkte Adressierung für die nächstfolgende LA und STA Operation im vollen Adressbereich von 16 Bit (65 K), während normalerweise nur 15 Bit (32 K) für Adressen zur Verfügung stehen, da das Adressbit 0 für die mehrfache indirekte Adressierung verwendet wird.

4. MODIFIKATIONEN ZUR VERBESSERUNG DES DURCHSATZES BEI MULTI-USER-BETRIEB

Bei der weiteren Intensivierung des Multi-User-Betriebs im letzten Jahr stellten wir die folgenden Einschränkungen oder Schwächen der Hardware des PACER 100 fest, die einen zeiteffektiven Multiprogrammbetrieb mehr oder weniger behinderten.

a) FPP SAVE Register

Der Digitalrechner besitzt einen Hardware Gleitkommaprozessor (FPP), der die Rechenzeit für Gleitkommaarithmetik erheblich herabsetzt. Leider ist der FPP aber nicht in den Instruktionssatz der CPU integriert, sondern muß mit internen I/O-Befehlen über den Standard I/O Kanal angesprochen werden. Die Systemsoftware hat zwar Zugriff zu den zwei entscheidenden 32-Bit Registern, doch muß für das Retten und Abspeichern dieser Register (wie es bei jedem Multiprogramming-Cycle notwendig wird, falls beide Exekutionslevel den FPP benützen) eine 90 µsec dauernde Softwaresequenz durchlaufen werden, die den Overhead des Multiprocessor Moduls des Betriebssystems bei jedem Cycle verdoppeln würde.

Wir haben deshalb den FPP modifiziert, indem wir für die wichtigsten 32-Bit Register eine Save und Restore Operation auf Registerenebene implementiert haben.

Statt einer Verdopplung des Overheads haben wir in gewissem Sinn eben die Register verdoppelt. Das Retten der FPP-Register geschieht jetzt über einen speziellen I/O-Befehl an den FPP und vergrößert den Overhead nur um ganze 4 μ sec. Die ganze Zusatzhardware konnte auf zwei freien Einschubkarten im FPP untergebracht werden. Strom- und Schaltungsprobleme wurden durch den Aufbau in Low-Power-Schottky Technologie vermieden. Darüber hinaus ist diese Modifikation voll software-kompatibel mit früher erstellten Programmen. Die Einschränkungen für den Multi-User-Betrieb bei Verwendung des FPP fallen hiermit weg.

b) Enddevice

Ein weiteres Problem stellten softwarebedingte "hang-ups" der CPU dar, falls ein Hybridprogramm einen I/O-Befehl zu einer nicht existierenden peripheren Einheit exekutieren wollte. Das tritt häufig bei einer falschen (nämlich zu kurzen) Parameterliste beim Aufruf eines Unterprogramms auf, wo dann eine "Ø" als Operationscode (Record-In Teletype) interpretiert wird. Durch eine Modifikation am Standard I/O Kanal haben wir ein fiktives peripheres Gerät, die "Enddevice", eingebaut. Meldet sich 1Ø msec nach einem I/O-Befehl kein Device Controller mit "ready", dann bedient die "Enddevice" die entsprechende Ready-Line und erzeugt einen Alarminterrupt, der vom Betriebssystem mit einem Programmexit und der Meldung PROGRAM HANG UP beantwortet wird.

c) Magnetband als allgemeiner Massenspeicher

Bisher konnte die Magnetbandstation nur in der bekannten Weise als serieller Datenspeicher verwendet werden, d.h. ein Ändern einzelner Records auf einem beschriebenen Band war ohne Zerstörung der nachfolgenden Information nicht möglich. Daher konnte das Mag Tape nur über spezielle Systemroutinen und nicht als allgemeiner District verwendet werden (z.B. für große Data Plotter Files, etc.). Durch eine Modifikation im EAI-Controller haben wir den von PERTEC vorgesehenen "Edit"-Mode für unsere Magnetbandstation zugänglich gemacht. Seither kann das Tape mit den gleichen Unterprogrammen wie Plattenfiles als District 16 angesprochen werden.

5. Im folgenden soll noch gezeigt werden, wie durch selbstentwickelte Modifikationen I/O Einheiten verbessert bzw. Geldmittel eingespart werden konnten.

- Obwohl die vorhandenen Datensichtgeräte (Videos) über eine verhältnismäßig schnelle Datenleitung (480Ø bd) mit der CPU verbunden sind, ergaben sich bei der Systemkonsole erhebliche Schwierigkeiten, rasch genug die laufenden Systeminformationen auszugeben. Normalerweise kennt das Sichtgerät nur den einfachen 64 Zeichen ASCII Satz und einige einfache CURSDR und EDIT Befehle. Um z.B. den Cursor von der HOME-Position in die Mitte des Bildes zu setzen, waren, selbst bei trickreichster Programmierung, 5Ø Zeichen zu übertragen. Dem erweiterten ASCII Zeichensatz, der normalerweise für Kleinbuchstaben verwendet wird, haben wir eine neue Bedeutung zukommen lassen: die absolute Zeilen- bzw. Spaltenpositionierung des Cursors. Im Falle des oben erwähnten Beispiels wären jetzt nur mehr 2 Zeichen zu übertragen.
- Beim Lochstreifenleser wurde ein automatischer Stop bei einer aufgetretenen "Missed Information" eingebaut, sodaß in keinem Fall mehr Information verloren gehen kann.
- Beim Synchron Interface zur CYBER 74 konnte die Anschaffung zweier Modems vermieden werden, da aufgrund der kurzen Distanz ein selbst eingebauter zusätzlicher Timer die Synchronisationssignale sowohl für den Multiplexer auf der CYBER Seite als auch für den CCM auf der PACER Seite erzeugt. Diese Verbindung läuft seit einem halben Jahr problemlos.

DIE OPTIMIERUNG DER ELASTISCHEN LAGERUNG EINES VERBRENNUNGSMOTORS MIT HILFE DER HYBRIDRECHENANLAGE AN DER TU-WIEN

AUFGABENSTELLUNG

Zur Simulation der Schwingungen eines elastisch gelagerten Verbrennungsmotors, Bild 1, wurde ein Modell erstellt, welches die Optimierung der Auflagerung ermöglicht.

Für die Optimierung waren folgende Gesichtspunkte maßgeblich:

- a) Die sich aus dem Feder-Massen-System ergebenden Eigenschwingungszahlen sollen außerhalb des Betriebsbereiches liegen.
- b) Die in das Fundament abgeleiteten Kräfte sollen möglichst klein sein.
- c) Die durch die Schwingungen angeregten Massenkkräfte von Motoranbauteilen sollen nicht zum Bruch derselben führen.

Nach der digitalen Berechnung der Eigenwerte und Eigenvektoren erfolgte die Lösung des Differentialgleichungssystems für die Schwingungssimulation des Motors im Analogteil. Nach Abklingen des Einschwingvorganges konnten die Bewegungsabläufe verschiedener Motorpunkte, die durch Ortsvektoren dargestellt werden, studiert werden. Maximale Schwingungsamplituden wurden digital ausgelesen und weiterverarbeitet.

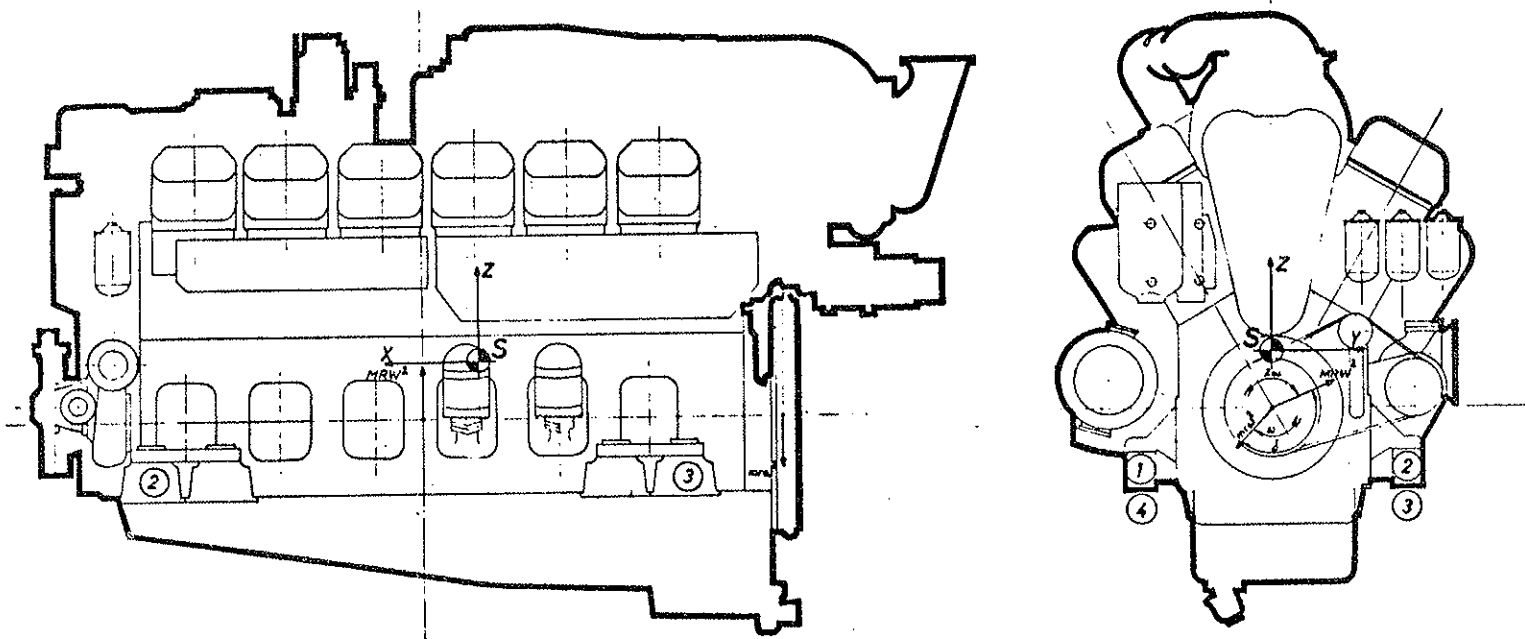


Bild 1
Koordinatensystem, Federanlenkpunkte und Massenkrafterregung
des Verbrennungsmotors

ANNAHMEN

- Der Motor ist ein starrer Körper mit 6 Freiheitsgraden.
- Annäherung der Form durch einen Quader gleichen Gewichtes und gleicher Massenträgheitsmomente. Keine Berücksichtigung der im Betrieb veränderlichen Massenverteilung.
- Hauptträgheitsachsen sind ident mit den System-Achsen und der Koordinatenursprung liegt im Schwerpunkt.
- Abstützung über Federelemente, die beliebig geneigt werden können. Federsteifigkeit in den drei Hauptrichtungen variabel.
- Lineare Federkennung.
- Geschwindigkeitsproportionale Dämpfung umgekehrt proportional zur Erregerfrequenz (charakteristisch für Werkstoff Gummi).
- Schwingungsamplituden klein im Verhältnis zu den Abmessungen.
- Harmonische Erregung durch die freien Massenkräfte der hin- und hergehenden Motorteile (Kolben, Pleuel, etc.) bzw. Unwuchten.

LÖSUNGSWEG

Um die Eigenschwingungszahlen der Motorlagerung zu bilden, werden die Eigenwerte des folgenden Differentialgleichungssystems ^{*)} bestimmt:

$$M\ddot{x} + Cx = D \quad (1)$$

mit der Massenmatrix M und Federmatrix C.

Unter den oben erwähnten Annahmen hat die Massenmatrix Diagonalform:

$$M = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & J_x & 0 & 0 \\ 0 & 0 & 0 & 0 & J_y & 0 \\ 0 & 0 & 0 & 0 & 0 & J_z \end{bmatrix} \quad (2)$$

mit der Körpermasse m und den Trägheitsmomenten J_x , J_y , J_z um die Koordinatenachsen.

Die Federmatrix hat die Blockform

$$C = \begin{bmatrix} C_0 & C_I' \\ C_I & C_{II} \end{bmatrix} \quad (3)$$

Dabei beschreiben die Matrizen:

$$\begin{aligned} C_0 &= \sum C_i && \text{Federungsmomente nullten Grades} \\ C_I &= \sum Y_i C_i && \text{Federungsmomente 1. Grades} \\ C_I' &= \sum (Y_i C_i)' && \text{Transponierte zu } C_I \\ C_{II} &= \sum Y_i (Y_i C_i)' && \text{Federungsmomente 2. Grades} \end{aligned}$$

mit dem schiefsymmetrischen Axiator

$$Y_i = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} i \quad (4)$$

*) s.a.: Appel, H.: Freie und erzwungene Schwingungen von elastisch gelagerten Fahrzeugmotoren; Automobil-Industrie 3/68 und 4/68.

Werden die Gummi-Federelemente gegen das System (x,y,z) gedreht, wird eine Ähnlichkeitstransformation notwendig, um die für die Rechnung benötigten Federmatrizen in (x,y,z) zu erhalten. Die Federsteifigkeiten in den Haupt-richtungen (ξ,η,ζ) des Federelementes i sind durch die Matrix \overline{C}_i beschrieben:

$$\overline{C}_i = \begin{bmatrix} C_\xi & 0 & 0 \\ 0 & C_\eta & 0 \\ 0 & 0 & C_\zeta \end{bmatrix}_i \quad (5)$$

Die Drehungen des Feder-Hauptachsensystems (ξ,η,ζ) gegen (x,y,z) beschreibt die orthogonale Transformationsmatrix T , welche die Richtungscosini von Bezugskoordinaten und gedrehten Koordinaten enthält.

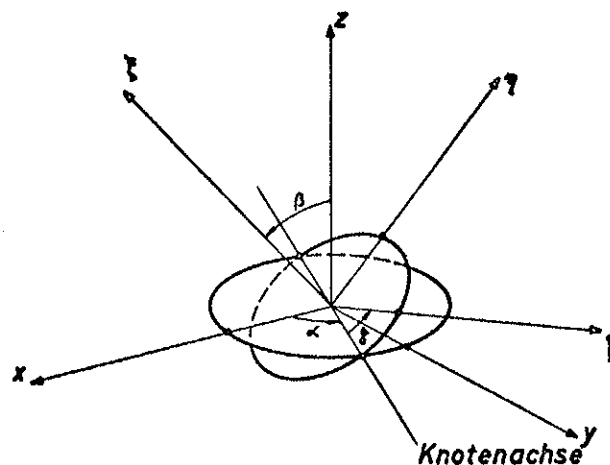


Bild 2
Eulersche Winkel α, β, γ

Die gegenseitige Lage von gedrehten Koordinatensystemen ist durch drei unabhängige Größen, die drei Eulerschen Winkel α, β, γ bestimmt, Bild 2.

Die Gesamtdrehung des Systems (x,y,z) in das System (ξ,η,ζ) kann als Aufeinanderfolge von drei Einzeldrehungen dargestellt werden:

- 1) Drehung mit α um die z -Achse: T_α
- 2) Drehung mit β um Knotenachse K : T_β
- 3) Drehung mit γ um die ζ -Achse: T_γ

$$T_\alpha = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad T_\beta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix};$$

$$T_\gamma = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Für die Transformationsmatrix oder Gesamtdrehung wird

$$T = T_{\alpha} \cdot T_{\beta} \cdot T_{\gamma} \quad (7)$$

sowie

$$C = T' \cdot \bar{C} \cdot T \quad (8)$$

mit der Transponierten von T, T'.

Um die erzwungenen Schwingungen des Motors zu simulieren, muß in Gleichung (1) noch die Dämpfung und Erregung berücksichtigt werden. Das Differentialgleichungssystem wird daher:

$$M\ddot{X} + D\dot{X} + CX = R \quad (9)$$

Aufgrund der Werkstoffeigenschaften von Gummi kann die Dämpfungsmatrix D in erster Näherung proportional zur Federsteifigkeitsmatrix C angenommen werden. Der Proportionalitätsfaktor ist umgekehrt proportional zur Erregerfrequenz ω , womit

$$D = \frac{\theta}{\omega} \cdot C \quad (10)$$

wird.

Der mechanische Verlustwinkel θ ist abhängig von der Shore-Härte des Gummis.

In der einspaltigen Erregungsmatrix R werden die ersten drei Elemente durch die im Schwerpunkt angreifenden Erregerkräfte, die zweiten drei Elemente durch die Erregungsmomente gebildet. In Bild 1 sind der mit der doppelten Frequenz umlaufende Vektor der freien Massenkräfte 2. Ordnung, $M\omega^2$, sowie der mit einfacher Frequenz und einem Phasenwinkel δ versehene Vektor einer an der Schwungscheibe angreifenden Unwucht $m\omega^2$ eingetragen. Die y- und z-Komponenten dieser Vektoren ergeben die entsprechenden Elemente der Matrix R.

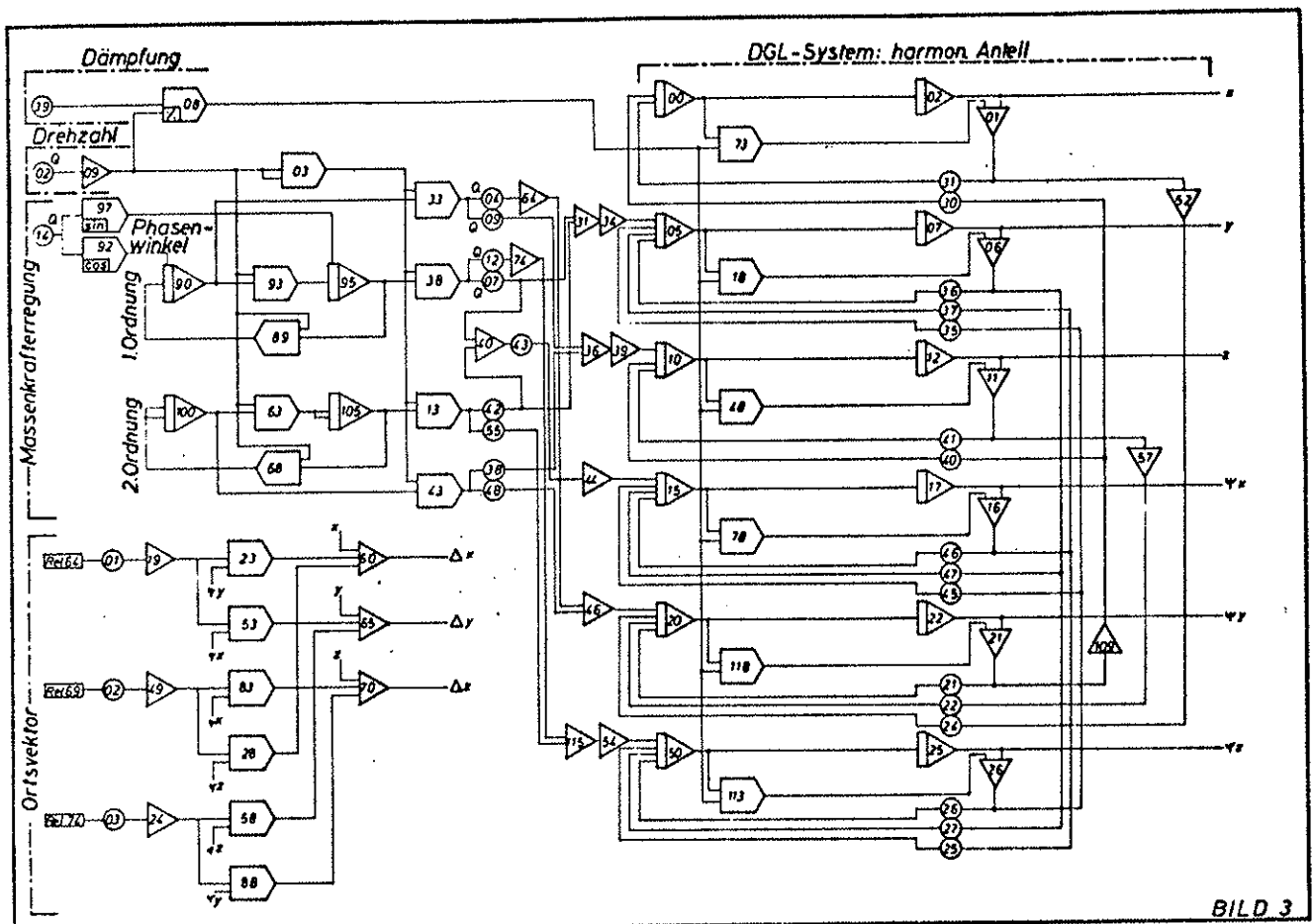


BILD 3

Bild 3

RECHENSCHALTUNG

Der in Bild 3 gezeigte Schaltplan stellt das Ersatzsystem für den Einmassenschwinger mit 6 Freiheitsgraden dar. Dämpfung und Massenkrafterregung sind berücksichtigt.

Zur Bestimmung der Schwingungsamplituden einzelner Motorpunkte wurde das äußere Produkt aus Ortsvektor und Schwerpunktsvektor überlagert, wodurch die Amplituden $\Delta x, \Delta y, \Delta z$ eines Punktes bestimmt werden konnten.

Mit der Maximum-Verfolgungsschaltung des Bildes 4 können nach Abklingen des Einschwingzustandes die Größtwerte, z.B. einer Motorprätzenbewegung, ausgelesen und in den Digitalteil des Hybridrechners übertragen werden. Dort werden sie mit den entsprechenden Federsteifigkeitswerten multipliziert und so erhält man die ins Fundament (bzw. Motorrahmen) eingeleiteten dynamischen Kräfte.

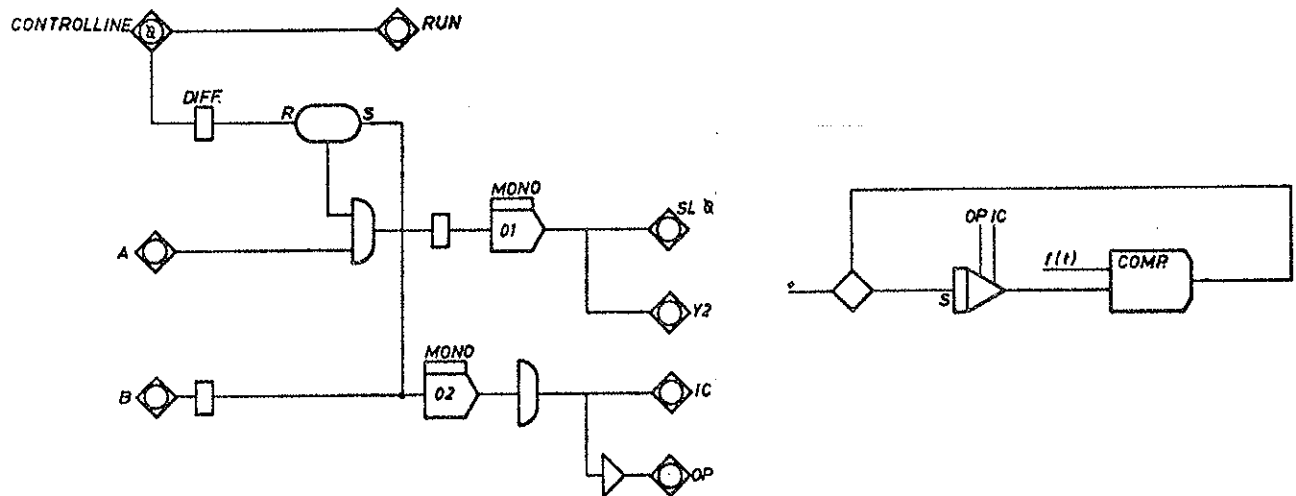


Bild 4
Maximumsuchschaltung und Logikschaltplan

ERGEBNISSE

Die folgenden Bilder stellen eine kleine Auswahl der gerechneten Varianten dar. Über dem Neigungswinkel β der Federelemente sind für eine Erregerdrehzahl von $n=1500$ U/min die maximalen Schwingungsamplituden in y-Richtung für den Schwerpunkt und zwei Ortsvektoren dargestellt. Vektor 3 ist die Bewegung des Einspritzpumpen-Reglers, Vektor 4 jene der Motorprätze Nr. 4.

Die Federsteifigkeiten wurden in x- und y-Richtung gleich groß angenommen, die Steifigkeit in z-Richtung variiert.

Für Vektor 4 wurden auch die ins Fundament eingeleiteten dynamischen Kräfte dargestellt.

Folgende Symbole gelten für die Diagramme:

	$F_z = 100$ kp/mm
	150 "
	200 "
	250 "
	300 "

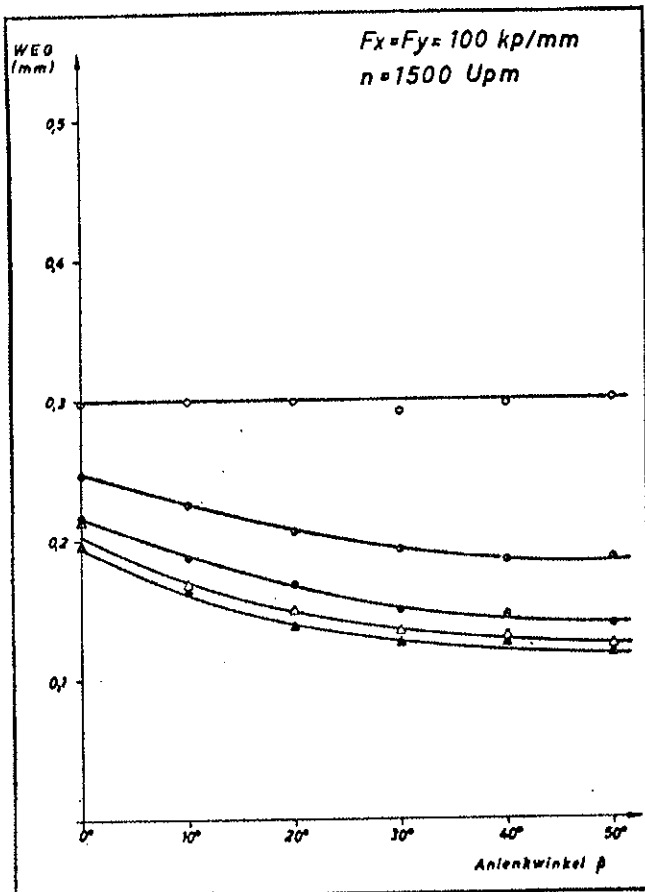


Bild 5

Schwerpunktsbewegung in Y-Richtung

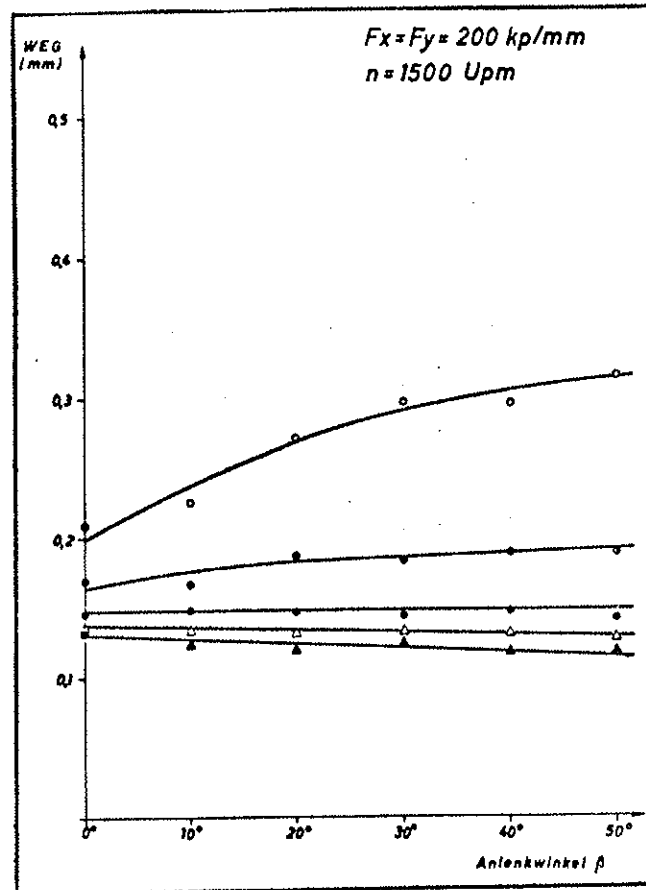


Bild 6

Schwerpunktsbewegung in Y-Richtung

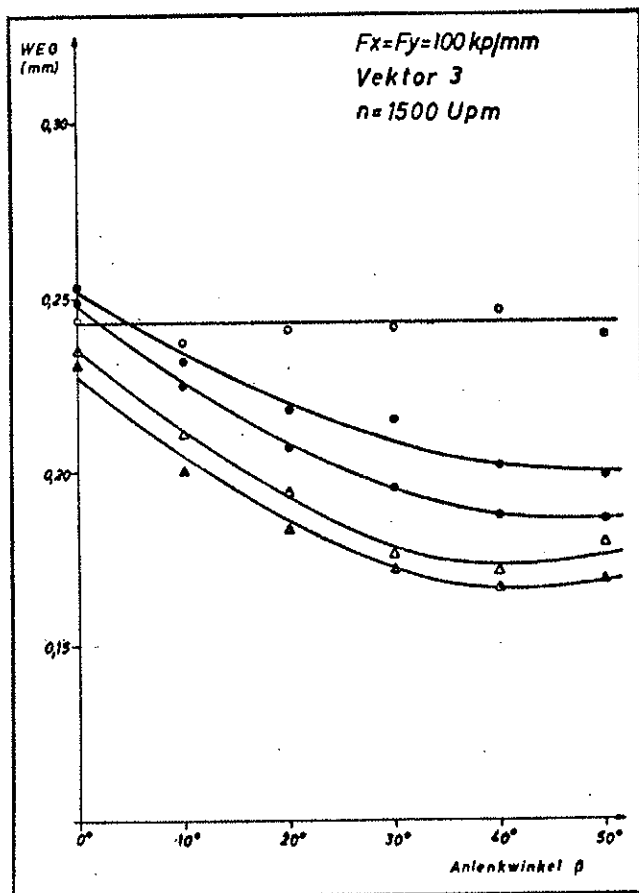


Bild 7

Bewegung in Y-Richtung

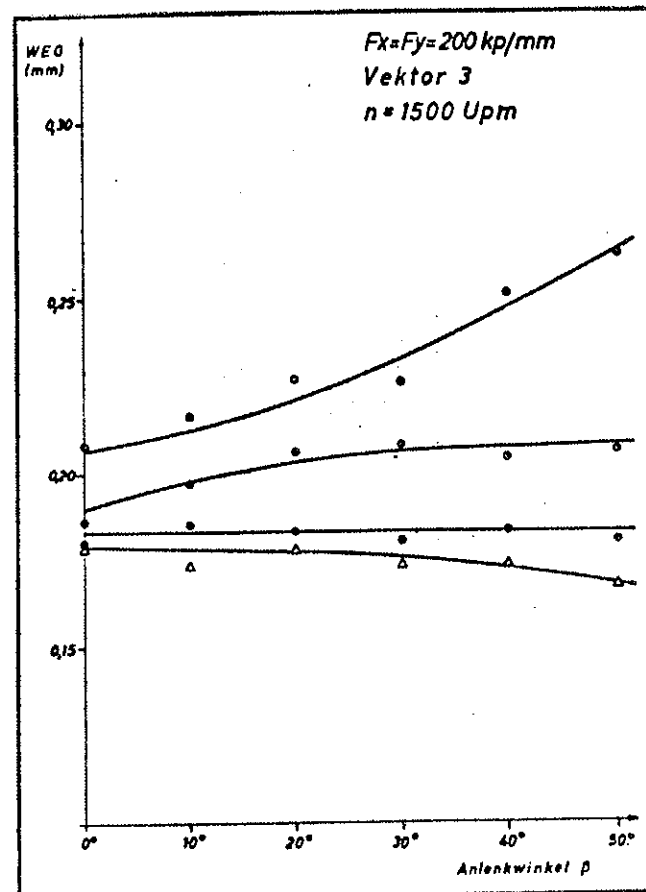


Bild 8

Bewegung in Y-Richtung

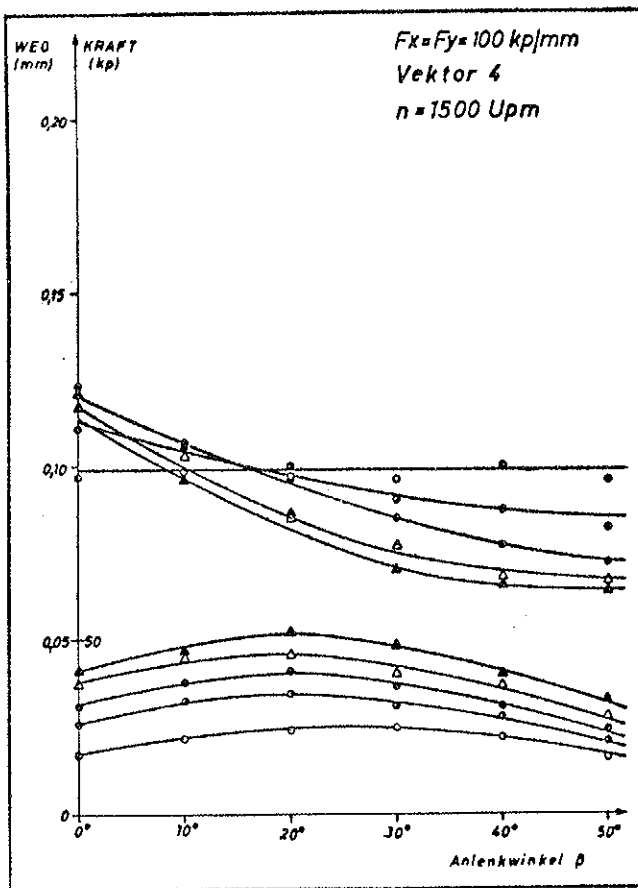


Bild 9

Bewegung in Y-Richtung

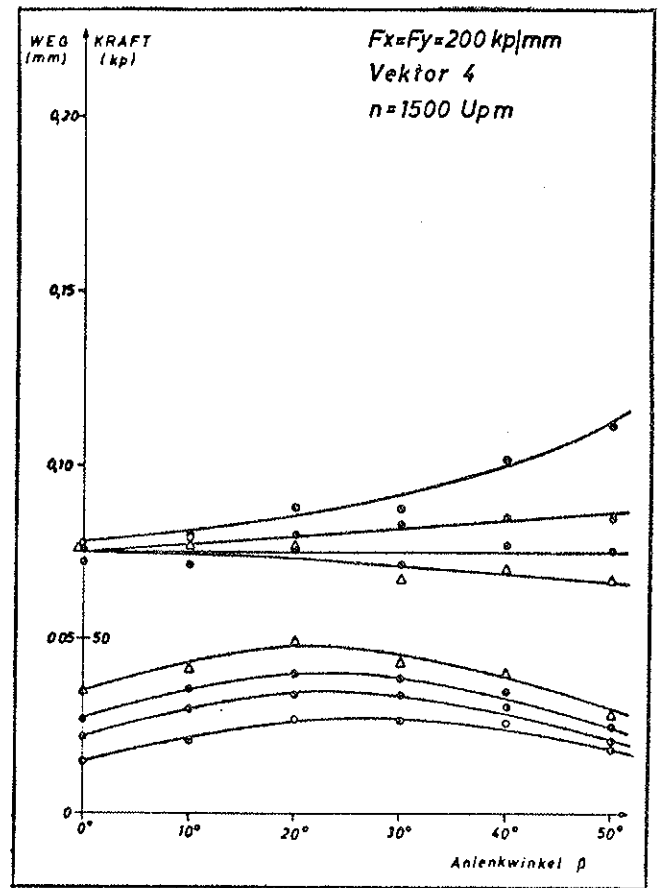


Bild 10

Bewegung in Y-Richtung

Aus den Bildern ist zunächst der triviale Fall zu erkennen, daß ein Schiefstellen der Federelemente bei gleich großen Steifigkeiten in allen drei Richtungen keinen Einfluß auf die maximalen Schwingungsamplituden hat. Weichere Federn in der Hochachse (z) bringen eine Verschlechterung der Ausschläge (Bilder 6,8 und 10), jedoch eine Verkleinerung der im Federelement wirkenden Querkräfte. Ferner sieht man, daß mit steigendem Neigungswinkel β der Einfluß (bei $F_z > F_x$) der Hochsteifigkeit abnimmt.

Dipl.-Ing. H. Filipisky
 Dipl.-Ing. H. Aibel
 Simmering-Graz-Pauker A.G.

Dipl.-Ing. F. Rattay
 Institut für Technische Mathematik
 Technische Universität Wien

MODELL ZUR SIMULATION DER ALTERSABHÄNGIGKEIT DER DOSIS IM ATEMTRAKT BEI DER INHALATION NATÜRLICH RADIOAKTIVER NUKLIDE (TEIL 1)

1. EINLEITUNG

Das in der Luft von Wohn- und Arbeitsräumen stets vorhandene Radon (^{222}Rn) und seine kurzlebigen Zerfallsprodukte verursachen in Teilen des Atemtraktes die höchste Strahlenbelastung, die im menschlichen Körper bei der Inhalation natürlich radioaktiver Nuklide auftreten kann (1,2,3). Gerade im Zusammenhang mit der Diskussion über die zusätzliche Strahlenbelastung beim Betrieb von Kernkraftwerken kommt der Frage der Inhalation der in unserer Umwelt stets vorhandenen natürlichen Radionuklide erhöhte aktuelle Bedeutung zu.

Die in der Atemluft befindlichen und zum größten Teil an das vorhandene Aerosol angelagerten Zerfallsprodukte werden im gesamten Atemtrakt wie auf einem Filter abgelagert und akkumuliert, wobei wegen des radioaktiven Zerfalls bei Dauerinhalation für jedes Nuklid charakteristische Gleichgewichtswerte erreicht werden. Die örtliche Verteilung der Nuklide im Atemtrakt und damit auch die Dosisverteilung hängt von der Art des Aerosols und von verschiedenen physiologischen Parametern ab.

Meist werden zur Berechnung der Dosis im Atemtrakt entsprechend den Vorstellungen der ICRP (International Commission on Radiological Protection)

- a) das Depositions- und Retentionsmodell der ICRP-Task Group on Lung Dynamics (4)
- b) die physiologischen Daten des ICRP-Reference Man (5)

verwendet.

Durch die Bezugnahme auf einen erwachsenen Referenz-Menschen werden jedoch die Änderungen des wachsenden Organismus und die daraus resultierende Altersabhängigkeit der Dosis nicht berücksichtigt. Kinder und Jugendliche sind aber für den Strahlenschutz einer Bevölkerung von ganz besonderer Bedeutung, da der wachsende Organismus im allgemeinen strahlensensibler ist.

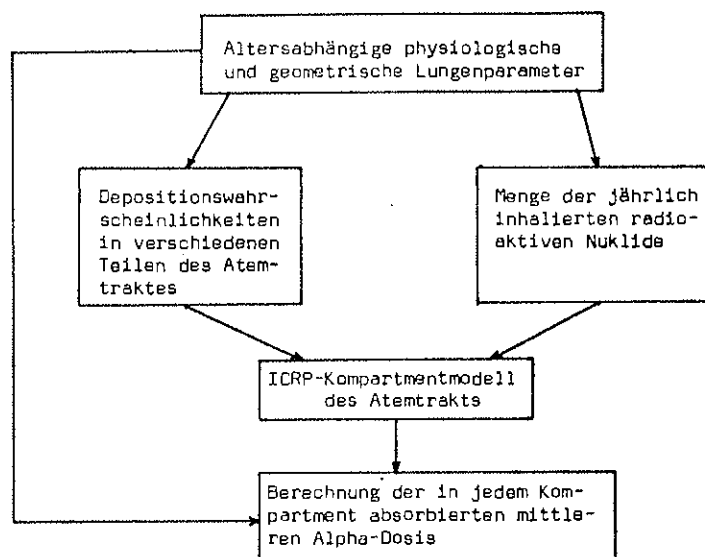


Abb.1 Flußdiagramm zur Berechnung der Dosis im Atemtrakt als Funktion des Lebensalters

Abb.1 zeigt die einzelnen Schritte, die zur Berechnung der Dosen in den verschiedenen Teilen des Atemtrakts in Abhängigkeit vom Alter notwendig sind.

Auf der Grundlage des ICRP-Kompartimentmodells, das infolge seiner Einfachheit gerade für Strahlenschutz Zwecke besonders geeignet ist, soll der Einfluß altersabhängiger Lungenparameter auf die Dosis im Atemtrakt untersucht werden.

2. KOMPARTIMENTMODELL DES ATEMTRAKTS

Zur einfacheren mathematischen Beschreibung wird der Atemtrakt in drei Bereiche, sogenannte Kompartments, unterteilt, die sich hinsichtlich ihrer anatomischen und funktionellen Eigenschaften unterscheiden:

- 1) Nasopharyngealer Bereich (N-P), der sich vom Nasen-Rachenraum bis zum Kehlkopf erstreckt.
- 2) Tracheobronchialer Bereich (T-B), bestehend aus der Trachea und dem mit einem Ciliarepithel versehenen Teil des Bronchialbaums.
- 3) Pulmonärer Bereich (P), der die respiratorischen Bronchioli und die gesamte Alveolarregion umfaßt.

Bei der Kinetik der Ausscheidung der abgelagerten Nuklide aus diesen Kompartments muß die chemische Verbindung des Nuklids berücksichtigt werden, da davon Art und Geschwindigkeit der Transportprozesse entscheidend abhängen. Die natürlich radioaktiven Nuklide, angelagert an das natürliche Aerosol, zählen zu den gut löslichen Verbindungen und werden deshalb der entsprechenden Retentionsklasse, der Klasse D, zugeordnet. In Abb.2 ist das ICRP-Lungenmodell in seiner revidierten Form für Aerosole der Klasse D dargestellt.

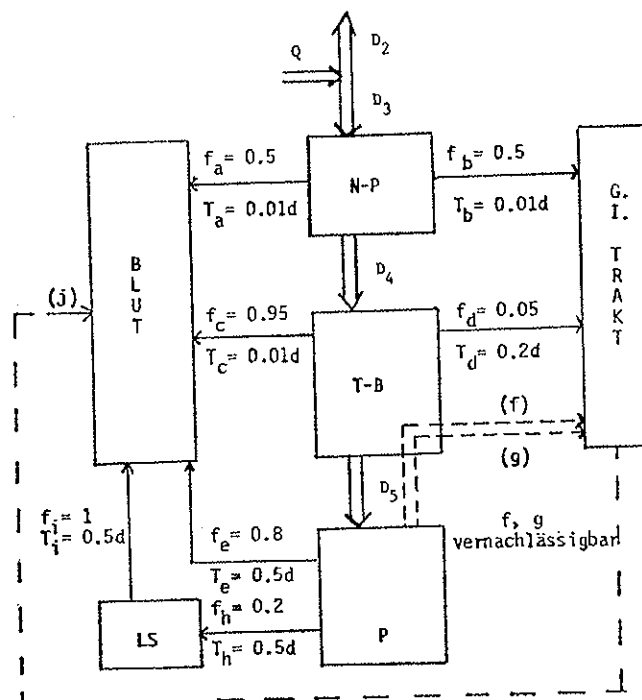


Abb.2 ICRP-Lungenmodell für Aerosole der Klasse D

D_3, D_4 und D_5 sind die Bruchteile der insgesamt inhaliierten Aktivität Q , die in den einzelnen Komponenten abgelagert werden, während D_2 den exhaliierten Anteil angibt. Die Buchstaben a-j kennzeichnen die verschiedenen Transportwege in das Blut, den Magen-Darm-Trakt und das Lymphsystem. Für jeden Weg ist die biologische Halbwertszeit T und derjenige Teil f des abgelagerten Materials, der über diesen Weg abtransportiert wird, angegeben. Die beiden Transportwege f und g vom P-Bereich über den T-B-Bereich in den Magen-Darm-Trakt können für Aerosole der Klasse D vernachlässigt werden*).

Im Falle der Inhalation von Nukliden einer Zerfallsreihe muß neben radioaktivem Zerfall, Deposition und Abtransport noch die radioaktive Nachbildung berücksichtigt werden. Überdies ändern sich bei der Untersuchung der Altersabhängigkeit die Depositionswahrscheinlichkeiten D_3, D_4 und D_5 sowie die jährlich inhaliierte Aktivität Q als Funktion des Lebensalters.

Mathematisch kann die zeitliche Änderung der Aktivitäten in den einzelnen Kompartments durch ein System linearer Differentialgleichungen beschrieben werden. Zur Vereinfachung des Gleichungssystems werden die Kompartments noch weiter in Subkompartments, entsprechend der Anzahl der Transportwege, unterteilt (6).

Als Beispiel kann die zeitliche Änderung der Aktivität im Tracheobronchialbereich durch folgende Differentialgleichungen dargestellt werden:

$$1. \quad \frac{dQ_c^n}{dt} = -Q_c^n (\lambda_c + \lambda_r^n) + f_c (Q_c^{n-1} + Q_d^{n-1}) \lambda_r^n + f_c \sum_{i,j} D_{4,i}^n Q_{i,j}^n$$

$$2. \quad \frac{dQ_d^n}{dt} = -Q_d^n (\lambda_d + \lambda_r^n) + f_d (Q_c^{n-1} + Q_d^{n-1}) \lambda_r^n + f_d \sum_{i,j} D_{4,i}^n Q_{i,j}^n$$

$$Q_{TB}^n = Q_c^n + Q_d^n$$

mit $Q_{TB}^n = AMV \cdot k_i \cdot T_{i,j} \cdot C_j^n$

- $Q_{c(d)}^n$... Aktivität des Nuklides n im Subkompartment c (d)
- $\lambda_{c(d)}$... biologische Zeitkonstante für den Transportweg c (d)
- λ_r^n ... radioaktive Zerfallskonstante des Nuklides n
- $f_c(d)$... Teil der Aktivität, die über den Weg c (d) abtransportiert wird
- $D_{4,i}^n$... Depositionswahrscheinlichkeit des Nuklides n im Tracheobronchialbereich bei physischer Aktivität i
- $Q_{i,j}^n$... jährlich inhaliierte Aktivität des Nuklides n bei physischer Aktivität i am Ort j
- AMV ... Atemminutenvolumen im Ruhezustand ($i=1$)
- k_i ... Skalierungsfaktor des AMV für verschiedene physische Aktivitäten i
- $T_{i,j}$... Aufenthaltszeit am Ort j bei physischer Aktivität i
- C_j^n ... Konzentration des Nuklides n in der Atemluft am Ort j

*) Für den Transportweg e ist jedoch nach den Untersuchungen von Pohl und Pohl-Rüling (3,11) die Halbwertszeit im ICRP-Modell viel zu groß angenommen.

Dabei gilt folgendes:

Nuklidart:	n=0	Rn	Anfangsbedingung:
	n=1	RaA	für n=0 sind sämtliche $Q_i = 0$
	n=2	RaB	
	n=3	RaC-RaC'	

physische Aktivität:

i=1	Ruhe
i=2	leichte Arbeit
i=3	schwere Arbeit

mit den Skalierungsfaktoren: $k_1=1, k_2=3, k_3=6$

Ort der Inhalation:

j=1	Raumluft
j=2	Freiluft

(für i=1 würde immer j=1 gesetzt)

Für die übrigen beiden Kompartments lassen sich analoge Gleichungen aufstellen. Man erhält somit für 6 Subkompartments und 3 Nuklidarten 18 Differentialgleichungen gleichen Aufbaus, die sich nur durch verschiedene Parameterwerte unterscheiden. Damit bietet sich ein iteratives Verfahren für die Lösung des Gleichungssystems an.

Aus den Lösungen für die Q_i und den Massen der Kompartments für jedes Lebensalter kann die Dosis berechnet werden. Dabei muß praktisch nur die Alpha-Strahlung berücksichtigt werden, die infolge ihrer hohen Energie und kurzen Reichweite eine hohe Dosis an das umgebende Gewebe abgibt. Die Beta-Strahlung des RaB kann dabei gegen die Alpha-Strahlung des RaA und des RaC' (im Gleichgewicht mit RaC) vernachlässigt werden.

Unter der Annahme der homogenen Bestrahlung eines Kompartments kann die Dosis nach folgender Formel berechnet werden:

$$D_{\alpha} [\text{mrad a}^{-1}] = 18.69 \frac{Q_1 [\text{pCi}] E_{\alpha} [\text{MeV}]}{m [\text{g}]}$$

mit $E_{\alpha} (\text{RaA}) = 6.0$

$E_{\alpha} (\text{RaC}') = 7.68$

In Wirklichkeit ist die Dosisverteilung innerhalb eines Kompartments keineswegs homogen. Zum Beispiel erhalten die höchsten Dosen wegen der flächenhaften Deposition und der geringen Reichweite der α -Strahlung die Basalzellen des Bronchialepithels der segmentalen und subsegmentalen Bronchien. Die Berechnung der von diesen Zellen erhaltenen Dosis ist einer späteren Arbeit vorbehalten.

3. SCHALTUNGS- UND PROGRAMMBESCHREIBUNG

Für die gegebene Problemstellung bietet ein Hybridrechner wesentliche Vorteile, da er sowohl Lösungen der Differentialgleichungen für alle Zeiten auf einfache Weise, als auch eine rasche und übersichtliche Änderung einzelner Parameter ermöglicht.

Zur Durchführung der Berechnungen wurde das an der Hybridrechenanlage zur Verfügung stehende Skalierungsprogramm zur automatischen optimalen Skalierung benützt (7). Die Skalierung der im Modell auftretenden Rechengrößen

wurde dadurch wesentlich erleichtert, obwohl diese aufgrund der physikalischen Konstanten oft um einige Zehnerpotenzen differieren.

Da die Berechnung der NP-,TB- und P-Bereiche ohne Datenfluß zwischen den Kompartments möglich ist, die dafür notwendigen Anlogschaltungen völlig identisch sind und sich rechentechnisch nur durch andere Übergangsparameter und Tabellen für die Depositionswahrscheinlichkeiten und die inhalierten Aktivitäten unterscheiden, wurde die Anlogschaltung nur einmal für die synchrone Auswertung von $n=1,2,3$ aufgebaut (Abb.3).

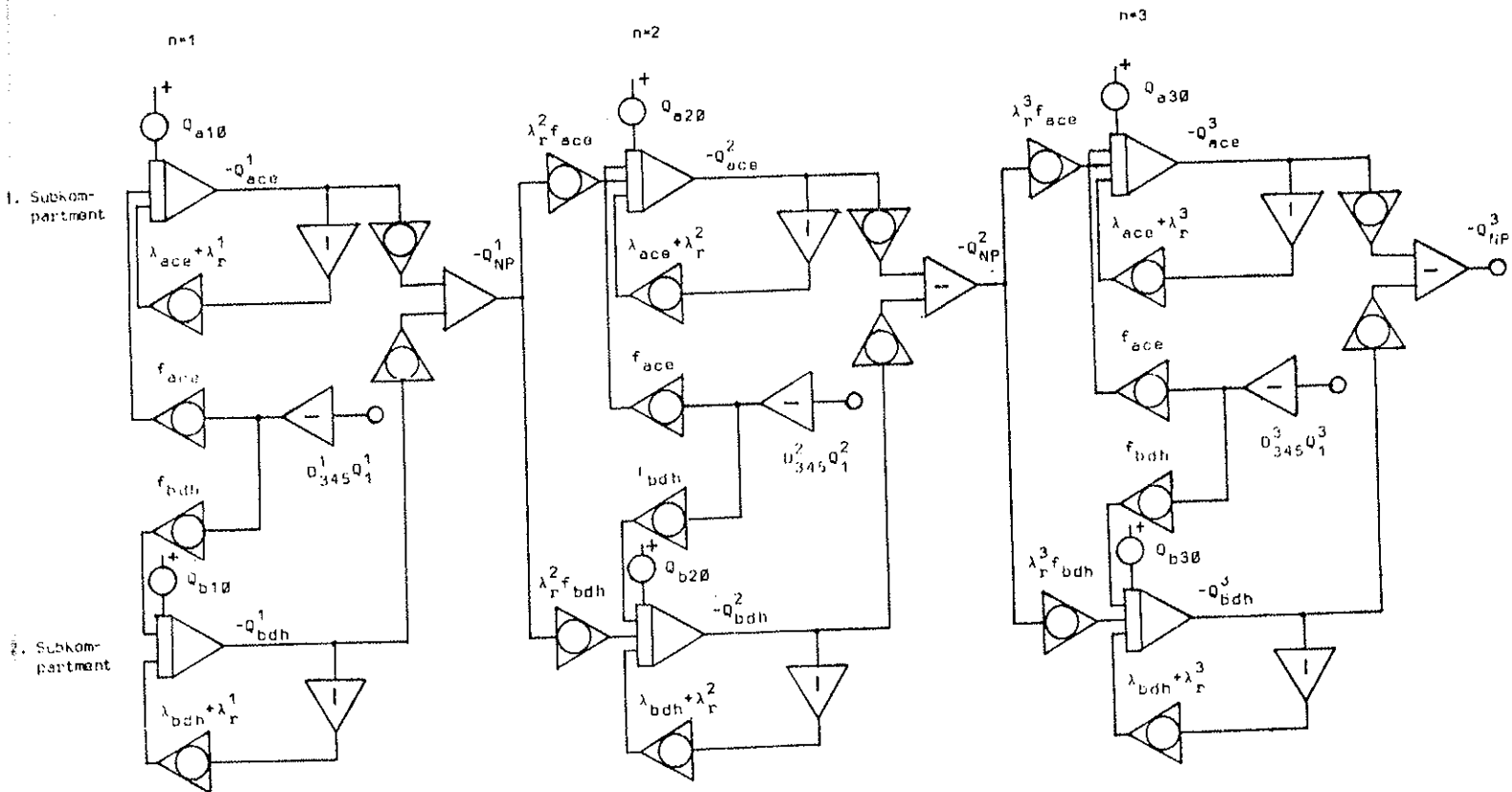


Abb.3 Analoger Schaltplan des Modells

Der gleichzeitige Aufbau für je zwei Subkompartments und alle 3 Nuklide war deshalb vorteilhaft, da in jeder Differentialgleichung mit dem radioaktiven Zerfall eines Nuklids der gleichzeitige Aufbau eines Folgeproduktes verbunden ist, und somit einen direkten Transfer zwischen den einzelnen Nukliden nötig macht. Um rechnerisch von einem Kompartiment zum anderen zu wechseln, ist nur der Austausch der Tabellen und Parameter-sätze notwendig. Die Absicht, die pro Kompartiment nötigen 3 Tabellen für die 3 Nuklide mit Hilfe von DCFG's (digital einstellbaren Funktionsgebern) mit leichter Umsetzbarkeit auf andere Tabellen synchron zu erzeugen, konnte nicht realisiert werden, da am Rechner derzeit nur 2 DCFG's zur Verfügung stehen.

Es wurde daher die Möglichkeit untersucht, die Tabellen über den direkten Datenkanal zu übertragen. Das ist mit Hilfe einer im Skalierungsprogramm bereits vorgesehenen Dummy-Subroutine TRANSR möglich, die durch eine vom Benutzer frei programmierte Routine ersetzt werden kann und jedesmal vor dem Start des Analogrechners durchlaufen wird.

In dieser Routine müssen somit die Datenkarten der Tabellen gelesen, die Tabellen vorbereitet und interpoliert und der direkte Speicherkanal initialisiert werden.

Aus Gründen der Effektivität im Multiprogramming Betriebssystem ist die maximale Transferzeit für einen Transfer am direkten Speicherkanal für den Hybridbenutzer automatisch beschränkt. Da das Skalierungsprogramm über Zeittransformationen aber Werte für die Maschinenzeit wählen kann, welche zu einer Überschreitung der maximal erlaubten Transferzeit führen, könnte es dadurch zu einem Abbruch des Übertragens der Tabellen für die Nuklide kommen.

Aus diesem Grund wurden Interrupts zum Übertragen der Daten verwendet. In der Subroutine TRANSR, welche anstelle der DUMMY-Subroutine nun vom Benutzer problemspezifisch geschrieben und geladen wurde, werden folgende Operationen durchgeführt:

- o Einlesen der Datenkarten für die Tabellen ⁺⁾
- o Ausdruck der gelesenen Tabellen zu Kontrollzwecken ⁺⁾
- o Multiplikation der Q-Tabellen mit den Depositionswahrscheinlichkeiten und Ausdruck dieser Ergebnisse zu Kontrollzwecken ⁺⁾
- o Einlesen des Status der Sense-Switches (Wahl des Kompartments)
- o Interpolation der nun in Tabellenform vorhandenen, an den Analogrechner zu Übertragenden Produktfunktion nach Akima (nur beim ersten Aufruf von TRANSR pro Programm-
lauf, oder, wenn ein anderes Kompartment mit den zugehörigen anderen Tabellen angewählt wurde)
- o Auswertung der Interpolationsfunktion für jene Zahl von Stützwerten, die der vom Skalierungsprogramm gewählten Maschinenzeit entspricht (nur beim ersten Aufruf von TRANSR pro Programmlauf, oder, wenn ein anderes Kompartment gewählt wurde, oder, wenn bei gleichbleibendem Kompartment durch (automatische) Wahl einer anderen Zeittransformation die Stützstellenanzahl und damit die Zahl der notwendigen interpolierten Werte verändert wurde)
- o Nach jeder neuen Akima-Interpolation werden automatisch die Maxima der neu berechneten Werte bestimmt, die Funktionswerte werden damit normiert und auf einem Arbeitsfeld für die Datenübertragung bereitgestellt
- o Die ermittelten Skalierungsfaktoren der Tabellen werden an das Skalierungsprogramm über Parameter weitergegeben und dort sofort berücksichtigt
- o In TRANSR werden Interrupts initialisiert. Da die Datenübertragung aber erst nach Verlassen von TRANSR und zugleich mit dem Beginn des Analoglaufes beginnen darf, werden sie in TRANSR durch einen besonderen Befehl verboten und erst beim Verlassen der Routine erlaubt

Wenn das kontrollierende Skalierungsprogramm unmittelbar nach Beendigung der Routine TRANSR den analogen Rechengang startet, wird das Arbeitsfeld mit den bereitgestellten Tabellenwerten über die Interrupts so abgearbeitet, daß zu jedem Zeitpunkt drei Werte entsprechend den drei Tabellen für die Nuklide $n=1,2,3$ eines Kompartments übertragen werden. Da das Arbeitsfeld linear abgearbeitet wird, müssen die Tabellen dort nach Art eines dreidimensionalen Feldes abgespeichert werden.

In der im Skalierungsprogramm bereits enthaltenen, aber ebenfalls speziell modifizierten Routine USERHP, mit der die Ausgänge beliebiger Verstärker geplottet werden können, können die gezeichneten Daten dann wahlweise auch unter einem frei bestimmbar File-Namen abgespeichert werden. Diese Files werden von einem Hilfsprogramm gelesen, welches vorher die altersabhängigen Massen in den Kompartments Akima interpoliert, und über interpolierte Massenwerte und abgespeicherte Werte des analogen Laufes wird die mittlere absorbierte α -Dosis als Funktion des Alters errechnet.

⁺⁾ nur beim allerersten Aufruf von TRANSR pro Programmlauf

4. DEPOSITIONSWAHRSCHEINLICHKEIT UND INHALIERTE AKTIVITÄT

Altersabhängige Daten von Lungenparametern dienen als Ausgangsbasis für die Berechnung der Depositionswahrscheinlichkeiten in den verschiedenen Bereichen des Atemtrakts und der jährlich inhaliierten Aktivität. Es gibt darüber aber nur unvollständige Angaben in der vorhandenen Literatur, die überdies oft große Differenzen aufweisen (5,8). Waren nicht genügend Daten vorhanden, so wurde angenommen, daß die Lunge eines Kindes eine maßstäbliche Verkleinerung der Lunge eines Erwachsenen darstellt. Im allgemeinen zeigen die Kurven der meisten Lungenparameter als Funktion des Alters folgenden Verlauf: steiler Anstieg während der ersten 10 Lebensjahre, gefolgt von einem langsamen Übergang in einen Sättigungswert zwischen 20 und 30 Jahren. Auf diese Weise wurde ein altersabhängiges Lungenmodell nach Landahl (9) aufgestellt, das Funktionen für alle notwendigen Parameter, von den geometrischen Dimensionen der Lunge bis zu den Atmungsgrößen, angibt. In letzter Zeit sind wir dazu übergegangen, das Landahl-Modell durch das detailliertere Weibel-Modell (10) zu ersetzen.

Aufgrund der verschiedenen Geometrie des Atemtraktes und der pro Zeiteinheit eingeatmeten Luftmenge eines Kindes im Vergleich zum Erwachsenen ändern sich auch die Strömungsverhältnisse in den Atemwegen, was zu altersabhängigen Depositionswahrscheinlichkeiten führt. Die Depositionswahrscheinlichkeiten für die verschiedenen Regionen wurden nach Landahl (9) mit einigen Verbesserungen berechnet. Die Ergebnisse dieser Berechnungen zeigten, daß sich die Depositionswahrscheinlichkeiten trotz der großen Unterschiede infolge des Wachstums in den ersten 30 Lebensjahren nicht wesentlich änderten (11).

Für die Bestimmung der inhaliierten Aktivität wurde ein typischer altersabhängiger Tagesablauf definiert. Die individuelle physische Aktivität, die das Atemminutenvolumen stark beeinflußt, wurde in Ruhe, leichte und schwere Arbeit unterteilt. Überdies wurde die Aufenthaltszeit in Räumen oder im Freien wegen der unterschiedlichen Nuklidkonzentrationen berücksichtigt. Auf diese Weise wurden Aktivitätsfunktionen für 2 Extremfälle[†]) berechnet (11).

Die mit einem Kleincomputer der Type Digital PDP-8/F am Institut für Physik berechneten Werte für die Depositionswahrscheinlichkeiten und die jährlich inhaliierten Aktivitäten wurden dann in Tabellenform in das Gleichungssystem des ICRP-Kompartimentmodells am Hybridrechner eingesetzt.

5. ERGEBNISSE

Die hier als Beispiel dargestellten Ergebnisse wurden unter Annahme folgender Referenzatmosphäre erhalten:

Raumluft: Rn-Konzentration 0.1 pCi/l
RaA:RaB:RaC=0.9:0.6:0.4

Freiluft: Rn-Konzentration 0.1 pCi/l
RaA:RaB:RaC=0.9:0.5:0.3

Physische Aktivität des Erwachsenen: leichte Arbeit (Beamter)
AMAD (activity median aerodynamic diameter) der an das Aerosol angelagerten Nuklide: 0.2 μm
AMAD der freien Atome (8.5 % RaA, 1.0 % RaB, 0.6 % RaC) : 0.001 μm

Abb.4 zeigt die Lösungen des Differentialgleichungssystems für den Tracheobronchial- und den Pulmonär-Bereich.

[†]) Minimum und Maximum der inhaliierten Aktivitäten, die wir bei unseren Untersuchungen über die Umweltradioaktivität in Wohn- und Arbeitsräumen in Salzburg gefunden haben (13)

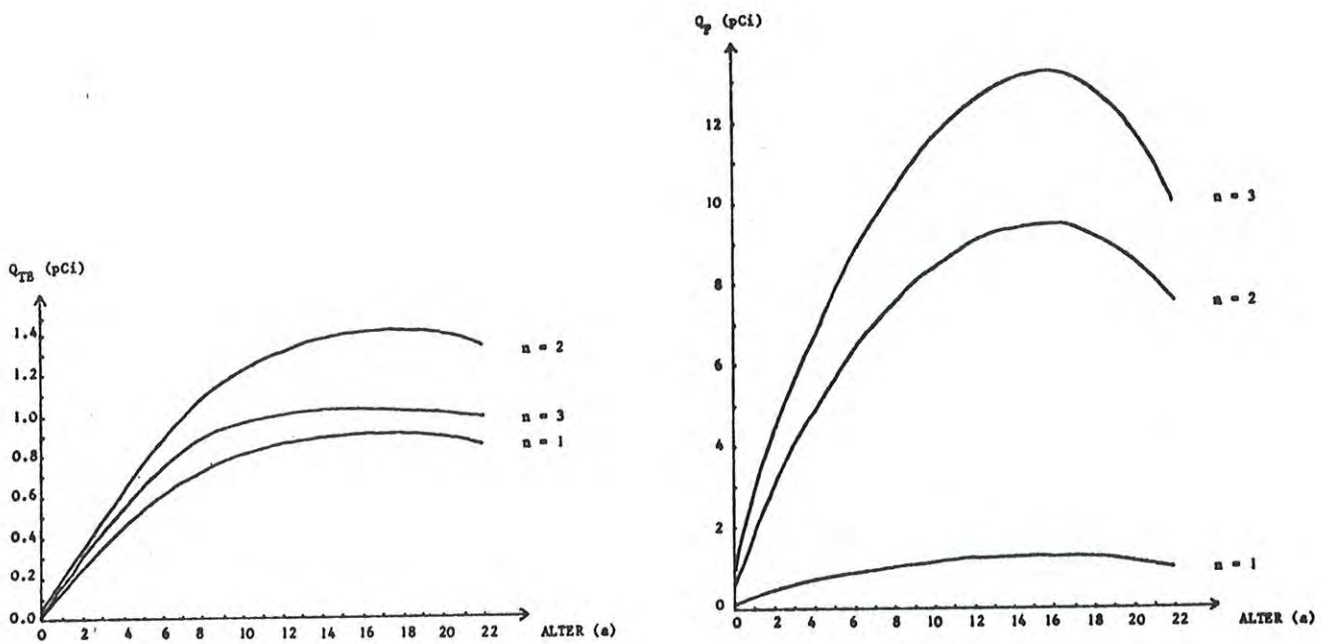


Abb.4 Lösungen des Gleichungssystems für das TB- und P-Kompartment

Die Gleichgewichtsaktivitäten in beiden Kompartments nehmen mit steigendem Alter zu, erreichen zwischen 15 und 20 Jahren ein Maximum und sinken dann gegen 30 Jahre hin wieder leicht ab.

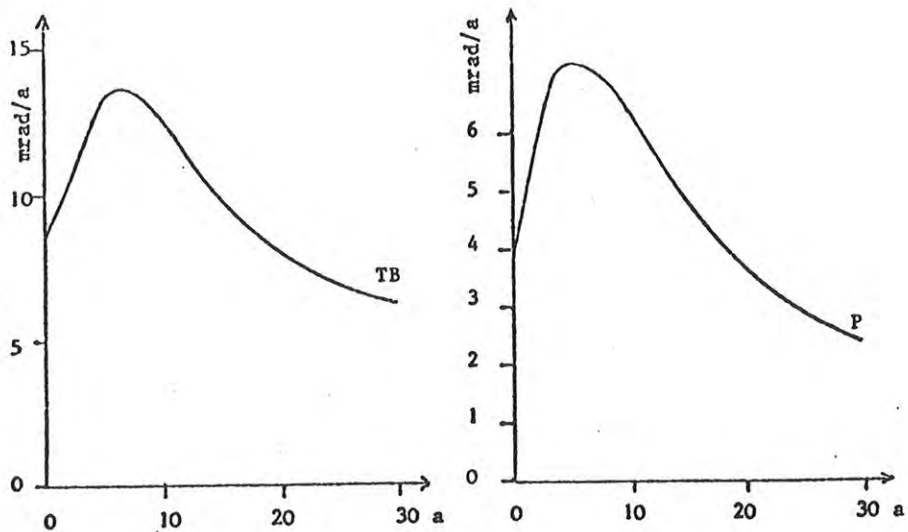


Abb.5 Integrale Alpha-Dosis (mrad/a) als Funktion des Alters für das Tracheobronchial- und Pulmonär-Kompartment

Aus diesen Lösungskurven für RaA und RaC und den entsprechenden Werten für die Massen der Kompartments kann schließlich die absorbierte mittlere α -Dosis für beide Bereiche berechnet werden (Abb.5).

Die Ergebnisse der Berechnungen zeigen eine deutliche Altersabhängigkeit der Dosis in beiden Kompartments mit einem signifikanten Maximum im Alter von etwa 6 Jahren. Dieser Maximalwert liegt bis zu einem Faktor 3 höher als die entsprechende Dosis für einen Erwachsenen, der unter denselben Umweltbedingungen lebt. Berechnungen für andere Referenzatmosphären und Lebensbedingungen zeigen ähnliche Kurvenverläufe, wenn auch mit anderen absoluten Werten.

Experimentelle Untersuchungen (12) zeigten, daß rund ein Drittel der im Alveolarraum abgelagerten Aktivität sehr rasch in den Blutkreislauf gelangt. Dies steht im Widerspruch zu der im ICRP-Modell angenommenen Halbwertszeit von 0.5 Tagen (vergleiche Fußnote Seite 39).

Simuliert man diese Ergebnisse am Hybridrechner, so erhält man Halbwertszeiten zwischen 28 und 43 Minuten je nach Nuklidart und angenommener Halbwertszeit für den Transport über das Lymphsystem. Das bewirkt eine Verminderung der Aktivität und somit auch der Dosis im Alveolarbereich auf ungefähr die Hälfte.

6. ZUSAMMENFASSUNG

Der Einfluß altersabhängiger Lungenparameter auf die Dosis im Atemtrakt bei der Inhalation natürlich radioaktiver Nuklide wurde am ICRP-Kompartimentmodell mit Hilfe eines Hybridrechners untersucht. Dabei zeigte sich - von der Geburt bis zu einem Alter von 30 Jahren - eine deutliche Altersabhängigkeit der Dosis, die im Alter von etwa 6 Jahren ein Maximum erreicht. Die Berechnung der absorbierten Dosen in den Basalzellen des Bronchialepithels ist noch im Gange.

Weitere laufende Untersuchungen befassen sich mit der Anwendung dieser Methode auf die Berechnung der Dosen bei der Inhalation von Thoron (^{220}Tn) und seiner Zerfallsprodukte.

SUMMARY

In our environment the inhaled natural radioactive nuclides Radon (^{222}Rn) and its short-lived decay products cause the highest radiation burden in parts of the respiratory system. The local dose distribution in the respiratory tract is dependent on several parameters, characterising aerosol and lung physiology. Due to the age dependence of the latter, children and youths receive a radiation burden different from that of an adult. For the mathematical description the respiratory tract was divided into 3 defined compartments with 6 subcompartments. The temporal variation of the compartmental activities for 3 types of radioactive nuclides was described by a system of 18 linear differential equations. From the solutions of the subcompartmental activities and the specific masses of the compartments the dose from α -irradiation was calculated for each year of age. Most of the calculations were carried out on a hybrid-computer system, using an automatic optimum-scaling program. The analog circuits necessary for the calculations in the 3 compartments were developed. A special subroutine (TRANSR) performed data input and output, choice of compartment, Akima -interpolation of functions and calculation of scaling factors. In addition, the age dependent deposition probabilities and annually inhaled activities were calculated using a digital mini-computer. The results show that from birth to the age of 30 years there is a significant age dependence of radiation dose to the respiratory system.

A maximum dose is reached at the age of about 6 years. This dose is up to a factor 3 higher than the corresponding dose value for an adult, living under the same conditions.

W. Hofmann, F. Steinhäusler

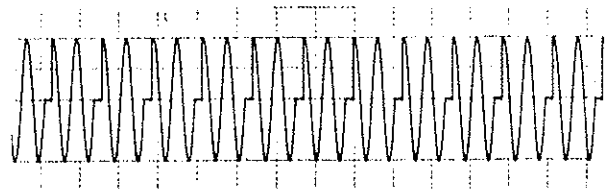
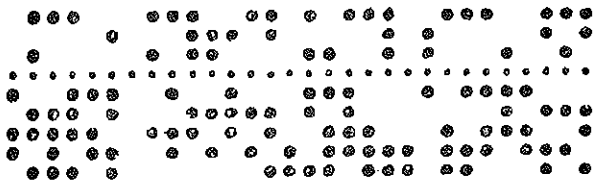
Institut für Physik, Universität Salzburg und
Forschungsinstitut Gastein der Österreichischen
Akademie der Wissenschaften

H. Hummer

Hybridrechenanlage der TU Wien

LITERATUR

1. Hofmann, W.: Modell zur Simulation des Verhaltens natürlich radioaktiver Nuklide in der Luft von Wohn- und Arbeitsräumen mit Hilfe eines Hybridrechners. Interface 8, 19 (1977)
2. Hofmann, W., Steinhäusler, F.: Die Strahlenbelastung des Atemtrakts bei der Inhalation natürlich radioaktiver Nuklide. Ber. Nat. Med. Ver. Sbg. 2 (1976)
3. Pohl, E., Pohl-Rüling, J.: Dose calculations due to the inhalation of ^{222}Rn , ^{220}Rn and their daughters. Health Phys. (in press)
4. Task Group on Lung Dynamics: Deposition and retention models. Health Phys. 12, 173 (1966)
5. Report of the Task Group on Reference Man. ICRP No. 23, Pergamon Press (1975)
6. Voillequé, P.G.: AERIN, a code for acute aerosol inhalation exposure calculations. Health Phys. 19, 427 (1970)
7. Solar, D.: Softwarepaket zur digitalen Unterstützung von Simulationen am Analogrechner. Interface 7, 18 (1976)
8. Hötter, G.J.: Die Lungenfunktion unter besonderer Berücksichtigung des wachsenden Organismus, Steinkopff Verlag (1975)
9. Landahl, H.D.: On the removal of air-borne droplets by the human respiratory tract: I. The lung. Bull. Math. Biophys. 12, 43 (1950)
10. Weibel, E.R.: Morphometry of the human lung, Springer Verlag (1963)
11. Hofmann, W., Steinhäusler, F.: Dose calculations for infants and youths due to the inhalation of radon and its decay products in the normal environment. Proc. 4. Int. IRPA Congr. (1977)
12. Pohl, E., Pohl-Rüling, J.: Die Strahlenbelastung bei der Inhalation von Radon, Thoron und deren Zerfallsprodukten. Strahlentherapie 136/6, 738 (1968)
13. Pohl, E., Steinhäusler, F., Hofmann, W., Pohl-Rüling, J.: Methodisches zur Messung und Berechnung der Strahlenexposition des Menschen durch natürliche Quellen ionisierender Strahlung am Beispiel einer Untersuchung an der Bevölkerung der Stadt Salzburg. ÖVS-Fachtagung 1976 "Strahlenbelastung der Bevölkerung und ihre Erfassung" (1977)



BENÜTZERFORUM

EIN VERGLEICH EINIGER VERFAHREN DER MERKMALBESTIMMUNG FÜR SPRACHERKENNUNG

I. Schaltungstechnische Bestimmung der Merkmale für Sprachanalyse

Romualdas Gudonawitschius
Politechnische Hochschule in Kaunas, UdSSR

Irmgard Husinsky
Hybridrechenanlage
Technische Universität Wien

Einleitung

Zur automatischen Spracherkennung werden verschiedene Verfahren der Merkmalgewinnung verwendet. In dieser Arbeit wird die Aufgabe gestellt, einen objektiven Vergleich ihrer Leistungsfähigkeit durchzuführen. Zur Merkmalgewinnung wurden folgende Verfahren verwendet:

- a) Analyse durch Autokorrelationsfunktion (Korrelationskoeffizienten),
- b) Analyse durch Lineare Prädiktion (Prädiktorkoeffizienten),
- c) Analyse durch Koeffizienten in Phasendiagrammen.

Im ersten Teil dieses Berichtes werden nur die schaltungstechnischen Realisierungen der Merkmalgewinnung in Echtzeit für obengenannte Verfahren mit Hilfe des EAI 680 Analogrechners besprochen. In der nächsten Nummer von "Interface" werden die theoretischen Grundlagen dieser Verfahren erklärt und ein objektiver Vergleich der Ergebnisse mit Hilfe digitaler Programme durchgeführt.

Unterscheidung von Sprachsignalen und Sprachpausen

Bei der automatischen Spracherkennung muß das kontinuierliche Signal in kleinere Einheiten, als sie einem ganzen Wort entsprechen, aufgeteilt werden. Eine mögliche Einteilung sind z.B. Sprach- und Pausebereiche, stimmhafte und stimmlose Bereiche usw.. Die naheliegendste Gliederung des Signals ist eine Aufteilung in Bereiche, die kein eigentliches Sprachsignal enthalten, die Sprachpausen, und in Bereiche mit Sprachsignalanteilen. Die Pausen tragen keine Information und können für die weitere Verarbeitung eliminiert werden, was z.B. bei isoliert gesprochenen Wörtern eine erhebliche Einsparung von Speicher- und Analyseaufwand bedeutet.

Ein einfaches Verfahren zur Unterscheidung von Signalen und Pausen ist folgendes: die vom Mikrofon abgegebene Spannung wird verstärkt und mit Referenzspannung im Komparator verglichen. Der Komparator gibt ein Ausgangssignal logisch 1 ab, solange die Referenzspannung überschritten wird. Das nachgeschaltete Halteglied verlängert die Impulse um 12.5 msec. Diese Schaltung gewährleistet keine sichere Unterscheidung von Sprachsignalen und Pausen, da bei konstanter Referenzspannung eine Schwelleüberschreitung der Signale, die nicht sprachspezifisch sind, möglich ist. Weiters besteht die Möglichkeit, daß die Amplitude von Lauten mit geringer Energie die Umschaltsschwelle des Komparators nicht erreicht. Zu diesem Zwecke zeigen die Untersuchungen von Reddy [1], daß aufgrund der Anregung stimmhafter Laute im Spracherzeugungssystem in der Umgebung eines signifikanten Maximums im Sprachsignal immer auch ein signifikantes Minimum auftritt, deren zeitlicher Abstand unabhängig vom Sprecher immer kleiner als 3.5 msec ist. Als signifikantes Maximum und Minimum sind die betragsmäßig größten Extremwerte einer Sprachgrundperiode definiert.

Diese Untersuchungen liefern ein sprachspezifisches, sprecherunabhängiges Merkmal zur Unterscheidung von Sprachsignalen und Pausen. Zu diesem Zweck wird eine Schaltung in Echtzeit realisiert, die in Abbildung 1 gezeigt ist. Die Mikrofonspannung wird verstärkt. Mit Hilfe bekannter Spitzengleichrichterschaltungen, Komparatoren K2, K3 und Differenzierer D2, D3, bekommt man die Impulse, die den größten Extremwerten einer Sprachgrundperiode entsprechen. Dann wird durch zwei monostabile Kippstufen (M1 und M2) bei jedem Maximum bzw. Minimum der Signalspannung ein Impuls von 3.5 msec Länge abgegeben. Ein UND-Gatter (3) entscheidet, ob der zeitliche Abstand aufeinanderfolgender Maxima und Minima kleiner als 3.5 msec ist.

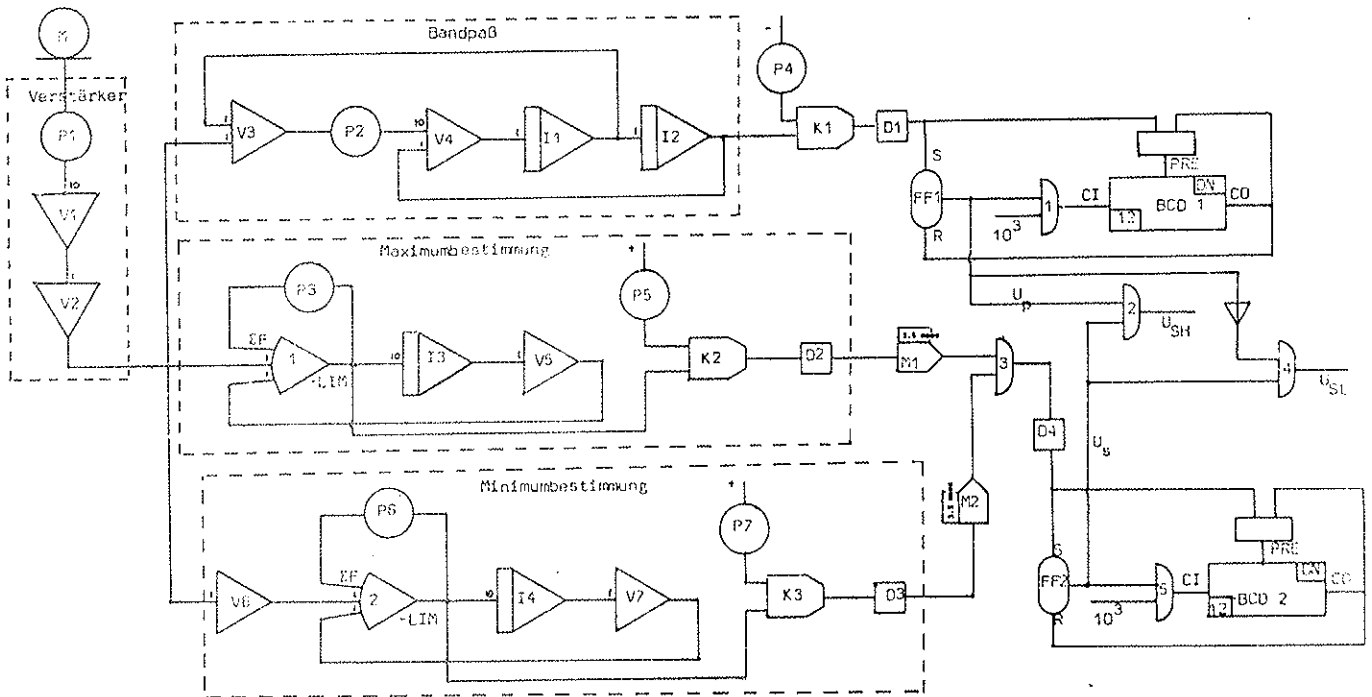


Abbildung 1

Ist dies der Fall, so wird der BCD-Zähler 2 gesetzt, dessen Haltezeit gleich der längsten Grundperiodendauer stimmhafter Laute (12,5 msec) ist. Der Vorteil einer derartigen Schaltung besteht darin, daß außer der Amplitudenschwelle von Maximum- und Minimumbestimmung ein für das Sprachsignal charakteristisches Kriterium zur Unterscheidung von Sprachsignalen und Pausen angewandt wird. Die Schaltung ist unempfindlich gegenüber niederfrequenten Störsignalen, auch 50Hz-Störspannungen, weil im allgemeinen die Bedingung eng benachbarter Maxima und Minima nicht erfüllt ist.

Abbildung 2 zeigt das Oszillogramm des Wortes /Eins/ sowie die Ausgangsspannung der Schaltung zur Unterscheidung von Sprachanteilen und Pausen. Auch Laute mit geringer Energie /s,f/ werden einwandfrei erkannt. Abbildung 3 zeigt das Ausgangssignal für das Wort /Acht/. Dabei ist die vor einem Explosivlaut auftretende Pause im Sprachsignal zu erkennen.

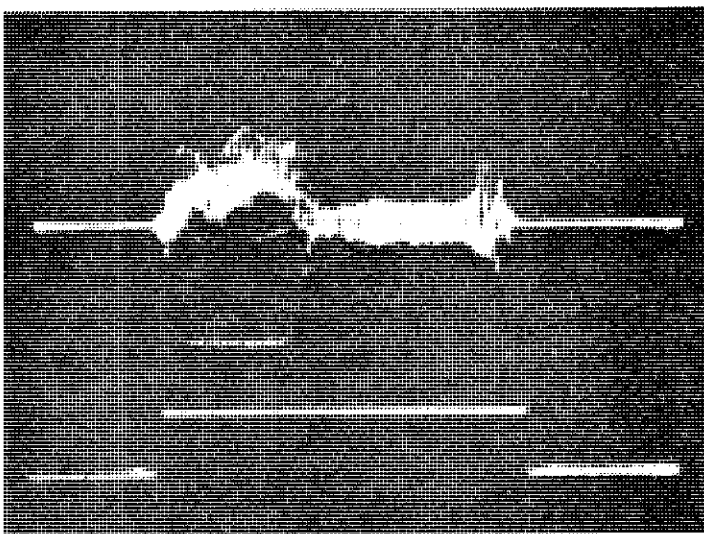


Abbildung 2
(Sprecher: RG)

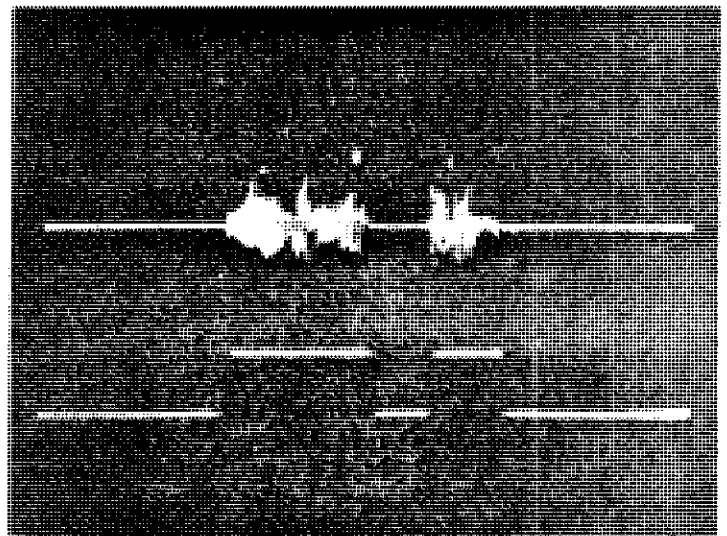


Abbildung 3
(Sprecher: III)

Unterscheidung von stimmhaften und stimmlosen Lauten

Sprachsignale umfassen zwei Lautgruppen: stimmhafte Laute, die durch Stimmbandschwingungen angeregt werden, und stimmlose Laute, bei denen die periodische Anregung fehlt und die dem zeitlichen Verlauf von Rauschsignalen ähnlich sind. Diese Lautgruppen werden als Entscheidungseinheiten für die automatische Spracherkennung benützt.

Es sind viele verschiedene Verfahren zur Unterscheidung von stimmhaften und stimmlosen Lauten bekannt, die entweder im Zeitbereich oder im Frequenzbereich arbeiten. Bei Verfahren, die im Zeitbereich arbeiten, wird die Trennung von stimmhaften und stimmlosen Lauten durch Auswertung der Extremwert- oder der Nulldurchgangshäufigkeit bzw. durch eine Kombination beider Merkmale im nicht bandbegrenzten Signal vorgenommen. Die Untersuchungen verschiedener Verfasser [2], [3] haben gezeigt, daß weder Extremwerthäufigkeit noch Nulldurchgangshäufigkeit befriedigende Ergebnisse liefern.

Die Anwendung der Cepstrummethode zur stimmhaft-stimmlos-Unterscheidung ist gut geeignet [4], aber der Hardwareaufwand für Analyse in Echtzeit ist nach dem heutigen Stand der Technik unvertretbar hoch. Daher haben wir in Verbindung mit der schon beschriebenen Anordnung der Unterscheidung von Sprachsignalen und Sprachpausen zusätzlich eine Schaltung realisiert, die einfache und einwandfreie stimmhaft-stimmlos-Unterscheidung in nicht bandbegrenzten Sprachsignalen ermöglicht (Abbildung 1). Die verstärkte Mikrofonspannung wird auf einen Bandpaß (Verstärker V3, V4 und Integrierer I1, I2) mit einem Durchlaßbereich von 80 Hz bis 400 Hz und weiter auf Komparator K1 geführt, der dann am Ausgang logisch 1 liefert. Mit Hilfe des BCD-Zählers 1 werden die abgegebenen Impulse um 12.5 msec, d.h. gleich der längsten Grundperiodendauer in stimmhaften Lauten, verlängert. Über ein UND-Gatter (2) der Ausgangsspannung U_p und der Ausgangsspannung U_s wird U_{SH} dann logisch 1, wenn ein Sprachsignal vorhanden ist. Dies ist nur bei stimmhaften Lauten der Fall. Wenn der Laut stimmlos ist, so ist nur U_s logisch 1 und daher U_p logisch 0. Durch $U_s \wedge \overline{U_p}$ (UND-Gatter 4) sind stimmlose Laute gegeben. Abbildung 4 zeigt anschaulich das Dszillogramm des Wortes /Eins/, wo die stimmhaften und stimmlosen Sprachanteile erkannt werden.

Weiterentwicklung der Verarbeitung des Sprachsignals

Das Sprachmaterial besteht aus einer Stichprobe von Zahlwörtern (Ziffern 0 bis 9), die von 3 Sprechern in einem reflexionsarmen

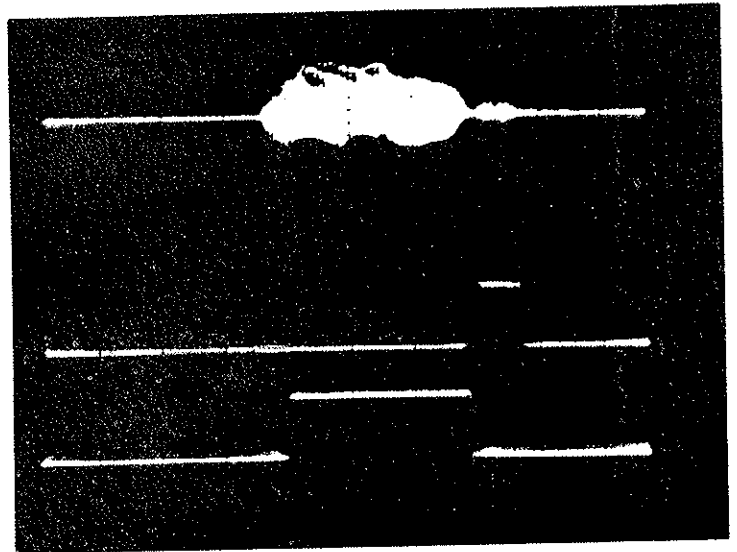


Abbildung 4
(Sprecher: RG)

Raum aufgenommen wird. Zur Gewinnung der Autokorrelationskoeffizienten und der Prädiktorkoeffizienten werden die Sprachanteile mit einer Frequenz von 12 kHz abgetastet und für die weitere Verarbeitung zum Digitalrechner übertragen. Zur Bestimmung der Koeffizienten in Phasendiagrammen für stimmhafte und stimmlose Sprachbereiche werden die topologischen Eigenschaften des Phasenbildes des Sprachsignals über seiner Ableitung nach der Zeit oder über den verschiedenen mathematischen Transformationen des Sprachsignals, die der Verarbeitung des Sprachsignals im Ohr entsprechen, untersucht. Über die schaltungstechnische Messung dieser Koeffizienten als Merkmale für die Spracherkennung wurde schon in [5] geschrieben.

Literatur:

- [1] REODY D.R.: Pitch Period Determination of speech sounds. Communications of the ACM 10(1967) 6, S. 343.
- [2] STOCK D., RUPPRATH R.: Die stimmhaft-stimmlos-Distinktion im Dawid II.7. International Congress on Acoustics (1971), Budapest 20C11.
- [3] HESS W.: Digitale grundfrequenzsynchrone Analyse von Sprachsignalen als Teil eines automatischen Spracherkennungssystems. Diss. TU-München (1972)
- [4] TIME M., IDLER H., LAY TH.: Auswertung von Echtzeit-Cepstra zur schnellen Detektion stimmhafter Laute. NTZ 26 (1973) 7, S. 312.
- [5] GSANDTNER A., GUDONAWITSCHIUS R.: Die Bestimmung der Merkmale für Vokalerkennung. INTERFACE 3 (1975) S. 7.

INTERFACE Juni 1977