

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

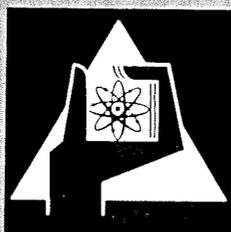
November 1969

KFK 1142
EUR 3695 d

Institut für Reaktorentwicklung
Projekt Schneller Brüter

DAS-2 - Ein dynamischer Simulator mit TOTZEIT-Gliedern
für Digitalrechner

C. Koepp



GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

November 1969

KFK 1142

EUR 3695 d

Institut für Reaktorentwicklung
Projekt Schneller Brüter

DAS-2 - Ein dynamischer Simulator mit TOTZEIT-Gliedern
für Digitalrechner

von

C. Koepp ^{*})

Gesellschaft für Kernforschung m.b.H., Karlsruhe

^{*}) delegiert von EURATOM, Brüssel

Zusammenfassung

Das DAS - 2 - Programm ist ein Digitalrechnerprogramm zur Berechnung von dynamischen Problemen, deren Bestimmungsgleichungen (algebraische Gleichungen, gewöhnliche Differentialgleichungen und Totzeitbeziehungen) in einem Rechenblock-Schalbild darstellbar sind. Das Programm enthält 22 verschiedene Rechenblockarten, deren Rechenfunktion erläutert wird. Der Totzeitrechenblock wird detailliert beschrieben und sein Flußdiagramm wird angegeben. Die Ausgabe der Ergebnisse kann sowohl als gedruckte Ausgabe, wie als Zeichnungskurve erfolgen.

Abstract

The DAS 2 - code is a digital computer programme for evaluation of dynamic problems, defined by a system of algebraic and differential equations and deadtime relations. The problems are to be represented in a block diagram. The code allows for 22 different block functions. The deadtime block is described in detail. The flowdiagram of its implementation is given. A choice of printout or plot output of the results is provided.

I n h a l t

1. Einleitung
2. Rechenblöcke
3. Integrationsverfahren
4. Totzeitrechenblöcke
5. Zeichnungsausgabe
6. Beispiel
7. Literaturverzeichnis
8. Anhang: Programmbeschreibung für DAS 2

1. Einleitung

Das DAS 2-Programm ist ein Digitalrechnerprogramm zum Berechnen von dynamischen Problemen, die durch ein System von algebraischen Gleichungen, von Gleichungen mit Verzögerungs- und Totzeitgliedern und von gewöhnlichen Differentialgleichungen mit Anfangsbedingungen erfaßt werden können. Mit dem DAS 2-Programm können alle Berechnungen (nicht jedoch Aufgaben, da z.B. eine Eingabe von Daten die vom Meßwertgeber kommen, nicht vorgesehen ist), die mit dem Analogrechner durchgeführt werden können, gelöst werden. Darüber hinaus können Probleme, die Funktionen enthalten, die nicht mit dem Analogrechner (z.B. Abruf der Dampftabelle, Lösung von komplizierten algebraischen Gleichungen) aber mit dem Digitalrechner gelöst werden können, bearbeitet werden (hierfür ist der Rechenblock FUNC vorgesehen). Eine kurze Aufzählung der hauptsächlichlichen Vor- und Nachteile von Digital- und Analogrechner ergibt:

Digitalrechner: Vorteile: - einfache Aufbewahrung der Programmdateien (Datenkarten) eines bestimmten Problems für eine spätere zusätzliche Berechnung (z.B. mit geänderten Konstanten)

- hohe Genauigkeit
- keine Dimensionierungsprobleme
- Variablen dürfen sich über viele Dekaden ändern
- viele mathematische Funktionen sind einfach zu berechnen

Nachteile: - niedrige Rechengeschwindigkeit bei Integrationen

- lange Umlaufzeiten und damit langes Warten auf die Ergebnisse

Analogrechner: Vorteile: - hohe Rechengeschwindigkeit

- Eingreifen in den Rechenablauf möglich
- Einsatz von externen technischen Elementen (z.B. Regler, Meßwertgeber) möglich
- sofortige Angabe der Ergebnisse

Nachteile: - geringe Genauigkeit

- Darstellung nichtlinearer Funktionen schwierig und ungenau
- Ausfallrate einzelner Rechenglieder höher
- hoher Aufwand, wenn sich Variablen über mehr als 4 Dekaden ändern
- Lösung einer höheren Anzahl (insbesondere bei komplizierten Gleichungen) von algebraischen Gleichungen schwierig (Instabilität)

Das DAS 2-Programm ist ein blockorientierter Simulator, das heißt: das Rechenmodell der zu lösenden Aufgabe muß in einem Schaltbild (Blockdiagramm) von Rechenblöcken vorliegen. Jeder Rechenblock stellt eine oder mehrere Rechenfunktionen dar. Die einzelnen Rechenblöcke werden in Abschnitt 2 erläutert.

Mit Hilfe des Blockdiagramms werden die Datenkarten angefertigt, die die Verknüpfung der Rechenblöcke untereinander angeben. Außerdem enthalten die Datenkarten die Titelzeile, die Rechenzeit (Problemzeit), die Schrittweite, die Rechengenauigkeit, den Zeitintervall für die geschriebene Ausgabe, die Daten, die ausgeschrieben bzw. gezeichnet werden sollen und die Kontrollparameter. Das Programm führt bei jedem Rechenschritt die durch die Rechenblöcke angegebenen Rechenoperationen in der Reihenfolge aus, in der sie in der Eingabe vorliegen.

Das DAS 2-Programm basiert auf den in [1] beschriebenen Algorithmus. Es ist eine Weiterentwicklung des DAS-Programmes, das von den Herren Janßen (IRE) und A. Peé (IRE) aus [1] entwickelt wurde. Es wurden z.B. einige Rechenblöcke in ihrer Rechenfunktion erweitert, die maximale

Anzahl der Konstanten wurde erhöht, der Totzeitblock und die Zeichnungsausgabe ist hinzugekommen, es wurden weitere Kontrollparameter eingebaut.

In dieser Beschreibung wurde angenommen, daß die unabhängige Veränderliche (im folgenden mit t bezeichnet) die Zeit darstellt. Selbstverständlich kann je nach Art des Problems eine andere physikalische oder mathematische Größe als unabhängige Veränderliche auftreten.

Die Integration wird im DAS2-Programm nach dem Kutta-Merson-Verfahren durchgeführt, das in Abschnitt 3 kurz beschrieben wird. Das Programm hat eine Rechenschrittautomatik, von der die Integrationsschrittweite automatisch dem gestellten Problem angepaßt wird. Die Schrittweite hängt von der geforderten Rechengenauigkeit und von der kleinsten Zeitkonstante in der gestellten Aufgabe und bei großer Schrittweite vom Zeitintervall der geschriebenen Ausgabe (mindestens ein Rechenschritt pro Zeitintervall) ab. Hier soll sogleich darauf hingewiesen werden, daß es aus praktischen Gründen nicht möglich ist, mit sehr kleinen Rechenschritten über große Problemzeiten hinweg zu rechnen, da sonst selbst bei den modernsten Großrechnern unerträglich lange Maschinenzeiten auftreten.

Die benötigte Maschinenzeit hängt natürlich hauptsächlich von der Anzahl der Integrierblöcke (oder Blöcke, die eine Integration beinhalten, so die TC-, TTC- und die DTC-Blöcke) und insbesondere der Totzeitblöcke ab (hier wird viel Zeit zum Umspeichern vom internen Speicher in den externen Speicher und umgekehrt verbraucht). In den meisten Fällen dürfte auch bei den heute schnellsten Rechenmaschinen die Grenze bei 10^4 bis 10^5 Integrationsschritten liegen. Die Zeichnungsausgabe und der Totzeitblock sind für 130.000 Rechenschritte ausgelegt.

Mit dem DAS2 - Programm können Probleme, die durch ein Blockdiagramm mit bis zu 200 Rechenblöcke dargestellt werden, berechnet werden. Das Blockdiagramm kann von jedem Blocktyp bis zu 100 Stück enthalten. Ausgenommen sind der Zeitblock, die Konstanten- und die Totzeitblöcke. Im Blockdiagramm können bis zu 20 Totzeit-, 198 Konstanten- und nur 1 Zeitblock vorhanden sein. Beim Integrationsblock und bei seinen verwandten Blöcken (TC, TTC- und DTC-Blöcken) besteht eine weitere Begrenzung: von diesen 4 Rechenblockarten dürfen zusammen nur höchstens 100 Blöcke im Block-

diagramm vorhanden sein.

Das DAS2 - Programm benötigt 225 K internen Speicherplatz auf der IBM-Rechenmaschine 360/65. Da es für das Integrieren viel Rechenmaschinenzzeit braucht, kann man mit dem DAS2 - Programm nur bei großen Digitalrechnern der dritten Generation (z.B. die IBM-Rechner 360/65 und 360/91) rechnen. Das heißt, daß das Rechnen mit dem DAS2 - Programm zur Zeit noch wesentlich teurer ist als mit dem Analogrechner. Bei einigen Aufgaben läßt sich der Einsatz des Digitalrechners mit dem DAS2 - Programm zur Zeit schon rechtfertigen, so z.B. für die Berechnung von dynamischen Untersuchungen eines gesamten Kernkraftwerkes, für dessen Berechnung es nur wenige Analog- und Hybridrechner mit ausreichender Kapazität in der Welt gibt, oder bei bewährten Schaltungen, wo nur die Parameter geändert werden. Durch den blockorientierten Aufbau ist das DAS2 - Programm leicht ausbaubar. Das erfordert aber jeweils zusätzlichen internen Speicherplatz. Es können noch weitere Rechenblocktypen (z.B. cosh, arctan...), die Rechenfunktionen darstellen, die mit dem Digitalrechner ohne zu viel Aufwand berechnet werden können, in das Programm aufgenommen werden.

2. Rechenblocktypen

Das DAS2 - Programm hat 22 Rechenblocktypen. Die Kennzeichnung der Rechenblöcke erfolgt über einen symbolischen Namen. Die Eingänge der Rechenblöcke können jeweils mit dem Ausgang eines anderen Blockes belegt werden. Nur beim Integrationsblock ist eine Belegung eines Einganges mit seinem eigenen Ausgang möglich. Zwei Blocktypen, der Zeit- und der Konstantenblock haben nur einen Ausgang und keine Eingänge.

Nachfolgend werden die Ein- und Ausgänge der Blöcke mit U_i bezeichnet. In den folgenden Gleichungen soll U_i den Wert der Variable bzw. Konstante und U_i' den Wert der Ableitung der Variablen nach t , der an dem Ein- bzw. Ausgang U_i liegt, darstellen.

Ein Rechenblock hat einen Ausgang und 5 Eingänge. Aber nicht bei jedem Rechenblock sind alle Eingänge belegbar. Aus der nachfolgenden Tabelle

der Rechenblöcke geht hervor, welche Eingänge belegt werden können.

U_1 ist der Ausgang des Blockes

U_2, U_3, \dots, U_6 sind die Eingänge des Blockes, die, wenn sie belegt sind, jeweils mit dem Ausgang eines anderen Blockes (beim INT-Block auch Verbindung mit eigenem Ausgang möglich) verbunden sind, d.h.

der Wert am Ausgang des anderen Blockes liegt auf dem entsprechenden Eingang. Die Belegung der Eingänge ist nicht beliebig, sondern durch die in der folgenden Tabelle angegebenen Erläuterungen festgelegt.

t ist die unabhängige Variable (in den meisten Fällen die Zeit).

Blocktyp	Name	Funktion
Integration	INT	$U_1 = \int_0^t (U_2 \times U_3 + U_4 \times U_5) dt + U_6$ <p>U_6 muß eine Konstante sein, bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen U_2, U_4 und U_6 und der Wert 1 den nichtbelegten Eingängen U_1 und U_5 zugewiesen.</p>
Verzögerungsglied 1. Ordnung	TC	$U_1 = \int_0^t \frac{U_5}{U_4} (U_2 \times U_3 - U_1) dt + U_6$ <p>U_3 und U_6 müssen Konstanten sein und $\frac{U_5}{U_4}$ muß > 0 sein</p> <p>bei Nichtbelegung der Eingänge \longrightarrow wie beim INT-Block</p>
Diff.-Glied mit Verzögerung 1. Ordnung	TTC	$U_1 = \int_0^t (U_2 \times U_3 - U_1) \frac{U_5}{U_4} dt + U_6$ <p>U_3 und U_6 müssen Konstanten sein und $\frac{U_5}{U_4}$ muß > 0 sein</p> <p>bei Nichtbelegung der Eingänge \longrightarrow wie beim INT-Block</p>
Allg. Übertragungsfunktion 1. Ordnung	DTC	$U_1 = \int_0^t \frac{1}{U_5} (U_2 \times U_3 + U_2 \times U_3 \times U_4 - U_1) dt + U_6$ <p>U_3 und U_6 müssen Konstanten sein und U_5 muß > 0 sein</p> <p>bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zugewiesen</p>

Blocktyp	Name	Funktion
Multiplikation	M	$U_1 = (U_2 + U_3 - U_4) \times (U_5 + U_6)$ bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zugewiesen
Division	D	$U_1 = (U_2 + U_3 - U_4) / (U_5 + U_6)$ bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zugewiesen
Addition	SUMA	$U_1 = U_2 + U_3 + U_4 - U_5 - U_6$ bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zugewiesen
	SUMM	$U_1 = U_2 \times U_3 + U_4 \times U_5 + U_6$ bei Nichtbelegung der Eingänge \longrightarrow wie beim INT-Block
Logarithmus	LOG	$U_1 = \ln \sqrt{(U_2 + U_3 - U_4) \times (U_5 + U_6)}$ bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen U_2, U_3, U_4 und U_6 und der Wert 1 dem nichtbelegten Eingang U_5 zugewiesen
Cosinus	C	$U_1 = \cos \left(\frac{U_2}{U_3} + U_4 \times U_5 \right)$ bei Nichtbelegung der Eingänge \longrightarrow wie beim INT-Block
Sinus	S	$U_1 = \sin \left(\frac{U_2}{U_3} + U_4 \times U_5 \right)$ bei Nichtbelegung der Eingänge \longrightarrow wie beim INT-Block
Quadrat	SQ	$U_1 = \left(\frac{U_2}{U_3} + U_4 \times U_5 \right) \times \left(\frac{U_2}{U_3} + U_4 \times U_5 \right)$ bei Nichtbelegung der Eingänge \longrightarrow wie beim INT-Block

Blocktyp	Name	Funktion
e-Funktion	EXPN	$U_1 = e \left(\frac{U_2}{U_3} + U_4 \times U_5 \right)$ <p>bei Nichtbelegung der Eingänge —————> wie beim INT-Block</p>
Funktion	FUNC	$U_1 = \text{SUB}(U_2, U_3, U_4, U_5, U_6)$ $U_2 = \text{Steuerindex}$ <p>mit diesem Block können FORTRAN - Unterprogramme abgerufen werden</p>
Konstanten	K	$U_1 = \text{Wert, der beim Lesen der}$ $\text{Datenkarten zugeordnet wurde}$
Potenzieren	PWR	$U_1 = \left(\frac{U_2 + U_4 \times U_5}{U_3} \right)^{U_6}$ <p>bei Nichtbelegung der Eingänge —————> wie beim INT-Block</p>
Absolutwertbildung	ABS	$U_1 = \left U_2 + U_3 + U_4 - U_5 - U_6 \right $ <p>bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zuge- wiesen</p>
Verzweigung	IF	$U_1 = U_2 \quad \text{wenn } U_4 < U_5$ $U_1 = U_3 \quad \text{wenn } U_4 \geq U_5$ <p>bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zuge- wiesen</p>
Begrenzung	LIM	$U_1 = U_4 \quad \text{wenn } (U_2 + U_3) > U_4$ $U_1 = U_5 \quad \text{wenn } (U_2 + U_3) < U_5$ $\text{sonst } U_1 = U_2 + U_3$ <p>bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen zuge- wiesen</p>

Blocktyp	Name	Funktion
Zeit	ZEIT	Dieser Block hat den Ausgang $U_1 = t$
Totzeit	TOT	$U_1(t) = U_3 \times U_2 \left(t - \frac{U_4}{U_5} \times U_6 \right)$ bei Nichtbelegung wird der Wert 0 den nichtbelegten Eingängen U_2 und U_4 und der Wert 1 den nichtbelegten Eingängen U_3, U_5 und U_6 zugewiesen.

3. Integrationsverfahren

Bei den INT- und den TC-, TTC- und DTC-Rechenblöcken, die implizit eine Integration enthalten, wird die Integration nach dem Kutta-Merson-Verfahren, einem Verfahren 5. Ordnung, durchgeführt. Mit diesem Integrationsverfahren können gewöhnliche Differentialgleichungen erster Ordnung mit Anfangsbedingungen gelöst werden. Die Integration erfolgt schrittweise, wobei die Schrittweite sich automatisch nach der geforderten Rechengenauigkeit und der kleinsten Zeitkonstante des Problems einstellt.

Für jede Integrationsvariable wird die Integration nach folgender Gleichung durchgeführt:

$$y_{tn+1} = y_{tn} + K$$

wobei

$$y_{tn} = \int_0^{tn} \frac{d_y(t)}{dt} dt$$

$$y_{tn+1} = \int_0^{tn+1} \frac{d_y(t)}{dt} dt$$

$$\frac{dy(t)}{dt} = f(t, y_1^i, y_2^i, \dots, y_n^i)$$

$$h = t_{n+1} - t_n = \text{Schrittweite}$$

$$K = f(t, h, y)$$

K (= Produkt von Schrittweite x Ableitung von y nach t an der Stelle t_{n+1}) wird für jede Integrationsvariable in 5 Rechenschritten nach folgendem Lösungsschema berechnet:

$$\begin{array}{ll} K_1 = f(t, y_{t_n}) & y_1 = y_{t_n} + \frac{h}{3} K_1 \\ K_2 = f(t + \frac{h}{3}, y_1) & y_2 = y_{t_n} + \frac{h}{6} (K_1 + K_2) \\ K_3 = f(t + \frac{2h}{3}, y_2) & y_3 = y_{t_n} + \frac{h}{8} (K_1 + 3K_3) \\ K_4 = f(t + \frac{h}{2}, y_3) & y_4 = y_{t_n} + \frac{h}{2} (K_1 - 3K_3 + 4K_4) \\ K_5 = f(t+h, y_4) & y_5 = y_{t_n} + \frac{h}{6} (K_1 + 4K_4 + K_5) \end{array}$$

Diese Rechenoperationen werden bei jedem Integrationsrechenschritt durchgeführt. Ein Integrationsrechenschritt ist erfolgreich, wenn der gerechnete Wert aller INT-, TC-, TTC- und DTC-Blöcke in den geforderten Genauigkeitsgrenzen liegt. Dann wird:

$$y_{t_{n+1}} = y_5$$

$$t_{n+1} = t_n + h \text{ gesetzt}$$

Als Vergleichswert für die Rechengenauigkeit wird folgende Größe herangezogen:

$$\begin{array}{ll} \mathcal{J}_1 = 0.2 \times |y_5 - y_4| & \text{bei absoluter Genauigkeit} \\ \text{bzw. } \mathcal{J}_2 = \frac{1}{|y_4|} & \text{bei relativer Genauigkeit} \end{array}$$

Ist bei allen Integrationsvariablen \mathcal{J}_1 bzw. \mathcal{J}_2 kleiner oder gleich der geforderten Genauigkeit, so ist der Integrationsrechenschritt erfolgreich. Im anderen Fall wird die Schrittweite halbiert und der Rechenschritt mit dieser Schrittweite wiederholt. Erfüllt \mathcal{J}_1 bzw. \mathcal{J}_2 dann noch nicht die Genauigkeitsforderungen, so wird die Schrittweite weiter halbiert, bis die Genauigkeitsforderungen eingehalten werden. Ist bei allen Integrations-

variablen δ_1 bzw. δ_2 $1/40$ der geforderten Genauigkeit, so wird beim nächsten Integrations-schritt mit der doppelten Schrittweite gerechnet. Auf diese Weise wird die Integrations-rechenschrittweite dem gestellten Problem angepaßt.

Beim ersten Rechenschritt des Problems wird mit der in den Datenkarten vorgegebenen Rechenschrittweite gerechnet.

4. Totzeitrechenblock

Der Totzeitrechenblock führt folgende Rechenoperation durch:

$$U_1(t) = U_3 \times U_2 (t - \bar{t})$$

wobei U_2 = Wert der Variable , die am Eingang 2 liegt

U_3 = Wert der Konstante, die am Eingang 3 liegt

$\bar{t} = U_4 \times \frac{U_6}{U_5}$ = momentane Totzeit

funktioniert nur für $\bar{t}_{n+1} - \bar{t}_n \leq h_n$

\bar{t} wird aus dem Produkt $U_4 \times \frac{U_6}{U_5}$ gebildet, um die Möglichkeit zu haben, Zeitverzögerungen die als Produkt von einer Konstante mit einer Zeitfunktion oder aus dem Verhältnis einer Konstante durch eine Funktion der Zeit (dieser Fall tritt z.B.auf, wenn die Transportzeit in Rohrleitungen sich in Funktion des Durchflusses ändert) vorliegen, ohne zusätzlichen Rechenblock aufschalten zu können. Der Eingang 4 vom Totzeitblock muß immer belegt werden, da er sonst vom Programm gleich Null gesetzt wird.

Der Aufbau des Totzeitblockes erfolgt nach folgendem Schema:

- der Wert $U_3 \times U_2(t)$ wird bei jedem erfolgreichen Rechenschritt in einen Teil des inneren Speichers, hier Schreibspeicher genannt, abgespeichert. Da der Schreibspeicher nur eine begrenzte Kapazität hat, erfolgt eine Umspeicherung der Werte in einen anderen Teil des inneren Speichers, hier Lesespeicher genannt, oder in einen externen Speicher, wenn der Schreibspeicher voll besetzt ist. Ob die Umspeicherung zum Lesespeicher oder zum externen Speicher erfolgt, hängt davon ab, ob \bar{t}_{max} (größte momentane Totzeit) kleiner oder größer als $t_{sp} = t_{End} - t_{Anf}$ (t_{Anf} = erster Wert und t_{End} =

letzter Wert der unabhängigen Variable im vollbesetzten Schreibspeicher) ist. Nach der Umspeicherung wird der Schreibspeicher von neuem belegt.

- Der Wert $U_1(t)$ wird bei jedem Rechenschritt der Integrationsberechnung aus dem Schreib- oder Lesespeicher abgelesen. Ist $(t - \bar{t})$ kleiner als der erste Wert von t , der im Schreibspeicher enthalten ist, so befindet sich der gesuchte Wert im Lesespeicher oder im externen Speicher. Wenn sich $U_1(t)$ im externen Speicher befindet, wird er vor dem Ablesen in den Lesespeicher umgespeichert.

Beim DAS 2 - Programm wird die Berechnung des Ausgangswertes $U_1(t)$ des Totzeitrechenblockes in einem Unterprogramm durchgeführt, das

- bei Beginn des Rechnens ($t=0$) die Anfangswerte der Rechenblöcke ($U_1(t=0)$) berechnet. Zusätzlich zum Berechnen der Anfangswerte werden bei den Rechenblöcken, die implizit eine Integration beinhalten, die Ableitung von $U_1(t)$ nach t an der Stelle $t = 0$ berechnet
- beim dynamischen Rechnen ($t>0$) für jeden Schritt der Integrationsberechnung (pro Rechenschritt viermal, wenn der Rechenschritt nicht erfolgreich und fünfmal, wenn der Rechenschritt erfolgreich war) die Werte der Rechenblöcke berechnet. Bei den Rechenblöcken, die implizit eine Integration beinhalten, wird zusätzlich K_1 bzw. $K_2 \dots K_5$ (siehe Abschnitt 3) berechnet.

Die Berechnung des Totzeitrechenblockes (der die Berechnung von maximal 20 Totzeitglieder ermöglicht) benötigt etwa 100 K internen Speicherplatz.

Flußdiagramm

IB	Speicherplätze des Schreibspeichers
ITOT	laufende Indexnummer
KTOT	Nummer, die den Totzeitblock kennzeichnet
K(KTOT, I)	$U_1(t)$ -Wert des I-ten Totzeitblockes
NTOT	Anzahl der Totzeitblöcke
TOTB(2IB)	Lesespeicher, der in 2 x MTOT-Sätze unterteilt ist. In den ungeradzahligen Sätzen werden die Werte U_1 und in den geradzahligen Sätzen die dem vorhergehenden Satz entsprechenden t -Werte eingegeben

TOTF(IB) Zwischenspeicher, der beim Umspeichern vom externen Speicher zum Lesespeicher benutzt wird

TOTM(20) Speicher, in dem die Anfangswerte für die Totzeitblöcke gespeichert werden

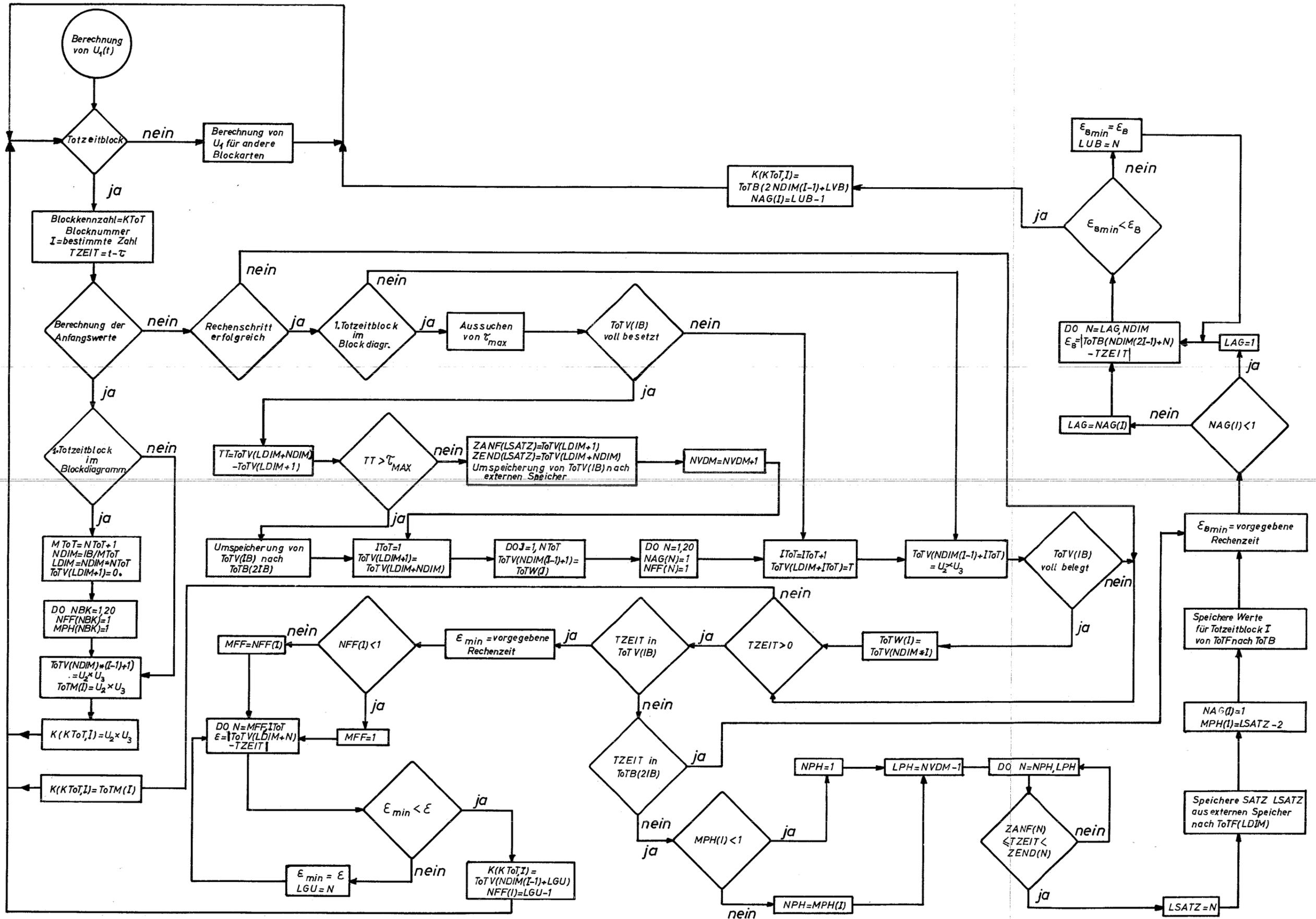
TOTV(IB) Schreibspeicher, der in MTOT Sätze unterteilt ist. In den ersten NTOT-Sätzen werden die $U_2 \times U_3$ -Werte und im MTOT-ten Satz die entsprechenden t-Werte abgespeichert

TOTW(20) Zwischenspeicher, der beim Umspeichern des letzten $U_2 \times U_3$ -wertes zum ersten $U_2 \times U_3$ -wert bei voll besetzten Schreibspeicher benutzt wird

τ_{\max} größte Totzeitkonstante

ZANF Speicher, in dem der 1. Wert von t eines jeden Satzes des externen Speichers gespeichert wird

ZEND Speicher, in dem der letzte Wert von t eines jeden Satzes des externen Speichers gespeichert wird



Flußdiagramm

5. Zeichnungsausgabe

Die Zeichnungsausgabe erfolgt durch das Unterprogramm PLOTA. Es können bis zu 20 Bilder mit insgesamt 100 Kurven gezeichnet werden, wobei ein Bild maximal 6 Kurven enthalten kann. Je nach Anzahl der Rechenschritte besteht eine Kurve aus 500 bis 1500 Punkten (vorausgesetzt, daß die Anzahl der Integrationsschritte ≥ 500 ist; die Punkteanzahl wird ausgedruckt), die, wenn die Kurve als ausgezogene Linie dargestellt wird, untereinander linear verbunden werden.

Die Abszissenachse ist die Achse der unabhängigen Variable. Da die unabhängige Variable meistens die Zeit ist, ist die Abszissenachse mit T [sec] beschriftet. Der kleinste Abszissenwert ist Null und der größte Abszissenwert ist gleich der vorgegebenen Länge der Rechenzeit in Problemsekunden. Wählbar bei der Abszissenachse ist der Abszissenwert eines Zeichnungsschrittes, der Abstand zwischen 2 Skalenstrichen und das Format für die Skalenbeschriftung.

Die Ordinatenachse ist die Achse der Veränderlichen. Wählbar bei der Ordinatenachse ist der Ordinatenwert eines Zeichnungsschrittes, der Abstand zwischen 2 Skalenstrichen, das Format der Skalenbeschriftung, die Beschriftung der Achse (maximal 20 Zeichen), der größte und der kleinste Ordinatenwert.

Weiterhin sind bei den Bildern der Typ der gewünschten Darstellung (punktweise, ausgezogene Linie, ausgezogene Linie mit Punkten), der Abstand der Punkte bei der ausgezogenen Linie mit Punkten, das Bildformat, die Ausführung mit Tinte oder Tusche und die Bildbeschriftung (maximal 80 Zeichen) wählbar.

6. Beispiel.

Im Zeitintervall $t = 0$ bis 80 Sekunden soll folgendes einfache Gleichungssystem berechnet werden:

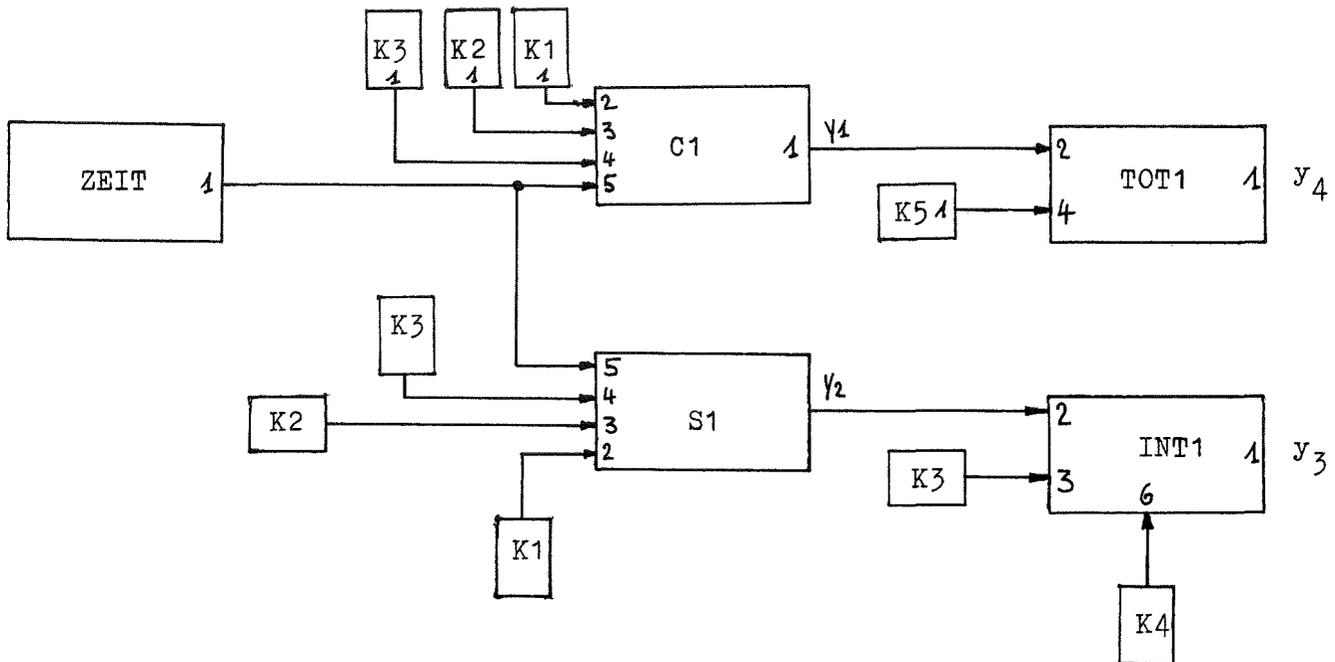
$$y_1 = \cos \left(\frac{4}{2} + 0.5 t \right)$$

$$y_2 = \sin \left(\frac{4}{2} + 0.5 t \right)$$

$$y_3 = 0.5 \int_0^t \cos \left(\frac{4}{2} + 0.5 t \right) dt = \sin \left(\frac{4}{2} + 0.5 t \right)$$

$$y_4 = y_1 (t - 4 \pi)$$

BLOCKDIAGRAMM



wobei

K_1	= 4.
K_2	= 2.
K_3	= 0.5
K_4	= 0.9093
K_5	= 12.5664

Abbild 00001 zeigt die Ergebnisse für die Schrittweite, die sich aus der kleinsten Zeitkonstante des Problems und aus dem Genauigkeitsparameter ergibt. Der Zeitintervall für die gedruckte Ausgabe beträgt 1 Sekunde, die Anzahl der erfolgreichen Schritte beträgt 254, die Anzahl der Punkte, aus denen jede Kurve gebildet wird, beträgt 254 und die Rechenmaschinenzzeit beträgt 3,78 Sekunden auf der IBM 360/65 der GfK. Man sieht, daß die Übereinstimmung der Kurven für y_1 und y_4 für $t \geq 12,56$ Sekunden nicht sehr gut ist. Das liegt am Auswahlverfahren des y -Wertes des TOT-Blockes, das bei großer Schrittweite (groß im Vergleich zur Änderung von y in Funktion von t) ungenaue Werte liefert. In Abbild 00002 sind die Ergebnisse für eine wesentlich kleinere Schrittweite, die durch das Herabsetzen des Intervalls für die gedruckte Ausgabe auf 0.01 Sekunden erzielt wurde, angegeben. Die Anzahl der erfolgreichen Rechenschritte beträgt 8014, die Zeichnungspunkteanzahl beträgt 1198 und die Rechenmaschinenzzeit beträgt 1 Minute 32,20 Sekunden. In Abb.00002 ist die Übereinstimmung von y_1 und y_2 für $t > 12,56$ Sekunden sehr gut.

7. Literaturverzeichnis

[1] J.L.Dineley, C.Preece:

Kaldas, an algorithmically based digital simulation of analogue computation.

Computer Journal, Vol.9 (1966), p.181

[2] S. Heine (nicht veröffentlicht).

8. Anhang: Programmbeschreibung für DAS2

In dieser Programmbeschreibung werden die notwendigen Angaben gemacht, um ein Problem mit dem DAS2-Programm auf der IBM-Rechenmaschine 360/65 der GfK rechnen zu können. Bei anderen Digitalrechnern müssen eventuell die FILE-Nummern für die Ein- und Ausgabe und die Zwischenspeicherung geändert werden, und das Unterprogramm für die Zeichnungsausgabe im DAS2-Programm muß an das zur Verfügung stehende Unterprogramm für das Zeichnen angepaßt werden. Als Programmiersprache wurde Fortran IV für IBM Betriebssystem/360 benutzt (siehe IBM Form 79807-3).

1. Eingabe - Datenkarten

Alphanumerische Wörter sind linksbündig zu schreiben

Kartentyp (FORMAT)	Spalte	Inhalt
K1 (20A4)	1 - 80	<u>Titelzeile</u> Der Inhalt dieser Karte wird als Überschrift ausgedruckt <u>Kontrollparameter</u>
K2 (6I6)	1 - 6	LINENO = Anzahl der Blöcke im Block- schaltbild (ohne Zeitblock und K-Blöcke) (muß größer Null sein)
	7 - 12	PRINTN = Anzahl der Blöcke, deren Aus- gang gedruckt werden soll (muß größer Null sein)
	13 - 18	CONSTN = Anzahl der Konstanten(muß größer Null sein)
	19 - 24	KPLOT = Anzahl der zu zeichnenden Bilder
	25 - 30	Steuerparameter für die Genauigkeits- abfrage 1 = Prüfung auf absolute Genauigkeit 2 = Prüfung auf relative Genauigkeit

Kartentyp (FORMAT)	Spalte	Inhalt
	31 - 36	<p>NEXT = Steuerparameter für die Beendigung des Rechnens</p> <p>= 1 Berechnung wird beendet</p> <p>= 2 ein weiteres Problem wird berechnet</p> <p>= 3 dasselbe Problem wird mit geänderten Konstanten (eine oder mehrere) nochmals gerechnet</p>
K3 (4G13.6)	1 - 13	Länge der Rechenzeit in Problemsekunden
	14 - 26	Zeitintervall für die gedruckte Ergebnisausgabe
	27 - 39	Integrationsschrittweite bei Beginn des Rechnens. Wenn die Spalten 27-39 leer sind, wird die Anfangsschrittweite = RANGE gesetzt
	40 - 52	Genauigkeitsparameter
		<u>Konstanten</u> (K-Block)
K4 (6G13.6)	1 - 13 14 - 26 usw. jeweils 6 Werte pro Karte	Es werden CONSTN Werte eingelesen. Die Eingabe der Konstanten muß in der Reihenfolge, wie sie im Blockschaltbild numeriert sind, erfolgen
		<u>Blockverknüpfungen</u> (PATCH DATA)
K5 (6(A4,I3,3X))	1 - 4	Name des Blockes, dessen Ausgang berechnet wird
	5 - 7	laufende Nummer dieses Blocktypes
	11 - 14	Name des Blockes, dessen Ausgang mit dem Eingang U_2 des Blockes, der an erster Stelle der Karte steht, verbunden ist
	15 - 17	laufende Nummer dieses Blocktypes
	21 - 24	Name des Blockes, dessen Ausgang mit dem Eingang U_3 des Blockes, der an erster Stelle der Karte steht, verbunden ist
	25 - 27 usw.bis	laufende Nummer dieses Blocktypes
	51 - 54	Name des Blockes, dessen Ausgang mit dem Eingang U_6 des Blockes, der an erster Stelle der Karte steht, verbunden ist

Kartentyp
(FORMAT)

Spalte

Inhalt

55 - 57

laufende Nummer dieses Blocktyps

Es werden LINENO Karten eingelesen. Der Zeitblock und die K-Blöcke haben keine Eingänge, so daß für diese Blöcke keine Verknüpfungskarten erstellt werden. Die Rechnung wird in der Reihenfolge dieser Karten durchgeführt. Deshalb ist es unbedingt notwendig die Karten so anzuordnen, daß die Eingänge des Blockes (außer beim TC-, INT-, TTC- und DTC-Block), dessen Ausgang berechnet wird, schon durch die vorhergehenden Karten definiert sind. Die nichtbelegten Eingänge der Blöcke werden je nach Blocktyp mit 0 oder 1 belegt.

Gedruckte Ergebnisausgabe (OUT PUT LIST)

K6
(2(A4,I3,3X))

1 - 4

Name des Blockes, dessen Ausgang gedruckt werden soll

5 - 7

laufende Nummer dieses Blocktyps

11 - 14

Kennzeichnung des Blockausganges, die maximal aus 4 alphanumerischen Zeichen bestehen kann. Diese Kennzeichnung erscheint in der Ausgabe als Überschrift über dem entsprechenden Wert des Blockausganges. Die Ausgabe der gedruckten Werte erfolgt in Intervallen von Range. Zusätzlich zu den in der OUTPUT LIST angegebenen Blöcken werden STEP (Anzahl der versuchten Rechenschritte) und ERFS (Anzahl der erfolgreichen Rechenschritte) ausgedruckt.

Zeichnungsausgabe

Die Zeichnungsausgabe wird durch die Angabe der zu zeichnenden Blockausgänge (PLOT LIST) und die Festlegung der Steuergrößen (PLOT AUXILIARY DATA) bestimmt.

PLOT LIST

K7
(6(A4,I3,3X))

1 - 4

Name des Blockes, dessen Ausgang in Kurve 1 des entsprechenden Bildes gezeichnet wird

5 - 7

laufende Nummer dieses Blocktypes

usw.bis

51 - 54

Name des Blockes, dessen Ausgang in Kurve 6 des entsprechenden Bildes gezeichnet wird

Kartentyp
(FORMAT)

Spalte

Inhalt

55 - 57

laufende Nummer dieses Blockes

Diese Karte gibt an wieviel Kurven (maximal 6) pro Bild gezeichnet werden sollen. Es werden K PLOT Bilder in der Reihenfolge dieser Karte gezeichnet. Die Bilder werden laufend numeriert von 00001 bis maximal 00020. In diesen Karten muß die Belegung fortlaufend sein, d.h. daß z.B. bei 3 Kurven für ein Bild die Spalten 1-4, 5-7, 11-14, 15-17, 21-24 und 25-27 belegt sind.

PLOT AUXILARY DATA

K8

1 - 5

(3I5, 2G13.6, 2A4)

NT = Typ der gewünschten graphischen Darstellung
= 1 punktweise
= 2 als ausgezogene Linie
= 3 als ausgezogene Linie mit Punkten

6 - 10

NPA= Abstand der zu zeichnenden Punkte für NT = 3

11 - 15

NX = Indikator für Skalenbeschriftung der Abszissenachse
= 0 keine Zifferbeschriftung
= 1 jeder Skalenstrich wird beschriftet
= N jeder N-te Skalenstrich wird beschriftet

16 - 28

SX = Abszissenwert eines Zeichnungsschrittes
($\frac{1}{100}$ inch \approx $\frac{1}{4}$ mm)

29 - 41

DX = Abstand zwischen 2 Skalenstrichen der Abszissenachse

42 - 50

NFX= Format der Skalenbeschriftung der Abszissenachse

Diese Karte wird nur einmal pro Problem angefertigt, d.h. alle Bilder haben die gleiche Abszissenachse

K 9

(2I5, 4G13,6,2A4) 1 - 5

INDZ=Indikator für Bildformat(Absolutwert) und Bildausführung (ohne Vorzeichen oder +-Vorzeichen → Ausführung in Tinte, - Vorzeichen → Ausführung in Tusche)

Kartentyp
(FORMAT)

Spalte

Inhalt

= 1	$\frac{\text{maximale Ausdehnung der Ordinatenachse}}{\text{maximale Ausdehnung der Abszissenachse}}$	= 1
= 2	"	= 2
= 3	"	= 3
= 4	"	= 4
= 5	"	$\frac{1}{1,5}$

Die Ausdehnung der Ordinatenachse beträgt 10 Inch
($\approx 25,4$ cm)

6 - 10	NY	= Indikator für Skalenbeschriftung der Ordinatenachse Bedeutung wie NX
11 - 23	YMAX	= größter Wert der Ordinatenachse
24 - 36	YMIN	= kleinster Wert der Ordinatenachse
37 - 49	SY	= Ordinatenwert eines Zeichnungsschrittes ($\approx \frac{1}{4}$ mm)
50 - 62	DY	= Abstand zwischen 2 Skalenstrichen der Ordinatenachse
63 - 70	NFY	= Format der Skalenbeschriftung der Ordinatenachse
K10 (15A4)	1 - 60	NTEXT = Beschriftung des Bildes, das maximal aus 60 alphanumerischen Zeichen bestehen darf
K 11 (5A4)	1 - 20	NTXT2 = Beschriftung der Ordinatenachse, die maximal aus 20 alphanumerischen Zeichen bestehen darf

Die Karten 9, 10 und 11 werden pro Bild jeweils einmal mit den Daten, die zur entsprechenden Karte in der PLOT LIST gehören, erstellt. Sie werden dann jeweils als Paket von 3 Karten in der Reihenfolge der Karten in der PLOT LIST zusammengestellt.

Detailliertere Erläuterungen der PLOT AUXILIARY DATA sind unter den angegebenen Namen (NT,NPA...) in [2] enthalten.

Anschließende Berechnung des Problems mit geänderten Konstanten

K12 (2I5)	1 - 5	Anzahl der Konstanten die bei NEXT=3 geändert werden.
--------------	-------	---

Kartentyp (FORMAT(Spalte	Inhalt
	6 - 10	Zahl 3 (Next=3) wenn anschließend das Problem nochmals mit erneut geänderten Konstanten gerechnet wird, sonst Zahl 1
K13 (I5,G13,6)	1 - 5	Nummer der Konstante, die geändert wird
	6 - 18	neuer Wert der Konstante Diese Karte muß so oft erstellt werden als Konstanten geändert werden. Die geänderten Werte der Konstanten werden ausgedruckt.

2. Fehlernachrichten und Warnungen

Folgende Fehlernachrichten bzw. Warnungen werden ausgedruckt:

2.1 ÜBERSCHREITUNG DER BEGRENZUNGEN:

Diese Fehlernachricht wird ausgedruckt wenn:

- in der Karte 2 der angegebene Wert für

LINENO > 200

oder CONSTN > 198

oder PRINTN > 100

oder KPLOT > 20

- im PATCH DATA

- die Anzahl der Blöcke eines Typs (ausgenommen ZEIT-, TOT-, INT-, TC-, TTC-, DTC- und K-Block) größer als 100 ist

- die Anzahl der INT-, TC-, TTC- und DTC-Blöcke zusammen größer als 100 ist

- die Anzahl der TOT-Blöcke größer als 20 ist

2.2 ERROR IN ABOVE DATA

Diese Fehlernachricht wird ausgedruckt, wenn im Patch Data:

- derselbe Block zweimal an erster Stelle einer Karte steht
- in einer Karte eine falsche Bezeichnung oder eine falsche Zuordnung in den Spalten enthalten ist

- ein Block keine Nummer oder eine negative Zahl als Nummer hat

2.3 WARNUNG INTEGRATIONSSCHRITTWEITE = XXX

Diese Warnung wird ausgedruckt, wenn die Rechenschrittweite $< 10^{-8}$ wird.

2.4 ANFANGSABLEITUNG DES nnn-Blockes mmm = XXX

Diese Warnung wird unter dem Titel "INDICATION OF LARGE INITIALDERIVATIVES" ausgedruckt, wenn die Anfangsableitung (für $t=0$) eines INT-,TC-,TTC-bzw.DTC-Blockes:

- größer als der geforderte Genauigkeitsparameter ist, wenn der Ausgangswert des Blockes zwischen -1 und 1 liegt
- größer als das Produkt von Genauigkeitsparameter mal Ausgangswert des entsprechenden Blockes bei $t = 0$ ist, wenn der Ausgangswert des Blockes ≥ 1 oder ≤ -1 ist

nnn = Blocktyp

mmm = Blocknummer

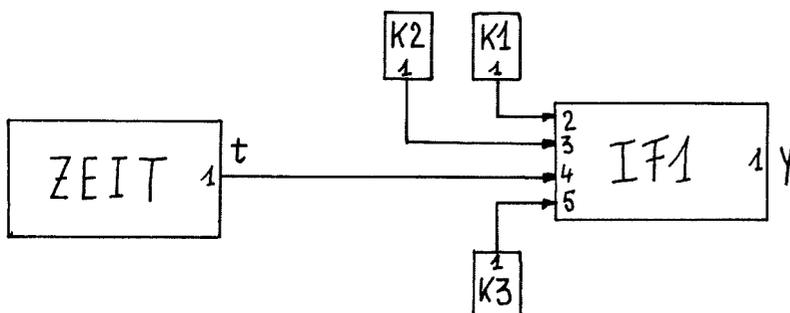
XXX = Wert der Anfangsableitung

2.5 RANGE WARNING, ERROR IN ARG XXX, usw.

Diese Fehlernachrichten und Warnungen werden für die Zeichnungsausgabe ausgedruckt. Die genauen Angaben über diese Fehlernachrichten und Warnungen sind [2] zu entnehmen.

3. Eingabe einer Sprung- oder Rampenfunktion

Veränderliche, die sich in Funktion der Zeit sprung- oder rampenförmig ändern, werden im Blockschaltbild mit Hilfe des IF- und des Zeit-Blockes realisiert. Soll z.B. der Ausgangswert einer Variablen sprungförmig von 1 auf 0.5 für $t = 10$ Sekunden geändert werden, so wird folgendes Schaltbild benutzt:



wobei K 1 = 1
 K 2 = 0.5
 K 3 = 10

4. F U N C - Block

Mit dem FUNC-Block können vorhandene digitale Unterprogramme benutzt werden. Die Unterprogramme werden über den FUNC-Block und eine FUNCTION mit dem Namen SUB, der vom Benutzer zu stellen ist, aufgerufen. Die FUNCTION SUB ist nur einmal vorhanden. Von SUB werden die gewünschten Unterprogramme mit ihrem echten Namen und ihrer erforderlichen Argumentenliste aufgerufen. Der errechnete Wert wird über den Namen SUB an den Block FUNC zurückgegeben. Über einen Steuerindex werden die einzelnen Unterprogramme den entsprechenden FUNC-Blöcken zugeordnet.

Die Eingabe Karte des FUNC-Blockes hat folgenden Aufbau:

```
FUNC n   k   i   U3   U4   U5   U6
```

wobei i der Steuerindex ist und U₃, U₄, U₅ und U₆ sind die Argumente, die an das Unterprogramm übergeben werden.

Beispiel für einen FUNC-Block

```
FUNC     10   K 3   D 4   SUMA 9   INT 2   K 23
```

Mit dem FUNC-Block Nr.10 wird das 3.Unterprogramm in der FUNCTION SUB mit den Ausgangswerten der Blöcke D 4, SUMA 9, INT 2, K 23 als Argumenten aufgerufen.

Beispiel für FUNCTION SUB

```
FUNCTION SUB (A, B, C, D, I)
GO TO (1,2,3,4,...),I
1 SUB = SINH(A*B+C/D)
RETURN

2 VOL = WUDZ (A, B,1,1)
SUB = 1.E-03/VOL
RETURN
```

```

3 SUB=TS(A)
  RETURN
4 CALL BETAN(A,B,SUB)
  RETURN
  END

```

5. Steuerkarten

5.1 Blockschaltbild ohne FUNC-Block

```

//      JOB-Karte
// EXEC FHLG,LIB=IRE,PARM.C='CDS=XXXX',
//      REGION.G=250K,TIME.G=nn,PAGES=mmm
//L.SYSIN DD *
      INCLUDE LOAD(DAS2)
/*
//G.FT07FO01 DD SYSOUT=P,SPACE=(80,XXXX)
//G.FT10FO01 DD UNIT=SYSDA,SPACE=(6124,(500)),
//  DISP=(NEW,DELETE),DCB=(BLKSIZE=6124,RECFM=VB,LRECL=404)
//G.FT11FO01 DD UNIT=SYSDA,SPACE=(7260,(100,100)),DISP=(NEW,DELETE)
//G.SYSIN DD *

```

Eingabedatenkarten

```

/*
//      JOB-Trennkarte

```

XXXX = eine vierstellige positive ganze Zahl, die kleiner als 10 000 ist. Sie gibt die maximale Anzahl der Karten an, die im GO-Step für die Herstellung der Zeichnungsausgabe notwendig sind. Sie braucht nur angegeben zu werden, wenn die Anzahl der Karten größer als 2000 ist.

nn = ein- bis dreistellige positive ganze Zahl, die die Rechenmaschinenzeit in Minuten angibt, die maximal für den GO-Step zur Verfügung steht.

mmm = positive ganze Zahl, die die maximale Anzahl der Druckausgabeseiten festlegt. Sie muß angegeben werden, wenn die Anzahl der Seiten größer als 60 ist.

5.2 Blockschaltbild mit FUNC-Blöcken

```
//JOB-Karte
// EXEC FHCLG,LIB=IRE.PARM,G='CDS=XXXX',REGION.G=250K,
// TIME,G=nn,PAGES=mmm
//C.SYSIN DD *
```

Karten für FUNCTION SUB

```
/*
//eventuell DD-Karten zum Überschreiben der vorhandenen DD-
//Anweisungen der FHCLG-Prozedur im L-Step
//L.SYSIN DD *
  INCLUDE LOAD(DAS2)
  ENTRY MAIN
/*
//G.FT07FO01 DD SYSOUT=P.SPACE=(80,XXXX)
//G.FT10FO01 DD UNIT=SYSDA,SPACE=(6124,(500)),
// DISP=(NEW,DELETE),DCB=(BLKSIZE=6124,RECFM=VB,LRECL=404)
//G.FT11FO01 DD UNIT=SYSDA,SPACE=(7260,(100,100)),DISP=(NEW,DELETE)

//G.SYSIN DD *
```

Eingabedatenkarten

```
/*
//JOB-Trennkarte
```

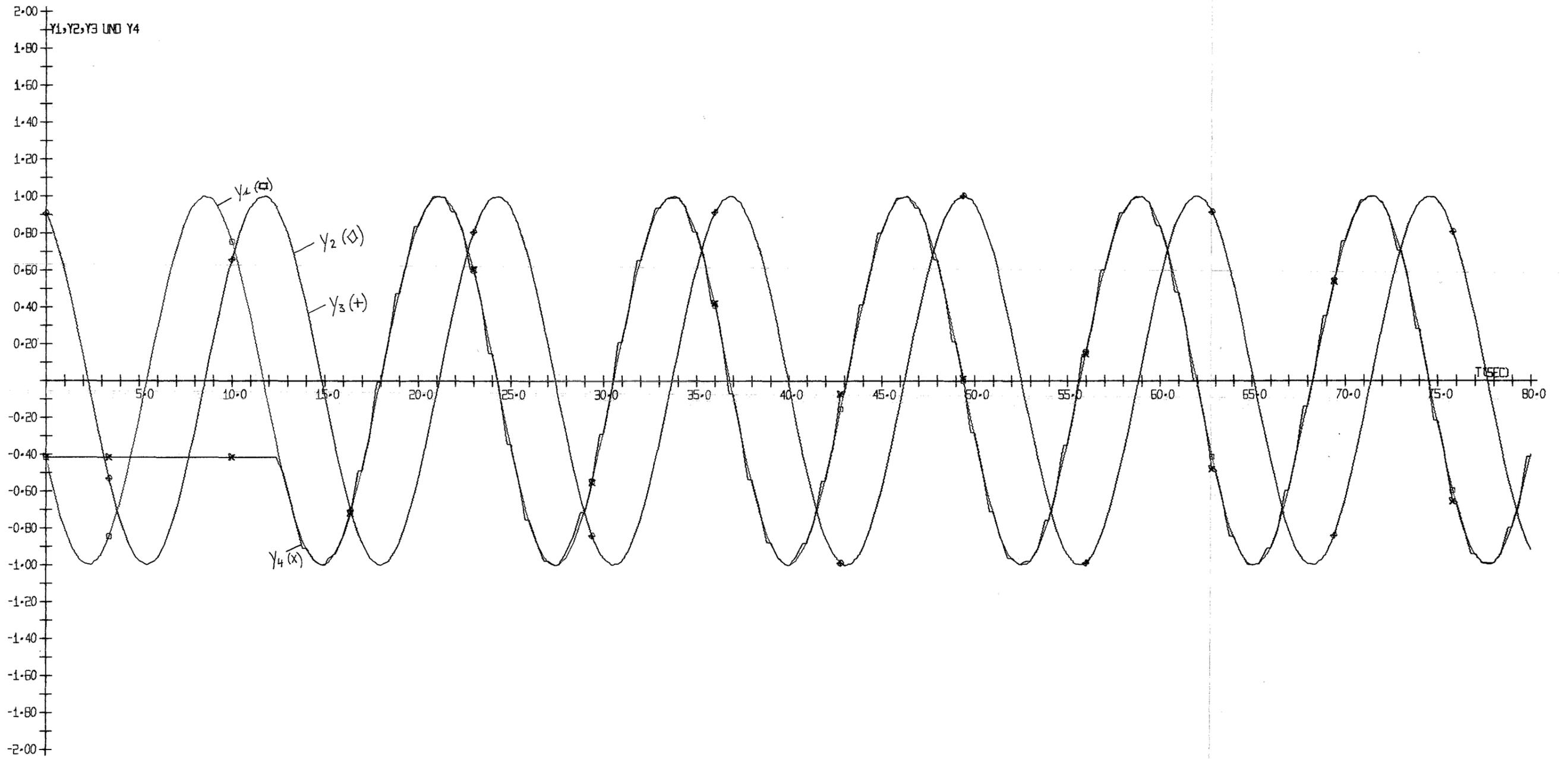


ABB-00001 Y1=CO5F, Y2=SINF, Y3=0.5INT(CO5F), Y4=Y1(T-12.56) FUER F=2*0.5T

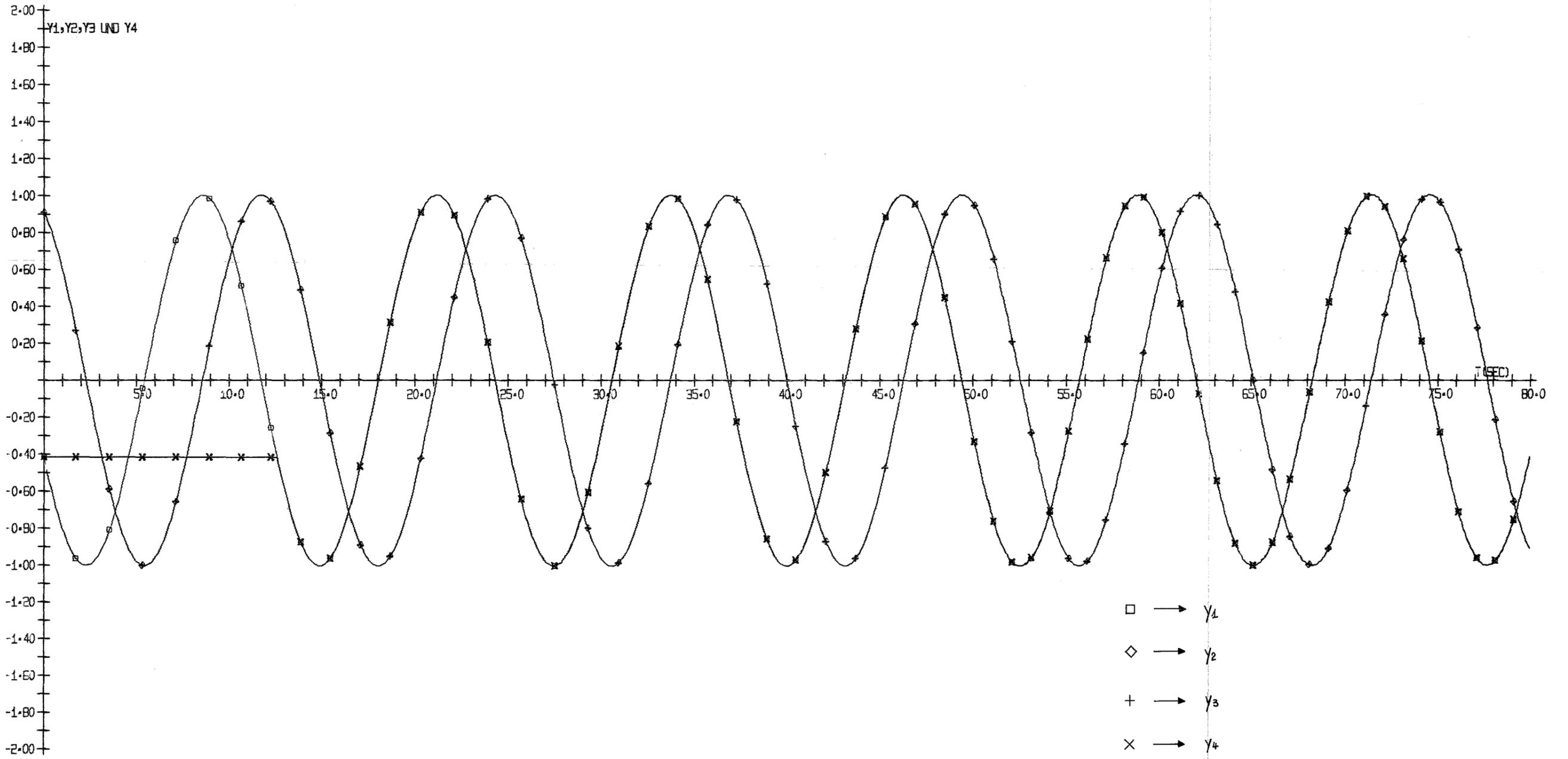


ABB-00002 Y1=CO5F, Y2=5INF, Y3=0.5INT(CO5F), Y4=Y1(T-12.56) FUER F=2+0.5T