

Echtzeitsysteme

Einleitung

19. Oktober 2009

Überblick

Einleitung

Echtzeitbetrieb

Fallbeispiel Wärmetauscher

Zusammenfassung

Bibliographie

DIN 44300

Ereignis- oder zeitgesteuerte Programmverarbeitung

- ▶ *Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind.*
- ▶ *Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorbestimmten Zeitpunkten anfallen.*

Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. *real-time processing*)

Zustandsänderung von Programmen wird zur Funktion der **realen Zeit** [2]

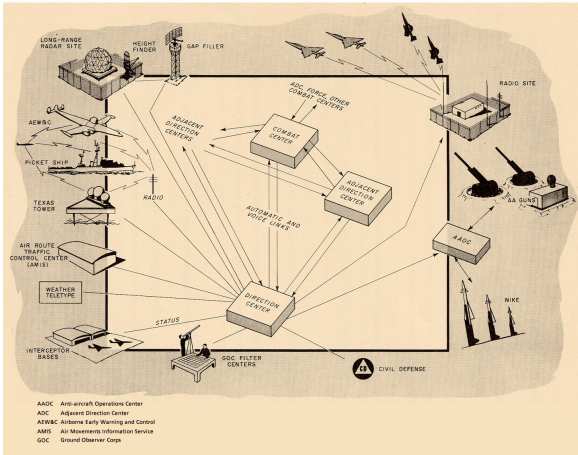
- ▶ korrektes Verhalten des Systems hängt nicht nur von den logischen Ergebnissen von Berechnungen ab
- ▶ zusätzlicher Aspekt ist der **physikalische Zeitpunkt** der Erzeugung und Verwendung der Berechnungsergebnisse

☞ Whirlwind (MIT, 1951), AN/FSQ-7 (Whirlwind II, IBM, 1957)

☞ SAGE (*semi-automatic ground environment*, 1958–1983)

SAGE und AN/FSQ-7 „Whirlwind II“

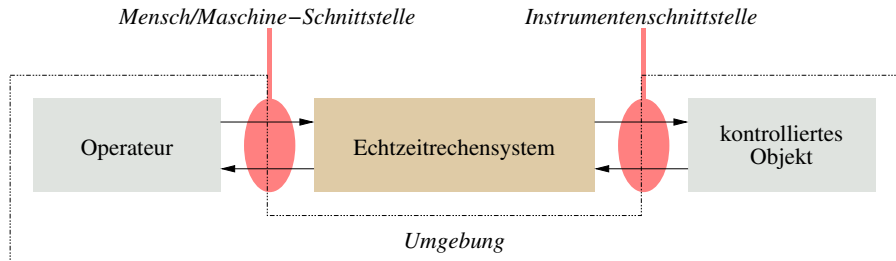
Erstes Echtzeitrechensystem als Schöpfung des „Kalten Krieges“



- ▶ 27 Installationen über die USA verteilt
 - ▶ *Nonstop*-Betrieb
 - ▶ 25 Jahre
- ▶ durch Datenfernleitungen miteinander gekoppelt
 - ▶ Telefonleitungen
 - ▶ Internet-„Mutter“
- ▶ pro Installation...
 - ▶ 100 Konsolen
 - ▶ 500 KLOC Ass.

Komponenten eines Echtzeitsystems

Echtzeitrechnungssystem und seine Umgebung



- ▶ das Echtzeitrechnungssystem muss auf Stimuli innerhalb von Zeitspannen reagieren, die durch die Umgebung vorgegeben sind
- ▶ der Zeitpunkt, zu dem ein Ergebnis vorliegen muss, wird als **Termin** oder **Frist** (engl. *deadline*) bezeichnet

Echtzeitfähigkeit bedeutet Rechtzeitigkeit

Zuverlässige Reaktion des Rechensystems auf Umgebungsereignisse

Geschwindigkeit liefert keine Garantie, um rechtzeitig Ergebnisse von Berechnungen abliefern und Reaktionen darauf auslösen zu können

- ▶ asynchrone Programmunterbrechungen (engl. *interrupts*) können **unvorhersagbare Laufzeitvarianzen** verursachen
- ▶ schnelle Programmausführung ist bestenfalls hinreichend für die rechtzeitige Bearbeitung einer Aufgabe

Zeit ist keine intrinsische Eigenschaft des Rechensystems

- ▶ die im Rechenystem verwendete Zeitskala muss mit der durch die Umgebung vorgegebenen identisch sein
- ☞ die **Terminvorgaben** der Umgebung gelten als weich, fest oder hart

Arten von Echtzeitsystemen

Verbindlichkeit von Terminvorgaben

weich (engl. *soft*) auch „schwach“

- ▶ das Ergebnis einer zu einem vorgegebenen Termin nicht geleisteten Arbeit ist weiterhin von Nutzen
- ▶ Terminverletzung ist tolerierbar

fest (engl. *firm*) auch „stark“

- ▶ das Ergebnis einer zu einem vorgegebenen Termin nicht geleisteten Arbeit ist wertlos und wird verworfen
- ▶ Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

hart (engl. *hard*) auch „strikt“

- ▶ das Versäumnis eines fest vorgegebenen Termins kann eine „Katastrophe“ hervorrufen
- ▶ Terminverletzung ist keinesfalls tolerierbar

Arten von Echtzeitsystemen (Forts.)

Fest \leftrightarrow Hart

fest/hart \mapsto Terminverletzung ist nicht ausgeschlossen¹

- ▶ die Terminverletzung wird vom Betriebssystem erkannt

fest \leadsto plangemäß weiterarbeiten

- ▶ das Betriebssystem bricht den Arbeitsauftrag ab
- ▶ der nächste Arbeitsauftrag wird gestartet
- ▶ ist transparent für die Anwendung

hart \leadsto sicheren Zustand finden

- ▶ das Betriebssystem löst eine **Ausnahmesituation** aus
- ▶ die Ausnahmebehandlung führt zum sicheren Zustand
- ▶ ist **intransparent für die Anwendung**

¹Auch wenn Ablaufplan und Betriebssystem auf dem Blatt Papier Determinismus zeigen, kann das im Feld eingesetzte technische System von Störeinflüssen betroffen sein, die ggf. die Verletzung auch eines harten Termins nach sich ziehen.

Arten von Echtzeitsystemen (Forts.)

Radikale Unterschiede im Systementwurf zeichnen sich ab...

hard real-time computer system

- ▶ ein Rechensystem, das mind. einen strikten Termin erreichen muss
 - ▶ garantiert unter allen (spezifizierten) Last- und Fehlerbedingungen
 - ▶ das Laufzeitverhalten ist ausnahmslos deterministisch
- ▶ typisch für ein **sicherheitskritisches Echtzeitrechensystem**
 - ▶ engl. *safety-critical real-time computer system*

soft real-time computer system

- ▶ ein Rechensystem, das keinen strikten Termin erreichen muss
- ▶ es ist erlaubt, gelegentlich Termine zu verpassen

Vorhersagbarkeit des Laufzeitverhaltens

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch

Determiniertheit

Bei ein und derselben Eingabe sind verschiedene Abläufe zulässig, alle Abläufe liefern jedoch stets das gleiche Resultat.

- ▶ Transparenz von Programmunterbrechungen
 - ▶ *Interrupts verursachen* vom „normalen Ablauf“ verschiedene *ausnahmebedingte Abläufe*
 - ▶ Verzögerungen in der Programmausführung wirken sich nicht nachteilig aus
- ▶ Terminvorgaben der Umgebung werden eingehalten

Determinismus

Zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird.

Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [3, S. 25]

rein zyklisch \leadsto nur periodische *Tasks*, *Polling*-Betrieb

- ▶ nahezu konstanter Betriebsmittelbedarf von Periode zu Periode

meist zyklisch \leadsto überwiegend periodische *Tasks*

- ▶ das System muss auf externe Ereignisse reagieren können

asynchron und irgendwie vorhersagbar \leadsto kaum periodische *Tasks*

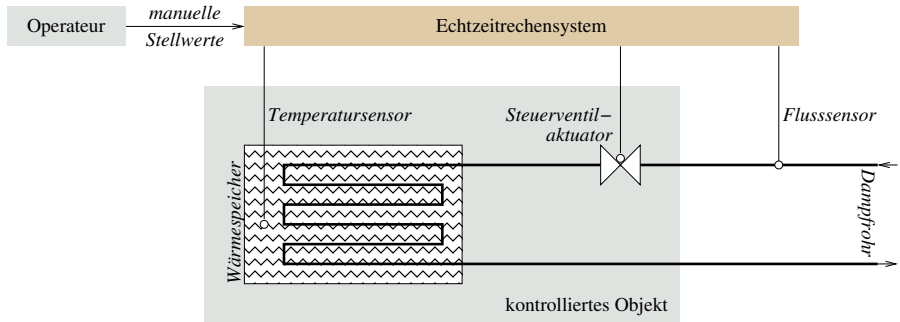
- ▶ aufeinanderfolgende Aktivierungen können zeitlich stark variieren
- ▶ Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik

asynchron und nicht vorhersagbar \leadsto aperiodische *Tasks*

- ▶ Anwendungen reagieren auf asynchrone Ereignisse
- ▶ hohe, nicht deterministische Laufzeitkomplexität einzelner *Tasks*

Aufbau des Demonstrators

Elementare Kontrollschleife



Aufgabe des Echtzeitrechnungssystems: Kontrolle des Steuerventils

- ▶ Dampffluss durch den Wärmetauscher bestimmen
- ▶ Temperatur im Wärmespeicher im vorgewählten Bereich halten

Kontrolliertes Objekt

Schrittfunktion (engl. *step function*) und Antwortfunktion (engl. *response function*)

Erhöhung des Dampfes (Schrittfunktion) verändert die Temperatur im Wärmespeicher (Antwortfunktion), bis **Gleichgewicht** eingestellt ist:

- ▶ die **Objektdynamik** bestimmt sich durch die Flüssigkeitsmenge im Wärmespeicher und den Dampfdruck des Wärmetauschers

Zeitparameter zur Charakterisierung der Schritt-/Antwortfunktion:

d^{object} Zeitdauer bis die Temperatur zu steigen beginnt

- ▶ hervorgerufen durch die (initiale) Trägheit des Objektes
- ▶ auch als Prozessverzögerung (engl. *process lag*) bezeichnet

d^{rise} Zeitdauer bis zum (erneuten) Temperaturgleichgewicht

Kontrollierendes Rechensystem

Echtzeitrechensystem

Temperaturwerte sind periodisch abzutasten, um Abweichungen des aktuellen Wertes von dem eingestellten Wert zu erkennen:

- d^{sample} Zeitabstand (konstant) zwischen zwei Abtastungen
- ▶ analoge auf digitale Werte abbilden \leadsto A/D-Wandlung
 - ▶ diskretes System sich quasi-kontinuierlich verhalten lassen
 - ▶ Faustregel: $d^{sample} < (d^{rise}/10)$

f^{sample} Abtastfrequenz, entspricht $1/d^{sample}$

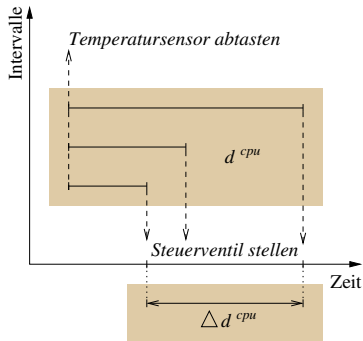
Abweichung (Ist-/Sollwert) bestimmen und dem Regelungsalgorithmus zur Berechnung des neuen Stellwertes zuführen:

- d^{cpu} Zeitdauer bis zur Ausgabe des neuen Stellwertes
- ▶ digitale auf analoge Werte abbilden \leadsto D/A-Wandlung
 - ▶ Randbedingung: $d^{cpu} < d^{sample}$

Δd^{cpu} Differenz zwischen Minimum und Maximum von d^{cpu}

Kontrollierendes Rechensystem (Forts.)

Schwankung (engl. *jitter*) in den Messergebnissen



d^{cpu} ist trotz konstantem Rechenaufwand zur Stellwertbestimmung variabel

- ▶ verdrängende Einplanung
- ▶ überlappende Ein-/Ausgabe
- ▶ Programmunterbrechungen
- ▶ Busüberlastung, DMA

Δd^{cpu} fügt Unschärfe zum Zeitpunkt der Temperatursensorabtastung hinzu

- ▶ bewirkt zusätzlichen Fehler
- ▶ beeinträchtigt die Dienstgüte

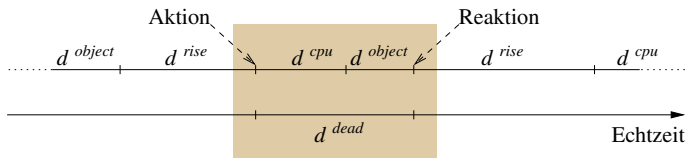
- ▶ **unbekannte variable Verzögerungen** können bei der Regelung nicht kompensiert werden, aber bekannte konstante Verzögerungen
- ▶ Randbedingung: $\Delta d^{cpu} \ll d^{cpu}$

Kontrolliertes Objekt \leftrightarrow Kontrollierendes Rechensystem

Totzeit des offenen Regelkreises

d^{dead} Zeitintervall zwischen Start der Aktion zur Stellwertberechnung und Wahrnehmung einer Reaktion nach erfolgter Steuerung

- ▶ setzt sich zusammen aus d^{cpu} und d^{object} , d.h.:
 1. der Implementierung des kontrollierenden Rechensystems
 2. der Dynamik des kontrollierten Objektes



- ▶ beeinträchtigt Güte und **Stabilität** der Kontrollschleife
 - ▶ insbesondere in Anbetracht der mit d^{cpu} gegebenen Varianz
- ▶ gibt einem relative Ungewissheit über die erzielte Wirkung

Resümee

Echtzeitbetrieb eines Rechensystems in seiner Umgebung

- ▶ Komponenten eines Echtzeitsystems:
 - ▶ Operateur, Echtzeitrechensystem, kontrolliertes Objekt
- ▶ Verhalten von Echtzeitanwendungen:
 - ▶ rein/meist zyklisch
 - ▶ asynchron und irgendwie/nicht vorhersagbar
- ▶ schwache, starke oder strikte Echtzeitbedingungen

Fallbeispiel (Wärmetauscher)

- ▶ Schritt- und Antwortfunktion, Abtastrate, Zeitparameter
- ▶ Schwankungen in den Messergebnissen, Totzeit

Literaturverzeichnis

- [1] Deutsches Institut für Normung.
Informationsverarbeitung — Begriffe.
DIN 44300. Beuth-Verlag, Berlin, Köln, 1985.
- [2] Hermann Kopetz.
Real-Time Systems: Design Principles for Distributed Embedded Applications.
Kluwer Academic Publishers, 1997.
- [3] Jane W. S. Liu.
Real-Time Systems.
Prentice-Hall, Inc., 2000.