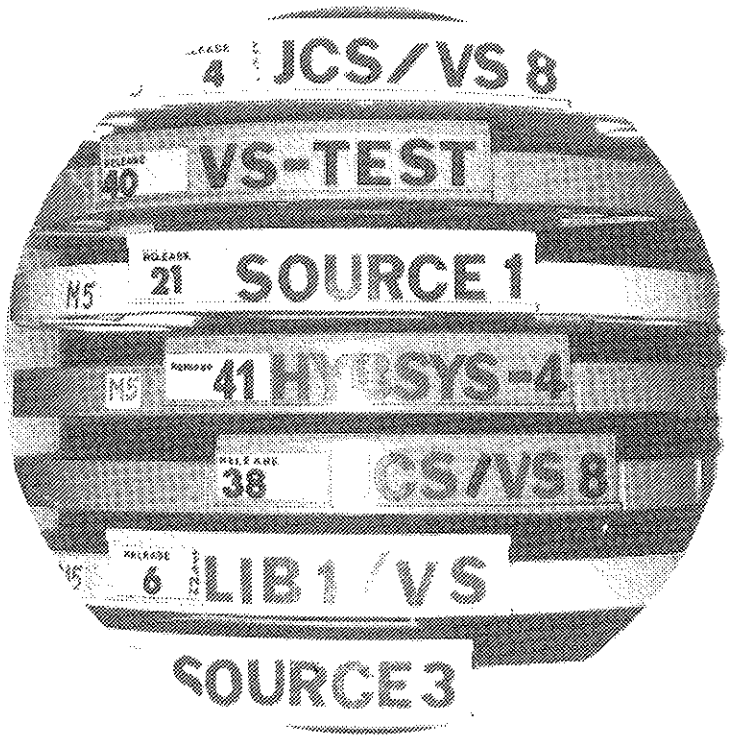
Interface

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

Nummer 17
April 1981



Umstellung auf das Betriebssystem JCS/VS 8

INHALTSVERZEICHNIS

Redaktion: Irmgard Husinsky

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

ERSTE ERFAHRUNGEN MIT DEM AUTOPATCH-BETRIEB

Die am 2. März 1981 erfolgte Umstellung auf das neue Betriebssystem JCS/VS 8 stellt, obwohl auf den ersten Blick gar nicht ersichtlich, einen bemerkenswerten Einschnitt in der Entwicklung der Hybridrechentechnik an der Technischen Universität Wien dar. Einerseits markiert sie den letzten Schritt in der Umstellung auf ausschließlichen AutoPATCH-Betrieb für hybride Benützer, andererseits ist damit ein erster äußerst wichtiger Schritt in Richtung hybrides Time-Sharing gesetzt worden.

Die Entwicklung nahm in den letzten 15 Monaten einen stürmischen Verlauf. Es ist nämlich noch nicht lange her, daß der Hybridrechner an der Technischen Universität Wien ausschließlich in der konventionellen Art händisch programmiert werden konnte. Am 19.11.1979 wurde die Lieferung und Installation der AutoPATCH-Erweiterung der Hybridrechenanlage von der Firma EAI erfolgreich abgeschlossen. Die vertraglich vereinbarten Leistungstests konnten im Jänner 1980 erfolgreich beendet werden. Unser hybrider Sprachprozessor HYBSYS wurde bis Ende Februar 1980 "AutoPATCH-fähig" gemacht und ab 1. März 1980 konnten am Rechner im Sommersemester neben den händisch gesteckten Schaltungen bereits hybride Simulationen mit dem AutoPATCH-System in einer maschinenunabhängigen Syntax vom Terminal aus programmiert werden. In der Übergangsphase bis Ende Juni stand allerdings nur ein erster Testsatz von analogen Makros zur Verfügung (z.B. gab es nur 12 Integrierer und 6 Summierer). Doch konnten schon mit dieser minimalen Konfiguration wichtige Betriebserfahrungen gesammelt werden.

In den Sommerferien wurde der Übergang auf einen ausschließlichen AutoPATCH-Betrieb vorbereitet. Dazu waren auch weitgehende Hardwaremodifikationen notwendig. So wurden 32 zusätzliche Konstantengeber (CDACs) entwickelt und eingebaut und als Ersatz für die langsamen Servopotentiometer wurden 60 digitale Potentiometer unter Verwendung der neuesten Technologie von uns selbst entwickelt und implementiert, sodaß nunmehr alle 120 Koeffizienteneinheiten digital setzbar sind. Zur Steuerung der individuellen Zeitkonstanten der Integrierer war es auch notwendig, das Interface um 64 zusätzliche Kontrollleitungen zu erweitern.

Das neue AutoPATCH-System, dessen Makros nun die gesamte verfügbare Schaltkapazität ausnützen und die auch qualitativ gegenüber dem Testbetrieb erweitert (Anzahl und Art der Makros) wurden, konnte am EAI Users' Group Meeting im September 1980 bereits vorgestellt werden. Seit 1. Oktober 1980 ist ein ausschließlich AutoPATCH-orientierter Hybridbetrieb eingeführt.

Obwohl über den reinen AutoPATCH-Betrieb erst Daten über einen verhältnismäßig kurzen Zeitraum zur Verfügung stehen, ist es doch interessant, die für den Tätigkeitsbericht des Bundesministeriums für Wissenschaft und Forschung erhobenen Daten über den Einsatz der Sprachprozessoren in den Jahren 1979 und 1980 zu vergleichen:

	Verwendete Prozessoren in % der Programme	Verwendete Prozessoren in % der CPU-Zeit
1979	89% FORTRAN 10% Assembler 1% HOI	77% FORTRAN 13% Assembler 10% HOI
1980	61% FORTRAN 11% Assembler 28% HYBSYS	20.8% FORTRAN 6.5% Assembler 72.7% HYBSYS

Das hervorstechendste Ergebnis ist wohl, daß HYBSYS zunehmend in FORTRAN geschriebene Benützerechtzeitprogramme ersetzt. Da von unserem Jobaccountingsystem bisher nur die CPU-Zeit, aber nicht die Einschaltzeit, d.h. die Zeit, die von einem Benützer während seiner Simulationsläufe am Terminal verbracht wird, ermittelt werden konnte, liefern die angeführten Zahlen noch gar kein richtiges Bild über die tatsächliche Verwendungsintensität von HYBSYS. Tatsächlich ist der Prozentsatz von HYBSYS unter den verwendeten Sprachprozessoren, gemessen an der Einschaltzeit, weit über 90%. Seine Interpretereigenschaften bedingen aber auch, daß ein relativ großer Teil der Einschaltzeit durch Nachdenken des Benützers verstreicht und daher ein hybrides Time-Sharing-Verfahren weitere enorme Durchsatzsteigerungen für hybride Simulationen erwarten läßt.

W. Kleinert

EXPERIENCES WITH AUTOPATCH

The new operating system JCS/VS 8 which is running since March 1981 represents an important landmark in the development of hybrid computation at the Technical University of Vienna. It is the last step on the way to a complete autopatch system and at the same time a very important step towards hybrid time-sharing. It has not been such a long time that the hybrid computer at the Technical University of Vienna could be programmed only in the conventional manual way. In November 1979 the installation of the autopatch extension was successfully completed by EAI. Acceptance tests were completed in January 1980. Our hybrid interpreter HYBSYS was adapted for autopatch by February and from March 1980 on the autopatch system could be used by terminal users programming hybrid simulations in a problem-oriented syntax, where at first there was only a test set of analog macros. Extensive hardware modifications were necessary to complete the transition, 32 additional CDACs and 60 digital potentiometers were developed and implemented. 64 control lines had to be added to the interface. The new autopatch system could be presented at the EAI Users' Group Meeting in September 1980.

The table at the top of the page shows job accounting data comparing 1979 to 1980. The first column gives the percentage of programs using different processors. The second column gives the percentage of CPU-time used by different processors. The most outstanding result is that HYBSYS jobs more and more replace real-time user jobs written in FORTRAN. But this is only an account of the CPU-time and not of the actual time a user spends at the terminal using the interpreter HYBSYS. Regarding the actual time HYBSYS is active in more than 90% of the time. Considering that the user spends a lot of time at the terminal without responding immediately a hybrid time-sharing system will certainly increase the throughput for hybrid simulations.

aktuelle mitteilungen

ÖFFNUNGSZEITEN

Da sich aufgrund der größeren Leistungsfähigkeit des neuen Betriebssystems JCS/VS 8 höhere Anforderungen an den Operatorbetrieb ergeben und zur Zeit nur zwei Operatoren an der Hybridrechenanlage beschäftigt werden können, mußten die Öffnungszeiten um zwei Stunden reduziert werden. Die Hybridrechenanlage ist nun von

Montag bis Freitag
von
8 Uhr bis 18 Uhr

geöffnet.

RECHENZEITRESERVIERUNG

Rechenzeit zur Benützung des AutoPATCH-Systems mit dem Prozessor HYBSYS kann für die angegebenen Öffnungszeiten außer Montag von 8 bis 14 Uhr, wo Hardware- und Systemwartung durchgeführt wird, beim Operator reserviert werden. Ebenso ist bei Verwendung spezieller Hardware sowie für interaktive graphische Jobs eine Reservierung vorzunehmen.

TERMINALRESERVIERUNG

Für die Benützerterminals, die zum Editieren von Programmen und Daten sowie zur Programmvorbereitung zur Verfügung stehen, können beim Operator Reservierungen vorgenommen werden.

CLOSED-SHOP-BETRIEB

Closed-Shop-Programme werden nach Möglichkeit, auf jeden Fall aber einmal innerhalb von 24 Stunden gerechnet.

Neues Betriebssystem JCS/VS 8

Anfang März 1981 löste das neue Betriebssystem JCS/VS 8 das System JCS/TS 7 ab. Nähere Informationen befinden sich auf den Seiten 11 bis 22.

Die Verwendung von HYBSYS im Betriebssystem JCS/VS 8

Generell bringt das neue Betriebssystem keine Änderungen für den HYBSYS-Benützer mit Ausnahme der zu verwendenden Steuerkarten (/HYB). Von der Möglichkeit, Programme am Terminal zu entwickeln und von dort zu starten, sollte jeder HYBSYS-Benützer Gebrauch machen. Dabei kann die Deklaration der Modellgleichungen auf einem Source File (mit den entsprechenden Steuerkarten) vorbereitet werden. Erst nach Fertigstellung der Deklaration wird HYBSYS gestartet und es entfällt die Belegung einer Batch-Partition.
Ein weiterer Vorteil besteht darin, daß die Modellgleichungen auf der Deklarationsebene zugänglich bleiben.

Genauere Angaben bezüglich der jeweils geltenden HYBSYS-Release (Steuerkarten, Districts, Neuerungen, Fehler) sind dem HYBSYS-Bulletin zu entnehmen.

HYBSYS-Bulletin

Das HYBSYS-Bulletin ist eine Informationsschrift, die in Ergänzung zum HYBSYS-Manual alle notwendigen Angaben zur Bedienung von HYBSYS enthält. Vom HYBSYS-Bulletin erscheint bereits die dritte Ausgabe und es wird allen HYBSYS-Benützern , sowie auf Anfrage, zugesandt.

Personelles

Seit Beginn des Jahres 1981 ist Frau Irmgard Husinsky wieder an der Hybridrechenanlage tätig (halbtags) und übernimmt wieder die Redaktion des INTERFACE. Dipl.Ing. Dietmar Solar hat seit Jänner 1981 einen Ganztagsposten an der Hybridrechenanlage.

Drucksorten für Benützer der Hybridrechenanlage

Titel	Preis
Benützung des Betriebssystems JCS/VS 8	-
Terminalbenützung	-
JCS/VS 8 User Manual	5o.-
HYBSYS-Manual	5o.-
Graphische Grundsoftware	25.-
EAI-Assembler I Grundlagen der Assemblerprogrammierung	5o.-
EAI-Assembler II Hinweise zur fortgeschrittenen Programmierung	5o.-
Das Interpolations- und Plotterpaket IPP1 Teil I: Allgemeine Beschreibung	5o.-
Das Interpolations- und Plotterpaket IPP1 Teil II: Programmbeschreibungen	5o.-
GDP1 Programmpaket zur graphischen Darstellung von Lösungen gewöhnlicher Differentialgleichungen Teil I: Allgemeine Beschreibung	5o.-
GDP1 Programmpaket zur graphischen Darstellung von Lösungen gewöhnlicher Differentialgleichungen Teil II: Programmbeschreibungen	5o.-
POINT: Programmpaket zur numerischen Integration	-

Ferner gibt es Kurzbeschreibungen für alle Bibliotheksprogramme.

Programmberatung

Programmberatung wird von allen Mitarbeitern der Hybridrechenanlage nach Bedarf und Möglichkeit durchgeführt.

Benützerversammlung

Am 3. März 1981 wurde anläßlich der Betriebssystemumstellung eine Benützerversammlung der Hybridrechenanlage abgehalten, die großes Interesse fand. Die Benützer wurden sowohl über die organisatorische Durchführung der Betriebssystemumstellung als auch über die neuen Möglichkeiten, die Veränderungen und Erweiterungen, die das neue System bringt, und den Terminalbetrieb informiert. Anschließend wurden spezielle Benützerwünsche diskutiert.

kurse

RH3 HINWEISE FÜR FORTRAN-PROGRAMMIERER AN DER HYBRIDRECHENANLAGE

und

RH7 SOFTWAREUNTERSTÜTZUNG FÜR DIE BENÜTZUNG DES PACER 600 ALS PLOTTER SYSTEM

Termin: Oktober 1981

Vortragender: Dipl. Ing. F. Blöser

Die folgenden Kurse werden nach Vereinbarung abgehalten:

RH1 GERÄTETECHNIK EAI PACER 600

RH2 BENÜTZUNG DES BETRIEBSSYSTEMS JCS/VS 8

RH6 EAI ASSEMBLER

RH9 EINFÜHRUNG IN DAS HYBRIDE AUTOPATCH-SYSTEM

RH11 ASSEMBLER-PROGRAMMIERUNG FÜR FORTGESCHRITTENE MIT ÜBUNGEN

RH13 BEDIENUNG DES HYBRIDEN PROZESSORS HYBSYS
RH14 ZUR SIMULATION DYNAMISCHER SYSTEME AM AUTOPATCH-SYSTEM, TEIL 1 BZW. TEIL 2

RH16 PROGRAMMENTWICKLUNG AM TERMINAL

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

EAI Computer Users' Group Meeting in Wien

An der Technischen Universität Wien trafen am 8. September 1980 65 Teilnehmer aus 12 verschiedenen Ländern zum EAI Computer Users' Group Meeting 1980 ein.
Das Meeting wurde durch o. Univ. Prof. Dr. A. Weinmann eröffnet. Nach dem Mittagessen wurde das vor einem Jahr installierte PACER 600 AutoPATCH-System von Dr. W. Kleinert vorgeführt. Anschließend stellte Dipl.Ing. D. Solar den von ihm entwickelten Interpreter HYBSYS zur Unterstützung von hybriden Simulationen mit dem AutoPATCH-System vor.
Am Abend fuhren alle Teilnehmer zum eigentlichen Veranstaltungsort, dem Alpenhotel Gösing, ungefähr 120 km von Wien entfernt, wo sie die nächsten zwei Tage verbrachten. Es wurden 25 Vorträge gehalten, die sich mit Themen wie Simulationen an hybriden Systemen (im medizinischen, industriellen, sportlichen Bereich), RTOS, ECSSL, EAI 2000, neue Software für PACER 600 Systeme etc. beschäftigten.
In zwei EAI Sessions berichteten Mitarbeiter von EAI über ihre Produkte, es wurden zwei Filme über Power System Simulation und Power Plant Training Simulator präsentiert, und es gab Diskussionen zwischen EAI und den Mitgliedern der Users' Group.
Dienstag abend war die Möglichkeit gegeben, an einer von drei Working Groups teilzunehmen, in denen Fragen an EAI über die Themen ECSSL und RTOS, EAI-PDP-Connections und Hard- und Software ausgearbeitet wurden. Diese Working Groups wurden zum ersten Mal abgehalten und das Interesse der Teilnehmer war sehr groß. Es wurde vorgeschlagen, zukünftig das Arbeiten in kleinen Diskussionsgruppen über spezielle, genau abgegrenzte Themenbereiche zu fördern und während eines Meetings öfter zu ermöglichen. Einige der Teilnehmer ergriffen die Möglichkeit, an Hand von selbst vorbereiteten Beispielen on-line über ein graphisches Terminal unter Verwendung des Interpreters HYBSYS mit dem PACER 600 AutoPATCH-System zu rechnen.
Durch die Abgeschiedenheit des Veranstaltungsortes Gösing wurde der Kontakt zwischen den Benutzern und zwischen EAI und Benutzern gefördert.

EAI COMPUTER USERS' GROUP MEETING 1981

Das diesjährige EAI Computer Users' Group Meeting wird vom 15.-18. September 1981 in Glasgow, Schottland, stattfinden. Der Veranstaltungsort ist das Hotel Lomond Castle, das am See Loch Lomond, in der Nähe von Glasgow, liegt. Das Meeting wird von Anne MacKinnon organisiert.

E.Wittek

DARF ICH BEKANNTMACHEN?: IMACS

IMACS, die International Association for Mathematics and Computer in Simulation, ist eine der großen internationalen Vereinigungen, die es sich zum Ziel gesetzt haben, den wissenschaftlichen Erfahrungsaustausch zu fördern. Dies geschieht einerseits durch die Veranstaltung von - speziellen Problemstellungen gewidmeten - Symposien und des allgemeinen, alle drei Jahre stattfindenden Kongresses, andererseits durch die Herausgabe wissenschaftlicher Publikationen in Form von Tagungsberichten und der viermal jährlich erscheinenden IMACS-Transactions. Dem raschen Austausch von Informationen dient der in zwangloser Form erscheinende Newsletter, den alle Mitglieder von IMACS erhalten.

Im Gegensatz zu anderen derartigen Vereinigungen (wie z.B. IFAC) gibt es bei IMACS sowohl eine persönliche Mitgliedschaft von einzelnen Personen, als auch eine solche von Institutionen. Die persönliche Mitgliedschaft kann entweder eine sogenannte Grundmitgliedschaft sein (bfr 300.- jährlich, das sind etwa öS 150.-), die zum Bezug des Newsletter sowie zu Preisermäßigungen bei IMACS-Publikationen und einigen IMACS-Veranstaltungen berechtigt, oder aber eine Vollmitgliedschaft (bfr 800.- jährlich), die darüber hinaus auch den kostenlosen Bezug der IMACS-Transactions einschließt. Für Firmen und Institute sind spezielle Regelungen vorgesehen.

Für alle Anfragen sowie zur Entgegennahme von Anträgen auf Mitgliedschaften steht gerne zur Verfügung

Prof.Dr. I. Troch
Technische Universität Wien
Gußhausstraße 27-29
A-1040 Wien

PUBLIKATION SIMULATIONSTECHNISCHER ERGEBNISSE

Meist ist für den Naturwissenschaftler, Techniker usw. die Simulation nur Mittel zu dem Zweck, Ergebnisse auf dem eigenen und eigentlichen Arbeitsgebiet zu erhalten. Dennoch ist es häufig so, daß im Verlauf derartiger Untersuchungen - quasi als Nebenprodukt - auch interessante Ergebnisse auf dem Gebiet der Simulationstechnik anfallen:

Dies können z.B. mathematische, algorithmische, usw. Varianten bekannter Verfahren sein, die für eine spezielle Problemklasse besonders gut geeignet sind, sowie auch Erfahrungen bei der Anwendung verschiedener Verfahren auf eine solche. Eine Publikation dieser "Nebenergebnisse" gemeinsam mit den anderen Resultaten ist meist nicht möglich.

Gerade Benutzer der Hybridrechenanlage werden relativ häufig in die oben geschilderte Lage kommen und es sei darum darauf hingewiesen, daß viele derartige Ergebnisse in der wissenschaftlichen Zeitschrift der International Association for Mathematics and Computers in Simulation, den IMACS-Transactions, veröffentlicht werden können. Manuskripte (vorzugsweise in englischer Sprache) können an jedes Mitglied des Editorial Board gesandt werden, insbesondere übernimmt gern die Weiterleitung

Prof.Dr. I. Troch
Technische Universität Wien
Gußhausstraße 27-29
A-1040 Wien

NEUES BETRIEBSSYSTEM JCS/VS 8

Anfang März wurde an der Hybridrechenanlage ein neues Betriebssystem, JCS/VS 8, installiert. Es löst das seit Oktober 1976 laufende System JCS/TS 7 ab und bringt in erster Linie die erforderliche Leistungssteigerung, die der rapiden Entwicklungstätigkeit an der Hybridrechenanlage sowohl auf dem Gebiet der Hardware als auch der Software Rechnung trägt.

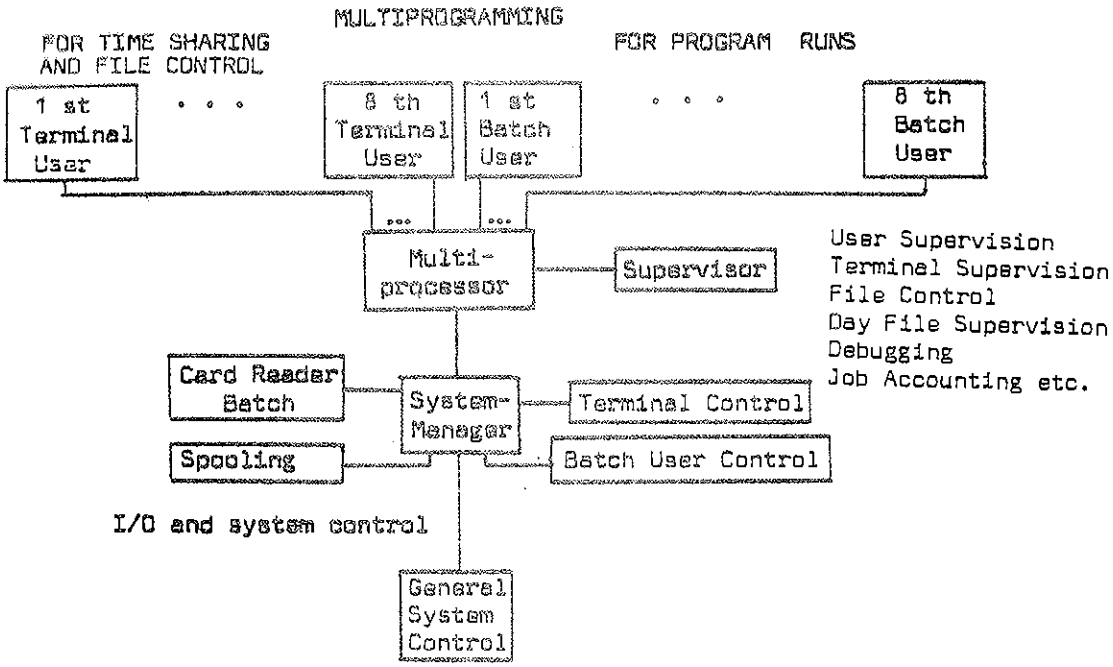
JCS/VS 8 ist eine Eigenentwicklung der Abt. Hybridrechenanlage und wurde nach mehr als zweijähriger Entwicklungszeit in seiner Version .1 fertiggestellt. JCS/VS 8 ist fast völlig aufwärtskompatibel zum alten System (siehe auch Seite 16), arbeitet aber intern völlig virtuell, um die Leistung auf einem Kleinrechner wie dem PACER 100 zu bewerkstelligen, d.h. neben dem virtuellen Massenspeicherkonzept (50 Districts) gibt es virtuelle Terminals, virtuelle Files und virtuelle Systemprozessoren, die unabhängig von der Anzahl der parallel arbeitenden Benutzer arbeiten.

JCS/TS 7	JCS/VS 8
2 Systemlevel	2 Systemlevel
2 User Level, inklusive der Terminal User	8 Batch User Level(Programme) und
	8 Terminal User (Time Sharing)
27 Districts	50 Districts
3 Terminals	9 Terminals

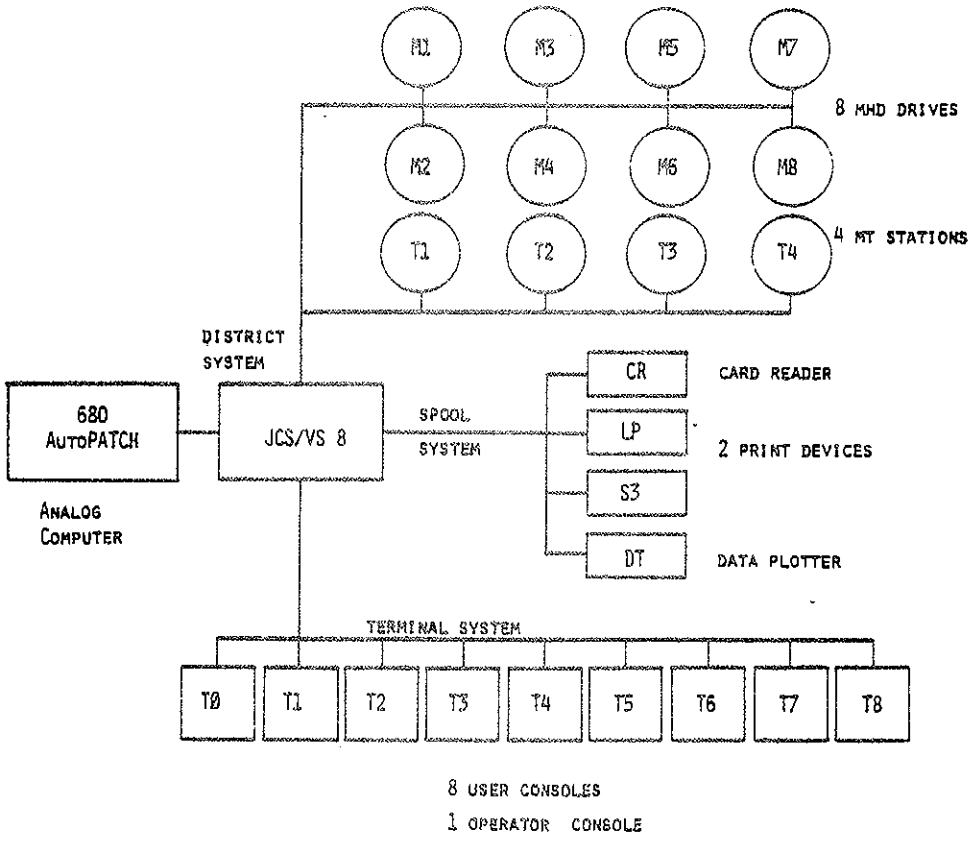
Wichtigste Leistungsunterschiede

Im Multiprogramming werden maximal 18 Level - 16 Benützer (User) und zwei Systemaktivitäten (Supervisor und System Manager) - parallel bedient. Die Anzahl der Level richtet sich nach der jeweiligen Anforderung. Dabei ermöglicht dynamisches Cycling eine effektive CPU-Aktivität. Sobald ein Exekutionslevel untätig (Idle) ist oder logisch warten muß, wird ein Cycle zu einem anderen Task erzwungen.

Das Input/Output System arbeitet unabhängig von anfallendem Input/Output, indem es Kernspeicherpuffer benützt, die die Information spoolen. Außerdem werden Card Reader Input, Line Printer- und Data Plotter Output und Output einer dritten Spool Device gespoolt und können daher ohne Einschränkung von allen User Tasks parallel durchgeführt werden.



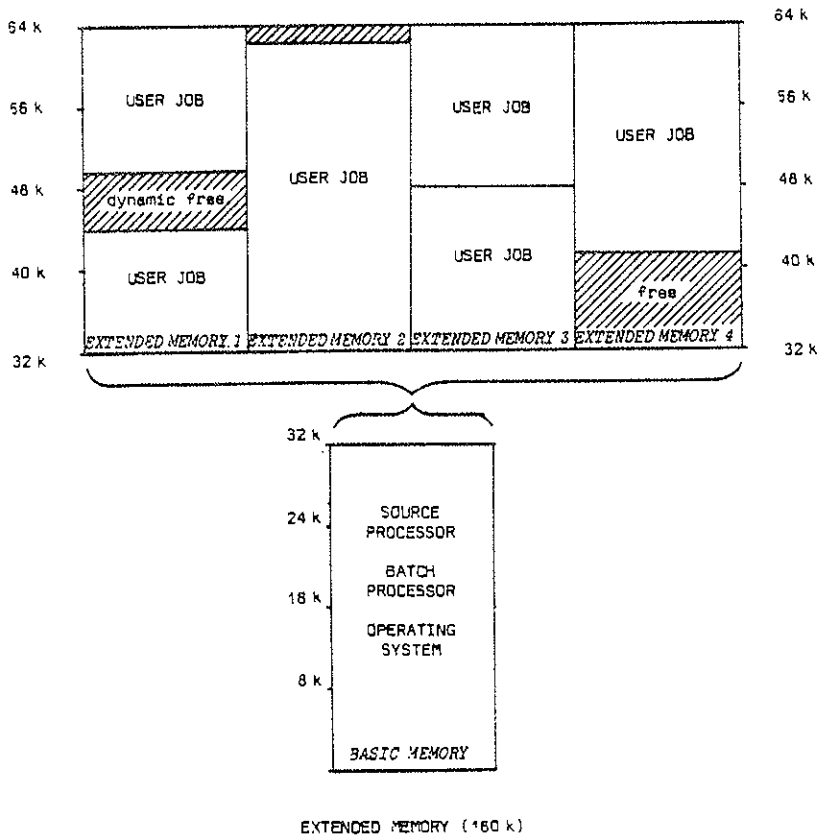
Organisation der Systemaktivitäten in JCS/VS 8



Systemunterstützte Peripherie in JCS/VS 8

Das zur Verfügung stehende Memory wird auf das System und die Benützeraktivitäten aufgeteilt.

Das System selbst belegt inklusive aller System Tasks 32K. Weitere 128K stehen den 8 User-Programmen zur Verfügung, wobei es vier Memory-Bänke zu je 32K gibt, die jeweils einen (volle 32K) oder 2 User aufnehmen können.



Memory Aufteilung in JCS/VS 8

Abgebaut wurde der Zugriff zu einer Festkopfplatte (längst nicht mehr rentabel) und zu variablen Card Codes und für Lochstreifen Ein/Ausgabe.

Hauptziel war es, daß jeder von seinem Arbeitsplatz aus die Programmentwicklung per Terminal ohne Systembelegung durchführt und von dort aus die Anweisungen zur Programmexekution gibt. Das Source Processing wird von JCSSOP, einem System Task, betrieben, der für alle 8 Terminal User insgesamt nur 7K Worte benötigt, wobei es keinerlei Limitierung der angesprochenen zu editierenden Files gibt.

Insgesamt bringt das neue System nicht nur eine enorme Leistungssteigerung, sondern auch eine beträchtliche Durchsatzsteigerung.

ÜBERBLICK ÜBER DIE EIGENSCHAFTEN DES SYSTEMS

- Real-Time Hybrid: Das System unterstützt voll den AutoPATCH-Betrieb mit HYBSYS und übernimmt alle entsprechenden hybriden Systemoperationen.

- Multiprogramming: Das System hat 2-18 gleichzeitig aktive Exekutionslevel. Acht Level sind durch User Jobs frei programmierbar, und ein Level ist für Systemprogramme mit beschränktem Kernspeicherplatz, die durch Overlays ausgetauscht werden können (Supervising).

- Batch Processing: Der Card Reader ist immer ready und liest jeden Job sofort ein. Die Aufteilung der Jobs auf die User Partitions zur Exekution im Kernspeicher erfolgt automatisch, kann aber auch händisch gesteuert

werden. Jeder Job wird auf einen File gelegt und kann daher öfter exekutiert und editiert werden, ohne neu eingelesen werden zu müssen.

- Source Processor: Dieser ermöglicht parallel zum Batch-Betrieb an maximal 8 Terminals das Erstellen, Editieren und Vorbereiten von Programmen und Unterprogrammen sowie die Bearbeitung von Daten. So können beliebige Kartenstapel verarbeitet und über eine Option als File zur Verfügung gestellt werden. Sämtliche Funktionen wirken virtuell auf die Informationen des Files und benötigen daher kein Memory. Es können vom Terminal direkt Jobs zur Exekution freigegeben werden.

- Districts: JCS/VS 8 liefert bis zu 50 virtuelle Massenspeicher, die logisch den vorhandenen Massenspeichern zugeteilt werden. Diese Districts sind flexibel dimensionierbar. So kann man auch mit einer einzigen Platte mit 21 offenen Files arbeiten. Die Districts sind zylindrisch aufgebaut, also ohne Anfang und Ende, und sind laufend belegbar. Die Districttransfers sind interruptgesteuert.

- Peripherie: Sämtliche peripheren Geräte sind über Kernspeicherpuffer laufend aktiv und sind durch logische Units programmierbar, die vom System den tatsächlichen Geräten zugeordnet werden. Austausch und Umorganisation der Geräte sind jederzeit möglich. Alle Units sind ASCII-standardisiert und werden interruptgesteuert: Card Reader, zwei Printer, Data Plotter, Betriebsvideokonsole, acht weitere Terminals über V24.

- Spooling: Line Printer, Data Plotter und eine dritte Spool Device werden im Scratch/Spooling Verfahren bedient. Sämtlicher Dutput auf diesen Geräten wird auf Files gelegt. Nach dem Ende der Exekution des Jobs ist der File automatisch oder händisch abrufbar. Mehrfaches Spoolen eines Files ist möglich, ohne den Job neu zu rechnen. Das Spoolen bringt kurze Exekutionszeiten und die Möglichkeit, die betroffenen Geräte zugleich von mehreren Exekutionsleveln anzusprechen.

- Datenschutz: Es können Files schreibgeschützt werden, oder es wird nur einem bestimmten Benützer der Zugriff zu einem bestimmten File gestattet. Ferner ist der Datenschutz eines bestimmten Files durch die Programmierung möglich.

- Job Accounting: Es wird vom System automatisch die Aktivität jedes Jobs kontrolliert und aufgezeichnet: verbrauchte CPU-Zeit, Anzahl der Aufrufe der einzelnen Prozessoren, Filezugriffshäufigkeit, Kernspeicherbelegung, Fehlerabbrüche, sowie der gesamte Betriebsmittelverbrauch auf jedem einzelnen Input/Dutput-Gerät. Das Job-Accounting läuft automatisch mit und wird durch eigene Unterstützungsprogramme kontrolliert und ausgewertet. Es besteht die Möglichkeit der Rechenzeit- und Betriebsmittelbeschränkung für jeden einzelnen Job (Programmsperre oder Programmabbruch).

- Dynamische Memoryeinteilung: Die Memoryeinteilung für die acht User Partitions wird je nach Jobanforderungen neu erstellt und während des Rechenbetriebs laufend verändert. Vollautomatische Kontrolle und Programmschutz sind selbstverständlich.

- Run-Time-Messages: Hunderte Systemmeldungen und Fehlermeldungen sind auf einem eigenen Systemfile resident und werden gegebenenfalls zum Run-Time-Zeitpunkt dem Benützer am Terminal bzw. dem Job mitgeteilt. Ebenso wird der Operator auf der Systemkonsole informiert. Je nach Problem wird der Job abgebrochen oder weitergerechnet. Alle Meldungen sind programmierbar.

- Supervising: Über System Overlays sind Kontrollmoduln für den Operator verfügbar, die die Kontrolle und die Funktion des Betriebes gewährleisten. So kann eine ständige optische Kontrolle der laufenden Aktivitäten erfolgen. Jobs können manuell gestartet, gestoppt oder abgebrochen werden.

Datum und Tageszeit werden angezeigt. Es können Wartepositionen im Programm mit Meldungen an den Operator (z.B. für Bandwechsel) programmiert werden, die bestätigt werden müssen, ehe das Programm fortsetzt. Es gibt die Möglichkeit, Programme zu tracen. Ferner sind folgende District-Operationen vorgesehen: File-Transport, automatisches Löschen und Platzoptimierung auf Districts, manuelles Löschen, Closen, Schützen und Umbenennen von Files. Ein Systemgenerator ermöglicht die Kontrolle der Systemparameter, Dokumentation der aktuellen Konfiguration und der System Release und die temporäre Veränderung von Systemparametern. Eine Kontrolle über alle Jobnummern gewährleistet ein weiterer Overlay.

- o System Traps: Das System JCS/VS bietet selbst intern eine Reihe leistungsfähiger Funktionen, die mnemonisch im Assembler kodiert werden können (z.B. Hybrid-, File- und Input/Output-Programmierung).

- o Options: Zahlreiche textfreundliche Options können zusammen mit den Steuerkarten einzelne Funktionen des Jobablaufs steuern. Alle Kommandos und Texte zur Steuerung sind informationsfreundlich und werden verständlich ausgeschrieben.

- o Compiler: Es sind FORTRAN IV, Assembler, Core Image Generator und der Anschluß an den hybriden Prozessor HYBSYS verfügbar.

- o File Control: Diese ermöglicht die Kontrolle und Manipulation von Files auf Districts parallel zu Batch Usern von jedem Terminal aus.

- o Bootstrap: Über Bootstrap kann das System JCS/VS automatisch geladen und gestartet werden (Hardware-PACER-Bootstrap-Prozedur). Das System ist ladbar von Card Reader, Moving Head Disk oder Magtape und generiert bzw. lädt sich zum Großteil selbst, je nach Systemkonfiguration.

- o Doppelte Integer Genauigkeit: wird durch eine entsprechende Software (einschließlich variables Format) ermöglicht.

- o Software zur Unterstützung von häufig gebrauchten Programmstrukturen: Bit-Manipulationen, Wort-, Halbwort- und Viertelwort-Manipulationen, Packen und Entpacken. Alle Routinen sind FORTRAN-aufrufbar.

- o Kontrollprogramme existieren zur Job-, System- und Funktionskontrolle, sowie für Save-Läufe und Dumps. Es gibt Software zum Fusionieren von Objekts zu einem Sammelfile und zum Modifizieren fertiger Programme auf Maschinencodeebene und viele weitere Unterstützungsprogramme.

- o Plotter Software: Zur Steuerung des Data Plotters oder eines graphischen Bildschirms gibt es FORTRAN-aufrufbare, CALCOMP-kompatible, geschwindigkeitsgeregelte, bewegungsoptimierte leistungsfähige Software. Es besteht die Möglichkeit der Softcopy einer an einem graphischen Bildschirm erstellten Zeichnung am Data Plotter auf Tastendruck.

- o Terminalunterstützung: JCS/VS 8 berücksichtigt die Mixed Hardware der Terminal-Typen sowie wahlweise System Echo oder Local Echo; Rubout von Ziffern und Buchstaben wird systemmäßig unterstützt.

- o Virtuelle Daten: Es können von FORTRAN aus Datenfelder virtuell angelegt und so Memory-Platz gespart werden.

- o Virtuelles System: JCS/VS 8, Batch Processor, Source Processor und File Controller sind reentrant.

- o Terminal Identification: Standardmäßig wird Terminal I/O auf das Gerät gelegt, an dem sich der Benützer befindet, unabhängig von der programmierten Terminal Device. Alle Terminals sind ständig an das virtuelle System angeschlossen.

A. Blauensteiner

Änderungen und Verbesserungen für den FORTRAN-Programmierer durch die Umstellung auf das Betriebssystem JCS/VS 8

Im Zuge der Umstellung auf das neue Betriebssystem JCS/VS 8 wurden in der Systemsoftware zahlreiche Änderungen und Verbesserungen durchgeführt. Für den Benutzer ergibt sich daher die Notwendigkeit, seine Programme auf das neue System bzw. die neue Systemsoftware umzustellen. Die folgenden Punkte sind dabei zu beachten:

Core Image-Programme

Core Image-Programme, die im System JCS/TS 7 (d.h. vor Anfang März) gebunden wurden, können nicht mehr gerechnet werden und müssen neu erzeugt werden. Objekt-Programme sind, sofern sie nicht von den folgenden Änderungen betroffen sind, kompatibel, sie sollten jedoch sicherheitshalber ebenfalls neu übersetzt werden.

Binärkarten

Die Eingabe von Binärkarten ist nicht mehr möglich. Zur Übertragung der Daten von der Digitalrechenanlage zur Hybridrechenanlage müssen Magnetbänder verwendet werden. Programmbeschreibungen für die entsprechenden Routinen sind in der Programmberatung erhältlich.

Zweiter Drucker

Als zusätzlicher Drucker steht der Matrixdrucker TALLY T1612 zur Verfügung. Der Drucker kann mit der FORTRAN-Device-Nummer 16 angesprochen werden. Er verfügt über 6 verschiedene Schreibdichten, die programmgesteuert gewählt werden können, wodurch es z.B. möglich ist, auf dem an der Hybridrechenanlage üblichen 80-Spalten-Papier bis zu 136 Zeichen/Zeile (exkl. Vorschubzeichen) auszugeben. Programmbeschreibungen zum Setzen der Schreibdichte sind erhältlich. Bei Verwendung dieses Druckers muß auf der /RUN-Steuerkarte die Option SPOOLING angegeben werden (z.B. /RUN,SPOOLING).

FORTRAN-Device-Nummern

Die FORTRAN-Device-Nummern für die wichtigsten Ein/Ausgabe-Einheiten sind im System JCS/VS 8 wie folgt:

	Eingabe	Ausgabe
Card Reader	15	-
Line Printer	-	15
Print Device (TALLY)	-	16
Terminal 3 (Graphic Display)	20	20
Nodevice	14	14

Ausgabe auf Terminals

Die Ausgabe mit einer FORTRAN-WRITE-Anweisung erfolgt nun auf allen Ausgabeeinheiten in gleicher Weise wie am Line Printer: Das erste Zeichen wird als Vorschubzeichen angesehen, das den Zeilenvorschub regelt. Zulässige Vorschubzeichen sind:

+	kein Vorschub
␣	nächste Zeile
Ø	eine Leerzeile
-	zwei Leerzeilen
1	neue Seite

Am Terminal bedeutet dabei der Vorschub auf eine neue Seite, daß der Bildschirminhalt gelöscht wird.

Bei der Ausgabe auf einem Terminal ist zu beachten, daß die Ausgabe von Leerzeichen als Zeichenkette das Löschen von eventuell vorhandenen Zeichen in den entsprechenden Spalten bewirkt, während das X-Format nur den Cursor bewegt und vorhandene Zeichen nicht löscht.

H-Format

Das Abschließen des H-Formats in einer FORTRAN-FORMAT-Anweisung durch einen Apostroph ist nicht mehr möglich. Die Anzahl der Zeichen des Strings muß entweder genau angegeben werden, oder der String muß durch zwei Apostrophe begrenzt werden:

```
z.B.    FORMAT(24H BETRIEBSSYSTEM JCS/VS 8)
        FORMAT(' BETRIEBSSYSTEM JCS/VS 8')
```

Library-Routinen

Die JCS/TS 7 Routinen INTEG und STRING sind durch die Routinen INTIO und TEXTIO (mit geänderter Parameterliste) zu ersetzen.

Die Routinen LINES und CLEAR (früher PAGE) wirken auch auf Terminals und haben als zusätzlichen Parameter die FORTRAN-Device-Nummer.

Mit der neuen File-Routine POSI kann geprüft werden, ob ein bestimmter File auf einem District existiert. Dadurch kann vermieden werden, daß beim Positionieren eines Files dieser File neu (mit nicht definiertem Inhalt) errichtet wird, falls es ihn aus irgendeinem Grund nicht gegeben hat.

Neue Systemroutinen

Zusätzlich zur Routine SYSØ34, die bei jeder Eingabe vor dem Carriage Return einen Beistrich einfügt, steht jetzt die Routine SYSØ73 zur Verfügung, die den Aufruf von SYSØ34 rückgängig macht.

Die Routine SYSØ22 liefert den Jobnamen, der auch auf der Titelseite des Line Printer Ausdrucks und in der Kennzeichnung von Plotterzeichnungen aufscheint.

Auf der /RUN-Steuerkarte kann die Option NAME=namexy angegeben werden (z.B. /RUN,NAME=FILEØ1). Der Aufruf der Routine SYSØ74 unmittelbar am Beginn eines Programms liefert den in der NAME-Option angegebenen Namen. Mit der NAME-Option und der Routine SYSØ74 kann so etwa der Name eines Datenfiles an ein Programm übergeben werden.

F. Blöser

TIME-SHARING IM BETRIEBSSYSTEM JCS/VS 8

Das Betriebssystem JCS/VS 8 ist für 8 Batch User (Programme) und 8 Terminal User (Time-Sharing) ausgelegt, wobei diese 16 User Level und 2 System Level (Supervisor und System Manager) parallel bedient werden (siehe Seite 11). Zusätzlich zu den gerade rechnenden Programmen können also bis zu 8 Benutzer im Time-Sharing-Betrieb arbeiten, ohne einen User Level und damit Memory zu belegen.

Für den Time-Sharing-Betrieb, der für die Programmentwicklung via Terminal gedacht ist, stehen der Source Processor JCSSOP und der File und District Controller JCSCON zur Verfügung. Diese beiden Betriebssystemprogramme sind reentrant, sie können von allen Terminal Usern aufgerufen werden. Außerdem können vom Terminal aus Source Files zur Exekution in die Batch Queue gestellt werden.

Beginn einer Terminal-Session

Am Terminal kann jederzeit (vor, während und nach einer Terminal-Session) mit CTL Q (CONTROL-Taste in Verbindung mit Q) abgefragt werden, ob das Terminal on-line ist. Man erhält in diesem Fall eine entsprechende Nachricht mit der Systemversionsnummer, der Terminalnummer, dem Datum und der Uhrzeit:

```
JCS/VS 8.1.xx  *  TERMINAL ON-LINE AS Ti  81/mm/dd  hh.mm
```

Eine Terminal-Session wird durch Eintippen von CTL I gestartet. Das System meldet sich mit

```
HELLO, WHO ARE YOU ?
```

und verlangt anschließend die Eingabe der Jobnummer des Benutzers. Danach wird sofort der Source Processor aktiviert.

Source Processor JCSSOP

Der Source Processor JCSSOP ermöglicht das Generieren und Manipulieren von Source Files. Ein Source File unterteilt sich in Source Lines, die maximal je 80 Zeichen enthalten können, wobei die Informationen in den Source Lines Daten für einen Programmlauf, ein Programm oder Steuerbefehle sein können. Ein Source File kann also auch die Informationen für einen ganzen Job (vom /JOB- bis zum /END-Steuerbefehl) enthalten. Soll ein bereits existierender Kartenstapel auf einem Source File abgespeichert werden, so braucht nur eine Karte

```
/SOURCE nameso
```

vor den Kartenstapel gelegt werden und das ganze Kartenpaket am Card Reader eingelesen werden. Das Betriebssystem erzeugt dann aus den nachfolgenden Lochkarten (bis inkl. der /END-Karte) einen Source File mit dem Namen nameso.

Source Processor-Befehle, die generell mit dem Zeichen "\" beginnen, können immer dann eingetippt werden, wenn sich der Source Processor mit

```
SOP:
```

gemeldet hat. Bei jedem richtigen Befehl wird das Befehlswort wiederholt, bei fehlerhafter Befehlseingabe wird

```
SOP COMMAND FAILED
```

ausgeschrieben. Mit der Taste DELETE können falsche Zeichen gelöscht und somit Tippfehler korrigiert werden.

Nach dem Einstieg in den Source Processor muß immer zuerst der District festgelegt werden, auf dem sich der zu bearbeitende Source File befindet. Für die Benutzer stehen zum Abspeichern die Districts 20-24 zur Verfügung. Neben der oben erwähnten Generierung eines Source Files durch Einlesen eines Kartenpakets kann ein neuer Source File natürlich auch durch Eintippen der einzelnen Lines am Terminal erzeugt werden. Außerdem ist es möglich, aus mehreren Source Files bzw. aus mehreren Teilen verschiedener Source Files einen neuen File zu generieren.

Viele Befehle von JCSSOP beziehen sich nur auf einen eingeschränkten Bereich des Source Files. Dieser Bereich umfaßt nach der Positionierung eines Source Files den gesamten File. Der Bereich kann durch Angabe der entsprechenden Line-Nummern eingeschränkt werden und im Extremfall eine einzige Line umfassen. Der Bereich kann aber auch in der Weise eingeschränkt werden, daß die Informationen, die in der ersten bzw. letzten Zeile des Bereichs stehen, angegeben werden. In einfacher Weise kann ein einmal gesetzter Bereich innerhalb eines Source Files nach vorn oder nach hinten verschoben werden.

Beim Listen werden üblicherweise alle Source Lines des gesetzten Bereichs ausgegeben. Es kann aber auch eine Maskierung vorgenommen werden, wobei nur jene Lines gelistet werden, die eine bestimmte Zeichenkette (z.B. einen Variablennamen) enthalten.

Zum Ändern eines Source Files gibt es die verschiedensten Befehle. Es können einzelne Lines gelöscht, geändert oder hinzugefügt werden. Außerdem können alle Lines eines Bereichs gelöscht, durch neu einzutippende Lines ersetzt oder an eine andere Position innerhalb des Source Files verschoben werden. Zusätzlich ist das Ersetzen einer bestimmten Zeichenkette durch eine andere (auch unterschiedlich lange oder leere) Zeichenkette möglich.

Ein spezieller Befehl erlaubt das Abschicken einer Nachricht an die Operator-Konsole. Erst wenn der Operator die Meldung zur Kenntnis genommen hat, kann in JCSSOP weitergearbeitet werden.

Soll ein Programm gerechnet werden, so kann mit dem Befehl

```
\BATCH:nameso
```

der Source File nameso vom angewählten District zur Exekution in die Batch Queue gestellt werden. Der Source File muß mit der /JOB-Line beginnen und mit /END abschließen und dazwischen Steuerbefehle und alle von den aufgerufenen Prozessoren und Programmen benötigten Informationen (z.B. FORTRAN-Programm, LOAD-Karten, Daten) enthalten. Sind diese Informationen auf eigenen Source-Files gespeichert, so müssen diese Source Files durch die INPUT-Option

```
INPUT:namess=d
```

auf den entsprechenden Steuerbefehlen angegeben werden.

Ein Beispiel dafür wäre etwa /FOR,INPUT:HAUPT=24.
Das FORTRAN-Programm ist in diesem Fall auf dem Source File HAUPT auf District 24 abgespeichert. Der File HAUPT darf aber außer dem Programm keine Informationen enthalten, da der gesamte File als Eingabe für den FORTRAN-Compiler genommen wird. Nach der /FOR-Line muß dann sofort der nächste Steuerbefehl (etwa /BUILD) folgen, da nach der Übersetzung aller Lines des Files HAUPT mit der Verarbeitung des nächsten Steuerbefehls im ursprünglichen, in die Batch Queue gestellten File weitergegangen wird.

JCSSOP-Befehle:	Steuerlines (Source File SCOMF auf District 23):	FORTRAN-Programm (Source File HAUPT auf District 24):
	/JOB,USER=jnr	REAL X(5Ø)
\DISTRICT:23	/FOR,INPUT:HAUPT=24	.
\BATCH:SCOMF	/BUILD	.
	LOAD,D21 UP	.
	/RUN	STOP
	/END	END

In entsprechender Weise können auch Daten, die sich auf einem Source File befinden, bei der Exekution eines Programms (durch Angabe der INPUT-Option beim /RUN-Befehl) verwendet werden.

Sobald ein Job, der in die Batch Queue gestellt worden ist, auf einem User Level mit der Exekution beginnt, erfolgt eine Meldung am Terminal. Entsprechende Meldungen erfolgen auch beim Starten eines Prozessors (FORTRAN-Compiler, Assembler, Core Image Generator) und beim Starten eines Benutzerprogramms. Arbeitet ein solches Programm mit Ein/Ausgabe auf einem Terminal, so wird automatisch die Ein/Ausgabe auf jenes Terminal verlegt, von dem der Job in die Queue gestellt worden ist, unabhängig davon, welche Terminal Device Nummer programmiert ist. Bei Ein/Ausgabe am Terminal ist es jedoch notwendig, für die Zeit der Exekution des Programms die Eingabe für JCSSOP bzw. JCSCON (auf dem Terminal User Level) zu sperren, da sonst sowohl das Programm als auch der Source Processor oder File Controller zur selben Zeit Eingabe erwarten. Das Sperren geschieht durch Angabe der Option TERMINAL beim /RUN-Befehl (z.B. /RUN,TERMINAL) und wirkt für die Dauer der Programmexekution (bis zum nächsten Steuerbefehl). Beim /FOR- und /BUILD-Befehl ist diese Option nicht sinnvoll, da diese Prozessoren standardmäßig keine Terminaleingabe erwarten, es kann also während dieser Zeit ohne weiteres weiter an Programmen editiert werden.

File und District Controller JCSCON

Der File und District Controller JCSCON ermöglicht Manipulationen mit Files auf den Districts. Er wird vom Source Processor aus durch Eintippen von CTL C aktiviert. Die Eingabe eines Befehls kann immer erfolgen, wenn sich der File Controller mit

SELECT TASK:

am Terminal gemeldet hat. Die Befehle werden durch Eintippen des Anfangsbuchstabens eingegeben. Der File Controller schreibt dann den vollen Befehlstext aus. Falsch eingetippte Zeichen von Filenamen oder District-Nummern können mit DELETE gelöscht werden.

Mit JCSCON können Filelisten aller Files eines bestimmten Districts erstellt werden. Es werden dabei unter anderem die Größe, das Generierungsdatum und der Typ der einzelnen Files ausgegeben. Zusätzlich werden Informationen über den District selbst geliefert, wie z.B. die Anzahl der eingetragenen Files und der freie Platz auf dem District.

Es können mit dem File Controller einzelne Files auf einem District gelöscht werden oder umbenannt werden. Außerdem besteht die Möglichkeit, einen File mit einem Schreibschutz und/oder einem Zugriffsschutz zu versehen. Damit kann vermieden werden, daß der File überschrieben wird, bzw. sichergestellt werden, daß der File nur von Jobs mit der gleichen Jobnummer wie der, unter der er errichtet wurde, angesprochen werden kann.

Durch Eintippen von CTL I kann vom File Controller aus wieder der Source Processor JCSSOP aktiviert werden, wobei alle gesetzten Informationen (District, Source File, Bereich, etc.) voll erhalten sind.

Beendigung einer Terminal-Session

Eine Terminal-Session kann standardmäßig maximal 2 Stunden dauern. Danach wird die Session automatisch beendet, wobei die laufende Aktivität des JCSSOP oder JCSCON noch durchgeführt wird und erst zu dem Zeitpunkt abgebrochen wird, zu dem die nächste Eingabe vom Benutzer erwartet wird.

Sowohl der Source Processor als auch der File Controller können vor Erreichen des Zeitlimits durch Eintippen von CTL O beendet werden. Am Terminal wird

THANK YOU, GOOD BYE !

ausgeschrieben. Damit ist der Time-Sharing-Betrieb für den betreffenden User beendet. Ein Benutzerprogramm mit Ein/Ausgabe auf dem Terminal kann aber selbstverständlich weitergerechnet werden. Daher sollten Benutzer, die den JCSSOP nur zum Starten eines interaktiven Programms (mit dem \BATCH-Befehl) verwenden und während der Exekution des Programms nichts im JCSSOP oder JCSCON rechnen können, den Terminal Level durch CTL O freigeben.

Abschließend sei darauf hingewiesen, daß genaue Beschreibungen des JCSSOP und JCSCON in der Programmberatung erhältlich sind. Für die Benutzerterminals, die zum Editieren von Programmen und Daten sowie zur Programmvorbereitung zur Verfügung stehen, können beim Operator Reservierungen vorgenommen werden.

F. Blöser

QUEST UND DIALOG LITERATURDATENBANKEN

Es wurde bereits mehrmals in den früheren Exemplaren von INTERFACE auf die Möglichkeit des Zugriffs zu Literaturdatenbanken an der Abt. Hybridrechenanlage hingewiesen. Mit 1981 haben wir die ausschließliche Betreuung von Kunden der Universität und der Technischen Universität übernommen, die früher auch von der Österreichischen Gesellschaft für Sonnenenergie und Weltraumforschung, ASSA, wahrgenommen wurde. Um einen kurzen Überblick über die Aktivitäten im ersten Quartal 1981 zu geben, seien im folgenden die in diesem Zeitraum betreuten Institute aufgeführt.

Institut	Thema
Institut für Feinwerktechnik	Räumliche Positionierungssysteme Positionsbestimmung
Institut für Hochfrequenztechnik	Laser Phase Locked Loops Costas Loops
Institut für Technische Mathematik	Darstellung, Struktur und asymptotisches Verhalten bestimmter Funktional-Differentialgleichungen
Institut für Angewandte Informatik und Systemanalyse	Schichtenmodelle virtueller Betriebssystemarchitektur

Außerdem wurde auf dem Gebiet der Satellitendynamik recherchiert.

Weitere Institute haben bereits die Bewilligung zur Durchführung von Suchläufen erhalten.

Für nähere Informationen wenden Sie sich bitte an die Abt. Hybridrechenanlage des EDV-Zentrums der Technischen Universität Wien, Tel.: 65-37-85/901 DW.

H. Hummer

Testen von HYBSYS-Makros mit Hilfe des Overlays CHECK

Es war naheliegend, nach Erweiterung des hybriden Prozessors HYBSYS in Richtung auf die automatische Erstellung von Schaltverbindungen mit Hilfe der Switch Matrix ein Testprogramm für den Analogrechner zu schaffen, das diese komfortable Software zum Prüfen der Analogrechenelemente nützt.
Wie bereits im Artikel "HYINFO - Aktuelle Informationen für Hybridbenützer" (INTERFACE 8) beschrieben, wurde die Evidenz über Fehler dieser Elemente bisher mit Hilfe der Benützer und deren Eintragungen in eine Fehlerliste geführt.
Die Behebung der Fehler wurde dann durch Mitarbeiter der Hybridrechenanlage durchgeführt. Das Erkennen der Fehler durch die Benützer erfolgte somit mehr oder weniger zufällig, je nach der Zahl der benützten Rechenelemente und der Benützungsfrequenz des Rechners. Die Hybridrechenanlage führte zusätzlich mit Hilfe fest verschalteter Steckbretter Tests durch. Da allerdings für jeden dieser elementspezifischen Tests ein eigenes Steckbrett erforderlich war und die Tests selbst sehr zeitraubend waren, da man sie nicht wirkungsvoll automatisieren konnte, und da außerdem während der Testzeiten kein normaler Rechenbetrieb für andere Hybridbenützer möglich war, wurde in der Praxis nicht so oft geprüft, wie wünschenswert gewesen wäre.

Um diese Situation zu bessern, wurde ein HYBSYS-Overlay geschrieben. Dieser verwendet die allen HYBSYS-Benützern zur Verfügung stehende Software zur Erzeugung von Overlays, deren Verwendung im HYBSYS-Manual genau beschrieben wird.
Die Prüfung erfolgt durch den Vergleich analog erzeugter Lösungen mit digital errechneten Vergleichsfunktionen bei den zu prüfenden HYBSYS-Makros.
Die Zahl der gesetzten Diskretisierungspunkte POINTS in HYBSYS wird mitberücksichtigt, und es werden mit dieser Zahl gemittelte Durchschnittswerte der Differenzen zwischen den beiden Funktionen an den Stützstellen und der an diesen Stellen aufgetretenen absolut größten Abweichungen gebildet und mit den Fehlerschranken CEPS und DEPS verglichen. CEPS bezieht sich dabei auf den Mittelwert der Differenzen, DEPS auf die absolut größte Abweichung.
Diese Variablen sind normale digitale HYBSYS-Parameter und können vom Benützer ausgelesen und verändert werden. Der Overlay CHECK erlaubt folgende Syntax:

a) CHECK

b) CHECK $var_1, var_2, var_3, \ldots, var_n$ $\quad 1 \leq n \leq 10$

var_i analoge Variable im Sinne von HYBSYS

c) CHECK $iop_1, iop_2, iop_3, \ldots, iop_n$ $\quad 1 \leq n \leq 10$

iop_i Operationscode eines Makros im Sinne von HYBSYS

Ad a) Es werden alle Makros in der gesetzten Schaltstruktur geprüft.

Ad b) Es werden die den Variablennamen var_i in der gesetzten Schaltstruktur zugeordneten Makros geprüft.

Ad c) Es werden alle Makros in der gesetzten Schaltstruktur geprüft, die einen der Operationscodes $iop_1,\ldots,iop_n$ haben.

Es gilt dabei jene Auswahl aus der Menge aller in HYBSYS definierten Makros, welche mit dieser Testmethode sinnvoll geprüft werden können, und, falls dies zutrifft, bereits in den Test aufgenommen wurden.

Dabei handelt es sich derzeit um folgende Makros:

INT	Integrierer	SQRT	Quadratwurzel
SUM	Summierer	ABS	Absolutbetrag
MULT	Multiplizierer	SIN	Sinus
SQR	Quadrierer	COS	Cosinus
DIV	Dividierer		

In der Argumentliste von CHECK dürfen Variablennamen und Operationscodes nicht gemischt auftreten, damit ein mehrfaches Prüfen ein und derselben Makros in einem Aufruf von CHECK verhindert wird.
Wird in der Parameterliste ein Operationscode iop in einer Liste von Variablen var_i festgestellt, erfolgt die Meldung

UNDEFINED VARIABLE iop

Wird in einer Liste von Operationscodes eine Variable var angeführt, so erfolgt die Meldung

UNDEFINED OPERATION var

Die fehlerhafte Zeile wird bis zur Position des Fehlers ausgeschrieben, und an dieser Stelle kann mit der Eingabe fortgefahren werden.
Wird in einer Variablenliste eine Variable var vom Typ DIGITL im Sinne von HYBSYS angeführt, erfolgt die Meldung

VARIABLE NOT TYPE ANALOG var

Gibt es eine in der Argumentliste angeführte Variable in der gesetzten Schaltstruktur nicht, so erfolgt die Meldung

UNDEFINED VARIABLE var

Gibt es einen angeführten Operationscode nicht, so erfolgt die Meldung

UNDEFINED OPERATION iop

Nicht im Test implementierte, in der Schaltung aber auftretende Makros werden bei ihrem Anführen in der Argumentliste von CHECK ohne Fehlermeldung akzeptiert und beim Testen übergangen.
Wird ein Operationscode angeführt, der im Test implementiert, in der Schaltung aber nicht enthalten ist, erfolgt die Meldung

PATCHING DOES NOT CONTAIN MACRO iop

Es wäre denkbar, beim Prüfen der Makros mit Testfunktionen zu arbeiten, welche die Prüfung des zu behandelnden Makros z.B. bei beliebigen gerade gesetzten Eingangsverbindungen des Rechenelementes erlaubten. Dies hätte den Vorteil, daß die gerade bearbeitete Schaltstruktur nicht durch den Test zerstört werden müßte. Durch derartige Anforderungen würde das Programm aber wesentlich kompliziert werden, und CHECK wurde in seinen Funktionen daher auf die Hardware-Testschaltung MACTST ausgerichtet, über die bereits in INTERFACE 15/16 unter dem Titel "68Ø-Elemente Test mit AutoPATCH-HYBSYS" berichtet wurde.

Für jeden Makro, der bei der Prüfung Abweichungen in mindestens einem der beiden Testkriterien liefert, welche die mit CEPS und DEPS vorgegebenen Grenzwerte überschreiten, wird eine Meldung

var iop f1 f2 BAD ELEMENT

ausgeschrieben, wobei f1 und f2 die ermittelten Fehler des geprüften Elementes mit den Schranken CEPS und DEPS bedeuten.
Wurden CEPS und DEPS nicht vor der Verwendung von CHECK deklariert, wird ein Hinweis in Form einer entsprechenden HYBSYS-Fehlermeldung ausgeschrieben.
In diesem Fall wird mit internen Defaultwerten Ø gearbeitet, alle geprüften Makros mit ihren Testergebnissen werden ausgeschrieben.

Die Vorgangsweise beim Verwenden von CHECK ist folgende:

USDSTR=y;	Speichern der bearbeiteten Schaltung x auf
STORE,x;	District y
USDSTR=19;	Holen der Testschaltung MACTST von District 19
GET,MACTST;	
DECLAR PAR:CEPS=c,DEPS=d;	Deklarieren der Fehlerschranken mit Wertzuweisung
PREPAR;	Routen der Testschaltung
GET,OV,CHECK;	Holen des CHECK-Overlays
CHECK SIN,COS;	Prüfen der Sinus- und Cosinusmakros
CHECK U5,U6,U7,U8;	Prüfen der U5 bis U8 zugeordneten Makros
USDSTR=y;	Wiederherstellung der ursprünglich
GET,x;	bearbeiteten Schaltung
PREPAR;	

Da aus der graphischen Darstellung der von den eventuell defekten Rechenelementen gelieferten Funktionen Rückschlüsse auf die Art der Fehler gezogen werden können, kann man die digital berechnete Referenz- und die Fehlerfunk-

tion auch zeichnen. Diese werden von CHECK unter den fest definierten Namen DIG und FEL behandelt, und sie sind vom Benützer als DACF-Funktionen dieses Namens zu deklarieren, falls er sie verwenden will.

Dazu muß wie folgt vorgegangen werden:

```
USDSTR=y;
STORE,x;
USDSTR=19;
GET,MACTST;
DECLAR DACF:DIG,FEL;          Deklarieren der Referenz- und der Fehlerfunktion
DECLAR PAR:CEPS=c,DEPS=d;
PREPAR;
GET,OV,CHECK;
CHECK SQR1;                   Prüfen z.B. des Quadrierers SQR1
PLOT SQR1,DIG,FEL;            Zeichnen von SQR1, der Referenz- und der Fehler-
                                                                  funktion
USDSTR=y;
GET,x;
PREPAR;
```

Die nach dem PREPAR-Befehl bei der Verwendung von DIG und/oder FEL ausgeschriebene Fehlermeldung

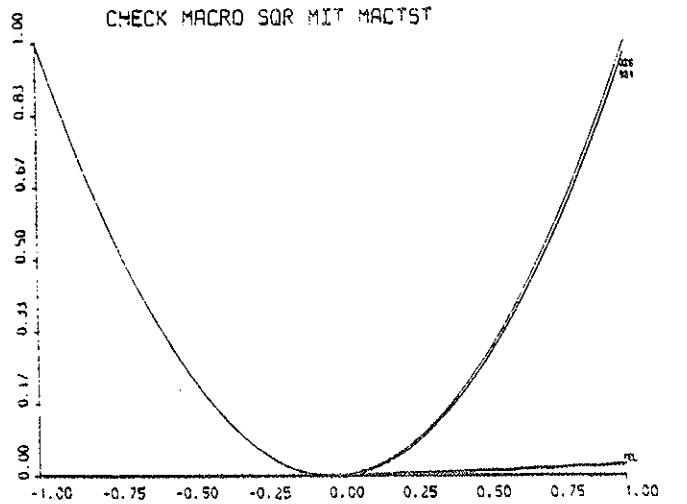
NO CONNECTION TO $\begin{matrix}\text{DIG}\\\text{FEL}\end{matrix}$ DEFINED

hat in diesem Fall nichts zu bedeuten. Erfolgt die Meldung

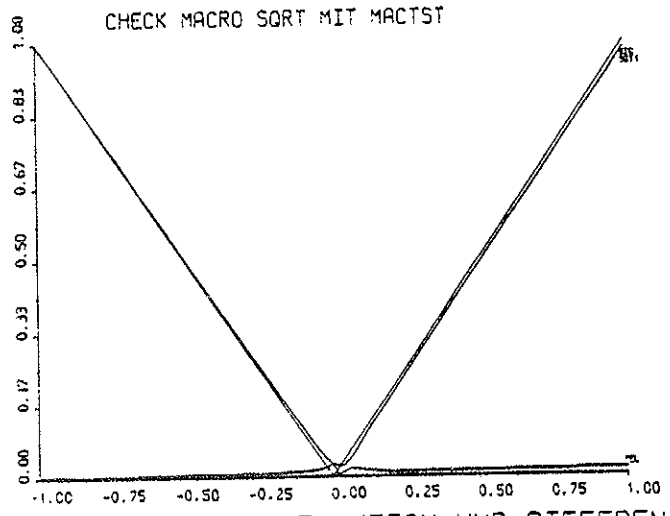
NO FREE MACRO iop

ist die Programmberatung aufzusuchen.

Mit einer der obigen Command-Folge äquivalenten Vorgangsweise erhält man bei Prüfen eines SQRT-und eines SQR-Makros folgende Zeichnungen, die mit GRAFIC=3 und Drücken der COPY-Taste auf den Plotter übertragen wurden.



SQUARER, DIGITALE FUNKTION UND DIFFERENZ



SQUARE ROOT, DIG. FUNKTION UND DIFFERENZ

H. Hummer

Die Neukonzeption des Terminalnetzes der Hybridrechenanlage und die neue Umschaltbox

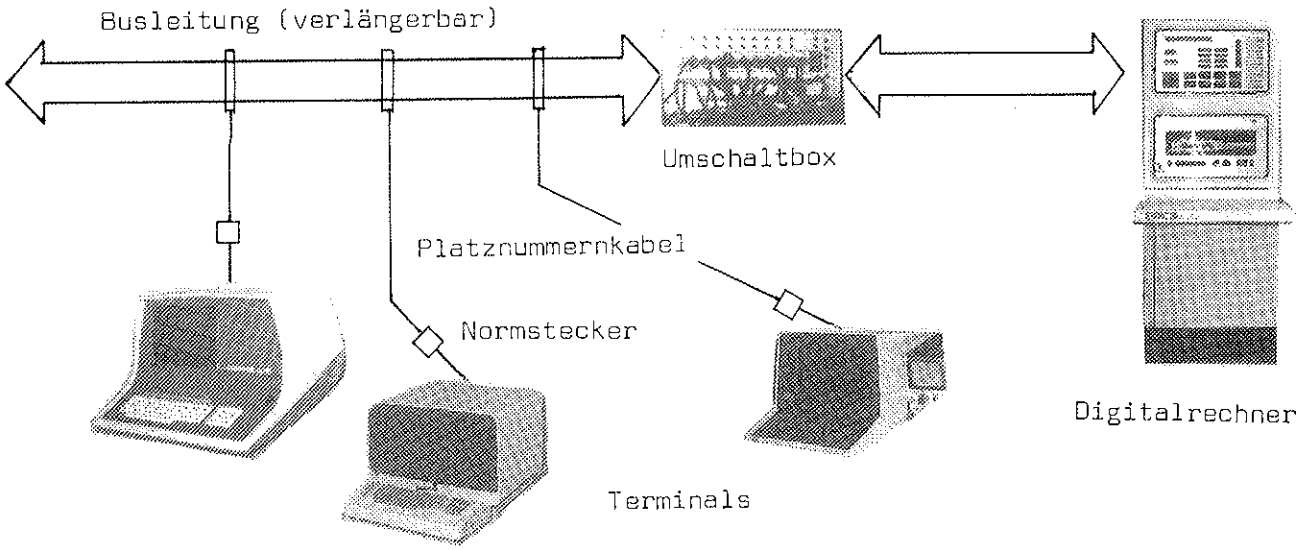
In INTERFACE 11 wurde unter dem Titel "Switch Panel zur Realisierung verschiedener Rechnerverbindungen mit vorhandener Hardware" eine Umschaltbox beschrieben, die an der Hybridrechenanlage zum Betrieb von Terminals an Controllern benutzt wurde, wobei die Anzahl der Terminals größer war als die Anzahl der Controller.Damit wurde die Bildung verschiedener Terminalkonfigurationen mit Verbindungen zu anderen Instituten und Rechnern ermöglicht.

Die Einrichtung erlaubte allerdings nur den Aufbau vorgeplanter und vorverdrahteter Konstellationen mittels Vierstufenschaltern. Einer der Hauptnachteile dieses Konzepts bestand in der Unübersichtlichkeit und Störanfälligkeit der diskreten Verdrahtung im täglichen Betrieb, und zwar insbesondere während der in letzter Zeit durchgeführten Geräteumstellungen im Maschinenraum, während derer die Kabelbäume unerwartet hohen Zugbelastungen ausgesetzt waren.

Die Verbindungen von der Schalteinrichtung zu den Terminals und den Controllern wurden durch je ein vierpoliges Kabel pro anzuschließender Einheit hergestellt, und Erweiterungen oder Änderungen erforderten somit jeweils mechanische Umbauten der Schalttafel und die Verlegung neuer Leitungen, die sternförmig zu den Anschlußkabeln geführt wurden. Die geringe Flexibilität war ein weiterer Nachteil. Trotzdem hatte sich das Gesamtkonzept beim Testbetrieb und bei rasch durchzuführenden Änderungen am Terminalsystem gut bewährt, und insbesondere bei Störfällen konnte rasch Abhilfe durch Umkonfigurationen geschaffen werden.

Da im jetzt neu installierten Betriebssystem JCS/VS 8 alle Terminals virtuell betrieben werden, d.h. von der Softwareseite her losgelöst von den Hardwareeigenschaften des jeweiligen Gerätes, und für den Programmierer daher sehr komfortabel mit den gleichen Softwarebefehlen zu programmieren sind, sollten ähnliche Eigenschaften der Hardware auch in eine neue Konzeption des Terminalnetzes und der Umschaltbox bei gleichzeitiger Ausmerzung alter Fehleranfälligkeiten einfließen.

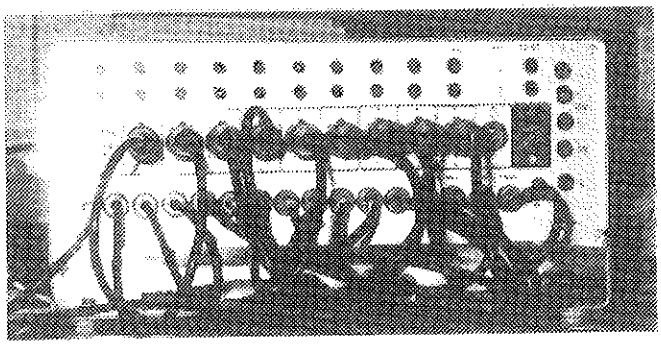
Das alte Konzept der sternförmig zu den Terminals laufenden Leitungen wurde durch eine 50-polige Flachbandleitung ersetzt, die als Ringleitung durch alle Räume verlegt wurde, in denen Anschlüsse von Terminals derzeit erforderlich sind oder erforderlich werden könnten. An diese neue Leitung sind maximal 24 Terminals anschließbar.



An den derzeit erforderlichen Terminalaufstellungsplätzen wurden in das Kabel Steckerleisten eingepreßt, die Zugang zu den im Kabel geführten Leitungen gestatten. Auf Wunsch können an beliebigen Stellen weitere solcher T-Abzweigungen eingefügt werden. Die eigentliche Verbindung zum Terminal besteht in einem 2-3m langen Kabel, das an der Terminalseite eine Miniatursteckverbindung mit Renkverschluß und auf der Seite des Flachbandkabels eine Federleiste trägt.

Durch die Anschlußbelegung der Federleiste wird das Verbindungskabel einer bestimmten, an der Umschaltbox bezeichneten Platznummer zugeordnet.
So können die Terminals örtlich beliebig umgestellt werden und erscheinen bei Mitnahme ihres auf die Platznummer "programmierten" Anschlußkabels für den Programmierer vom softwaremäßigen Ansprechen her unverändert.
Werden mehrere auf gleiche Platznummern programmierte Anschlußkabel an die Flachbandleitung angeschlossen, so kann ein und dasselbe Terminal z.B. beim Testen an verschiedenen Stellen aufgestellt werden, ohne daß das Programm geändert werden müßte. Da die Datenanschlüsse der Terminals selbst verschiedene Beschaltungen aufweisen und normalerweise terminalspezifische Anschlußkabel erfordern, wurden am anderen Ende dieser Anschlußkabel Miniatursteckverbindungen mit Renkverschluß montiert, die das Gegenstück zu den Steckverbindungen jener Kabel darstellen, welche an das Flachbandkabel angeschlossen sind.
Die Steckerbelegung der Miniatursteckverbindungen wurde nach einer selbst definierten Norm vorgenommen, in die auch die Kabelfarben einbezogen wurden.
Damit wurde die beliebige Austauschbarkeit der Terminals von der Beschaltung der Signalübertragung her erreicht.
Der Großteil aller bisher aufgetretenen Fehler konnte durch dieses übersichtliche Konzept ohne besonderen Meßaufwand gefunden werden.

Ein anderes Flachbandkabel führt von der Umschaltbox zu den Controllern des Rechners und zu Fremdrechneranschlüssen wie CYBER und VAX. Mit den Flachbandkabeln wurde außerdem eine an die Rechnerstromversorgung angeschlossene Netzleitung verlegt, die Störeinstreuungen und Spannungsschwankungen durch das An- und Abschalten rechnerfremder Geräte verhindert; vorher hatten solche Effekte zum zeitweiligen Flackern von Schirmbildern an Terminals geführt. Beide besprochenen Flachbandkabel führen in die Umschaltbox.



An der Vorderseite der Umschaltbox befinden sich die verschiedenen Gruppen von Elementen wie Steckbuchsen, Stecker und Anzeigen. Die Gruppe der Steckbuchsen ist mit den Controllern des Rechners verbunden und trägt entsprechende Nummerierungen. Die Gruppe der Stecker, die jeder für sich an kurzen Kabelstücken hängen, für die es im unteren Teil der Umschaltbox ein Ablagefach gibt, trägt Bezeichnungen, die ihre Zuordnung zu den bereits besprochenen platzmäßig programmierten Terminal-Anschlußstücken an das Flachbandkabel festlegen. Die Gruppe der Anzeigen besteht aus 24 Leuchtdioden, die an die Steckbuchsen der Controller angeschlossen sind und über eine Ansteuerungsschaltung das Vorhandensein von Signalen auf den Datenleitungen anzeigen, die der RS 232 C Norm entsprechen. Eine Überwachung des Datenverkehrs auf den Sende- und Empfangsleitungen ist damit leicht möglich. Durch Stecken einer Kabelbuchse in einen Einbaustecker wird an der Umschaltbox eine Verbindung Controller-Terminal aufgebaut. Sonderfälle wie Fremdrechner (CYBER, VAX) und Modems besitzen sowohl Kabel- als auch Einbausteckbuchsenpositionen, da sie je nach Konfiguration im Datenverkehr wie Controller oder Terminals an den Anschlußpunkten erscheinen müssen. Zudem enthält die Umschaltbox noch Testbuchsen und Stecker, die ein komfortables Einschleifen von Meßgeräten und Signalen in beliebige Verbindungen, die Vertauschung von Signalleitungen für bestimmte Zwecke, diverse Pegelüberwachungen und Auftrennen der Signalwege ermöglichen.

Für das laufende Jahr ist eine Erweiterung der Umschaltbox geplant, die die Schaltung zusätzlicher Verbindungen mit den vorhandenen Kabeln ermöglichen soll.

H. Hummer

H. Stallbaumer

COMPUTERUNTERSTÜTZTER UNTERRICHT IM HÖRSAAL X

Zur Erweiterung der didaktischen Möglichkeiten im Rahmen der Lehrveranstaltungen des Instituts für elektrische Regelungstechnik, Vorstand Prof. Dr. A. Weinmann, wurde ein direkter Informationsaustausch zwischen der Hybridrechenanlage und einer mikroprozessorgesteuerten Videoanlage im Hörsaal X durch die Erstellung einer Übertragungsanlage ermöglicht, die in einem früheren INTERFACE-Artikel bzgl. der Hardware und der peripheren Firmware besprochen wurde. Der vorliegende Beitrag soll grundlegenden Überlegungen sowie der Problemformulierung im Dialog Vortragender - Hybridrechner gewidmet sein.

Die wesentliche Einsicht, die als notwendige Basis für jede weiterführende Beschäftigung mit Problemstellungen der Regelungstechnik anzusehen ist, besteht darin, daß der Student den weitreichenden Einfluß des Prinzips der Rückkopplung begreift und für die Lösung spezifischer Aufgaben anzuwenden lernt. Dieser Einfluß der Rückkopplung, der auch schon in einfachster Form für den Lernenden sicherlich einige überraschende Effekte zeigt, muß aus einer statischen Betrachtungsweise von Vergleich und Anpassung gerückt und dynamisiert werden. Es sollte gelingen, den Blick des Adressaten dafür zu schärfen, daß in vielen Fällen das dynamische Verhalten eines apparativ gesehen äußerlich offenen Systems durch seine inneren Rückkopplungen bestimmt wird und daß das Auftrennen auch nur einer einzelnen Signalkette tiefgreifende Änderungen im Verhalten des betrachteten Systems zeigen wird. Dieser Ansatz zeichnet bereits den Weg geeigneter Wissensvermittlung im regelungstechnischen Unterricht vor.
Er muß folgende Komponenten umfassen:

1) Vermittlung der begrifflichen Grundlagen (rationaler Einstieg)
2) phänomenologische Behandlung dynamischer Effekte (emotionaler Einstieg)
3) Vertiefung des Wissens durch Abstraktion und rechnerische Erfassung der Vorgänge (kognitive Phase)

Der Erfolg dieser Bemühungen wird sowohl von der fachlichen und persönlichen Qualität des Lehrenden als auch in hohem Maße von den eingesetzten Unterrichtsmitteln bestimmt. Es kann folgender Forderungskatalog für ein geeignetes Lehrmittel aufgestellt werden:

1) unmittelbare Verfügbarkeit an beliebiger Stelle des Unterrichtsgeschehens
2) Anschaulichkeit und Anpassungsfähigkeit
3) Darstellungsmöglichkeit aller wesentlichen Behandlungsweisen für Regelsysteme in Signal-, Frequenz- und Parameterraum
4) leichte Bedienbarkeit

Dies sind im wesentlichen die gleichen Forderungen, die ein Ingenieur an ein modernes Entwicklungsinstrumentarium stellt.
Es wurde daher ein interaktives Simulationssystem mit verschiedenen Videoausgängen geschaffen, das in Verbindung mit dem universitären Hybridrechner eine Visualisierung regelungstechnischer Vorgänge gestattet. Besonderes Augenmerk wurde darauf gelegt, daß die Handhabung dieser Einrichtung die Konzentration auf den Vortrag nicht allzusehr beeinträchtigt. Es wurde daher von Seiten der Software ein Dialogkonzept verwirklicht, daß die Wahl zwischen vier verschiedenen Entscheidungsstufen ermöglicht, die sich im Hinblick auf die Eingriffsmöglichkeiten und daher auch hinsichtlich der Komplexität des Dialogs unterscheiden. Diese vier Entscheidungsstufen können folgendermaßen beschrieben werden:

1) Abarbeiten programmierter Systemstrukturen mit fixem Parametersatz bei freier Wahl der Darstellungsweise
2) wie 1), erweitert um die Möglichkeit beliebiger Änderung der Systemparameter
3) wie 2), erweitert bezüglich einer größeren Freiheit in der Gestaltung der Bildschirminhalte
4) wie 3), erweitert um die Möglichkeit der Änderung der Systemstruktur; Möglichkeit der Kreation eigener Beispiele.

Als zusätzliche Erleichterung für den Vortrag wurde ein Programm eingefügt, das es gestattet , Beispiel- und Darstellungs-

ketten zu einem bestimmten Thema vorzubereiten und zu editieren, die dann während des Vortrages schrittweise abrufbar sind. Trotzdem besteht jederzeit die Möglichkeit, auf die oben erwähnten Entscheidungsebenen unterschiedlichen Freiheitsgrades umzusteigen, wenn eine Zwischenfrage noch eine speziellere Darstellungsweise erfordert, als sie im automatischen Ablauf vorgesehen war. Jedoch wurde mit Absicht auf eine stoßfreie Rückkehrmöglichkeit in den automatischen Programmablauf verzichtet, da unzulässige Parameteränderungen (bezüglich der weiteren abgespeicherten Beispiele) nicht ausgeschlossen werden können.

An jeder beliebigen Stelle der Vorführung kann der Ausdruck eines Protokolls veranlaßt werden, das alle System- und Darstellungsparameter enthält. Dies setzt den Benutzer in die Lage, genaue Unterlagen für die nachträgliche Evaluation des Unterrichtserfolges zu besitzen, um eventuelle Änderungen im didaktischen Aufbau vornehmem zu können.

Der momentane Stand des Programmpaketes umfaßt alle gebräuchlichen Darstellungshilfen zur Untersuchung linearer kontinuierlicher Systeme mit konzentrierten Parametern. Der modulare Aufbau ermöglicht beliebige Erweiterungen. Als nächster Schritt sind Problemlösungen zur Behandlung diskreter linearer Systeme mit konzentrierten Parametern sowie Spektralbereichsmethoden nichtlinearer Systeme mit separierbarer Nichtlinearität und die Darstellung der Systembewegungen in der Phasenebene vorgesehen.

M. Haider

Institut für
elektrische Regelungstechnik

Berechnung der Skalierung 'abhängiger Variablen' und Berechnung der Maschinenkoeffizienten

Für die Skalierung sind die Verstärker des Analogrechners als parallele Prozessoren mit relativer Festkommaarithmetik zu betrachten, deren Ausgänge auf das Intervall (-1,+1) beschränkt sind. Diese Verstärker sind nach Zuordnung zu den auftretenden Problemvariablen entsprechend dem zu lösenden Problem miteinander verbunden. Wegen des beschränkten Wertebereichs treten die einzelnen Problemvariablen mit unterschiedlicher Skalierung an den Verstärkerausgängen auf. Die Skalierung der Verstärkereingänge muß dabei vor Durchführung der jeweiligen Operation der Skalierung des Ausgangs angepaßt werden, um eine möglichst hohe Rechengenauigkeit zu erzielen. Dazu werden in der Regel an den Verstärkereingängen anliegende Maschinenkoeffizienten verwendet, die eine tatsächliche Umskalierung der Eingänge bewirken. Diese Koeffizienten sind ebenfalls auf das Intervall (-1,+1) beschränkt, können aber bei bestimmten Verstärkertypen (z.B. Integrierer) mit einem zusätzlichen Faktor multipliziert werden (Eingangsverstärkung, Feedbackverstärkung). Bei einigen Verstärkertypen ist eine derartige Umskalierung nicht notwendig, da sie keine Erhöhung der Rechengenauigkeit bringt. In diesem Fall pflanzt sich die Skalierung bis zum nächsten auftretenden Maschinenkoeffizienten fort (z.B. $|x(t)|, \max_{0 \leq \tau \leq t}(x(\tau))$).

Das im letzten INTERFACE beschriebene Skalierungsverfahren ("Automatische On-line-Skalierung", INTERFACE 15/16,pp.33-43) setzt voraus, daß aus den bei jeder Iteration erhaltenen Skalierungsfaktoren die Maschinenkoeffizienten berechnet und auch eingestellt werden können. Diese beiden Voraussetzungen, Berechenbarkeit und Einstellbarkeit, sind für das Funktionieren einer automatischen On-line-Skalierung grundlegend.

Berechenbarkeit:

Bezüglich der Berechenbarkeit sind die beiden folgenden Variablen-Merkmale entscheidend:

Skalierungsunabhängigkeit: Die Skalierung des zugehörigen Verstärkerausganges ist unabhängig von der Skalierung seiner Eingänge, d.h. sein Skalierungsfaktor ist frei wählbar.

Skalierungsabhängigkeit: Die Skalierung des zugehörigen Verstärkerausganges hängt direkt von der Skalierung seiner Eingänge ab, d.h. sein Skalierungsfaktor muß aus der Skalierung seiner Eingänge berechnet werden.

Bei einer Schaltung, die nur skalierungsunabhängige Variable enthält, ist die Reihenfolge der Berechnung der Maschinenkoeffizienten beliebig und damit vom speziell gegebenen Problem unabhängig, da alle Skalierungsfaktoren "gleichzeitig" zur Berechnung bereitstehen.

Enthält die Schaltung skalierungsabhängige Variable, so muß zunächst deren Skalierung aus der Skalierung ihrer Eingänge berechnet werden, um anschließend die Berechnung der Maschinenkoeffizienten bei den skalierungsunabhängigen Variablen zu ermöglichen. Die Skalierung der Eingänge der abhängigen Variablen muß dabei zum Zeitpunkt der Berechnung bereits bekannt sein. Daraus ist leicht ersichtlich, daß bei algebraischen Schleifen, die nur aus abhängigen Variablen bestehen, dieselbe Schleife für die Skalierungsfaktoren gelöst werden muß. Diese Notwendigkeit entfällt jedoch, wenn jede auftretende algebraische Schleife zumindest eine skalierungsunabhängige Variable enthält.

Einstellbarkeit:

Die aus der jeweiligen Skalierung berechneten Koeffizienten können theoretisch jeden beliebigen reellen Wert annehmen. Die tatsächlichen Schwankungen dieser Koeffizienten während des Skalierungsverfahrens hängen von der Startskalierung, der Konvergenzgeschwindigkeit und der mit der jeweiligen Skalierung erzielten Rechengenauigkeit ab. Bezüglich letzterem spielen die Komplexität des Systems (insbesondere Nichtlinearitäten) und die Genauigkeit, mit der die Maschinenkoeffizienten eingestellt werden können, eine wesentliche Rolle.

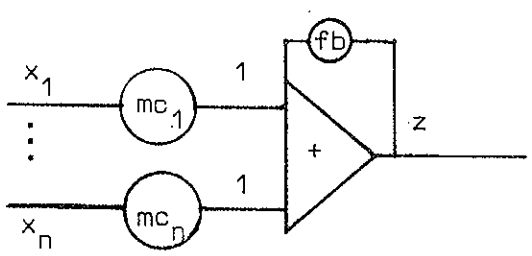
Hinsichtlich der Möglichkeiten, den einstellbaren Koeffizientenbereich zu regulieren, ist zwischen drei Verstärkergruppen zu unterscheiden:

Integrierer: Die Einstellbarkeit kann mit der Eingangsverstärkung (1,10,100), der lokalen Zeiteinheit (NS,FS,NMS,FMS), der globalen Zeiteinheit (NS, FS,NMS, FMS) und der globalen Zeittransformation ($\beta \in R$) gesteuert werden. Diese Möglichkeiten unterscheiden sich durch ihren Einfluß auf andere Maschinenkoeffizienten:
- Eine Änderung der Eingangsverstärkung bewirkt nur eine Änderung des dazugehörigen Maschinenkoeffizienten.
- Eine Änderung der lokalen Zeiteinheit bewirkt eine Änderung sämtlicher am Eingang des Integrierers anliegenden Maschinenkoeffizienten (Ausnahme IC).
- Eine Änderung der globalen Zeittransformation bewirkt eine Änderung aller an Integrierern anliegenden Maschinenkoeffizienten sowie eine Änderung der Maschinenzeit.

Summierer (und andere lineare Verstärker): Die Einstellbarkeit der Maschinenkoeffizienten kann mit der Eingangsverstärkung und einem Feedback-Koeffizienten fb ($0<fb\leq1$) gesteuert werden.
- Eine Änderung der Eingangsverstärkung bewirkt nur eine Änderung des dazugehörigen Maschinenkoeffizienten.
- Eine Änderung des Feedback-Koeffizienten bewirkt analog zur lokalen Zeittransformation bei Integrierern eine Änderung aller Eingangskoeffizienten.

Nichtlineare und andere skalierungsabhängige Verstärker: Bei dieser Gruppe können keine allgemein gültigen Aussagen getroffen werden, die Einstellbarkeit der Koeffizienten ist für jede Operation speziell zu behandeln.

Eine Variable ist dann skalierungsunabhängig, wenn ihr Verstärker an den Eingängen Koeffizienteneinheiten besitzt und die bei beliebig vorgegebener Ausgangsskalierung auftretenden Maschinenkoeffizienten eingestellt werden können. So ist z.B. ein Summierer mit Koeffizienteneinheiten und Eingangsverstärkung 1 an allen Eingängen sowie einer Koeffizienteneinheit in der Rückführung immer skalierungsunabhängig, da die Maschinenkoeffizienten bei beliebiger Ausgangsskalierung immer eingestellt werden können:



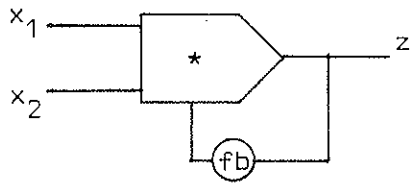
$$x_i = sx_i \cdot \bar{x}_i \qquad sx_i \in R$$

$$z = sz \cdot \bar{z} \qquad sz \in R \ (\neq \emptyset)$$

$$mc_i = \frac{sx_i \cdot fb}{sz} \qquad \begin{matrix} 0 < fb \leq 1 \\ -1 < mc_i \leq 1 \end{matrix}$$

Bei vorgegebenen Skalierungsfaktoren sx_i und sz kann der Koeffizient fb immer so gewählt werden, daß alle mc_i im verfügbaren Intervall liegen. Gibt man aus Gründen der Rechengenauigkeit eine untere Schranke ε für fb an, so können die Koeffizienten nur dann eingestellt werden, wenn $|(sx_i/sz).\varepsilon| \leq 1$ gilt, d.h. $|sz|$ muß größer oder gleich $|sx_i.\varepsilon|$ sein und die Variable ist nur mehr teilweise skalierungsunabhängig. Werden an den Eingängen zusätzlich Relays zur Anwahl einer Eingangsverstärkung 1 oder 10 verwendet, so vergrößert sich der für die Skalierung von z zulässige Bereich auf $[(sx_i.\varepsilon)/10,\infty]$.

Multiplizierer mit Feedback-Koeffizienten sind immer teilweise skalierungsunabhängig, da die Skalierung der Ausgänge immer kleiner oder gleich dem Produkt der Eingangsskalierung sein muß:



$$fb = \frac{sz}{sx_1 . sx_2}$$

$$0 < fb \leq 1$$

Wird sz größer als $sx_1 . sx_2$, so muß $sz = sx_1 . sx_2$ gesetzt werden.

Bei Schaltungen für Absolutbetrag oder Quadratwurzel treten keine Koeffizienteneinheiten auf. Sie sind vollständig skalierungsabhängig und ihre Skalierung muß aus der Skalierung der Eingänge berechnet werden ($sz = |sx|$ bei $z = |x|$ oder $sz = \sqrt{|sx|}$ bei $z = \sqrt{|x|}$).

Die Eigenschaft der Skalierungsabhängigkeit wird durch die verwendete Hardware festgelegt und führt im weiteren auf das Problem der Berechenbarkeit. Die im AutoPATCH-System verwendeten Makros wurden im Rahmen der vorhandenen Hardware so konzipiert, daß größtmögliche Skalierungsunabhängigkeit erreicht wird, sodaß auch Probleme mit sehr ungünstiger Anfangsskalierung am Analogrechner aufgebracht und anschließend mit dem Skalierungsverfahren optimal skaliert werden können:

- Alle Integrierer sind mit zusätzlichen Controllines zur automatischen Anwahl der lokalen Zeiteinheit sowie Relays an den Eingängen zur Anwahl der Eingangsverstärkung (1 oder 10) ausgestattet.
- Alle Summierer sind mit Feedback-Koeffizienteneinheiten versehen, die Eingangsverstärkung ist jedoch mit 1 festgelegt.
- Alle Multiplizierer besitzen Feedback-Koeffizienten, Dividierer besitzen am Eingang des Zählers eine Koeffizienteneinheit.
- Alle übrigen Makros sind skalierungsabhängig.

Dieses Makrokonzept ist von der vorhandenen Hardware wesentlich mitbestimmt und kann noch nicht als endgültig angesehen werden. Besonders bezüglich der Rechengenauigkeit hat sich gezeigt, daß bei den Summierern die Möglichkeit der Anwahl verschiedener Eingangsverstärkungen an allen Eingängen notwendig wäre.

D. Solar

Die Verwendung von Digitally Controlled Function Generators (DCFGs) in HYBSYS am Beispiel eines User Overlays

Der Prozessor HYBSYS unterstützt die Verwendung von DCFGs, doch ist diese noch nicht in der endgültig vorgesehenen Weise möglich. DCFGs haben unter allen Komponenten eine gewisse Sonderstellung. Diese Sonderstellung bezieht sich auf die Skalierung der DCFGs, da diese von der automatischen Skalierung in HYBSYS ausgenommen sind.

Der Benutzer muß derzeit noch zusätzliche Vorkehrungen treffen, um die Möglichkeiten der DCFGs in HYBSYS voll ausnützen zu können.

Laden eines DCFGs

Die DCFGs können je nach Bedarf auf zwei verschiedene Arten mit der gewünschten Funktion geladen werden.

a) mit Hilfe des Overlays LDFUN:

LDFUN fname:

oder

LDFUN fname;

ladet die DCFG-Variable fname mit der Funktion F(X), die mit bis zu 50 Stützpunkten definiert werden kann.

Der Eingabedialog läuft folgendermaßen ab:

```
SX   SF   (2E15.6)
sfx,sff,↵
X(I) F(I) (2E15.6)
x1,f1,↵
 :
xn,fn,↵
xend,↵
END
```

sfx Skalierungsfaktor der Abszisse

sff Skalierungsfaktor der Ordinate

xi unskalierte Abszissenwerte

fi unskalierte Funktionswerte

xend xend < x1 zeigt Ende der Liste der Stützpunkte an

1 ≤ n ≤ 50

Falls LDFUN mit ":" aufgerufen wird, werden sowohl die eingegebenen Stützpunkte als auch die ausgewählten Stützpunkte (maximal 16), auf die der DCFG dann tatsächlich gesetzt wird, auf der angesprochenen graphischen Device geplottet, wobei die Achsenbeschriftung dem aktuellen Skalierungsfaktor der Abszisse Rechnung trägt (siehe auch HYBSYS User Manual, Version 4 M/A, Kap. 11.4).

b) mit Hilfe eines User-Overlays:

Der Prozessor HYBSYS erlaubt und unterstützt durch sein Overlaykonzept die Verwendung von vom Benutzer geschriebenen FORTRAN- oder Assembler-Programmen, die speziell als User-Overlays gebunden werden (siehe HYBSYS User Manual, Version 4 M/A, Kap. 11.6). Mittels eines User-Overlays, der entsprechende DCFG-Routinen aus der Library aufruft, können DCFGs mit den gewünschten Funktionen geladen werden, wie das im folgenden angeführte Beispiel demonstriert (siehe auch INTERFACE 13, "Die Verwendung von digital setzbaren Funktionsgebern").

Die Skalierung der DCFGs

Der Benutzer muß selbst auf die richtige Skalierung achten. Da dies die häufigste Fehlerquelle bei der Verwendung von DCFGs ist, soll hier speziell darauf hingewiesen werden. Man muß in beiden oben angeführten Fällen pro DCFG einen Hilfsparameter deklarieren, der auf den reziproken Wert der Eingangsskalierung des entsprechenden DCFGs gesetzt wird. Da DCFGs keinen IO-er Eingang besitzen, müssen gegebenenfalls sogar Hilfssummierer deklariert werden, um dies zu ermöglichen.

Außerdem muß jeder DCFG selbst skaliert werden.
Dies kann mittels Wertzuweisung in HYBSYS geschehen oder aber in einem User-Overlay, wo der Skalierungsfaktor eines DCFGs YSCAL aus dem Commonblock CDCFG gewonnen werden kann und mittels des Unterprogramms SXIN in die Schaltdaten abgespeichert werden kann. Diese Möglichkeit muß auch dann angewandt werden, wenn ein Skalierungsfaktor eines DCFGs nicht bekannt ist. (Siehe Beschreibung der DCFG-Routinen DCFG,LFGS,LFGR,LFGFD).

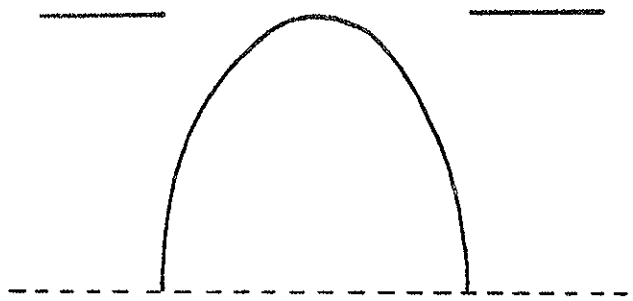
Achtung: Anschließend muß zur Berechnung der Potentiometerwerte ein SET exekutiert werden.

Beispiel

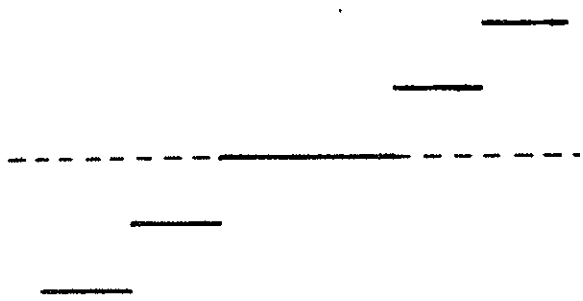
Nachstehendes Beispiel soll zeigen, wie mittels eines User-Overlays zwei DCFGs (DF und DG) auf beliebige Funktionen gesetzt werden können. (Es wurden zwei Sprungfunktionen FUN1, FUN2 und zwei Hysteresen FUN3, FUN4 ausgewählt.) PF und PG sind jene Hilfsparameter , die auf den reziproken Wert der Eingangsskalierung (T=10) gesetzt wurden.

Als Eingang der DCFGs wurde eine Sägezahnfunktion verwendet, die mit Hilfe des Summierers S und des Absolutbetrages A nur aus T erzeugt wurde. Der Summierer S1 wird für das Problem selbst nicht, sondern nur zum Plotten der Hysterese (PLOT DF, DG OVER S1) verwendet. Da zur graphischen Darstellung von Hysteresen sowohl die aufsteigenden als auch die fallenden Abszissenwerte benötigt werden, kann nicht über T geplottet werden.

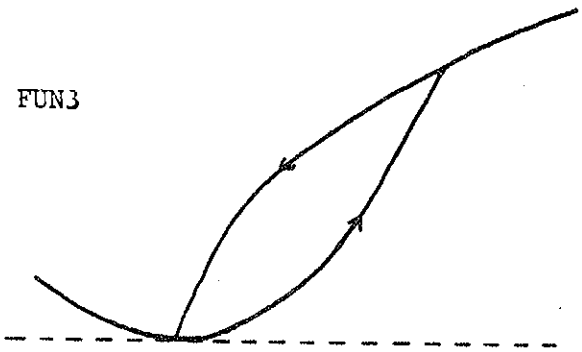
Funktionen:



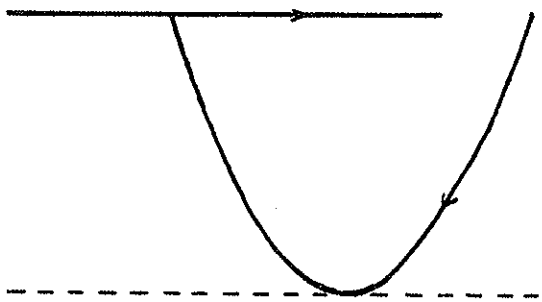
FUN1



FUN2



FUN3



FUN4

Schaltdaten:

```
TIME INTERVAL :  (=.1000E+02, .1000E+02)
TIME TRANSFORMATION:   .1000E+01
MACHINE TIME:  .2000E+02   MMS

POINTS:   1001
DISTANCE:   20 MYS

                    VARIABLES

T          .1000E+02  FS  INT     5 L     =.9967    .9854

S          .1000E+02      SUM    51        =.9922    .9983
S1         .1000E+02      SUM     6        =.9950    .9972

DF         .1000E+01      DCFG    1        =.0022    .9994
DG         .1000E+01      DCFG    2         .0617    .9998
A         =.1000E+02      ABS    49        =.9967    .0000

TA        =.1000E+02
TEND       .1000E+02
BETA       .1000E+01
K1         .1000E+01
PF         .1000E+00
PG         .1000E+00

              PATCHING CONNECTIONS / COEFFICIENT SETTINGS

                CDAC  0    .1000E+01  IC        INT     5    T       *  TA
                DC4   5   =.1000E+01  10    4                          K1

SUM   51        DC4  51    .1000E+01  SF        SUM    51    S       *  S
ABS   49        DC4  53    .1000E+01  1                                A
ABS   49        DC4  52    .1000E+01  1                                A
                DC4  33    .1000E+01  1                                TA
SUM    6        DC4   8    .1000E+01  SF        SUM     6    S1      *  S1
SUM   51        DC4   7    .1000E+01  1                                =S

SUM   51        DC4  45   =.1000E+01  1         DCFG    1    DF      *  =S     * PF
SUM   51        DC4  32   =.1000E+01  1         DCFG    2    DG      *  =S     * PG
INT    5                              1         ABS    49    A       *  T
```

Programm:

```
      SUBROUTINE HYOVSR(IRTCD)
C
C  OVERLAY ZUM SETZEN EINES DCFG
C  AUF EINE SPEZIELLE FUNKTION
C
      INTEGER DECZHL,NAM(3)
      DIMENSION H(16)
      LOGICAL LER
      COMMON/CDCFG/XSCAL,YSCAL,ISCAL
      COMMON/HYST/X1HY,X2HY,IHY,IR
      EXTERNAL FUN1,FUN2,FUN3,FUN4
C
C  DEKODIERUNG DER OVERLAYPARAMETER
C
      IF(IRTCD.NE.1) CALL REPEAT
      CALL STNAM(NAM,IRTCD,ICT,6,'240,2)
      IF(ICT.LT.1) CALL REPEAT
      IF(IRTCD.NE.1) CALL REPEAT
      CALL STNAM(K,IRTCD,ICT,1,'240,2)
      IF (ICT.LT.1) CALL REPEAT
      IF(IRTCD.GE.0) CALL REPEAT
      I=DECZHL(I)
      IF(I.GE.5) CALL NEXCOM
C
C  BERECHNUNG DER HARDWARE-ADRESSE DES DCFG
C
      IV=INDVRM(NAM,LER)
      IAM=IAMOUT(IV)
C
C  SETZEN DES DCFG AUF DIE GEWAEHLTE FUNKTION
C
      GOTO(10,20,30,40),I
   10 CONTINUE
      CALL DCFG(IAM,FUN1,-3.,3.,J,H,IER)
      CALL NEXCOM
C
   20 CONTINUE
      CALL DCFG(IAM,FUN2,-2.9,2.9,J,H,IER)
      CALL SXIN(IV,YSCAL)
      CALL NEXCOM
C
   30 CONTINUE
      IHY=1
      X1HY=-.5
      X2HY=.5
      CALL DCFG(IAM,FUN3,-1.,1.,J,H,IER)
      CALL NEXCOM
C
   40 CONTINUE
      IHY=3
      X1HY=-1.
      X2HY=2.
      CALL DCFG(IAM,FUN4,-3.,3.,J,H,IER)
      CALL NEXCOM
      RETURN
      END
```

```
FUNCTION FUN1(X)
FUN1=-1.
IF(X.GT.-1.58.AND.X.LT.1.58) FUN1=COS(X)
RETURN
END


FUNCTION FUN2(X)
FUN2=FLOAT(INT(X)*2)
RETURN
END


FUNCTION FUN3(X)
COMMON/HYST/X1HY,X2HY,IHY,IR
IF(IR.EQ.1) FUN3=(X+.5)*(X+.5)
IF(X.GT.-.5.AND.IR.EQ.-1) FUN3=SQRT(X+.5)
RETURN
END


FUNCTION FUN4(X)
COMMON/HYST/X1HY,X2HY,IHY,IR
FUN4=4.
IF(IR.EQ.-1) FUN4=(X-1.)*(X-1.)
RETURN
END
```

F. Berger

HYBRIDE SIMULATION EINES SCHWINGENDEN, EINSEITIG ERREGTEN SEILES MIT HYBSYS

Die vertikale Auslenkung u=u(x,t) eines schwingenden Seiles der Länge L, das an einem Ende festgehalten und am anderen Ende sinusförmig erregt wird, läßt sich durch die folgende partielle Differentialgleichung

$$u_{xx} = a.u_{tt}$$

mit den Anfangs- und Randbedingungen

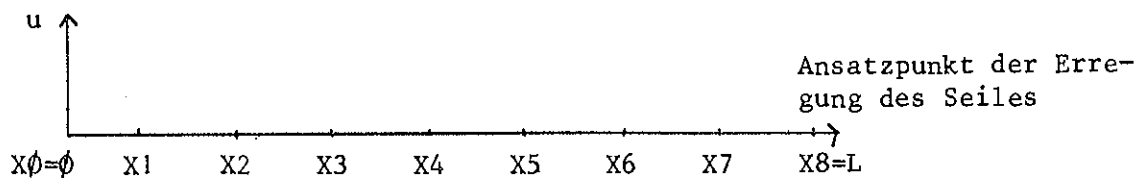
$$u(L,t) = b.\sin\omega t$$
$$u(x,\emptyset) = \emptyset$$
$$u_x(x,\emptyset) = \emptyset$$

beschreiben.

Anhand dieses mathematischen Modells sollen vor allem die Resonanzeigenschaften des Seils untersucht werden. Zu diesem Zweck ist besonders das Amplitudenmaximum in der Seilmitte in Abhängigkeit von der Erregerfrequenz ω im Intervall $[\emptyset,2]$ von Interesse.

In diesem Artikel wird ein besonderes Augenmerk darauf gelegt, das Aufbringen des Modells und die Modelluntersuchung mit Hilfe von HYBSYS zu demonstrieren.

Löst man diese partielle Differentialgleichung mit der Methode der Ortsdiskretisierung und wird die Länge des Seiles in acht gleiche Teile geteilt ($\Delta x=L/8$),



so erhält man nach Diskretisierung von $\partial^2/\partial x^2$ für die sieben Stützstellen das folgende System gewöhnlicher Differentialgleichungen

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{(\Delta x)^2} = au_i'' \qquad i=1,\ldots,7$$

mit den Anfangs- und Randbedingungen

$$u_i(\emptyset) = u_i'(\emptyset) = \emptyset \qquad x_i = i.\Delta x \qquad i=1,\ldots,7$$
$$u_i(t) = u(x_i,t)$$
$$u_o(t) = u(\emptyset,t) = \emptyset$$
$$u_8(t) = u(L,t) = b.\sin\omega t$$

Setzt man $A = 64/L^2 a$ und $B = A.b$, so erhält man

$$u_1'' = Au_2 - 2Au_1$$
$$u_i'' = Au_{i+1} - 2Au_i + Au_{i-1} \qquad i=2,\ldots,6$$
$$u_7'' = B\sin\omega t - 2Au_7 + Au_6 \qquad u_i(\emptyset) = u_i'(\emptyset) = \emptyset \quad i=1,\ldots,7$$

Bevor man mit den Deklarationen in HYBSYS beginnen kann, muß man das System von Differentialgleichungen als System von Integralgleichungen erster Ordnung schreiben (vergleiche mit den Schaltdaten):

$$u_i = \int_o^t u_i' d\tau \qquad i=1,\ldots,7$$
$$u_1' = \int_o^t (Au_2 - 2Au_1)d\tau$$
$$u_i' = \int_o^t u_i'' d\tau \qquad i=2,\ldots,6$$
$$u_7' = \int_o^t (Bu + 2Au_7 + Au_6)d\tau$$

(u_i'' sind Summierer: $u_i'' = Au_{i+1} - 2Au_i + Au_{i-1}$)

wobei der Sinus selbst mit Hilfe einer Integralgleichung erzeugt wird.

$$u = \int_o^t \omega v d\tau$$
$$v = \int_o^t -\omega u d\tau + 1$$

Nun kann die Deklaration der Schaltdaten in HYBSYS erfolgen:

```
DECLAR PAR:A,B,W,KØ=Ø,K2=2;
DECLAR INT:U,V,U1,U2,U3,U4,U5,U6,U7,U1',U2',U3',U4',U5',U6',U7';
DECLAR SUM:U2",...,U6",UØ,AØØ1;
DECLAR MAXT:MAU3,MAU4;
DIGITL;
DECLAR MULT:A2;
DECLAR A2=K2,A;
ANALOG;
DECLAR U1=U1';
    :
DECLAR U7=U7';
DECLAR U2"=A*U1,-A2*U2,A*U3;
    :
DECLAR U6"=A*U5,-A2*U6,A*U7;
DECLAR UØ=KØ;
DECLAR U1'=-A2*U1,A*U2;
DECLAR U2'=U2";
    :
DECLAR U6'=U6";
DECLAR U7'=-A2*U7,A*U6,B*U;
DECLAR U=W*V;
DECLAR V,IC=K1,-W*U;
DECLAR MAU3=U3;
DECLAR MAU4=U4;
```

Die im folgenden wiedergegebene Liste der Schaltdaten enthält einige Modellerweiterungen, doch werden diese Ergänzungen in diesem Artikel nicht besprochen, und daher wurden auch die entsprechenden DECLAR-Operationen ausgelassen.

Schaltdaten:

```
TIME INTERVAL :  ( .0000E+00, .2000E+02)
TIME TRANSFORMATION:    .5074E+01
MACHINE TIME:  .1014E+03   NMS

POINTS:     200
DISTANCE:    51 SMY

                        VARIABLES

T          .2000E+02  NS   INT     5 L      -.0011    .9933
U1         .5000E+01       INT    35       -.7024    .7681
U2         .4200E+01       INT    10       -.8788    .7513
U3         .4000E+01       INT    40       -.6912    .2655
U4         .5200E+01       INT    20       -.5733    .6985
U5         .5000E+01       INT    15       -.6446    .6339
U6         .5400E+01       INT    45       -.4879    .2661
U7         .4500E+01       INT    50       -.5677    .5064
U1'        .8700E+01       INT    25       -.4486    .4413
U2'        .7700E+01       INT    30       -.4453    .5019
U3'        .6200E+01       INT    75       -.3285    .3150
U4'        .9300E+01       INT    70       -.3913    .3790
U5'        .7000E+01       INT   105       -.4514    .3891
U6'        .8100E+01       INT   110       -.2717    .2448
U7'        .6400E+01       INT    55       -.3459    .4727
U          .2100E+01       INT    80       -.5868    .3863
V          .1800E+01       INT    85       -.5604    .5648

U2"        .1200E+02       SUM    51       -.2341    .3419
U3"        .1000E+02       SUM    41       -.1662    .2420
U4"        .1470E+02       SUM    56       -.2695    .2122
U5"        .1140E+02       SUM    36       -.2336    .2212
U6"        .1240E+02       SUM    96       -.1572    .1942
A000       .5000E+01       SUM     6       -.6435    .6322
A001       .4700E+01       SUM    16       -.6423    .4829
U0         .1000E+01       SUM    22        .0000    .0000

MAU3      -.4000E+01       MAXT   60       -.2689    .0016
MAU4      -.5200E+01       MAXT   90       -.7041    .0022

T0         .0000E+00
TEND       .2000E+02
BETA       .5074E+01
K1         .1000E+01
A          .1000E+01
U10        .0000E+00
U20        .0000E+00
U30        .0000E+00
U40        .0000E+00
U50        .0000E+00
U60        .0000E+00
U70        .0000E+00
K2         .2000E+01
GEW        .0000E+00
W          .1000E+01
B          .2000E+01
C          .0000E+00
H          .1000E+01
K0         .0000E+00

A2         .2000E+01       MULT
```

PATCHING CONNECTIONS / COEFFICIENT SETTINGS

		CDAC	0	.0000E+00	IC		INT	5	T	=	T0			
		DC4	5	-.9853E+00	10	4					K1			
		CDAC	1	.0000E+00	IC		INT	35	U1	=	U10			
INT	25	DC4	35	-.3428E+00	1	34					U1'			
		CDAC	7	.0000E+00	IC		INT	10	U2	=	U20			
INT	30	DC4	10	-.3612E+00	1	9					U2'			
		CDAC	4	.0000E+00	IC		INT	40	U3	=	U30			
INT	75	DC4	40	-.3052E+00	1	39					U3'			
		CDAC	13	.0000E+00	IC		INT	20	U4	=	U40			
INT	70	DC4	20	-.3524E+00	1	19					U4'			
		CDAC	6	.0000E+00	IC		INT	15	U5	=	U50			
INT	105	DC4	15	-.2758E+00	1	14					U5'			
		CDAC	11	.0000E+00	IC		INT	45	U6	=	U60			
INT	110	DC4	101	-.2955E+00	1	44					U6'			
		CDAC	14	.0000E+00	IC		INT	50	U7	=	U70			
INT	55	DC4	50	-.2802E+00	1	49					U7'			
INT	35	DC4	25	.2265E+00	1	24	INT	25	U1'	=	-U1	*	A2	
INT	10	DC4	29	-.9513E-01	1	29					U2	*	A	
SUM	51	DC4	96	-.3071E+00	1		INT	30	U2'	=	U2"			
SUM	41	DC4	75	-.3178E+00	1	74	INT	75	U3'	=	U3"			
SUM	56	DC4	72	-.3114E+00	1	69	INT	70	U4'	=	U4"			
SUM	36	DC4	105	-.3209E+00	1	104	INT	105	U5'	=	U5"			
SUM	96	DC4	110	-.3016E+00	1	109	INT	110	U6'	=	U6"			
INT	50	DC4	59	.2771E+00	1	59	INT	55	U7'	=	-U7	*	A2	
SUM	16	DC4	55	-.1447E+00	1	54					A001			
INT	85	DC4	80	-.1689E+00	1	79	INT	80	U	=	V	*	W	
		CDAC	15	-.5555E+00	IC		INT	85	V	=	K1			
INT	80	DC4	89	.2299E+00	1	89					-U	*	W	
INT	85	DC4	85	.0000E+00	1	84					-V	*	C	
SUM	51	DC4	51	.1000E+01	SF		SUM	51	U2"	=	U2"			
INT	35	DC4	53	-.4166E+00	1						U1	*	A	
INT	10	DC4	52	.6999E+00	1						-U2	*	A2	
INT	40	DC4	83	-.3333E+00	1						U3	*	A	
SUM	41	DC4	41	.1000E+01	SF		SUM	41	U3"	=	U3"			
INT	10	DC4	76	-.4200E+00	1						U2	*	A	
INT	40	DC4	99	.8000E+00	1						-U3	*	A2	
INT	20	DC4	90	-.5200E+00	1						U4	*	A	
SUM	56	DC4	56	.1000E+01	SF		SUM	56	U4"	=	U4"			
INT	40	DC4	58	-.2721E+00	1						U3	*	A	
INT	20	DC4	57	.7074E+00	1						-U4	*	A2	
SUM	6	DC4	111	-.3401E+00	1						A000			
SUM	36	DC4	36	.1000E+01	SF		SUM	36	U5"	=	U5"			
INT	20	DC4	66	-.4561E+00	1						U4	*	A	
INT	15	DC4	69	.8771E+00	1						-U5	*	A2	
INT	45	DC4	60	-.4736E+00	1						U6	*	A	
SUM	96	DC4	30	.1000E+01	SF		SUM	96	U6"	=	U6"			
INT	15	DC4	98	-.4032E+00	1						U5	*	A	
INT	45	DC4	97	.8709E+00	1						-U6	*	A2	
INT	50	DC4	93	-.3629E+00	1						U7	*	A	
SUM	6	DC4	6	.1000E+01	SF		SUM	6	A000	=	A000			
INT	15	DC4	7	-.1000E+01	1						U5	*	A	
		DC4	8	.0000E+00	1						-GEW			
SUM	16	DC4	16	.8703E+00	SF		SUM	16	A001	=	A001			
INT	45	DC4	18	-.1000E+01	1						U6	*	A	
INT	80	DC4	17	-.7777E+00	1						U	*	B	
SUM	22	DC4	21	.1000E+01	SF		SUM	22	U0	=	U0			
		DC4	22	.0000E+00	1						K0			
INT	40				1		MAXT	60	MAU3	=	U3			
INT	20				1		MAXT	90	MAU4	=	U4			

MULT A2 = K2
A

Durch die HYBSYS-Befehle

```
ONLINE;
PREPAR;
```

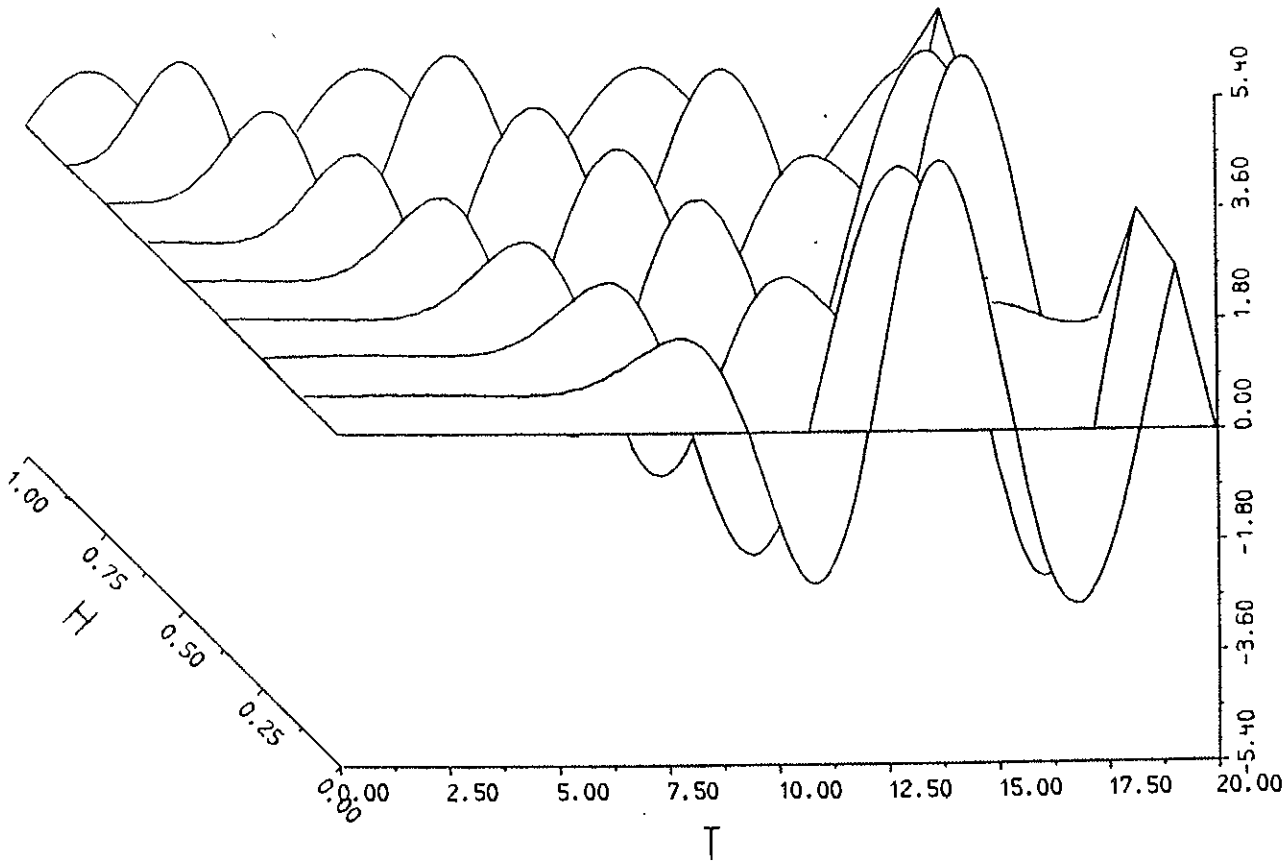
wird die Schaltung auf dem Analogrechner aufgebracht und die automatische Skalierung im interessierenden Bereich $\omega(\omega=\emptyset$ bis 2) erfolgt durch die HYBSYS-Befehle

```
W=2;SET;OPSCAL;SCALE;RESCAL;W=Ø,1.8,.2!SET;SCALE;
```

Folgende Darstellung der einzelnen u_i, die für $\omega=1$ über der Zeit t mit $t\in[\emptyset,2\emptyset]$ und der Seillänge geplottet wurde, erhält man durch den HYBSYS-Befehl

```
PLOT UØ,U1,U2,U3,U4,U5,U6,U7,U OVER T AND H=Ø,1;
```

wobei H eine Dummy-Variable ist und VIEW=2 gesetzt wurde.



Zum Studium der Resonanzeffekte wurden die Stellen x_4 (Seilmitte) und x_3 (=3L/8) herangezogen. Für die zugehörigen Amplituden u_4 und u_3 wurden die beiden Maximumdetektoren MAU4 und MAU3 definiert:

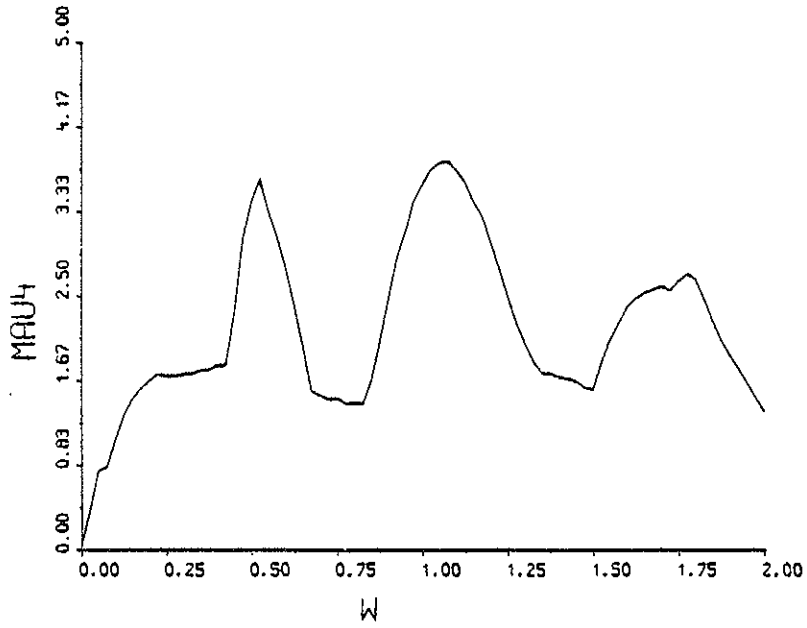
$$MAU4 = \max_{t \in [0,20]} u_4(t) \quad \text{bzw.}$$

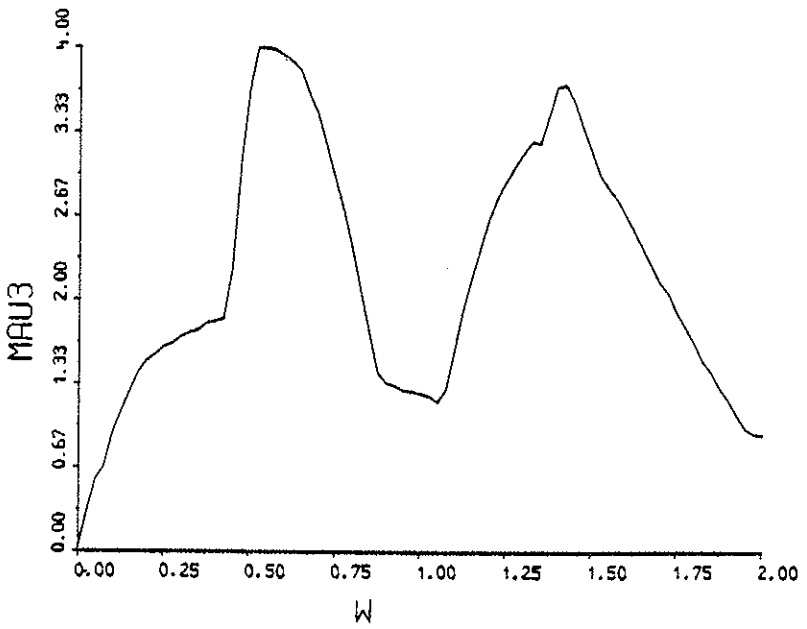
$$MAU3 = \max_{t \in [0,20]} u_3(t)$$

Die Plotbefehle

```
PLOT MAU4 OVER W=0.,2.,0.025;
PLOT MAU3 OVER W=0.,2.,0.025;
```

liefern die im folgenden wiedergegebenen Zeichungen. Zu beachten sind die 3 Resonanzspitzen in der Seilmitte, während beim unmittelbar benachbarten Punkt x_3 nur mehr 2 Resonanzen zu beobachten sind.





W. Kleinert

F. Berger

aus dem praktikum

HYBSYS mit AutoPATCH im Praktikumsbetrieb:

Im Sommersemester 1980 testete eine Gruppe des Fortgeschrittenen-Praktikums den Prozessor HYBSYS mit und ohne AutoPATCH. Die Praktikanten hatten zunächst noch mit Schwierigkeiten technischer Natur zu kämpfen (z.B. starkes Rauschen von Summierern wegen ungünstiger Feedback-Schaltung), die dann im Lauf des Semesters bereinigt wurden.

Trotzdem waren die übrigen Erfahrungen so positiv, daß im Wintersemester 1980/81 alle Praktika unter Verwendung von HYBSYS mit AutoPATCH am Hybridrechner durchgeführt wurden. Nur die Studenten des Einführungspraktikums verwendeten während der ersten acht Wochen noch die Kleinrechner EAI-1000 (da sie ja auch einen klassischen Analogrechner kennenlernen sollten), bevor sie ihre Arbeiten am Großrechner fortsetzten.

Der Praktikumsbetrieb im Wintersemester 1980/81 verlief weitgehend reibungslos. Kleinere Schwierigkeiten entstanden manchmal durch Informationsmängel bei der Umstellung auf neue Versionen des HYBSYS-Prozessors.

Die Vorteile der Verwendung von HYBSYS mit AutoPATCH sind mannigfaltig und könnten noch vergrößert werden:
Verkürzte die Verwendung des AutoPATCH die Vorbereitungszeit zur Lösung eines Beispiels wesentlich, so ermöglichte das automatische Skalieren eine große Verminderung der interaktiven Arbeitszeit am Hybridrechner. Das hatte zur Folge, daß in den Praktika mehr Probleme gelöst werden konnten und gleichzeitig auch die Qualität der Beispiele gehoben werden konnte. Dies traf besonders auf größere Simulationen physikalischer und biologischer Vorgänge zu, wo nun rasche Untersuchungen möglich waren und man sich ganz auf die Problemstellung konzentrieren konnte, ohne sich um programmiertechnische Fragen kümmern zu müssen.

Schwierigkeiten traten allerdings bei Verwendung von Schaltern auf; eine logische 'AND'-Verknüpfung erforderte trickreiche Ersatzschal-

tungen. Die projektierte Integration der Logik (PBP) in naher Zukunft wäre daher dringend wünschenswert, um einerseits komplexe Problemstellungen (z.B. Unstetigkeiten) zu untersuchen und andererseits auch optische Aufbereitungen wieder am großen Rechner durchführen zu können.

Ein nicht zu übersehender Vorteil für den Praktikumsbetrieb ist die Möglichkeit des raschen Wechsels verschiedener Probleme (Schaltungen). So konnte die reservierte Rechenzeit von 2 Gruppen parallel genutzt werden, wobei die eine aktiv rechnete, die andere weitere Änderungen, Zeichnungen usw. überlegte. Diese parallele Arbeitsweise wird im Betriebssystem JCS/VS 8 noch effizienter werden.

Relativ viel Zeit nimmt die Suche nach geeigneten Parameterwerten, vor allem bei großen Parametervektoren, in Anspruch. Einheitliches Skalieren für große Parameterbereiche, Datenübertragung und Zeichnungen hierfür kosten viel Zeit und bringen praktisch nichts. Eine Möglichkeit, den Parameterraum rasch und interaktiv abzutasten, um einen Bereich günstiger bzw. interessanter Werte zu finden, fehlt derzeit noch. Einfache "digitale Potentiometer", die die alten Handpotentiometer ersetzen und die Betrachtung der parameterabhängigen Änderungen der Lösung am Oszillographen ermöglichen, würden hier Abhilfe schaffen. Eine derartige Einrichtung hätte auch für die Untersuchung der Abhängigkeit der Problemlösung (z.B. einer optimalen Steuerung) von den Systemparametern entscheidende Bedeutung, da Empfindlichkeitsanalysen in einfacher und zweckmäßiger Weise interaktiv durchgeführt werden sollen. Diese (ebenfalls projektierten) "Potentiometer" sollten möglichst bald installiert werden, da sie vor allem für die Lehre große Bedeutung haben.

Zusammenfassend kann gesagt werden, daß die Verwendung von HYBSYS mit AutoPATCH im Praktikumsbetrieb die wesentlich raschere Lösung auch von komplexen Aufgaben bestimmter Problemklassen ermöglicht. Gewisse Verbesserungen (Parametervariation, komplexe Logik) könnten diese Problemklasse wesentlich erweitern.

Prof. Dr. I. Troch
Dr. F. Breitenecker
Dr. F. Rattay
Arbeitsbereich für
Hybridrechentechnik

SIMULATION DES OPTISCHEN EINDRUCKES BEIM BEFAHREN EINER BOBBAHN

Dr. F. Rattay
Institut für Technische Mathematik
Technische Universität Wien

An den Arbeitsbereich für Regelungstheorie und Hybridrechentechnik wurde von Prof. Kerner der Wunsch herangetragen, mit möglichst einfachen Mitteln eine Simulation des optischen Eindruckes, der beim Befahren einer Bobbahn besteht, durchzuführen.

Aus verschiedenen Modellen, die für die Kleinrechner EAI 1000 entworfen wurden, war das im folgenden beschriebene jenes, das das optische Empfinden des Bobfahrers - bei möglichst geringem gerätetechnischen Aufwand - am besten wiedergibt. Zunächst wurden folgende vereinfachende Voraussetzungen gemacht:

1. Die Bobbahn wird dadurch beschrieben, daß eine Parabel entlang einer vorgegebenen Leitkurve bewegt wird.
2. Das Bild, das den Eindruck des Befahrens der Bahn vermittelt, wird für einen Betrachter erzeugt, dessen relative Lage bezüglich der Parabel, in der sich der Bob gerade befindet, unverändert bleibt.
3. Den Zusammenhang zwischen der Gestalt der Leitkurven und den Aktivitäten des Bahnbenützers mit dem daraus resultierenden Bewegungsablauf ist nicht Gegenstand dieser Untersuchung, sondern es werden die daraus resultierenden Daten für die Geschwindigkeit und die Gestalt, mit der scheinbar die Bobbahn auf den Fahrer zukommt, als gegeben angenommmen.

Um die Geschwindigkeit des Bobs optisch zu erfassen wurde die Bahn äquidistant durch Parabeln segmentiert. Das erste vor dem Betrachter liegende Stück wurde durch einen parabolischen Zylinder angenähert, der perspektiv dargestellt wird.

Der dargestellte Zylinder mit zeitabhängiger Erzeugendenlänge wird also von der durch den Bob gelegten Ebene , die eine feste Parabel P aus der Bahn herausschneidet, sowie durch die, scheinbar auf den Betrachter zukommende, parabelförmige Begrenzung P_1 des ersten Segments bestimmt. Die Darstellung wird somit durch drei externe zeitabhängige Variable bestimmt, nämlich den beiden Koordinaten für den Fluchtpunkt, die die Richtung und Neigung der Bahn bestimmen sowie die durch die Fahrgeschwindigkeit bedingte Bewegung der herankommenden nächsten Segmentbegrenzung.

Liegt F oberhalb von P_0 - und nur solche Fälle wurden als zulässig angenommen .- sind die vollen Parabeln P_0 und P_1 sowie die beiden begrenzenden Erzeugenden sichtbar. P_1 ist zu P_0 ähnlich und auf das r-fache verkleinert.(Abb. 1)

Zur Darstellung am Oszillographen wurden abwechselnd die Parabel P_0 und P_1 erzeugt, wobei die Rechenzeit mit x identifiziert und der dynamische Eindruck durch schnell repetierendes Rechnen erreicht wurde.

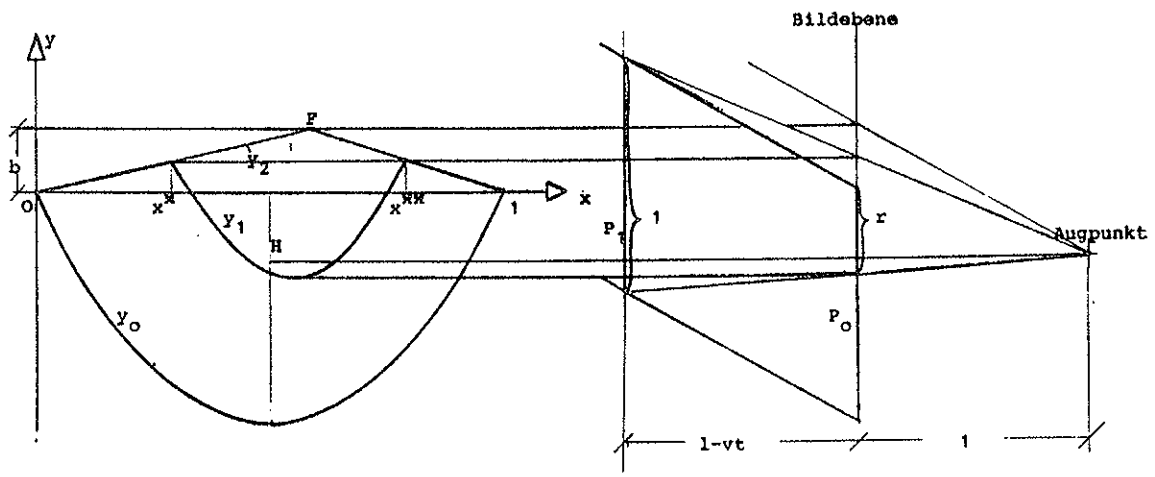


Abb. 1

Perspektive Darstellung des parabolischen Zylinders und Seitenansicht zum perspektiven Bild.

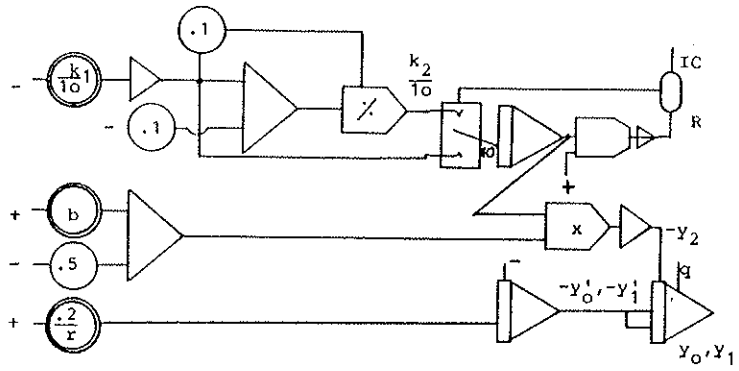


Abb.2

Prinzip der analogen Schaltung zur alternierenden Erzeugung von y_0 und y_1 mit $y_1=y_2$ für $x\in[0,x^*]\cup[x^{**},1]$. Die doppelt gezeichneten Pots enthalten die Eingabedaten für den Anstieg von y_2, Abszisse des Fluchtpunktes der Erzeugenden und den Verkleinerungsfaktor für y_1.

Die speziellen Simulationsdaten waren:

für P_0: $y''=4$, $y_0(0)=0$, $y_0'(0)=-2$, $x\in[0,1]$;
$y^2=2(x^2-1)$

und für P_1: $y''_1=\frac{1}{r}y''_0=\frac{4}{r}$, $y_1(IC)=y_2(x^*)$
$y_1'(IC)=-2$; $r\in[\frac{1}{5},1]$

Aus dem Seitenriß zum perspektiven Bild (Abb. 1) wird für den Verkleinerungsfaktor direkt die Proportion

$$\frac{1}{r}=\frac{1-L-vt}{1}$$

entnommen, wobei gilt
L: Länge eines Bahnsegments
v: Fahrgeschwindigkeit
t: physikalische Zeit

Das Erzeugendenpaar y_2 setzt sich aus $\overline{OF}$ und $\overline{F1}$ zusammen und wird zunächst für $x\in[0,x^*]$ dargestellt, dann wird der Schaltkreis für y_1 aktiviert und y_1 solange ins Display geführt, bis zum Zeitpunkt x^{**} $y_1>y_2$ wird und danach wieder y_2 gezeigt. Abwechselnd werden dann das aktuelle r und r=1 für P_0 erzeugt, sodaß dadurch die Abb. 3 entsteht, wo durch Komponentenungenauigkeit der zweite Umschaltpunkt x^{**} nicht exakt herauskommt. Selbstverständlich lassen sich mit geringem Mehraufwand auch mehrere Segmentierungen der Bahn erreichen, wodurch der optische Eindruck noch verbessert wird. Leider können die gezeigten Bilder (Abb. 3 und 4) nicht den dynamischen Eindruck wiedergeben, den die Simulation am Oszillographen vermittelt.

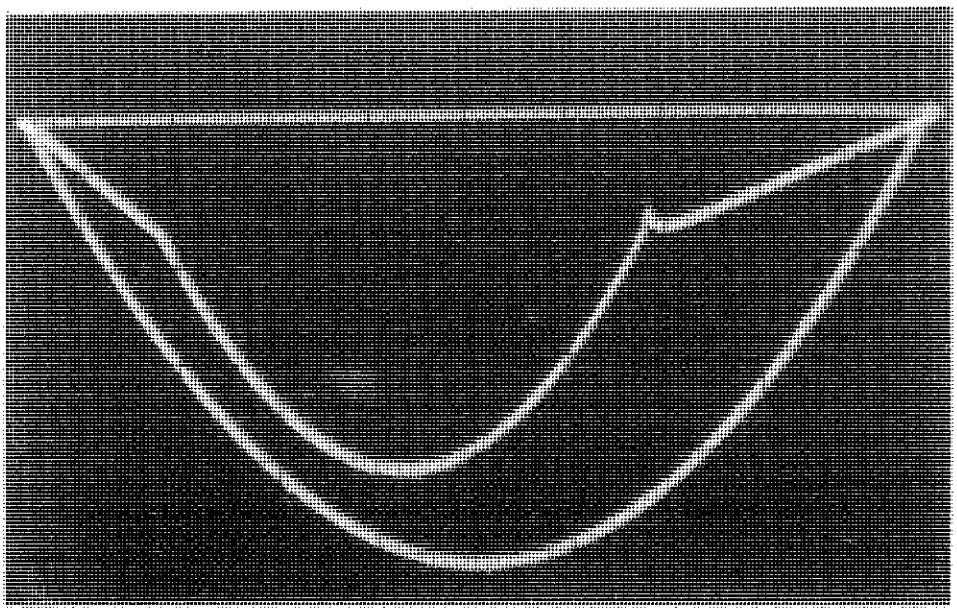


Abb. 3

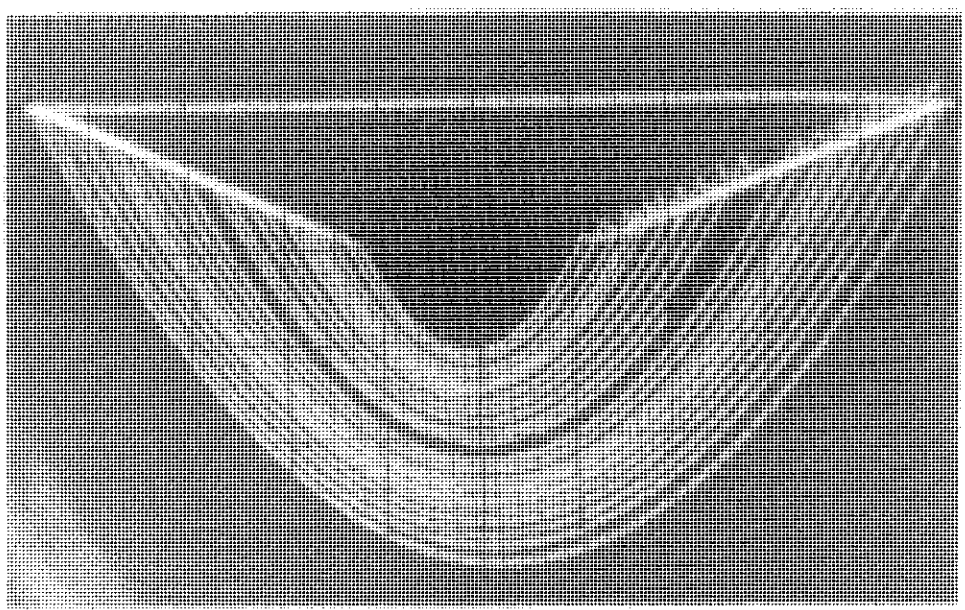


Abb. 4

INTERFACE April 1981