

GERHARD BERGHOLZ

**VERHALTENSMODELLE
VON PROZESSRECHNERN**

AKADEMIE - VERLAG · BERLIN

TP33
B6

3065886

VERHALTENSMODELLE VON PROZESSRECHNERN

von

GERHARD BERGHOLZ, Dresden

Mit 132 Abbildungen und 20 Tabellen



E8065886



AKADEMIE-VERLAG · BERLIN

1980

Erschienen im Akademie-Verlag, DDR-108 Berlin, Leipziger Straße 3-4
Lektor: Dipl.-Phys. Gisela Lagowitz
© Akademie-Verlag Berlin 1980
Lizenznummer: 202 · 100/418/79
Gesamtherstellung: VEB Druckerei „Thomas Müntzer“, 582 Bad Langensalza
Bestellnummer: 762 4398 (6424) · LSV 1085
Printed in GDR
DDR 42,- M

VORWORT

Mit der immer breiteren Anwendung der Rechentechnik gewinnt die effektive Nutzung der durch sie bereitgestellten Ressourcen eine zunehmende Bedeutung. Sowohl die Anwender von Rechenanlagen als auch die Geräte- und Systemunterlagenentwickler benötigen eine komplexe Bewertung einer Rechenanlage bezüglich ihrer Effektivität. Mehr noch, sie brauchen den Zusammenhang zwischen Faktoren, die sie beeinflussen können, und Bewertungsgrößen, die eine komplexe Einschätzung der gesamten Rechenanlage gestatten. Die Herstellung eines solchen Zusammenhangs ist nicht allein auf der Grundlage des „gesunden Menschenverstandes“ möglich, sondern erfordert die Aufstellung und Nutzung von Verhaltensmodellen.

Wenn wir eine Rechenanlage in der Phase ihrer Nutzung mit dem Ziel der Effektivitätserhöhung betrachten, dann steht der Verhaltensaspekt im Vordergrund. Wir untersuchen die Rechenanlage in ihrer Einheit von Geräten, Systemprogrammen und Anwenderprogrammen. Dabei abstrahieren wir

- bezüglich der gerätespezifischen Besonderheiten
- bezüglich der programmierungsspezifischen Besonderheiten und
- bezüglich der anwendungsspezifischen Besonderheiten.

Auf dieser Grundlage entsteht etwas Einheitliches, das die allgemeinen Verhaltenseigenschaften der Rechenanlage beschreibt. Dieses Einheitliche heißt Verhaltensmodell der Rechenanlage.

Die Aufstellung von Verhaltensmodellen dient der Herausarbeitung einer Theorie der Betriebssysteme oder bei einer breiteren Betrachtung, einer Theorie der Rechner-systeme.

Dieses Buch hat die Aufgabe, durch die Ausarbeitung und Nutzung von Verhaltensmodellen für Prozeßrechner einen Beitrag zu diesem Gebiet zu leisten.

Nach der Betriebsart werden in der Rechentechnik drei Klassen von Rechenanlagen unterschieden, die Stapelverarbeitungssysteme, die Teilnehmersysteme und die Prozeß-Echtzeitrechner.

Diese Arbeit ist auf Prozeßrechner orientiert. Dadurch, daß Prozeßrechner im Gegensatz zu Stapelverarbeitungs- und Teilnehmersystemen in den verschiedenen Einsatzfällen sehr unterschiedliche gerätetechnische Konfigurationen und Echtzeitoperationssysteme besitzen, ergibt sich für die Prozeßrechner die Aufgabe, Methoden der strukturellen Modellbildung zu entwickeln. Da andererseits dieses Gebiet in der Literatur bisher vernachlässigt wurde, nimmt in diesem Buch die strukturelle Modellbildung einen relativ breiten Raum ein. Einen Schlüssel für die strukturelle Modellbildung stellt der vom Autor geschaffene Forderungslaufplan dar.

Als Untersuchungsmethoden für die Effektivitätserhöhung von Rechnersystemen verwendet man

- die Messung der Bewertungsgrößen und bestimmter Einflußfaktoren am genutzten Rechnersystem; man spricht hier auch von Systemmessungen

- die Messung der Bewertungsgrößen am simulierten Rechnersystem (Simulation) und
- die Berechnung der Bewertungsgrößen mit Hilfe der analytischen Theorie.

Im Vordergrund der Arbeit stehen analytische Untersuchungen unter sinnvoller Einbeziehung der Simulation und von Systemmessungen. Dabei kann auf den mathematischen Apparat der Wahrscheinlichkeitstheorie und mathematischen Statistik, insbesondere der Bedienungstheorie, der Theorie stochastischer Netze und der Simulationstechnik aufgebaut werden. Über Ähnlichkeitsbetrachtungen wird im Buch die Verbindung zur Theorie der Signalübertragungssysteme hergestellt.

Die Anregung zu dieser Arbeit erhielt der Autor aus seiner langjährigen Tätigkeit auf dem Gebiet des Prozeßrechnereinsatzes. Die Ergebnisse dieser Arbeit entstammen der eigenen Forschungstätigkeit. Einen Einfluß auf die Gestaltung des Buches hatte die seit 1973 vom Autor gehaltene Vorlesung „Echtzeitoperationssysteme“ vor Studenten des letzten Studienjahres für Informationsverarbeitung.

GERHARD BERGHOLZ

INHALTSVERZEICHNIS

A. Prozeßrechner	1
1. Zur Anwendung von Prozeßrechnern	1
2. Echtzeitaufträge	2
2.1. Begriffe	2
2.2. Ankunftszeitabstände	4
2.3. Auftragsverweilzeit und Reaktionszeit	4
2.4. Echtzeitbetrieb	8
3. Aufbau einer Prozeßrechenanlage	11
3.1. Multiprogrammbetrieb des Prozessors	12
3.2. Parallelbetrieb der Informationsaustauschkanäle und des Prozessors	18
3.2.1. Gerätetechnischer Aufbau der verschiedenartigen Informationsaustauschkanäle	
3.2.2. Programmtechnische Ergänzung der Informationsaustauschkanäle	
3.3. Multiprogrammbetrieb mit externspeicherresidenten Applikationsprogrammen	24
4. Parallele und quasiparallele Abfertigung der Forderungen eines Echtzeitauftrags.	26
4.1. Algorithmen und Algorithmenetz	26
4.2. Forderungsverband eines Echtzeitauftrags	29
B. Modell der Zustandsänderungen eines Prozeßrechners	32
5. Systemelemente und ihre Zustände	32
5.1. Bedienungsressourcen als Systemelemente	32
5.2. Forderungen als Systemelemente	33
6. Forderungslaufplan	34
6.1. Prozesse	34
6.2. Operationen für Zustandsänderungen	35
6.3. Forderungslaufplan von m kopplungsfreien Applikationsprogrammen.	37
6.4. Wechselwirkung zwischen den Applikationsprogrammen eines Prozessors über Forderungen	38
6.5. Synchronisation der Forderungsläufe von Applikationsprogrammen	40
6.6. Wechselwirkung eines Applikationsprogrammes mit einem parallelen Informationsaustauschkanal	42
6.7. Kopplung eines Applikationsprogrammes mit dem Informationsaustauschkanal bei Programmauslagerung	45
6.8. Zur Synchronisation der ereignisorientierten Reaktionsaufträge und der Prozeßereignisforderungen	47
7. Steuerung der Bedienungsressourcen	48
7.1. Austragungsfreie Steuerung der Applikationsprogramme	49

7.2.	Steuerung der Applikationsprogramme mit Austragung.	55
7.3.	Steuerung des parallelen Informationsaustauschs	59
C.	Stochastische Rechnermodelle.	61
8.	Stochastische Modelle des Prozeßrechners.	61
8.1.	Stochastische Systeme	61
8.2.	Verschiedene Modellansätze für die Prozeßrechenanlage.	64
8.3.	Einflußfaktoren und Bewertungsgrößen für den Echtzeitbetrieb eines Prozeßrechners	69
9.	Ankunftsprozesse der Echtzeitaufträge und der Prozeßereignisse einer Prozeßrechenanlage	72
9.1.	Prozeßrechner als offene und als geschlossene Bedienungssysteme	73
9.2.	Eigenschaften der Ankunftsströme für die Prozeßrechenanlage als offenes Bedienungssystem	75
9.3.	Eigenschaften der Quellverweilzeiten für die Prozeßrechenanlage als geschlossenes Bedienungssystem	83
10.	Eigenschaften der Bedienungsprozesse einer Prozeßrechenanlage	85
10.1.	Eigenschaften der Programmflüge als Bedienungsprozesse.	85
10.2.	Eigenschaften der Zeitverzögerungsprozesse eines Prozeßrechners	92
10.3.	Eigenschaften der Informationsaustauschprozesse eines Prozeßrechners	92
11.	Simulationsmodelle	95
11.1.	Ereignisorientierte Simulation	95
11.2.	Zum Aufbau eines Simulationsprogrammsystems für die ereignisorientierte Simulation	97
11.3.	Zur Anwendung des VOPS SIMDIS für die Untersuchung des Echtzeitbetriebs eines Prozeßrechners.	100
11.3.1.	SIMDIS-Forderungslaufplan	
11.3.2.	Statistische Bestimmung der Bewertungsgrößen	
11.3.3.	Zur Programmierung in SIMDIS	
11.4.	Zur Simulationsversuchsplanung.	107
D.	Effektivitätsanalyse einer Prozeßrechenanlage als Einkanalbedienungssystem.	110
12.	Prioritätsfreie Bedienung der Echtzeitaufträge	110
12.1.	Untersuchung eines Prozeßrechners mit Hilfe des M/M/1/N-Bedienungsmodells.	111
12.2.	Untersuchung eines Prozeßrechners mit Hilfe des D/M/1-Bedienungsmodells	113
12.3.	Untersuchung eines Prozeßrechners mit Hilfe des M/G/1-Bedienungsmodells	118
12.4.	Vergleich von drei prioritätsfreien Bedienungsmodellen.	124
13.	Bedienung der Echtzeitaufträge mit Prioritäten	124
13.1.	Untersuchung eines Prozeßrechners mit relativer Prioritätsdisziplin	125
13.2.	Untersuchung eines Prozeßrechners mit programmierter Prioritätsdisziplin	131
13.3.	Untersuchung eines Prozeßrechners mit absoluter Prioritätsdisziplin	139
13.4.	Allgemeine Eigenschaften verschiedener Bedienungsdisziplinen des Multi-programmbetriebs	141
13.5.	Optimale Prioritätszuordnung.	143
14.	Einkanalssysteme mit Mehrphasenbedienung.	143
14.1.	Einstromsystem mit Mehrphasenbedienung durch m Applikationsprogramme ohne Verzweigungen	144

14.2.	Einstromsystem mit Mehrphasenbedienung durch m Applikationsprogramme mit Verzweigungen	151
14.3.	Zur optimalen Prioritätszuordnung für Einstromsysteme mit Mehrphasenbedienung	158
14.4.	Mehrstromsysteme mit Mehrphasenbedienung	158
14.4.1.	Mehrstromsysteme der Mehrphasenbedienung mit relativer und programmierter Prioritätsdisziplin	
14.4.2.	Mehrstromsysteme der Mehrphasenbedienung mit absoluter Prioritätsdisziplin	
E.	Effektivitätsanalyse einer Prozeßrechenanlage als stochastisches Netz	162
15.	Leistungsanalyse für das stochastische Netz einer Prozeßrechenanlage.	162
15.1.	Gegenstand der Leistungsanalyse	163
15.2.	Intensitätsübertragung in einem stochastischen Netz.	168
15.3.	Anwendungsergebnisse der Leistungsanalyse	173
15.4.	Schlußfolgerungen für die Gestaltung eines Prozeßrechners	176
16.	Verweilzeitanalyse für das stochastische Netz einer Prozeßrechenanlage.	177
16.1.	Verweilzeitmodelle.	177
16.2.	Zur Bestimmung der Verweilzeitverteilung für typische Forderungslaufplanabschnitte	182
17.	Nutzung der Ähnlichkeiten zwischen Verweilzeitmodellen und analogen Signalübertragungssystemen	189
17.1.	Ähnlichkeiten zwischen der Verteilungsfunktion der Verweilzeit und der Übertragungsfunktion analoger Signalübertragungssysteme	189
17.2.	Rationale Verweilzeitblöcke.	190
17.3.	Grundtypen von Verweilzeitblöcken	192
17.4.	Approximation der Verteilungsfunktion eines Verweilzeitblockes.	193
17.5.	Darstellung von Verweilzeitsystemen durch kontinuierliche Signalübertragungssysteme	195
17.6.	Diskontinuierliche Verweilzeitmodelle	198
F.	Effektivitätsanalyse einer Prozeßrechenanlage als Bedienungsnetz	201
18.	Untersuchung des Prozeßrechners als Bedienungsnetz mit unbegrenzten Bedienungsressourcen.	201
18.1.	Annahmen und ihre Konsequenzen	201
18.2.	Zur Bestimmung der Verweilzeitverteilung bei exponentiellen Bedienungszeitverteilungen der elementaren Verweilzeitblöcke.	203
18.3.	Zur Bestimmung der Verweilzeitverteilung bei konstanten Bedienungszeiten der elementaren Verweilzeitblöcke	205
18.4.	Anwendungsbeispiel für die Meßwerterfassung und -verarbeitung	208
19.	Untersuchung des Prozeßrechners als Bedienungsnetz mit mehreren Prozessoren	218
19.1.	Modelle der Bedienungsanlagen des Bedienungsnetzes	219
19.2.	Bedienungsnetze mit mehreren Prozessoren	220
19.3.	Serienkopplung von mehreren Prozessoren	222
19.4.	Zur Optimierung eines linearen Bedienungsnetzes von M/M/1-Wartesystemen	226
19.5.	Optimale Aufteilung der mittleren Bedienungszeit auf zwei Prozessoren bei einfacher Parallelkopplung	230
20.	Untersuchung des Prozeßrechners als Bedienungsnetz mit einem Prozessor	232

Literaturverzeichnis 240

Symbolverzeichnis 245

Sachwortverzeichnis 247

A. Prozeßrechner

1. ZUR ANWENDUNG VON PROZESSRECHNERN

Anwendung eines Prozeßrechners

Prozeßrechner werden in automatisierten Systemen für die Leitung und Steuerung ASU (автоматизированные системы управления) eingesetzt. Ein ASU besteht aus zwei Teilsystemen: dem Informationssystem und dem Basissystem. Speziell bei der Anwendung eines Prozeßrechners ist dieser Kernstück des Informationssystems, und das Basissystem enthält technische oder technologische Prozesse. Wenn im Basissystem nur technologische Prozesse vorkommen, dann wird auch die Abkürzung ASUTP (автоматизированные систему управления технологическими процессами) verwendet (s. [STA]). Die wichtigsten Anwendungen von Prozeßrechnern sind in ASUTP zu finden. Außerdem gibt es noch Anwendungen von Prozeßrechnern, in deren Basissystemen technische Prozesse nichttechnologischer Form auftreten.

Ein Prozeßrechner hat die Aufgabe die Prozesse in technischen und technologischen Basissystemen zu überwachen, zu steuern bzw. Informationen von diesen Prozessen zu verdichten. Aus diesem Grunde wurde auch der Begriff Prozeßrechner eingeführt und wird heute allgemein verwendet.

Man bezeichnet den Prozeßrechner auch noch als Prozeßrechenanlage (PRA). In diesem Buch wird immer, wenn von einem Prozeßrechner (oder einer Prozeßrechenanlage) die Rede ist, vorausgesetzt, daß er sich im Echtzeitbetrieb in der Anwendung befindet.

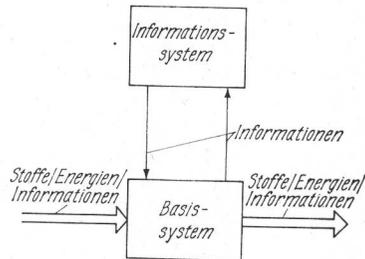


Abb. 1.1. Symbolische Darstellung der Anwendung eines Prozeßrechners

Vom Standpunkt der Steuerungstechnik werden technische und technologische Basissysteme auch Strecken und Prozeßrechner auch Einrichtungen genannt (s. [REI]). In Abb. 1.1 ist die Anwendung eines Prozeßrechners symbolisch dargestellt.

Anwendungsgebiete von Prozeßrechnern

Prozeßrechner werden heute in allen Bereichen der Volkswirtschaft und darüber hinaus angewendet. Die wichtigsten Anwendungsgebiete sind

- verfahrenstechnische Produktionsprozesse
- Prozesse der Energieerzeugung und -verteilung
- Fertigungsprozesse
- Transportprozesse
- Meß- und Prüfprozesse
- Prozesse in großen Informationsverarbeitungssystemen
- ausgewählte Prozesse im Gesundheitswesen und
- ausgewählte Lehr- und Lernprozesse im Hochschulwesen und in der Volksbildung.

Allgemein handelt es sich um stoffliche, energetische oder informationelle Prozesse.

2. ECHTZEITAUFRÄGE

2.1. Begriffe

Wir führen jetzt den Begriff Echtzeitauftrag ein (s. auch [BER 1]). Dabei verstehen wir unter einem Echtzeitauftrag die einmalige Anforderung an die Prozeßrechenanlage zur Bearbeitung von Programmen und zum Austausch der zugehörigen Informationen mit dem Basissystem. Dieser Anforderung ist die Lösung einer Prozeßrechneraufgabe zugeordnet.

Jeder Echtzeitauftrag besitzt einen Ankunftszeitpunkt t_a und einen Erledigungszeitpunkt t_v . Der Ankunftszeitpunkt t_a ist dabei der Zeitpunkt, zu dem der Echtzeitauftrag über ein Unterbrechungssignal beim Prozeßrechner ankommt. Der Erledigungszeitpunkt t_v ist der Zeitpunkt, zu dem der durch den Echtzeitauftrag bewirkte Lösungsprozeß der Prozeßrechneraufgabe abgeschlossen wird. Dabei kann dieser Lösungsprozeß zwischenzeitlich unterbrochen werden.

Auftragsströme

In einer PRA treffen während des Echtzeitbetriebs zeitlich nacheinander ständig neue Echtzeitaufträge ein.

Echtzeitauftragsstrom

Die nach den Ankunftszeitpunkten geordnete Folge von Echtzeitaufträgen

$$E_1, E_2, \dots, E_i, \dots \quad (2.1)$$

heißt Echtzeitauftragsstrom E .

Man kann eine Teilfolge aus dem Echtzeitauftragsstrom auswählen und eine solche nach den Ankunftszeitpunkten geordnete Teilfolge der Echtzeitaufträge wieder zu einem Auftragsstrom zusammenfassen.

Klassifizierung der Echtzeitaufträge

Wir klassifizieren die Echtzeitaufträge nach folgenden Gesichtspunkten: nach der Regelmäßigkeit ihrer Ankunft, nach der Auftragsquelle und nach der Auftragslenke. Nach der Regelmäßigkeit ihrer Ankunft teilen wir die Echtzeitaufträge in regelmäßige oder deterministische Echtzeitaufträge und in unregelmäßige oder stochastische Echtzeitaufträge ein.

Nach der Auftragsquelle unterscheiden wir Basisaufträge und Zeitgeberaufträge. Bei den Basisaufträgen ist unmittelbar das Basissystem die Auftragsquelle, während bei Zeitgeberaufträgen der Zeitgeber, der auch Echtzeituhr genannt wird, die Auftragsquelle ist. Die Erteilung der Zeitgeberaufträge erfolgt gewöhnlich zeitzyklisch, d. h. regelmäßig. Demgegenüber wird die Auftragserteilung bei Basisaufträgen meist unregelmäßig vorgenommen, so daß wir diese als stochastisch ansehen. Somit können wir die Klassifizierung nach der Regelmäßigkeit ihrer Ankunft und nach der Auftragsquelle zusammenfassen.

Nach der Auftragsquelle unterscheiden wir Verdichtungsaufträge und Reaktionsaufträge. Für Verdichtungsaufträge ist der Prozeßrechner selbst Auftragsquelle. Das trifft z. B. zu, wenn die Ergebnisse des durch den Echtzeitauftrag bewirkten Lösungsprozesses einer Prozeßrechneraufgabe in einem durch einen anderen Echtzeitauftrag ausgelösten Lösungsprozeß genutzt werden. Für Reaktionsaufträge ist das Basissystem die Auftragsquelle. Der Prozeßrechner reagiert, indem er ein Steuersignal an das Basissystem oder eine Information an den Prozeßbediener im Basissystem ausgibt. Der Zeitpunkt dieser Reaktion wird Reaktionszeitpunkt genannt und mit t_r bezeichnet.

Die Reaktionsaufträge können wir weiter in regelmäßige und ereignisorientierte Reaktionsaufträge unterteilen. Die regelmäßigen Reaktionsaufträge sind eine Teilfolge der regelmäßigen Echtzeitaufträge. Dabei werden regelmäßig mit jedem n-ten regelmäßigen Echtzeitauftrag Reaktionen an das Basissystem gegeben. Jeder ereignisorientierte Reaktionsauftrag ist genau einem Ereignis im Basissystem zugeordnet, auf das er reagieren muß. Dabei kann ein Strom ereignisorientierter Reaktionsaufträge entweder ein Strom stochastischer Echtzeitaufträge sein oder zu einem Strom regelmäßiger Echtzeitaufträge gehören. Im ersten Fall sind die Ereignisse selbst Quelle der Reaktionsaufträge, während im zweiten die Ereignisse erst durch die regelmäßigen Echtzeitaufträge überwacht und entdeckt werden.

In Abb. 2.1 wird die oben durchgeführte Klassifizierung der Echtzeitaufträge zusammenfassend dargestellt.

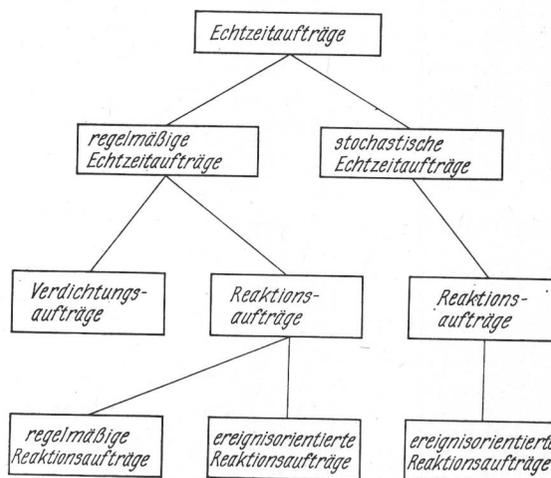


Abb. 2.1. Zur Klassifizierung der Echtzeitaufträge

Echtzeitaufgaben

Bei einer Prozeßrechenanlage bleiben — im Gegensatz zu Stapelverarbeitungssystemen und Teilnehmersystemen — die in einem Einsatzfall zu lösenden Aufgaben über lange Zeiträume (Monate, Jahre) konstant. Dabei kann man davon ausgehen, daß eine bestimmte Prozeßrechneraufgabe immer wieder durch dieselben Programme (Algorithmen) gelöst wird.

Wir nehmen an, daß es möglich ist, alle Reaktionsaufträge eines Einsatzfalles nach funktionellen Gesichtspunkten in eine endliche Zahl von Klassen aufzuteilen. Jeder Klasse ist also immer ein Strom von Reaktionsaufträgen und manchmal auch ein Strom von Verdichtungsaufträgen zugeordnet. Da die Klasseneinteilung nach funktionellen Gesichtspunkten erfolgt, entspricht jede Klasse einer Prozeßrechneraufgabe, die wir in diesem Zusammenhang Echtzeitaufgabe nennen wollen. Eine PRA löst also in einem bestimmten Einsatzfall eine Zahl von L Echtzeitaufgaben. Da jeder Echtzeitaufgabe je ein Echtzeitauftragsstrom zugeordnet ist, sprechen wir auch von einem Mehrstromsystem.

2.2. Ankunftszeitabstände

Die Ankunftszeitpunkte der Echtzeitaufträge bilden einen Ereignisstrom, den Ankunftsstrom

$$t_{a1}, t_{a2}, \dots, t_{ai}, \dots$$

Es gilt für den i -ten Echtzeitauftrag E_i

$$T_{ai} = t_{ai} - t_{ai-1}, \quad (2.2)$$

wobei T_{ai} der Ankunftsabstand der benachbarten Echtzeitaufträge ist. Bei stochastischer Auftragserteilung sind die Ankunftszeitabstände T_{ai} Zufallsgrößen, während bei regelmäßiger Auftragserteilung die Ankunftszeitabstände T_{ai} deterministische Größen sind. Im letzteren Fall haben gewöhnlich alle Zeitabstände den gleichen Wert $T_{ai} = T_a$, und wir haben es mit einem regelmäßigen Ankunftsstrom der Echtzeitaufträge zu tun. Bei zufälligen T_{ai} werden meist rekurrente Ströme betrachtet, die für alle T_{ai} die gleiche Verteilungsfunktion besitzen. Deshalb kann auch dann der Index i entfallen. Somit verzichten wir in unseren Betrachtungen gewöhnlich auf den Index i .

2.3. Auftragsverweilzeit und Reaktionszeit

Auftragsverweilzeiten

Jeder Echtzeitauftrag besitzt, unabhängig davon, ob es sich um einen Reaktions- oder Verdichtungsauftrag handelt, einen Auftrags erledigungszeitpunkt t_v . Jedem Echtzeitauftrag ist eine Auftragsverweilzeit T_v zugeordnet. Es gilt

$$T_v = t_v - t_a. \quad (2.3)$$

Die Verweilzeiten der verschiedenen Echtzeitaufträge können sich zeitlich überlappen (s. Abb. 2.2).

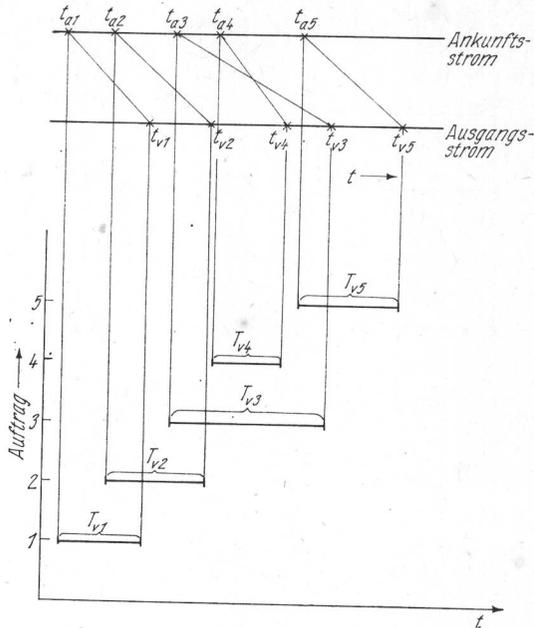


Abb. 2.2. Zeitliche Überlappung der Abfertigung der Echtzeitaufträge einer PRA

Reaktionszeiten für ereignisorientierte Reaktionsaufträge

Jedem Reaktionsauftrag kann eine Reaktionszeit zugeordnet werden. Ein Verdichtungsauftrag besitzt keine Reaktionszeit, aber eine Verweilzeit. Wir definieren jetzt die Reaktionszeit für einen ereignisorientierten Reaktionsauftrag als die Differenz

$$T_r = t_r - t_e, \quad (2.4)$$

wobei t_e der Zeitpunkt des Prozeßereignisses und t_r der Zeitpunkt der Reaktion der Prozeßrechenanlage auf das Prozeßereignis sind (s. Abb. 2.3a). Dabei wird angenommen, daß der Reaktionszeitpunkt eines Reaktionsauftrages mit dessen Erledigungszeitpunkt zusammenfällt, d. h., für einen Reaktionsauftrag gilt

$$t_v = t_r. \quad (2.5)$$

Wir betrachten zunächst die zeitzyklische Überwachung von Ereignissen im Basissystem. Es handelt sich also um ereignisorientierte Reaktionsaufträge. Da die Überwachung zeitzyklisch erfolgt, haben wir es mit einem Strom von regelmäßigen Echtzeitaufträgen zu tun, der den Strom der ereignisorientierten Reaktionsaufträge als Teilfolge enthält. Zwischen einem Ereignis t_e im Basissystem, auf das der Prozeßrechner zu reagieren hat, und dem Ankunftszeitpunkt t_a des Reaktionsauftrages, der auf dieses Ereignis reagiert, besteht kein Zusammenhang, da beide Ereignisformen völlig unabhängig voneinander auftreten. Bei der zeitzyklischen Überwachung von Prozeßereignissen entsteht zwischen dem Prozeßereigniszeitpunkt t_e und dem Ankunftszeitpunkt t_a des entsprechenden Reaktionsauftrages im allgemeinen eine

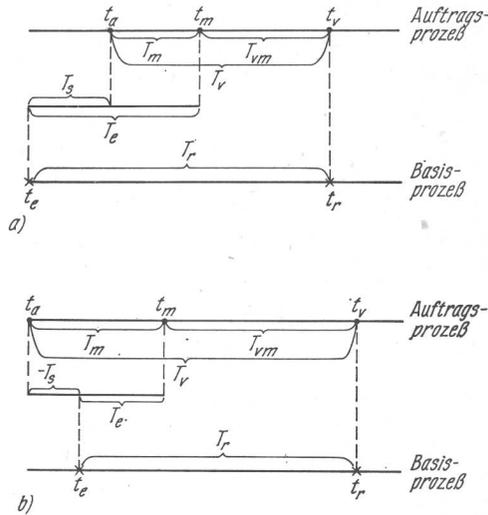


Abb. 2.3. Zusammenhang zwischen der Reaktionszeit und der Auftragsverweilzeit bei zeitzyklischer Überwachung von Prozeßereignissen

Zeitdifferenz, die wir Auftragsverschiebungszeit T_s nennen wollen, wobei gilt

$$T_s = t_a - t_e \quad (2.6)$$

(s. Abb. 2.3a).

Folglich besitzt die Reaktionszeit bei der zeitzyklischen Überwachung von Prozeßereignissen zwei Summanden (s. Abb. 2.3a)

$$T_r = T_s + T_v. \quad (2.7)$$

Ein Prozeßereignis zum Zeitpunkt t_e muß erst durch den entsprechenden Überwachungsalgorithmus in der PRA entdeckt werden. Wir bezeichnen den Entdeckungszeitpunkt des Prozeßereignisses mit t_m (s. Abb. 2.3a). Erst durch die Entdeckung des Prozeßereignisses zum Zeitpunkt t_m wird eine Synchronisierung der Arbeit der PRA mit dem Prozeß des Basissystems erreicht. Wir führen die Ereignisentdeckungszeit T_e ein, die als Differenz zwischen dem Entdeckungszeitpunkt t_m und dem Ereigniszeitpunkt t_e festgelegt wird, mit

$$T_e = t_m - t_e \quad (2.8)$$

(s. Abb. 2.3a). Dabei müssen wir davon ausgehen, daß i.a. der Entdeckungszeitpunkt t_m des Prozeßereignisses nicht mit dem Beginn des Reaktionsauftrages übereinstimmt, sondern um die Anfangsverweilzeit T_m verschoben ist. Es gilt (s. Abb. 2.3a)

$$T_m = t_m - t_a. \quad (2.9)$$

Aus (2.6), (2.8) und (2.9) folgt (vergleiche Abb. 2.3a)

$$T_e = T_s + T_m. \quad (2.10)$$

Wir nehmen an, daß der Entdeckungszeitpunkt eines Prozeßereignisses mit dem Erfassungszeitpunkt des entsprechenden Meßwertes zusammenfällt und die Meßwert-erfassungszeit dabei gleich 0 ist. Somit besteht die Möglichkeit, daß ein Prozeß-

ereignis, das von einem regelmäßigen Reaktionsauftrag entdeckt wird, erst nach dem Ankunftszeitpunkt dieses Echtzeitauftrages liegen kann, d. h., daß in diesem Fall

$$T_s < 0 \quad (2.11)$$

und

$$T_e < T_m \quad (2.12)$$

gilt (s. Abb. 2.3b).

Bei einer durch ein Prozeßereignis bedingten Auftragserteilung fällt der Auftragsankunftszeitpunkt mit dem Prozeßereigniszeitpunkt und dem Ereignisentdeckungszeitpunkt zusammen, d. h., es gilt in diesem Fall

$$t_e = t_a = t_m. \quad (2.13)$$

Folglich verschwinden die Auftragsverschiebungszeit, die Anfangsverweilzeit und die Ereignisentdeckungszeit

$$T_s = T_m = T_e = 0, \quad (2.14)$$

und die Reaktionszeit stimmt mit der Auftragsverweilzeit überein

$$T_r = T_v. \quad (2.15)$$

Reaktionszeiten für regelmäßige Reaktionsaufträge

Wie bereits angeführt, gibt es bei den Reaktionsaufträgen mit regelmäßiger Reaktion kein signifikantes Prozeßereignis, auf das die PRA zu reagieren hat. Hier führt jeder n -te Echtzeitauftrag, wobei n eine deterministische Größe ist, zu einer Reaktion an das Basissystem.

Für $n = 1$ ist jeder Echtzeitauftrag ein Reaktionsauftrag, und dem Ankunftszeitpunkt t_a des Echtzeitauftrages kann ein Reaktionszeitpunkt t_r zugeordnet werden. Die Reaktionszeit ist dann gleich der Auftragsverweilzeit entsprechend Gleichung (2.15).

Für $n > 1$ ergibt sich

$$i = (j - 1)n + l, \quad (2.16)$$

wenn mit i die laufende Nummer des Echtzeitauftrages, mit l die Nummer des Echtzeitauftrages innerhalb eines Reaktionszyklus und mit n die Nummer des Reaktionszyklus bezeichnet werden. Dabei ist

$$1 \leq l \leq n. \quad (2.17)$$

Jedesmal wenn

$$i = jn \quad (2.18)$$

und

$$l = n \quad (2.19)$$

gelten, handelt es sich um einen Reaktionsauftrag. Sind die Beziehungen (2.18) und (2.19) nicht gültig, dann haben wir es mit Verdichtungsaufträgen zu tun. Man kann auch in diesem Fall die Reaktionszeit jedes Reaktionsauftrages mit der Auftragsverweilzeit entsprechend Formel (2.15) ansetzen.

Beispiele für regelmäßige Reaktionsaufträge sind die bei einer Tastregelung (s. z. B. [BER 2]) oder einer zeitzyklischen optimierenden Steuerung (s. z. B. [KAP]) auftretenden Reaktionsaufträge.

In einer Tastregelung ist $n = 1$, und jeder von der Echtzeituhr erzeugte Echtzeitauftrag ist ein Reaktionsauftrag. Dabei wird mit jedem Echtzeitauftrag ein analoges Signal an das Basissystem gegeben.

Bei einer zeitzyklischen optimierenden Steuerung erfolgt die Optimierungsrechnung meist auf der Grundlage von zeitlichen Mittelwerten der Meßgrößen. In jedem Zyklus der Meßwerterfassung wird somit dem Prozeßrechner ein Echtzeitauftrag erteilt, und in jedem n -ten Erfassungszyklus, in dem die Optimierungsrechnung mit den Mittelwerten der Meßgrößen ausgeführt wird, ergibt sich ein Reaktionsauftrag, da im Anschluß an die Optimierungsrechnung die PRA in das Basissystem eingreift. Die deterministische Größe n ist dabei die Zahl der für jeden Mittelwert über die Zeit gesammelten Meßwerte.

2.4. Echtzeitbetrieb

Begriffsbestimmung

Unter Echtzeitbetrieb eines Prozeßrechners verstehen wir die Betriebsweise des Prozeßrechners, bei der die Lösungsprozesse der Echtzeitaufgaben schritthaltend mit den Prozessen im Basissystem ablaufen. In den nachfolgenden vereinfachten Beispielen wird gezeigt, was wir mit schritthaltendem Ablauf der Lösungsprozesse der Echtzeitaufgaben meinen.

Vereinfachte Beispiele

1. Wir betrachten die Füllstandssteuerung eines Behälters (s. Abb. 2.4a) und setzen voraus, daß Zuflußventil V geöffnet ist und über die Zuflußleitung Flüssigkeit in den Behälter strömt.

Die Aufgabe des Prozeßrechners besteht darin, das Ventil V der Zuflußleitung zu schließen, wenn der Füllstand h den Wert h_0 erreicht hat. Dazu wird der Füllstand mit Hilfe eines Füllstandskontaktes in der Höhe h_0 außerhalb des Prozeßrechners überwacht. Es sei angenommen, daß das Ereignis „Erreichen des Füllstandes h_0 “ zum Zeitpunkt t_e eintritt. Zu diesem Zeitpunkt erfolgt durch das Meßgerät eine Unterbrechungsmeldung, die dem Prozeßrechner über einen entsprechenden digitalen Eingang zugeführt wird. Während h eine analoge Größe ist, stellt die Größe x am Ausgang des Füllstandskontaktes eine digitale Größe dar, mit der Zuordnung

$$x = \begin{cases} 0 & \text{für } h < h_0, \\ L & \text{für } h \geq h_0. \end{cases} \quad (2.20)$$

In Abb. 2.4b ist der mögliche zeitliche Verlauf von h und x dargestellt. Hier ist erkennbar, daß zum Zeitpunkt t_e die Größe x von 0 in L sprunghaft übergeht. Um eine rechtzeitige Reaktion der Prozeßrechenanlage auf das Ereignis „Erreichen des Füllstandes h_0 “ zu erhalten, wird die Bearbeitung des gerade laufenden Programmes unterbrochen. Nach dieser Unterbrechung wird die Unterbrechungsmeldung übernommen und festgestellt, welches Maßnahmeprogramm zu aktivieren ist. In unserem Fall ist das ein Programm zum Schließen des Ventils V. Dieses Maßnahmeprogramm wird sofort bearbeitet, wenn es wichtiger als das unterbrochene Programm ist. Trifft das nicht zu, dann wird das Maßnahmeprogramm zur Abarbeitung vorgemerkt und das unterbrochene Programm fortgesetzt. Das Maßnahmeprogramm wird ent-