Anwendungsprogramme für den Computer Aided Design

Autor(en): **Vogel, J.**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins : gemeinsames Publikationsorgan des Schweizerischen Elektrotechnischen Vereins (SEV) und des Verbandes Schweizerischer Elektrizitätswerke (VSE)**

Band (Jahr): **62 (1971)**

Heft 21

PDF erstellt am: **29.05.2024**

Persistenter Link: https://doi.org/10.5169/seals-915864

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.
Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.
Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Ein Dienst der *ETH-Bibliothek*
ETH Zürich, Rämistrasse 101, 8092 Zürich, Schweiz, www.library.ethz.ch

http://www.e-periodica.ch

Anwendungsprogramme für den Computer Aided Design

Vortrag, gehalten an der Diskussionsversammlung des SEV vom 22. Juni 1971 in Zürich,
von *J. Vogel*, Zürich

681,3,01:62.002.612

1. Einleitung

Die Entwicklung leistungsfähiger Computer hat dem Elektroingenieur in den vergangenen Jahren ein attraktives Mittel in die Hand gegeben, einen Teil seiner Arbeit vom Laboratorium auf den Rechner zu verlagern. Dies betrifft sowohl zeitraubende Test- und Optimierungsaufgaben an bestehenden elektrischen Schaltungen und Netzwerken, als auch den Entwurf neuer Systeme. Da man vom Ingenieur nicht voraussetzen darf, dass er mit allen Finessen der Programmierung vertraut sei, muss man ihm besondere «Software», d. h. fertige Anwendungsprogramme zur Verfügung stellen, die er auf möglichst einfache Art für seine Probleme einsetzen kann.

Durch die Computerhersteller und die Elektroindustrie sind zu diesem Zweck Programme auf drei verschiedenen Einsatzstufen entwickelt worden: die Klasse der

a) Subroutinen, Prozeduren und Funktionsprogramme
b) die modularen Applikationsprogramme und
c) die Rahmenprogramme zur Entwicklung eigener Programmsysteme und Sprachen.

2. Subroutinen, Prozeduren, Funktionsprogramme

Von dieser heute verfügbaren Software nehmen die Subroutinen, Prozeduren und Funktionsprogramme den zahlenmässig grössten Umfang ein. Es handelt sich dabei nicht um selbständige, für sich allein anwendbare Programme, sondern um Programmteile, die in einer der gebräuchlichen höheren Computersprachen wie FORTRAN, ALGOL, PL/1, APL oder BASIC geschrieben sind. Diese Programmteile werden nun vom Benützer in Rahmenprogramme, die er selber entwickelt, eingesetzt. Dabei dürfen solche Unterprogramme in beliebiger Reihenfolge und beliebiger Häufigkeit aufgerufen werden.

Der Vorteil ihres Einsatzes liegt darin, dass der Benützer nicht selber besonders raffinierte Rechenalgorithmen suchen muss, welche Spezialisten der numerischen Mathematik entwickelt und bereits effizient programmiert haben; er erspart es sich so meistens, sich mit langwierigen und komplizierten Programmierarbeiten beschäftigen zu müssen.

Die Gebiete, auf welchen fertige Unterprogramme zur Verfügung stehen, sind vielfältig. Von besonderer Bedeutung für den Einsatz in der Elektrotechnik sind die Routinen für:

a) Matrizen-Arithmetik und Manipulation;
b) Berechnung von Eigenwerten;
c) Ermittlung der Wurzeln von Übertragungsfunktionen;
d) Integration und Differentiation;
e) Lösen von Gleichungssystemen;
f) Fourieranalyse;
g) Approximation und Interpolation;
h) Optimierung;
i) Statistik.

Sammlungen von Unterprogrammen, aus denen der Ingenieur die für ihn zweckmässigsten herausgreifen kann, existieren an den verschiedensten Stellen, insbesondere sind sie in praktisch allen grösseren Rechenzentrumsbibliotheken von Industriebetrieben vorhanden. Aber auch die beiden Technischen Hochschulen und die Universitäten von Zürich und Bern bieten den Benützern eine Vielfalt von Routinen in FORTRAN ALGOL und PL/1 an. Die Computerindustrie hat sich hier sehr verdient gemacht, indem sie ihren Kunden umfangreiche Sammlungen abgibt. Einige der gebräuchlichsten für FORTRAN sind SSP (Scientific Subroutine Package), MATHPACK, STATPACK und MATH, für PL/1 eine neuere Sammlung namens PL-MATH (Procedure Library Mathematics).

Voraussetzung dafür, dass ein Elektroingenieur von diesen Subroutinen zweckmässig Gebrauch machen kann, ist eine gute Programmiererfahrung in einer höheren Programmiersprache. Um das Rahmenprogramm schreiben zu können, ist es unumgänglich, dass er die notwendige Netzwerktheorie beherrscht und mit der numerischen Mathematik vertraut ist, um eben die Theorie in eine Form umzusetzen, aus der der Computer dann zuverlässige numerische Resultate berechnen kann. Schliesslich ist es auch fast immer nötig, dass der Ingenieur mit Matrizen umzugehen versteht.

3. Benützer-Programmsysteme

Man kann nun nicht von jedem Elektroingenieur voraussetzen, dass er alle Bedingungen für den effizienten Einsatz von Subroutinen erfüllt. Ausserdem darf der Computer Aided Design nicht nur ein Privileg des Hochschulingenieurs sein; auch Absolventen der Technika und der Gewerbeschulen sollten den Computer für die Entwicklung von elektrischen Schaltungen einsetzen können. Es wurden deshalb von den Computerherstellern, von Softwarefirmen, Hochschulen und Industriefirmen integrierte Programm-Systeme entwickelt. Solche Systeme existieren auf den verschiedensten, nachfolgend in vier Hauptklassen unterteilten Gebieten:

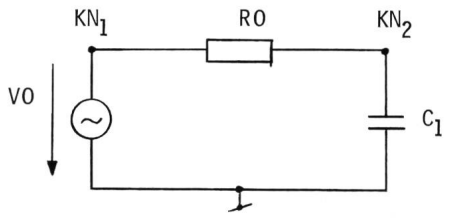
a) Programme für die Netzwerkanalyse;
b) Simulatoren für kontinuierliche Systeme;
c) Spezielle Designprogramme;
d) Programme für die Netzwerksynthese.

3.1 Netzwerkanalyse

Die wohl bekannteste und am weitesten verbreitete Klasse ist die der Netzwerk-Analysenprogramme. Wie der Name sagt, dienen sie dazu, Starkstromnetze oder elektronische Schaltungen zu analysieren, wobei bei diesen noch zwischen analogen und digitalen Schaltungen unterschieden wird.

Ein Analysenprogramm muss einer ganzen Reihe von Anforderungen genügen, damit es den Entwicklungsingenieur wirklich dazu verleitet, den Lötkolben aus der Hand zu legen, seinen Messplatz zeitweilig zu verlassen und einen Teil der Arbeit dem Computer zu überlassen.

Das Programm muss eine einfache, leichtverständliche Eingabesprache besitzen, mit der sich auch der Computerlaie in kürzester Frist, d. h. in einigen Stunden vertraut machen



R0, KN1—KN2, 100
C1, KN2—GROUND, 1E—6
V0, KN1—GROUND=0,75

Fig. 1
Topologische Netzwerkbeschreibung

kann. Es soll dem Benützer umfassende Information über die ihn interessierenden Grössen liefern, mindestens soviel Information wie er auch am Messplatz erhalten konnte. Es muss einfach sein, Parametermodifikationen durchzuführen, und auch Änderungen der Schaltungstopologie sollten erlaubt sein. Das Programm muss selber prüfen, ob die erhaltenen Resultate zuverlässig und reproduzierbar sind und gewissen Genauigkeitsanforderungen genügen.

Bei modernen Programmen ist es fast eine Selbstverständlichkeit, dass sie eingebaute Halbleitermodelle besitzen oder zumindest erlauben, derartige Modelle abzuspeichern. Weil die Werte von Bauelementen in der Produktion streuen, muss man die Möglichkeit haben, die Parameter mit Toleranzen zu versehen und statistische Untersuchungen durchzuführen.

Neben linearen Netzwerken will man oft auch nichtlineare Schaltungen untersuchen, und schliesslich muss dem Ingenieur die Option in die Hand gegeben werden, je nach Wunsch Gleich-, Wechselstrom- oder transiente Analysen an passiven und aktiven Schaltungen vorzunehmen.

Um dem Bedürfnis einer einfachen Eingabesprache nachzukommen, besitzen praktisch alle Analysenprogramme die Eigenschaft, dass ein Netzwerk topologisch beschrieben werden kann, und dass man nicht selber Kirchhoffsche Knoten- und Maschengleichungen aufstellen muss. Es ist nur noch notwendig, alle Knoten und Zweige mit einem eindeutigen Namen oder einer Zahl zu versehen und anzugeben, zwischen welchen Knoten sich die einzelnen Zweige bzw. Elemente befinden und welchen Wert sie haben.

Ein einfaches Beispiel zeit Fig. 1. Die Knoten sind mit KN1, KN2 und GROUND bezeichnet, die Elemente mit R0, C1 und V0.

Die topologische Beschreibung für den ersten Zweig lautet R0, KN1–KN2, 100 und sagt aus, dass sich der Widerstand R0 zwischen den Knoten KN1 und KN2 befindet und einen Wert von 100 besitzt. Analog befindet sich die Kapazität C1 zwischen den Knoten KN2 und GROUND und hat einen Wert von 10^{-6}. Schliesslich ist V0 eine Spannungsquelle mit dem Wert 0,75. Die Maßstabseinheiten kann der Benützer frei wählen.

Die Elemente, die in den gebräuchlichsten Analysenprogrammen Verwendung finden, sind: Widerstände, Kapazitäten, Induktivitäten, Gegeninduktivitäten, unabhängige Strom- und Spannungsquellen, abhängige, d. h. gesteuerte Strom- und Spannungsquellen, Dioden und Transistoren.

Für die Änderung von Parametern existieren Optionen wie MODIFY; für Topologieänderungen, wie z. B. das Kurzschliessen von Zweigen, Befehle wie CHANGE, TOPOLOGY.

Was nun die Informationen anbelangt, die der Benützer erhalten kann, sind diese von Programm zu Programm verschieden. Üblich sind numerische Werte von Strömen, Zweig- und Knotenspannungen, Leistungen, die Empfindlichkeit gegenüber Parametervariationen, die Ermittlung der schlimmsten Grenzwerte, welche Knotenspannungen annehmen können, wenn sich alle Parameter an den kritischen Grenzen ihrer Streuungsbereiche befinden und schliesslich statistische Angaben über Strom- und Spannungswerte, wenn die Parameter über

```
      TRANSFER FUNCTION     MS   =     ZNUM*N(S)*KN /    ZDEN*D(S)*KD
      ZNUM =      1.0000     (       )   "  N(S) = +V( 6)                        , KN        1.0000
      ZDHN =      1.0000     (       )   "  D(S) = NOT REQUESTED - SET = 1.0 ,   KD        1.0000

MS     FOLLOWS, MULTIPLIER = 1.000000E 00

  NUMERATOR COEFFICIENTS, DEGREE = 6.  DESCENDING ORDER

       3.2159974E 08        1.3246746E 09       2.8207462E 09        4.6146765E 09       2.7371256E 09
       3.4920200E 09        0.0

              REAL              IMAG

            0.0               0.0
            0.0               1.054092E 00
            0.0              -1.054092E 00
           -7.500001E-01     -1.780125E 00
           -7.500001E-01      1.780125E 00
           -2.619013E 00      0.0

DENOMINATOR COEFFICIENTS, DEGREE = 7.  DESCENDING ORDER

       1.6080000E 06        4.1719936E 08       2.1563159E 09        5.2469842E 09       7.5568251E 09
       4.3164672E 09        3.6144691E 09       0.0

              REAL              IMAG

            0.0               0.0
           -1.108414E-01      8.346870E-01
           -1.108414E-01     -8.346870E-01
           -1.177269E 00     -1.822900E 00
           -1.177269E 00      1.822900E 00
           -2.648344E 00      0.0
           -2.542278E 02      0.0
```

Fig. 2
Numerische Ausgabe von Übertragungsfunktion, Polen und Nullstellen beim Programm LISA-360

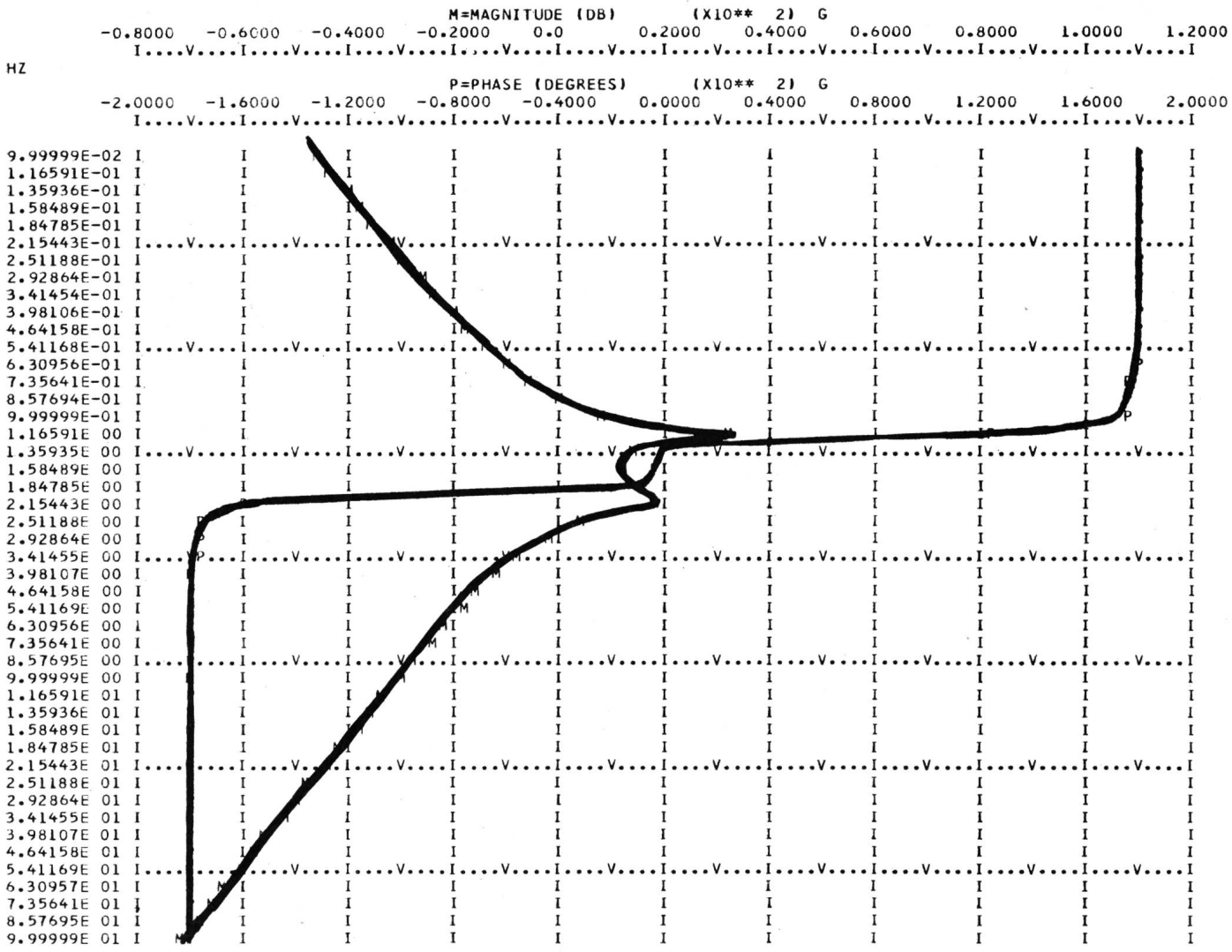


Fig. 3
Graphische Ausgabe (Bodediagramm) mit dem Programm LISA-360

gewisse Toleranzbereiche streuen und eine grosse Anzahl von Fällen ausgewertet wird. Es gibt einige Programme, die echte Monte-Carlo-Statistiken mit einer beliebig grossen Anzahl von gerechneten Fällen durchführen.

Bei Systemen, die in der Frequenzebene arbeiten, können meistens die Wurzeln der Schaltung, d. h. Pole und Nullstellen berechnet werden. Diese Grössen geben dem Ingenieur wertvolle Hinweise in bezug auf die Stabilität einer Schaltung. Mittels einer Root-Locus-Option ist es möglich, die Verschiebung der Wurzelorte in der Bildebene zu beobachten, wenn sich ein Netzwerkparameter ändert; insbesondere wird man verfolgen, ob sich irgendwelche Pole in die rechte Halbebene bewegen.

Zu jeder Analyse in der Frequenzebene gehört üblicherweise ein Bodediagramm und Angaben über die Gruppenlaufzeit. Transiente Analysen wird man vornehmen, wenn aperiodische Vorgänge, z. B. Schaltvorgänge, Ein- und Ausschwingvorgänge untersucht werden sollen und wenn sich Nichtlinearitäten im Netzwerk befinden. Fast alle modernen Programme, arbeiten sie nun im Zeit- oder im Frequenzbereich, erlauben solche transiente Analysen.

Was die Rechen- und Ausgabebefehle anbelangt, sind sie in der Sprache des Ingenieurs gehalten. Beispiele dafür sind:

```
PRINT, CURRENTS, WORST CASE
COMPUTE, BODE, VIN, IOUTPUT, H21
PLOT, VIN, DRIVER
```

Die Ausgabe der Resultate kann je nach der verfügbaren Hardware auf verschiedene Art und Weise erfolgen. Numerische Werte erhält man über Schreibmaschinenterminals, Zeilendrucker und auch über den Bildschirm. Wünscht sich der Benützer sofort ein Bild von gewissen Kurvenverläufen zu machen, so kann er auch dazu vielfach gerade den Zeilendrucker gebrauchen. Schönere und exaktere Bilder ergeben sich natürlich mittels Kurvenzeichner und graphischem Terminal.

Die Fig. 2 und 3 zeigen die numerische Ausgabe von Übertragungsfunktion, Polen und Nullstellen beim Programm LISA, sowie ein dazugehöriges Bodediagramm mit den Verläufen von Betrag und Phase.

Die Anzahl der bis heute vorwiegend in den USA entwikkelten Anwendungsprogramme ist sehr gross [1][1]). Leider sind viele dieser Programme in Europa nicht verfügbar, da sie militärischen Ausfuhr-Restriktionen der USA unterliegen. Die Programme, die in der Schweiz erhältlich und an verschiedenen Rechenzentren verfügbar sind, wurden in Fig. 4 unterstrichen.

Das wohl am meisten verbreitete und bekannteste ist ECAP, das Electronic Circuit Analysis Program, welches auch unter dem Namen GOCAP und PAN erhältlich ist. Es ist ein Programm der 1. Generation und leistet trotz seines für Computerprogramme schon fast ehrwürdigen Alters noch vielfach

[1]) Siehe Literatur am Schluss des Aufsatzes.

1. Generation	2. Generation
ECAP (GOCAP PAN)	LISA
NET - 1	CORNAP
PREDICT	PANE
SCEPTRE	SACSY
CALAHAN	TRAC
CIRCUS	ANCIR
NASAP	PCAP
	RIPS
	CADIC
	DCCAP
	SNAP
	AC-DC-CIP
	SYSCAP
	STAEN
	ECAP II

Fig. 4
Netzwerk-Analysenprogramm

gute Dienste, wenn auch seine grossen Mängel nicht übersehen werden können. NASAP ist ein interessantes Programm, das die Gemeinschaftsarbeit einer ganzen Reihe von amerikanischen Universitäten darstellt. Mittels NASAP wollte man erstmals versuchen, die Duplikation von gleichartigen Entwicklungen zu verhindern. Solche Koordinationen sollten auch in Zukunft vermehrt angestrebt werden, wenn man die Tatsache berücksichtigt, dass in den USA im Jahr 1969 etwa 100 Millionen Dollar allein für Software-Entwicklungen auf dem Gebiet der Elektrotechnik ausgegeben wurden.

Die beiden jüngsten Vertreter der Programme der zweiten Generation verdienen besondere Erwähnung. Bei STAEN handelt es sich um den europäischen Nachfolger von SCEPTRE, dem wohl erfolgreichsten Analysen-Programm in den USA. Es besitzt besondere Vorzüge bei der Behandlung von nichtlinearen Schaltungen und bei der Abspeicherung von beliebig komplexen Halbleitermodellen. ECAP II schliesslich ist der langersehnte Nachfolger von ECAP. Es wird Ende 1971 verfügbar sein und wesentliche Erweiterungen und Verbesserungen gegenüber der alten Version enthalten.

Die wichtigsten Merkmale, nach denen man heute solche Analysenprogramme beurteilt, sind die verwendeten Rechenmethoden, insbesondere zeitsparende Matrizentechniken oder State-Variable-Methoden, der Bereich, in dem die Programme arbeiten, die Möglichkeit Tabellen, z. B. für nichtlineare Funktionen oder Parameterwerte vorzusehen, die Möglichkeit der analytischen Beschreibung von Signalen, Funktionen und Netzwerkelementen und schliesslich Speicherbedarf und Rechengeschwindigkeit.

Detaillierte Angaben über die in Fig. 4 aufgeführten Programme finden sich in [1] und [2].

Was die verfügbaren Programme zur Untersuchung von digitalen Schaltungen anbelangt, ist die Auswahl hier nicht sehr gross. Wohl haben grössere Computerhersteller hervorragende Programme entwickelt, verwenden sie aber ausschliesslich für eigene Zwecke, so dass sie der Öffentlichkeit nicht zugänglich sind.

DICAP und TIPSI sind Programme, die kürzlich in der deutschen Literatur beschrieben wurden. TIPSI erlaubt es, taktgesteuerte Schaltungen zu untersuchen, und ist für den Terminal-Betrieb entwickelt worden. Bei LOGIC wird an einer verbesserten Version gearbeitet und ADN wurde aus einer Studienarbeit an der ETH Zürich weiterentwickelt.

Das «Digital System Modeling Program» stellt eine schweizerische Neuentwicklung dar. Es erlaubt die Simulation von 200 logischen Schaltkreisen mit synchroner oder asynchroner Steuerung und liefert auch gerade Unterlagen zur Erstellung von Schaltungen.

Bei den Systemen zur Berechnung von Starkstromnetzen sind in der Schweiz die Programme PSP (Power System Planning, IBM), ELEC (Univac) und LOADFLOW (CDC, Univac, IBM) erwähnenswert. Man kann mit diesen Programmen Lastflussberechnungen vornehmen, Kurzschlüsse simulieren und transiente Stabilitäten untersuchen. PSP besitzt ausserdem noch eine Reihe von Daten-Management-Routinen, die es gestatten, alle interessierenden Daten eines Starkstromnetzes zu speichern, nachzuführen und für Berechnungen auszuwerten.

3.2 Simulation kontinuierlicher Systeme

Die zweite grosse Klasse umfasst die Programme zur Simulation kontinuierlicher Systeme. Unter diesen versteht man Systeme, die sich in Form von Differentialgleichungen beschreiben oder mit Hilfe eines Funktions-Blockschemas darstellen lassen. Man hat schon seit langem solche Simulationen mit Hilfe von Analog- oder Hybridcomputern vorgenommen. Da aber derartige Rechner nicht jedermann zugänglich sind und ausserdem gewisse Nachteile in bezug auf Genauigkeit, Programm-Logik, Umfang der Systeme und Entwicklungskosten der Simulationsmodelle aufweisen, ist man dazu übergegangen, Simulationsprogramme für Digitalcomputer zu schreiben.

In Fig. 5 und 6 ist die Geschichte dieser Programme dargestellt, die 1955 begonnen hat. Die digitalen Analogsimulatoren haben weitgehend die Analogcomputer verdrängt. CSMP 1130 ist das am weitesten verbreitete Programm auf diesem Gebiet. Erst kürzlich wurde eine erweiterte Version, das CSMP II 1130, angekündigt, es wird in Kürze den Benützern des Systems IBM 1130, welches ein kleinerer Rechner für wissenschaftliche Zwecke ist, zur Verfügung stehen.

Name	Herkunft	Datum	Geschrieben für Computer
SELFRIDGE	USNOTS Inkoyern	1955	IBM 701
DEPI	Jet Propulsion Lab.	1957	Burroughs 204
ASTRAL	Convair	1958	IBM 704
DYSAC	University of Wisconsin	1961	CDC 1604
DYNASAR	General Electric	1961	IBM 7090
MIDAS	Wright-Patterson	1963	IBM 7090-94
PACTOLUS	IBM Research Lab.	1964	IBM 1620
CSMP	IBM	1966	IBM 1130
CSMP II	IBM	1971	IBM 1130

Fig. 5
Digitale Analogsimulatoren

Die Simulatoren für kontinuierliche Systeme verdrängen in vielen Fällen die Hybridrechner. Hier hat die Entwicklung 1965 begonnen. MIMIC und DSL wurden für Computer der 2. Generation geschrieben und haben sehr viele Anhänger gefunden. Man darf aber nicht übersehen, dass ihr Konzept heute überholt ist und ihre Flexibilität und Auswahl an Optionen zu wünschen übrig lässt. Verschiedene Firmen sind deshalb dazu übergegangen, MIMIC durch eigene Routinen zu ergänzen und den modernen Bedürfnissen in bezug auf Flexibilität und Auswahl an Optionen anzupassen. CSMP 360 (Continuous System Modeling Program) ist eine äusserst erfolgreiche Weiterentwicklung von DSL/90 für Computer der 3. Generation und momentan auf den Systemen IBM/360 und /370 verfügbar.

Vor kurzem wurden schliesslich die beiden jüngsten Glieder der Familie, CSMP III und CSMP III Graphic Feature, für dieselben Systeme angekündigt, wodurch der inter-

Name	Herkunft	Datum	Geschrieben für Computer
DSL / 90	IBM Develop. Lab	1965	IBM 7090
MIMIC	Wright-Patterson	1965	IBM 7090
SLASH	U.S. Air Force	1965	Burroughs B-5000
DSL / 40	IBM Develop. Lab	1966	IBM 7040
CSMP-360	IBM Develop. Lab	1967	IBM / 360

Fig. 6
Simulatoren für kontinuierliche Systeme

aktive Einsatz von Bildschirm-Terminals für die Simulation ermöglicht wird.

Hervorstechendes Merkmal der CSMP-Sprache ist ihre Ähnlichkeit mit der Sprache des Ingenieurs oder Mathematikers. Beispielsweise lautet eine CSMP-Instruktion:

GENERAL FORM	FUNCTION
Y = INTGRL (IC, X) Y (0) = IC INTEGRATOR	$Y = \int_0^t X\,dt + IC$ EQUIVALENT LAPLACE TRANSFORM: $\frac{1}{S}$
Y = DERIV (IC, X) $\dot{X}$ (t = 0) = IC DERIVATIVE	$Y = \frac{dX}{dt}$ EQUIVALENT LAPLACE TRANSFORM: S
Y = DELAY (N, P, X) P = DELAY TIME N = NUMBER OF POINTS SAMPLED IN INTERVAL P (INTEGER) DEAD TIME (DELAY)	Y (t) = X (t - P) $t \geq P$ Y = 0 $t < P$ EQUIVALENT LAPLACE TRANSFORM: e^{-PS}
Y = ZHOLD (X_1, X_2) ZERO-ORDER HOLD	Y = X_2 $X_1 > 0$ Y = LAST OUTPUT $X_1 \leq 0$ Y (0) = 0 EQUIVALENT LAPLACE TRANSFORM: $\frac{1}{S}(1 - e^{St})$
Y = IMPL (IC, P, FOFY) IC = FIRST GUESS P = ERROR BOUND FOFY = OUTPUT NAME OF LAST STATEMENT IN ALGEBRAIC LOOP DEFINITION IMPLICIT FUNCTION	Y = FUNCT (Y) |Y - FUNCT (Y)| ≤ P|Y|

Fig. 7
Funktionsblöcke des CSMP-360 (Continuous System Modeling Program): Mathematische Funktionen

A = A*COS(Z1)/EXP(Z2)+B*GAUSS (1, MITTEL, SIGMA)+INTGRL(ANFBED, Z3), hat also eine Form, die dem Fachmann sofort verständlich ist. Funktionsblöcke wie diejenigen zur Bildung von Zufallszahlen oder Integratoren werden durch Schlüsselwörter bezeichnet und aufgerufen. Der Benützer verfügt über eine Auswahl von etwa 40 derartigen Blöcken, die er in beliebiger Häufigkeit und beliebiger Reihenfolge in sein System einsetzen kann.

Aus den Fig. 7 und 8 ist eine kleinere Auswahl dieser Blöcke ersichtlich. Sie dienen zur Darstellung von Funktionselementen, die in der Elektrotechnik zum täglichen Brot gehören. Daneben kann der Benützer beliebige FORTRAN-Instruktionen für logische Entscheidungen und Verzweigungen in sein Programm einfügen und dieses auf sehr flexible Weise noch durch Subroutinen und Funktionsprogramme ergänzen. Der Aufbau und die Strukturierung von CSMP ist derart, dass der Ingenieur mit relativ wenig Aufwand ganze Optimierungsaufgaben vollständig automatisieren kann.

Dass daneben vielfältige Instruktionen für numerische und graphische Ausgabe nicht fehlen, versteht sich von selbst. Fig. 9 enthält eine graphische Ausgabe auf dem Zeilendrucker. Darüber hinaus ist es aber auch möglich, Bildschirmterminals für die Entwicklung von CSMP-Simulationsmodel-

GENERAL FORM	FUNCTION
Y = AFGEN (FUNCT, X) ARBITRARY FUNCT. GEN. (LINEAR INTERPOLATION)	Y = FUNCT (X)
Y = NLFGEN (FUNCT, X) ARBITRARY FUNCT, GEN. (QUADRATIC INTERPOLATION)	Y = FUNCT (X) $X_0 \leq X \leq X_n$
Y = LIMIT (P_1, P_2, X) LIMITER	Y = P_1 $X < P_1$ Y = P_2 $X > P_2$ Y = X $P_1 \leq X \leq P_2$
Y = QNTZR (P, X) QUANTIZER	Y = kP $(k-1/2)P < X \leq (k+1/2)P$ k = 0, ±1, ±2, ±3...
Y = DEADSP (P_1, P_2, X) DEAD SPACE	Y = 0 $P_1 \leq X \leq P_2$ Y = X - P_2 $X > P_2$ Y = X - P_1 $X < P_1$
Y = HSTRSS (IC, P_1, P_2, X) Y (0) = IC HYSTERSIS LOOP	Y = X - P_2 $(X - X_{n-1}) > 0$ AND $Y_{n-1} \leq (X - P_2)$ Y = X - P_1 $(X - X_{n-1}) < 0$ AND $Y_{n-1} \geq (X - P_1)$ OTHERWISE Y = LAST OUTPUT

Fig. 8
Funktionsblöcke des CSMP-360: Funktionsgeneratoren

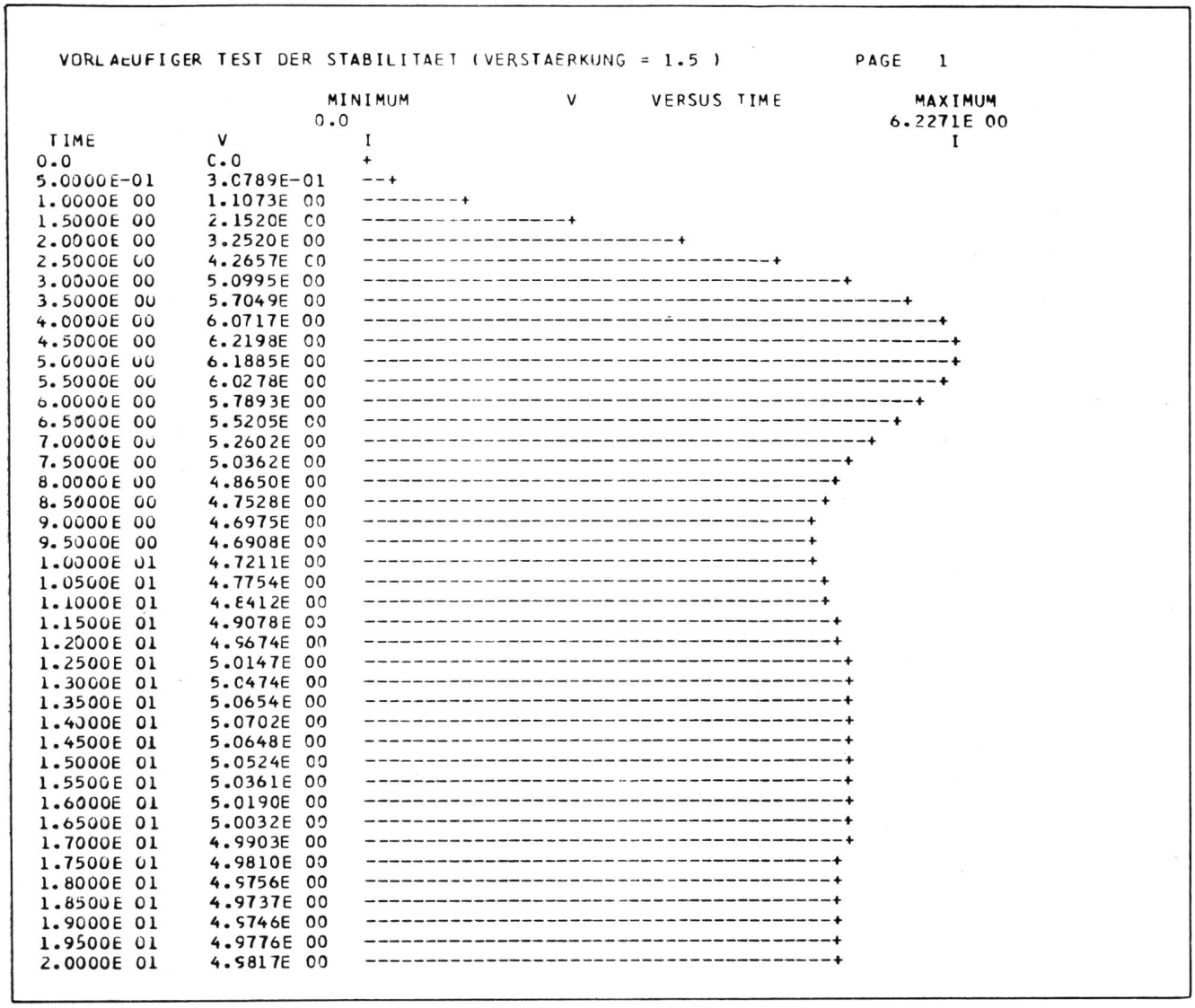


Fig. 9
Graphische Druckerausgabe mit CSMP-360 (Continuous System Modeling Programm)

len einzusetzen. Auf interaktive Weise, d. h. in Konversation mit dem Computer kann der Ingenieur eine Schaltung auf bauen und austesten. Die Ausgabe der Resultate erfolgt dann ebenfalls in numerischer oder graphischer Form direkt auf dem Bildschirm.

3.3 Spezielle Designprogramme

Eine weitere Klasse von Anwendungsprogrammen sind solche, die für besondere «Design-Probleme» entwickelt wurden. Die wichtigsten unter ihnen sind Programme, die es gestatten, die Auslegung von gedruckten Schaltungen zu optimieren und solche, die sich mit dem Design von integrierten Schaltungen befassen. Auch hier muss man sich wieder mit der Tatsache abfinden, dass die Computerhersteller sehr effiziente Programme auf diesen Gebieten entwickelt haben, sie aber praktisch ausschliesslich für den eigenen Gebrauch reservieren.

Immerhin sind in der Schweiz doch drei Programme, UCARDS, ACCEL und COPAR, verfügbar, welche dem Entwurf von gedruckten Schaltungen dienen. Sie kombinieren selbständig Bauteile zu grösseren Elementen, teilen den Print entsprechend den Ausmassen der grössten auftretenden Elemente ein, plazieren die Bauteile und optimieren dann die Länge der Verbindungswege durch passende Austauschverfahren. Ist einmal die beste Plazierung erreicht, so erfolgt das Ziehen der Verbindungen. Wo dies notwendig ist, werden Bohrungen von Löchern vorgesehen und die Verbindungslinien auf der Unterseite fortgesetzt. Als Resultate erhält der Benützer eine Zeichnung des Layouts, in Form von Lochstreifen oder Magnetbändern Angaben zur automatischen Steuerung von Bohrmaschinen sowie Ätzmasken für die Prints.

Mittels UCARDS kann beispielsweise die optimale Auslegung von maximal 250 Bauteilen zwischen 200 Knoten mit 40 Eingabe/Ausgabepunkten erfolgen. Das Programm benötigt einen Arbeitsspeicher in der Grösse von 240 kbit. Einen Anhaltspunkt für die notwendige Rechenzeit geben die nachfolgenden Erfahrungswerte. Auf einem System IBM/360–75 dauerte die Anordnung von 48 Bauteilen auf einem 60×60 Gitter 4,4 min, für 100 Bauteile auf demselben Gitter waren 15 min notwendig.

3.4 Syntheseprogramme

Unter einem idealen Syntheseprogramm würde man ein solches verstehen, das aus einem Satz von möglichen Bauelementen eine derartige Auswahl und Anordnung trifft, dass einerseits eine gewünschte Funktion innerhalb gewisser Toleranzen erzielt wird, anderseits die Anzahl der notwendigen Bauelemente minimal wird.

Leider ist es aber heute so, dass noch keine allgemein verwendbaren Programme existieren, die für komplexe nichtlineare

und aktive Schaltungen eingesetzt werden könnten. Auf dem Gebiet der Filtersynthese sind in der Industrie wohl sehr gute Programme entwickelt worden, und auch an technischen Hochschulen, u. a. an den Instituten für Fernmeldetechnik und Technische Physik der ETH Zürich, wurden in dieser Richtung hervorragende Arbeiten geleistet. Dass keine generell verwendbaren Programme existieren, heisst noch nicht, dass der Ingenieur deshalb die Flinte ins Korn werfen muss. Sofern er nämlich mit einer höheren Programmiersprache umgehen kann und daneben etwas von Optimierungsmethoden versteht, kann er ohne weiteres eines der bestehenden Analysen- oder Simulationsprogramme durch eigene Programmteile ergänzen und für Synthesen einsetzen. Dass es sich hier um keine Utopie handelt, sondern um eine reelle Möglichkeit, soll ein einfaches Beispiel mit CSMP zeigen (Fig. 10).

Die Idee bei der Synthese ist, dass das Programm eine Schaltung derart dimensioniert, dass es aus einer vorgegebenen Eingangsgrösse eine gewünschte Ausgangsgrösse erzeugt. Beginnt man mit irgendeiner Struktur mit willkürlich gewählten Parametern, so wird das effektive Ausgangssignal sich wesentlich vom gewünschten Ausgangssignal unterscheiden. Die Differenz, d. h. der Fehler, wird nun passend verarbeitet, so dass Korrekturen für die Netzwerkparameter entstehen. Beim zweiten Mal werden Ausgangssignal und gewünschtes Signal schon näher beieinander liegen und auf iterative Weise wird man so die optimale Schaltung synthetisieren, bei der der Fehler minimal wird.

Das Fehlerkriterium kann die Form

$$\int_R W(\varrho) \cdot F(\varrho)^n \cdot d\varrho \qquad (1)$$

haben. F ist der Fehler, n ein beliebiger Exponent und R der Arbeitsbereich, in welchem die Optimierung vorgenommen werden soll. Dieser kann ein Zeitintervall, ein Frequenz- oder ein Temperaturbereich sein. $W(\varrho)$ schliesslich ist eine Gewichtsfunktion, mit der gewisse Teile des Arbeitsbereiches, in denen der Fehler möglichst klein gehalten werden soll, besonders hervorgehoben werden können.

In Fig. 11 ist ein einfaches Basisnetzwerk dargestellt. Als Eingangssignal haben wir einen Dreieckimpuls und am Ausgang wird folgendes Signal gewünscht:

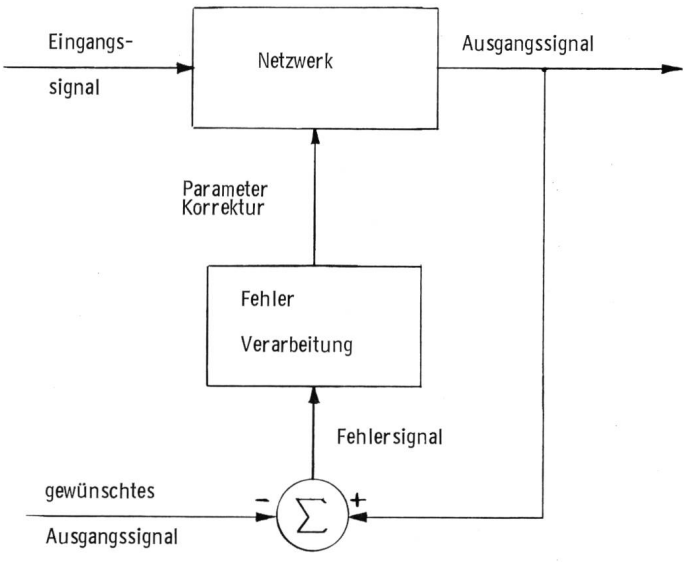


Fig. 10
Prinzipschema zur Netzwerksynthese

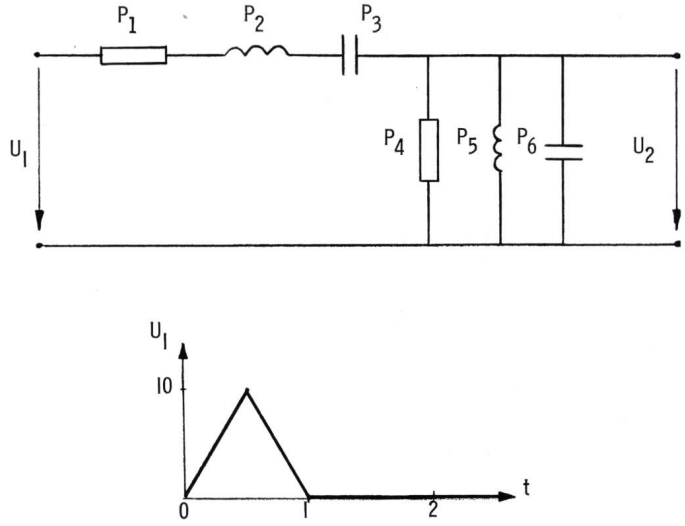


Fig. 11
Basisnetzwerk mit 6 zu optimierenden Netzwerkparametern

$$U_{2SOLL} = 3 \cdot t \cdot \exp(-t) \cdot \sin[2{,}5\, t \cdot \exp(-0{,}2\, t)]$$

also etwas ziemlich Ausgefallenes, von dem man a priori sagen würde, dass es nicht möglich ist, dies zu erreichen. Natürlich ist es nicht möglich, diese Funktion genau zu erreichen, aber man kann die 6 Parameter des Netzwerkes doch derart optimieren, dass der Unterschied zwischen dem effektiven U_2 und U_{2SOLL} möglichst klein wird. Das Fehlerkriterium lautet:

$$\text{Fehler} = \int_0^{t_{ENDE}} (U_2 - U_{2SOLL})^2\, dt \longrightarrow \text{Minimum} \qquad (2)$$

Man bildet das Quadrat des Fehlers und integriert über den interessierenden Zeitbereich.

Die Verwendung der Gradientenmethode nach *Fletcher-Powell* [3; 4] und einer linearen Suche nach *Fibonacci* [3] ergaben nachfolgende Resultate:

Zu Beginn wurden alle sechs Parameter zu 0,05 gewählt, der Fehler entsprechend Gl. (2) betrug 14,4. Nach 11 Iterationen war der Fehler bereits auf $9{,}8 \cdot 10^{-3}$ gesunken; die Parameterwerte hatten sich sehr weit von ihren ursprünglichen Werten entfernt und betrugen P1 = 1,4, P2 = 0,037, P3 = 10^{-6}, P4 = 1,2, P5 = 1,5 und P6 = 1,6. Die Rechenzeit für diese Optimierung war auf einem System IBM/360–40 6 min und auf dem Modell/360–65 50 s.

Es ist also möglich, bestehende Anwendungsprogramme für Synthese-Aufgaben einzusetzen, wobei aber anzunehmen ist, dass in einigen Jahren selbständige Programmsysteme verfügbar sein werden, welche ausschliesslich diesem Zweck dienen und mit einer Auswahl von Optimierungsmethoden ausgestattet sein werden.

4. Rahmenprogramme für Modularsysteme

Die Philosophie dieser Programme ist die, dass man dem Benützer eine Reihe von Werkzeugen zur Verfügung stellt, mit denen er ein eigenes grösseres Anwendungsprogramm schreiben kann, welches er mit einer eigenen ihm besonders aussagefähigen, höheren Programmiersprache versieht. PLAN — 360 (Problem Language Analyser) ist bisher das einzige verbreitete System, welches diese Idee verwirklicht und dem Anwendungsprogrammierer Hilfsmittel in Form eines speziellen Submonitors, eines Teils zur Sprach-Definition, eines Sprachinterpreters, eines umfangreichen Fortran-Subroutinen-Paketes und eines Diagnose-Überwachungsprogrammes zur Verfügung stellt.

Bisher wurden mittels PLAN vorwiegend Anwendungsprogramme aus der Mechanik und Optik entwickelt, die letzte Neuschöpfung stammt aber aus der Elektronik, indem ECAP II, das letzte der früher beschriebenen Netzwerkanalyser-Programme, auf PLAN basiert.

PLAN und ähnliche Programmsysteme werden voraussichtlich in der nächsten Zukunft noch stark an Verbreitung zunehmen, insbesondere dann, wenn sie es auch erlauben, graphische Bildschirmterminals in den Designprozess miteinzubeziehen und so dem Benützer eine dialogförmige, interaktive Arbeit mit dem Computer gestatten.

Wenn diese Zusammenstellung auch keinen Anspruch auf Vollständigkeit erhebt, so sollte sie doch eine Übersicht über den «State of the Art», den momentanen Stand der verfügbaren Software auf dem Gebiet des Computer-Aided-Design geben.

Literatur

[1] Electronics' guide to CAD programs. Electronics 43(1970)8, p. 109...112.

[2] *D. F. Dawson, F. F. Kuo* and *W. G. Magnuson:* Computer-aided design of electronic circuits. A user's viewpoint. Proc. IEEE 55(1967)11, p. 1946...1954.

[3] *J. W. Bandler:* Optimization methods for computer-aided design. Trans. IEEE MTT 17(1969)8, p. 533...552.

[4] *F. F. Kuo* and *W. G. Magnuson:* Computer oriented circuit design. Englewood Cliffs, N. J., Prentice Hall, 1969.

Adresse des Autors:

Dr. *J. Vogel*, dipl. El.-Ing. ETH, International Business Machines, Talstrasse 66, 8022 Zürich.

Commission Electrotechnique Internationale (CEI)

36. Haupttagung vom 9. bis 19. Juni 1971 in Brüssel [1])

CE 1, Terminologie

Le Comité d'Etudes 1 s'est réuni à Bruxelles les 11, 12 et 14 juin. Maintenant que la plus grande partie de l'élaboration du VEI se fait dans les Groupes de Travail réunissant les spécialistes des domaines concernés, les séances plénières du CE 1 sont consacrées presqu'exclusivement aux problèmes généraux, à la coordination des travaux et à l'examen des résultats des votes.

Après avoir approuvé le procès-verbal de la réunion de Washington, le CE 1 a discuté de la nouvelle numération des chapitres du VEI et de leur répartition en classes. Après de longues discussions, il a été décidé qu'il y aura au total 10 classes: 8 classes pour les nouveaux chapitres, une classe pour les éditions avancées et une pour incorporer la 2e édition du VEI. Les classes sont:

classe 0 chapitres de la 2e édition du VEI
classe 1 Concepts généraux
classe 2 Matériaux et composants
classe 3 Mesures, régulation et calculs
classe 4 Matériels électriques
classe 5 Matériels électroniques
classe 6 Techniques de l'énergie
classe 7 Technique des télécommunications
classe 8 Applications particulières
classe 9 Editions avancées

Le CE 1 a ensuite pris acte d'un rapport de son Secrétaire donnant l'état des travaux en cours, puis procédé à la révision du document *1(Bureau Central)1007,* Procédure applicable aux travaux d'élaboration du VEI, document auquel il a apporté quelques modifications d'ordre mineur.

La révision du document *1(Bureau Central)1003,* Directives générales pour les travaux d'élaboration du VEI, a été étudiée d'abord par un Groupe de Travail ad hoc. Les décisions principales finalement adoptées par le CE 1 sont: Le terme anglais «concept» sera toujours traduit en français par «concept» vu que le terme «notion» peut avoir en français des sens différents. Ceci se justifie par le fait que le terme français concept, autrefois seulement employé en philosophie, entre maintenant dans la langue courante. Le paragraphe 1.2.4 est supprimé, ses deux alinéas étant mis à la fin du paragraphe 1.0: on évite ainsi le titre «notions semblables» qui prêtait à confusion. L'impression des termes se fera avec la première lettre en majuscule ou minuscule telle qu'elle apparaît dans la langue considérée à l'intérieur d'une phrase. Les termes définis ailleurs dans le même chapitre ne seront pas imprimés en italique. Dans chaque chapitre, l'ordre des concepts est en général choisi tel que les définitions nécessaires à la compréhension d'un terme le précèdent et l'usage de caractères italiques ne ferait que compliquer la lecture et augmenter les frais d'impression. L'annexe à l'ancien document donnant les décisions de l'ISO, et qui en son temps était nécessaire pour expliquer certaines décisions, ne sera plus reproduite. Toutefois, un tableau au début du document donnera la liste des documents ISO se rapportant à la terminologie et un tableau récapitulatif donnera à la fin des exemples concernant l'utilisation des symboles et, en particulier, des parenthèses ou des crochets. Les autres modifications sont rédactionnelles ou d'ordre mineur.

La révision du document *1(Bureau Central)1008,* Répartition des chapitres du VEI et des travaux de terminologie, tiendra compte de la nouvelle classe introduite. La création d'un chapitre concernant le domaine de l'optique électronique a été décidée et un Groupe de Travail préparatoire correspondant sera constitué. Le CE 1 reprendra le petit Groupe de Travail constitué auprès du Comité d'Action sur les matériels d'éducation. La proposition du Comité National des Etats-Unis de substituer «Graphic Symbols» à «Graphical Symbols» a été retirée par le représentant de ce pays.

La discussion sur le résultat du vote du document *1(Bureau Central)1021,* soumis à la Règle des Six Mois et qui a rencontré une forte approbation pour la première question (définitions des concepts de: caractéristiques assignées/rating, valeur nominale/rated value, valeur limite/limiting value et valeur dénommée/nominal value), mais beaucoup d'opposition à la deuxième question, c'est-à-dire aux termes choisis, a donné lieu à un débat fort animé. Il a, en fin de compte, été décidé de faire un nouveau document soumis à la Règle des Six Mois. Les nouveaux termes choisis ne contiendront ni en anglais, ni en français, le terme nominal: «Rated value» sera traduit en français par «valeur assignée»; «valeur dénommée» sera remplacé par «valeur de désignation» en français et par «designating value» en anglais. Toutefois, un renvoi ou une remarque pourra attirer l'attention du lecteur, si nécessaire, sur le terme «nominal» anciennement employé pour le concept en question.

Un document suédois *02(Suède)18,* adressé au Comité d'Action, demandant un renforcement des moyens pour accélérer les travaux du VEI, a été longuement discuté au CE 1 en présence de M. Michoudet, président du Comité National français et membre du Comité d'Action. La France, qui détient le Secrétariat du CE 1, fera un effort encore accru pour l'énorme entreprise en plein développement qu'est la révision continue du VEI. Le Comité national français se rend compte que la tâche dépasse probablement le travail d'un seul Secrétaire et envisage de trouver le moyen de lui mettre de l'aide à disposition. Le Comité d'Action, de son côté, envisage la possibilité de donner au Bureau Central de la CEI les moyens d'absorber sans retard les documents qui lui seront transmis.

Grâce à une excellente préparation des séances, tant par le Président, M. Radulet, que par le Secrétaire, M. Nasse, un avan-

[1]) siehe auch Bull. SEV 62(1971)18, S. 903...907.